Ciphers for MPC and FHE[1]

Martin Albrecht[1], Christian Rechberger[2,4], Thomas Schneider[3], Tyge Tiessen[2], and Michael Zohner[3]

[1] Royal Holloway, University of London, UK
martinralbrecht@googlemail.com
[2] DTU Compute, Technical University of Denmark, Denmark
{crec,tyti}@dtu.dk
[3] TU Darmstadt, Darmstadt, Germany
{thomas.schneider,michael.zohner}@ec-spride.de
[4] IAIK, Graz University of Technology, Austria
christian.rechberger@iaik.tugraz.at

Abstract. Designing an efficient cipher was always a delicate balance between linear and non-linear operations. This goes back to the design of DES, and in fact all the way back to the seminal work of Shannon.
Here we focus, for the first time, on an extreme corner of the design space and initiate a study of symmetric-key primitives that minimize the multiplicative size and depth of their descriptions. This is motivated by recent progress in practical instantiations of secure multi-party computation (MPC), fully homomorphic encryption (FHE), and zero-knowledge proofs (ZK) where linear computations are, compared to non-linear operations, essentially "free".
We focus on the case of a block cipher, and propose the family of block ciphers "LowMC", beating all existing proposals with respect to these metrics. As examples, we give concrete instatiations for 80-bit, 128-bit, and 256-bit security. We sketch several applications for such ciphers and give implementation comparisons suggesting that when encrypting larger amounts of data the new design strategy translates into improvements in computation and communication complexity by up to a factor of 5 compared to AES-128, which incidentally is one of the most competitive classical designs. Furthermore, we identify cases where "free XORs" can no longer be regarded as such but represent a bottleneck, hence refuting this commonly held belief with a practical example.

Keywords: block cipher, multiplicative complexity, multiplicative depth, secure multiparty computation, fully homomorphic encryption [1]

1 Introduction

Motivation. Modern cryptography developed many techniques that go well beyond solving traditional confidentiality and authenticity problems in two-party communication. Secure multi-party computation (MPC), zero-knowledge proofs (ZK) and fully homomorphic encryption (FHE) are some of the most striking examples.

In recent years, especially the area of secure multi-party computation has moved from a science that largely concerned itself with the mere existence of solutions towards considerations of a more practical nature, such as costs of actual implementations for proposed protocols in terms of computational time, memory, and communication.

Despite important progress and existing proof-of-concept implementations, e.g. [MNPS04, PSSW09, HEKM11, NNOB12, KSS12, FN13, SS13], there exists a *huge cost gap* between employing cryptographic primitives in a traditional way and using them in the more versatile MPC context. As an example, consider implementations of the AES block cipher, a global standard for the bulk encryption of data. Modern processors achieve a single execution of the block

[1] An earlier version appeared in the proceedings of Eurocrypt 2015 [ARS+15]. This is an updated and extended version.

cipher within a few hundred clock cycles (or even less than 100 clock cycles using AES-NI). However, realizing the same cipher execution in the context of an MPC protocol takes many billions of clock cycles and high communication volumes between the participating parties, e.g. several hundreds of Megabytes for two-party AES with security against malicious adversaries [PSSW09, NNOB12, KSS12, FN13, SS13, DZ13, LOS14, DLT14].

Traditionally, ciphers are built from linear and non-linear building blocks. These two have roughly similar costs in hardware and software implementations. In CMOS hardware, the smallest linear gate (XOR) is about 2-3 times larger than the smallest non-linear gate (typically, NAND). When implemented in an MPC protocol or a homomorphic encryption scheme, however, the situation is radically different: linear operations come almost for free, since they only incur local computation (resp. do not increase the noise much), whereas the bottleneck are non-linear operations that involve symmetric cryptographic operations and communication between parties (resp. increase the noise considerably). Our motivation hence comes from implementations of ciphers in the context of MPC, ZK, or FHE schemes where linear parts are much cheaper than non-linear parts.

This cost metric suggests a new way of designing a cipher where most of the cryptographically relevant work would be performed as linear operations and the use of non-linear operations is minimized. This design philosophy is related to the fundamental theoretical question of the minimal multiplicative complexity (MC) [BPP00] of certain tasks. Such extreme trade-offs were not studied before, as all earlier designs – due to their target platforms – faired better with obtaining a balance between linear and non-linear operations.

In this work we propose to start studying symmetric cryptography primitives with low multiplicative complexity in earnest. Earlier steps in this direction [GGNPS13, PRC12, GLSV14] were aimed at good cost and performance when implemented with side-channel attack countermeasures, and are not extreme enough for our purpose. Our question hence is: what is the minimum number of multiplications for building a secure block cipher? We limit ourselves to multiplications in GF(2) and motivate this as follows:

- By using Boolean circuits we decouple the underlying protocol / primitive (MPC protocol / ZK protocol / FHE scheme) from that of the cipher. Hence, the same cipher can be used for multiple applications.
- GF(2) is a natural choice for MPC protocols based on Yao or GMW (in the semi-honest setting, but also for their extensions to stronger adversaries), ZK protocols, as well as for fully or somewhat homomorphic encryption schemes (cf. Section 2 for details).

By nature of the problem, we are interested in two different metrics. One metric refers to what is commonly called multiplicative complexity (MC), which is simply the number of multiplications (AND gates) in a circuit, see e.g. [BPP00]. The second metric refers to the multiplicative depth of the circuit, which we will subsequently call ANDdepth. We note that already in [DSES14a] it was observed that using ciphers with low ANDdepth is of central importance for efficient evaluations within homomorphic encryption schemes. Therefore, the authors of [DSES14a] suggest to study block cipher designs that are optimized for low ANDdepth, a task to which we provide a first answer. Our work is somehow orthogonal to Applebaum et. al [AIK06], where the question of what can in principle be achieved in cryptography with shallow circuits was addressed.

While our design approach is not specific to block ciphers but can be equally applied to other primitives like hash functions or stream ciphers, in this work we focus on the block cipher case. This all motivates the following guiding hypothesis which we will test in this paper: "When implemented in practice, a block cipher design with lower MC and lower ANDdepth will result in lower executing times". We note that the relatively low execution times often reported in

the literature are *amortized* times, i.e. averaged over many calls of a cipher (in parallel). This, however, neglects the often important *latency*. Hence, another design goal in this work is to reduce this latency.

Outline and contribution. In Section 2 we describe several schemes with "free XORs"and use-cases where block ciphers with a low ANDdepth and MC are required. Then, in Section 3, we focus on an extreme corner of the design space of block ciphers and propose a new block-cipher design strategy that minimizes the multiplicative size and depth of the circuit describing it, beating all existing candidates with respect to these metrics. In terms of ANDdepth, the closest competitor is PRINCE and Noekeon. In terms of MC, the closest competitor turns out to be Simon. We give a high-level overview over a larger field of competing designs in Section 4. We analyse the security of our constructions in Section 5 and provide experimental evidence for the soundness of our approach in Section 6. In particular, our implementations outperform previously reported results in the literature, often by more than a factor 5 in MPC and FHE implementation settings. They also indicate that in the design space we consider, "free XORs" can no longer be regarded as free but significantly contribute to the overall cost, hence refuting this commonly held belief with a practical example. Finally, we describe our optimisation strategies for implementing our designs in the MPC and FHE case, which might be of independent interest.

Main features and advantages of LowMC. LowMC is a very parameterizable design approach. Given any blocksize, a choice for the number of Sboxes per round, and security expectations in terms of time and data complexity, a concrete instantiation can be created easily. Notable features include:

- Low ANDdepth, and low MC, which positively impacts the latency and throughput of the FHE, MPC, or ZK evaluation of the cipher. We give example instatiations that minimize the ANDdepth, others that minimize the number of ANDs overall, and again others that minimize the number of ANDs per encrypted bit.
- Partial Sbox layer, which is partially responsible for the low multiplicative complexity (MC). Together with Armadillo[BDJ+10], Zorro [GGNPS13] is the first SPN cipher in the literature that uses a non-full Sbox layer and is related to LowMC in this respect. However, recent attacks on Zorro that exploit this particular property [WWGY13, RASA14, GNPW13, BDD+15], highlight the need to be very careful with this design strategy. In our analysis of LowMC in Section 5 we are able to take these into account.
- Security arguments against large classes of statistical attacks like differential attacks, similar to other state-of-the-art designs are given in Section 5.
- In contrast to many other constructions, it is easy to obtain tight bounds on the MC and ANDdepth.
- The design is very flexible and allows for a unified description regardless of the blocksize. In fact it would even allow an asymptotic study of the design and analysis of this class of ciphers, and is as such the first practical construction after this was proposed for SPN constructions in [MV12].
- We explicitly de-couple the security claim of a block cipher from the block size.
- We do not add an explicit security margin, but base our choice for the number of rounds on attack vectors known to us and our bounds on them.

Version history of LowMC. A first parameter set for LowMC (called v0 in this paper) was circulated since end of 2014 and after a seminar presentation at ESC 2015, Dmitry Khovratovich point out the omission of combined attacks in our formula to compute the number of

rounds, which led to a new set of parameters (called v1 in this paper) which also appear in the proceedings version of this paper at Eurocrypt 2015 [ARS+15]. After its presentation a number of works gave new insights on how higher-order properties can get extended because of non-full Sbox layers [DLMW15, DEM16] and novel optimization of interpolation attacks [DLMW15]. Results show that for LowMCv1 for fraktion of weak choices of the affine layer, attacks much better than brute-force are possible. For LowMC a concrete choice for the used affine layers should follow a "nothing-up-my-sleeves approach" and not be in the hand of a malicious party. Attacks are also extended to work for any choice of affine layer for our parameterset aiming at 128-bit security. In this more interesting case, attacks reported in [DLMW15] are estimated to have a time complexity that is about 1000 times faster than a naive brute-force key search. We note that as the memory complexity of that attack is very high, even a variant of a brute force key search that could use the same amount of memory would also be sped up noticeably, perhaps by a factor 10 to 100. Insights of these two cryptanalysis papers have been taken into account in this paper, and the new parameter sets for LowMC are now referred to as LowMCv2. For reference, the earlier parameter-sets for v0 and v1 can be found in Appendix A and B.

Subsequent work. For the FHE/SHE use-case, a number of designs followed-up on LowMC and provided alternatives. Keyvrium [CCF+15] is a proposal aimin at 128-bit security and is based on the stream-cipher design approach Trivium [DP08] which has 80-bit security and has in terms of AND-related metrics similar properties as LowMC, albeit LowMC can be parameterized in many ways whereas the the Keyvrium design is fixed. The very recent proposal FLIP [MJSC16] has a much lower depth (especially important for the FHE use-case), but a much higher number of ANDs per encrypted bit. All of them use operations in GF(2) as their main building block. The only cipher design in this domain which deviates, with basic operations either in $GF(2^n)$ or $GF(p)$, is MiMC [AGR+16, GRR+16], which aims for a small number of multiplications in larger fields, and finds applications in MPC and ZK protocols.

2 Schemes and Applications

In this section we list several schemes and applications for MPC, FHE, and ZK that benefit from evaluating our cipher. In all the example applications below, the complexity of the functionality being evaluated within the protocol is dominated by the complexity of the block cipher and hence using our cipher instead of AES or any other cipher results in immediate performance gains.

2.1 Multi-Party Computation (MPC)

Schemes. There are two classes of practically efficient secure multi-party computation (MPC) protocols for securely evaluating Boolean circuits where XOR gates are considerably cheaper (no communication and less computation) than AND gates.

The first class of MPC protocols has a constant number of rounds and their total amount of communication depends on the MC of the circuit (each AND gate requires communication). Examples are protocols based on Yao's garbled circuits [Yao86] with the free XOR technique [KS08]. To achieve security against stronger (i.e., malicious or covert) adversaries, garbled circuit-based protocols apply the cut-and-choose technique where multiple garbled circuits are evaluated, e.g., [LP07, AL07, LPS08, PSSW09, LP11, SS11, KSS12, FN13, Lin13, HKE13, SS13, FJN14, HKK+14, LR14]; also MiniLEGO [FJN+13] falls into this class.

The second class of MPC protocols has a round complexity that is linear in the ANDdepth of the evaluated circuit (each AND gate requires interaction) and hence the performance depends

on both, the MC and ANDdepth of the circuit. Examples are the semi-honest secure version of the GMW protocol [GMW87] implemented in [CHK+12, SZ13], and tiny-OT [NNOB12] with security against malicious adversaries.

Server-side one-time passwords. The following application is being commercialized by Dyadic Security.[5] In one-time password authentication schemes, a user authenticates with a freshly generated short 6-8 digit password. This helps alleviate the problem of users choosing bad passwords. The one-time password is generated by a device that contains a cryptographic key and computes the new password by applying a function like AES to the time or some other transient value. In order to verify the one-time password, a server has to compute the cryptographic function itself, derive the password and compare. The problem with this system is that if a server breach occurs, then all of the cryptographic keys can be stolen. In such a case, all user devices have to be replaced, which is extremely expensive (most of these devices cannot even be reprogrammed with a new key). This exact scenario happened to RSA, and Lockheed-Martin reported attacks on their systems that can be traced back to the server breach at RSA (note that devices were not replaced after this breach, probably because it was not clear exactly what was stolen and the cost would be too great). Using MPC, it is possible to mitigate the danger of such a server breach by splitting the server into two or more parts and giving each server a share of the cryptographic key for computing the one-time passwords. Then, one-time passwords can be verified by running a secure computation to compute AES. This forces the attacker to break into both servers which is much harder (of course, in order to ensure that it is indeed harder, they should be given different protection mechanisms and/or be at different locations; in addition, the shares should be refreshed periodically so that an attacker has to break into both simultaneously).

Oblivious pseudorandom functions and applications. An example for privacy-preserving keyword search is based on Oblivious Pseudorandom Functions (OPRFs), introduced in [FIPR05], where one party inputs a key and the other party obliviously obtains an encryption under the key. OPRFs have further applications in set intersection [HL08] and secure database join [LTW13]. Another interesting application of OPRFs was proposed in[BCDa15] in the area of brokered identification systems. When evaluated as a circuit, the OPRF could be instantiated with our cipher instead of AES resulting in better performance.

Secure storage. [DK10] describe further applications of evaluating a block cipher (AES in their case) within an MPC protocol, e.g., to allow the servers involved in an MPC computation to store symmetrically encrypted information in untrusted (cloud) storage. Dyadic Security builds on this idea for their "Application-Layer Data Encryption".[6]

2.2 Fully homomorphic encryption (FHE)

Schemes. In all somewhat and fully homomorphic encryption schemes known so far XOR (addition) gates are considerably cheaper than AND (multiplication) gates. Moreover, XOR gates do not increase the noise much, whereas AND gates increase the noise considerably (cf. [HS14]). Hence, as in somewhat homomorphic encryption schemes the parameters must be chosen such that the noise of the result is low enough to permit decryption, the overall complexity depends on the ANDdepth.

[5] see `http://www.breachworks.com/datasheets/mpc-primer.pdf` for details
[6] see `https://www.dyadicsec.com/wp-content/uploads/2015/06/dyadic-s-dsm-web-suite-use-cases.pdf`

Sending data to the cloud. Today's FHE schemes still have a very high ciphertext expansion rate. Therefore, as described in [NLV11,LN14], it is beneficial for privacy-preserving applications based on FHE that outsource computations on sensitive data to the cloud to make use of hybrid encryption: Instead of encrypting client's data directly with FHE, the client encrypts its data using symmetric encryption, e.g., with a block cipher such as LowMC, and then sends the encrypted data along with the FHE-encrypted symmetric key to the cloud. The cloud then decrypts the symmetrically encrypted data under FHE. Using this method, the network communication is lowered to the data size, which is optimal, plus a one-time setup for sending the FHE-encrypted symmetric key.

Verifiable computing. Another use case of our cipher is *verifiable computing* proposed in [GGP10] which allows to outsource computation to untrusted workers (such as the cloud) and verify that the result was computed correctly. [GGP10] propose to evaluate a garbled circuit within fully homomorphic encryption. Today's most efficient construction of garbled circuits of [BHKR13] requires one evaluation of a block cipher per AND gate. The block cipher can be instantiated with our cipher instead of AES for better efficiency.

2.3 Zero-Knowledge proof of knowledge (ZK)

Schemes. In several zero-knowledge proof protocols XOR relations can be proven for free and the complexity essentially depends on the number of AND gates of the relation to be proven. Examples for such protocols are [BC86, BDP00] and the recently proposed highly efficient protocol of [JKO13] that requires only one evaluation of a garbled circuit [Yao86] and can make use of the free XOR technique [KS08].

ZK proof of knowledge of symmetric encryption key. The following application was described in [JKO13]: In 2010 Julian Assange released a "thermonuclear file insurance", i.e., a 1.4GB file which was an AES encryption of highly sensitive information as a countermeasure to protect WikiLeaks from being shut down by the U.S. government. The file was widely distributed over peer-to-peer networks and simply releasing the short encryption key would have allowed a great number of people to have access to the information. In order to prove that the file actually contains sensitive information without releasing the decryption key, one needs a block cipher that "works well" with a zero-knowledge protocol.

ZK proof of knowledge of MAC key. Another application of our cipher would be to prove in zero-knowledge that one knows the key corresponding to a message and a MAC for a block cipher-based MAC, e.g., CBC-MAC.

3 Description of LowMC

LowMC is a flexible block cipher based on an SPN structure where the block size n, the key size k, the number of Sboxes m in the substitution layer and the allowed data complexity d of attacks can independently be chosen[7]. The number of rounds needed to reach the security claims is then derived from these parameters.

To reduce the multiplicative complexity, the number of Sboxes applied in parallel can be reduced, leaving part of the substitution layer as the identity mapping. Despite concerns raised regarding this strategy [WWGY13, RASA14, GNPW13, BDD+15], we will show that security is

[7] The number of Sboxes is limited though by the block size as the Sboxes need to fit into a block.

viable. To reach security in spite of a low multiplicative complexity, pseudorandomly generated binary matrices are used in the linear layer to introduce a very high degree of diffusion. A method to accountably instantiate LowMC is given in Section 3.3.

Encryption with LowMC starts with a key whitening, followed by several rounds of encryption where the exact number of rounds depends on the chosen parameter set. A single round is composed as follows:

$$\textsc{LowMCRound}(i) = \\ \textsc{KeyAddition}(i) \circ \textsc{ConstantAddition}(i) \circ \textsc{LinearLayer}(i) \circ \textsc{SboxLayer}$$

In the following we give a detailed description of the individual steps.

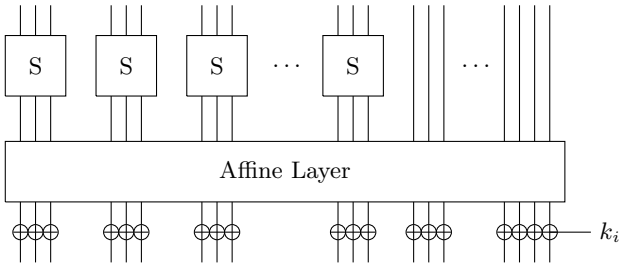

Fig. 1. Depiction of one round of encryption with LowMC.

SboxLayer is an m-fold parallel application of the same 3-bit Sbox on the first $3m$ bits of the state. If $n > 3m$ then for the remaining $n-3m$ bits, the SboxLayer is the identity. The selection criteria for the Sbox were as follows:

- Maximum differential probability: 2^{-2}
- Maximum square correlation: 2^{-2}
- Simple circuit description involving only 3 AND gates, with ANDdepth=1
- Algebraic degree 2 in each of the 8 non-zero component functions

The Sbox is specified in Figure 2, and coincides with the Sbox used for PRINTcipher [KLPR10]. A table representation of the Sbox can be found in Section C of the Appendix.

LinearLayer(i) is the multiplication in GF(2) of the state with the binary $n \times n$ matrix `Lmatrix[i]`. The matrices are chosen independently and uniformly at random from all invertible binary $n \times n$ matrices during the specification of a LowMC instantiation and are then fixed.

ConstantAddition(i) is the addition in GF(2) of `roundconstant[i]` to the state. The constants are chosen independently and uniformly at random from all binary vectors of length n during the specification of a LowMC instantiation and are then fixed.

KeyAddition(i) is the addition in GF(2) of `roundkey[i]` to the state. To generate `roundkey[i]`, the master key `key` is multiplied in GF(2) with the binary $n \times k$ matrix `Kmatrix[i]`. The matrices are chosen independently and uniformly at random from all binary $n \times k$ matrices of rank $\min(n, k)$ during the specification of a LowMC instantiation and are then fixed.

Decryption is done in the straightforward manner by an inversion of these steps.

$$\boxed{S(a,b,c) = (a \oplus bc, a \oplus b \oplus ac, a \oplus b \oplus c \oplus ab)}$$

Fig. 2. Specification of the 3-bit Sbox.

3.1 Pseudocode

`plaintext` and `state` are n-bit quantities. `key` is a k-bit quantity, which can both be larger or smaller than n. `r` is the number of rounds.

```
ciphertext = encrypt (plaintext,key)
  //initial whitening
  state = plaintext + MultiplyWithGF2Matrix(KMatrix(0),key)

  for (i = 1 to r)
    //m computations of 3-bit sbox,
    //remaining n-3m bits remain the same
    state = Sboxlayer (state)

    //affine layer
    state = MultiplyWithGF2Matrix(LMatrix(i),state)
    state = state + Constants(i)

    //generate round key and add to the state
    state = state + MultiplyWithGF2Matrix(KMatrix(i),key)
  end
  ciphertext = state
```

3.2 Parameters

As the design goal of LowMC is to be minimal with respect to complexity metrics based on the AND gates, we tried to minimize the number of rounds necessary to reach security. To this end we evaluated and estimated the security of LowMC with respect to a range of different attack vectors (see Section 5), and used this as a base to derive a formula that determines for any set of block size n, number of Sboxes m, key size k and allowed data complexity d, the number of rounds r necessary for security. Our selection of parameter sets offering different trade-offs can be found in Table 1. The formula used to derive the number of rounds can be found in Section 5.7.

3.3 Instantiation of LowMC

To maximize the amount of diffusion achieved by the linear layer, we rely on randomly generated, invertible binary matrices. As there exist no binary matrices of size larger than 1×1 that are MDS, and as it is generally an NP-complete problem to determine the branching number of a binary matrix [BMvT78], there is no obviously better method to reach this goal. The problem in the instantiation of LowMC is to find an accountable way of constructing the random matrices and vectors that leaves no room for the designer to plant backdoors.

Our recommended instantiation is a compromise between randomness, accountability and ease of implementation. It uses the Grain LSFR as a self-shrinking generator (see [HJMM08]

Table 1. A range of different parameter sets for LowMC instantiations offering different trade-offs. The first set is for PRESENT-like security parameters, the second set for AES-like security parameters. The third aims at a 128-bit post-quantum security level with $k = d = 256$. The ANDdepth corresponds to the number of rounds.

blocksize n	sboxes m	keysize k	data d	rounds r	# of ANDs	ANDs per bit
256	49	80	64	**12**	1764	6.89
128	31	80	64	**12**	1116	8.72
64	1	80	64	164	**492**	7.69
1024	20	80	64	45	2700	2.64
1024	10	80	64	85	2550	**2.49**
256	63	128	128	**14**	2646	10.34
196	63	128	128	**14**	2646	13.50
128	3	128	128	88	792	6.19
128	2	128	128	128	768	6.00
128	1	128	128	252	**756**	5.91
1024	20	128	128	49	2940	2.87
1024	10	128	128	92	2760	**2.70**
512	66	256	256	**18**	3564	6.96
256	10	256	256	52	1560	6.09
256	1	256	256	458	**1374**	5.37
1024	10	256	256	103	3090	**3.02**

and [MS94]) as a source of random bits. The exact procedure can be found in Section G in the Appendix.[8]

It must be mentioned though that it is principally possible to use any sufficiently random source to generate the matrices and constants. It is also not necessary that the source is cryptographically secure. Note that this of course assumes that no malicious party has influence over the choice of these constants.

4 Comparison with other ciphers

In the following we survey a larger number of existing cipher designs and study their ANDdepth and MC per encrypted bit which we summarize in Table 2. We both choose representative candidates from various design strategies, as well as the designs that are most competitive in terms of our metrics. We do this in two distinct categories: AES-like security (with key sizes of 128-bits and more and data security and block size of 128-bits and more), and lightweight security (data security and block size of 96 bits or below). Note that data security refers to the log_2 of the allowable data complexity up to which a cipher is expected to give the claimed securtiy against shortcut attacks. For LowMC we explicitly de-couple the data security from the block size of the cipher as the proposed design strategy favour larger block sizes but we don't see a new for larger data security than 128. For size-optimized variants we instantiate ℓ-bit adders using a ripple-carry adder which has $\ell - 1$ ANDs and ANDdepth $\ell - 1$; for depth-optimized variants we instantiate them with a Ladner-Fischer adder that has $\ell + 1.25\ell \log_2 \ell$ ANDs and ANDdepth $1 + 2 \log_2 \ell$, cf. [SZ13].

We first survey AES versions and then ciphers with related security properties. The Sbox construction of [BP12] has 34 AND gates and ANDdepth 4 (the size optimized Sbox construction of [BMP13] has only 32 AND gates, but higher ANDdepth 6). See also Canright [Can05]. To encrypt a 128-bit block, AES-128 has 10 rounds and uses 160 calls to the Sbox (40 for key

[8] A reference implementation of LowMC including functionality to generate instances as described above and a program to compute the number of rounds can be found at `https://github.com/tyti/lowmc`.

schedule), hence 5 440 AND gates, or 42.5 AND gates per encrypted bit. To encrypt a 128-bit block, AES-192 has 12 rounds and uses 192 calls to the Sbox (32 for key schedule), hence 6 528 AND gates, or 51 AND gates per encrypted bit. To encrypt a 128-bit block, AES-256 has 14 rounds and uses 224 calls to the Sbox (56 for key schedule), hence 7 616 AND gates, or 59.5 AND gates per encrypted bit.

AES is actually comparatively efficient. Other ciphers with a different design strategy can have very different properties. Threefish [FLS+10] is a cipher with large block size. Threefish with its 512-bit block size has 72 rounds with 4 additions modulo 2^{64} each resulting in 35.438 AND gates per encrypted bit and ANDdepth=4 536 (63 per round). Threefish with its 1 024-bit block size has 80 rounds with 8 additions each resulting in 39.375 AND gates per bit and ANDdepth=5 040 (63 per round). The recently proposed NSA cipher Simon [BSS+13] is also a good candidate to be of low multiplicative complexity. If b is the block size, it does $b/2$ AND gates per round, and ANDdepth is equal to the number of rounds. For a key size of 128 bit (comparable to AES) and block size 128 bit, it needs 68 rounds. This means, 4 352 AND gates, or 34 AND gates per bit.

In the lightweight category, we consider Present, but also Simon. The Present Sbox can be implemented with as little as 4 AND gates which is optimal [CHM11] and has ANDdepth 3. With $16 \cdot 31 = 496$ Sbox applications per 64 bit block we arrive at 31 AND gates per bit. A depth-optimized version of the Present Sbox with ANDdepth 2 and 8 ANDs is given in the full version of this paper. The 128bit secure version of Present differs only in the key schedule. Simon-64/96 has a 96 bit key, block size 64 bit and 42 rounds and Simon-32/64 has a 64 bit key, block size 32 bit and 32 rounds; see above for MC and ANDdepth. As another data point, the DES circuit of [TS] has 18 175 AND gates and ANDdepth 261. KATAN [CDK09] has 254 rounds. In KATAN32, the ANDdepth increases by two every 8 rounds resulting in an ANDdepth of 64; with 3 AND gates per round and a block size of 32 bit this results in 23.81 ANDs per bit, but similar to Simon-32/64 applications are limited due to the small block size. In KATAN48 and KATAN64 the ANDdepth increases by 2 every 7 rounds resulting in an ANDdepth of 74. KATAN48 has 6 ANDs per round and a block size of 48 bit resulting in 31.75 ANDs per bit. KATAN64 has 9 ANDs per round and a block size of 64 bit resulting in 35.72 ANDs per bit. Prince [BCG+12] has 12 rounds and each round can be implemented with ANDdepth 2, and each Sbox with 10 AND gates, cf. [DSES14b]. The low-latency design goal ends up making Prince the most competitive when it comes to the ANDdepth. NOEKEON [DPVAR00] is a competitive block cipher with 16 rounds and each round applies 32 S-boxes consisting of 4 AND gates with ANDdepth 2 each.

LowMC is easily parameterizable to all these settings, see also Table 1 in Section 3. It has at most (if $3m = n$) one AND gate per bit per round which results, together with a moderate number of rounds to make it secure, in the lowest ANDdepth and lowest MC per encrypted bit, cf. Table 2.

5 Resistance against cryptanalytic attacks

The ANDdepth, the number of ANDs per encrypted bit, and the total number of ANDs all critically depend on the number of rounds. As the goal of LowMC is to optimize the cipher with regard to these metrics, the goal of this security analysis is to determine the minimal number of rounds required to grant security for a given fixed set of parameters, i.e., for any choice of block size n, number of Sboxes m, allowed data complexity d, and key size k.

To determine the number of rounds needed for LowMC to be secure, we evaluate its vulnerability under a range of different attack vectors that cover a large portion of standard attacks. In particular, we will discuss differential cryptanalysis, linear cryptanalysis, boomerang attacks,

Cipher	Key size	Block size	Data sec.	ANDdepth	ANDs/bit	Sbox representation
AES-like security						
AES-128	128	128	128	40 (60)	43 (40)	[BP12] ([BMP13])
AES-192	192	128	128	48 (72)	51 (48)	[BP12] ([BMP13])
AES-256	256	128	128	56 (84)	60 (56)	[BP12] ([BMP13])
Simon	128	128	128	68	34	[BSS+13]
Simon	192	128	128	69	35	[BSS+13]
Simon	256	128	128	72	36	[BSS+13]
Noekeon	128	128	128	32	16	[DPVAR00]
Robin	128	128	128	96	24	[GLSV14]
Fantomas	128	128	128	48	16.5	[GLSV14]
Threefish	512	512	512	936 (4 536)	306 (36)	[FLS+10]
Threefish	512	1 024	1024	1 040 (5 040)	340 (40)	[FLS+10]
LowMC	128	256	128	**14**	10.34	AppendixC
LowMC	128	1024	128	92	**2.7**	AppendixC
Lightweight security						
PrintCipher-96	160	96	96	96	96	AppendixC
PrintCipher-48	80	48	48	48	48	AppendixC
Present	80 or 128	64	64	62 (93)	62 (31)	AppendixF ([CHM11])
Simon	96	64	64	42	21	[BSS+13]
Simon	64	32	32	32	16	[BSS+13]
Prince	128	64	64	24	30	[DSES14b]
KATAN64	80	64	64	74	36	[CDK09]
KATAN48	80	48	48	74	32	[CDK09]
KATAN32	80	32	32	64	24	[CDK09]
DES	56	64	56	261	284	[TS]
LowMC	80	256	64	**12**	6.89	AppendixC
LowMC	80	1 024	64	85	**2.49**	AppendixC
Allowing attacks with data and time complexity up to 2^{256}						
LowMC	256	512	256	**18**	6.96	AppendixC
LowMC	256	1 024	256	103	**3.02**	AppendixC

Table 2. Comparison of ciphers (excluding key schedule). We list the depth-optimized variants; size-optimized variants are given in () if available. Best in class are marked in bold.

higher-order attacks and interpolation attacks. The discussion is accompanied by experimental evaluation of small-scale version of LowMC.

Building up on this, we will finally formulate a relatively simple expression for deriving an estimate for the number of rounds needed for a desired security level.

5.1 Differential characteristics

In differential cryptanalysis [BS91], the principal goal is to find a pair (α, β) of an input difference α and an output difference β for the cipher such that pairs of input texts with difference α have an unusual high probability to produce output texts with difference β. Such a pair of differences is called a *differential*. A good differential can be used to mount distinguishing attacks as well as key recovery attacks on the cipher. For this it suffices if the differential does not cover the whole cipher but all except one or a few rounds.

As it is infeasible to calculate the probability of differentials for most ciphers, the cryptanalyst often has to be content with finding good *differential characteristics* i.e., paths of differences through the cipher for which the probability can directly be calculated. Note that a differential is made up of all differential characteristics that have the same input and output difference as the differential. The probability of a good differential characteristic is thus a lower bound for the related differential.

Allowing parts of the state to go unchanged through the Sbox layer clearly increases the chance of good differential characteristics. It is for example always possible to find a one round characteristic of probability 1. In fact, it is even possible to find $\lceil \frac{l}{3m} \rceil$-round characteristics of probability 1 where l is the width of the identity part and m the number of 3-bit Sboxes. Nonetheless, as we will prove in the following, this poses no threat. This is because of the

randomness of the linear layer which maps a fixed subspace to a random subspace of the same dimension: Most "good" differences i.e., differences that activate none or only few Sboxes, are mapped to "bad" differences that activate most of the Sboxes per layer. This causes the number of characteristics that only use "good" differences to decay exponentially with the number of rounds. In the case of a $\lceil \frac{l}{3m} \rceil$-round characteristic of probability 1, this means that the output difference is fixed to very few options, which makes it then already in the next round extremely unlikely that any one of the options is mapped onto a "good" difference.

We will now prove that good differential characteristics exist only with negligible probability in LowMC after a low number of rounds. The basic idea behind the proof is the following. We calculate for each possible good differential characteristic over r rounds the probability that it is realized in an instantiation of LowMC under the assumption that the binary matrices of the linear layer were chosen independently and uniformly at random. By evaluating the sum of these probabilities, which is an upper bound for the probability that any good characteristic exists, we can test that this probability is negligible for a given number of rounds.

Recall that m is the number of Sboxes in one Sbox layer in LowMC and that l is the bit-length of the identity part of the Sbox layer. We thus have $n = 3m + l$. Let $V(i)$ be the number of bit vectors of length n that correspond to differences that activate exactly i Sboxes. As we can choose i out of the m Sboxes, as for each active 3-bit Sbox there are 7 possible non-zero input differences, and as the bits of the identity part can be chosen freely, we have

$$V(i) = \binom{m}{i} \cdot 7^i \cdot 2^l \quad . \tag{1}$$

Let α_0 be an input difference and let α_1 be an output difference for one round of LowMC. Let a_0 be the number of Sboxes activated by α_0. As an active Sbox maps its non-zero input difference to four possible output differences each with probability $\frac{1}{4}$, and as a uniformly randomly chosen invertible binary $n \times n$ matrix maps a given non-zero n-bit vector with probability $\frac{1}{2^n-1}$ to another given non-zero output vector, the probability that the one-round characteristic (α_0, α_1) has a non-zero probability in a random instantiation of LowMC is

$$\frac{4^{a_0}}{2^n - 1} \quad . \tag{2}$$

Let $(\alpha_0, \alpha_1, \ldots, \alpha_r)$ now be a given characteristic over r rounds where the differences α_i are the output differences of round i and α_0 is the starting difference. Let $(a_0, a_1, \ldots, a_{r-1})$ be the numbers of Sboxes activated by each α_0, α_1,..., and α_{r-1}. We can now calculate the probability that this characteristic has a non-zero probability in a random instantiation of LowMC as

$$\frac{4^{a_0}}{2^n - 1} \cdot \frac{4^{a_1}}{2^n - 1} \cdots \frac{4^{a_{r-1}}}{2^n - 1} = \frac{4^{a_0 + a_1 + \cdots + a_{r-1}}}{(2^n - 1)^r} \quad . \tag{3}$$

Summing this probability now for all possible characteristics over r rounds that activate at most d Sboxes, gives us an upper bound for the probability that there exists any r-round characteristic with d or fewer active Sboxes. This sum is then

$$\sum_{\substack{0 \leq a_0, a_1, \ldots, a_{r-1} \leq m \\ a_0 + a_1 + \cdots + a_{r-1} \leq d}} V(a_0) \cdot V(a_1) \cdots V(a_{r-1}) \cdot (2^n - 1) \cdot \frac{4^{a_0 + a_1 + \cdots + a_{r-1}}}{(2^n - 1)^r} \tag{4}$$

where the factor $(2^n - 1)$ is the number of choices for the last difference α_r that can take any non-zero value.

With the knowledge that each active Sbox reduces the probability of a characteristic by a factor of 2^{-2}, we can now calculate for each parameter set of LowMC the number of rounds

after which any good differential characteristic is present only with a negligible probability. We consider as good differential characteristics those with a probability higher than 2^{-d}, where d is the allowed data complexity in the respective parameter set. We call a negligible probability a probability lower than 2^{-100}. Note that this probability only comes into play once when fixing an instantiation of LowMC.[9] The calculated bound for our choice of parameters can be found in Table 4.

Table 3. Example of how the probability bound p_{stat}, for the existence of differential or linear characteristic of probability at least 2^{-d}, evolves. The parameters are here $m = 42$, $d = 128$.

Rounds	1 - 6	7	8	9	10	11	12	13	14	15
$n = 256$	1.0	2^{-100}	2^{-212}	2^{-326}	2^{-442}	2^{-558}	2^{-676}	2^{-794}	2^{-913}	-
$n = 1024$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2^{-26}	2^{-145}	2^{-264}

5.2 Linear characteristics

In linear cryptanalysis [Mat93], the goal of the cryptanalyst is to find affine approximations of the cipher that hold sufficiently well. As with differential cryptanalysis, these can be used to mount distinguishing and key recovery attacks. The approximation is done by finding so-called *linear characteristics*, a concatenation of linear approximations for the consecutive rounds of the cipher. Similar to differential characteristics, linear characteristics activate Sboxes that are involved in the approximations.

The proof for the absence of good differential characteristics is directly transferable to linear characteristics because of two facts. Firstly, the maximal square correlation of an approximation over one Sbox is 2^{-2}, just the same as the maximal differential probability. Secondly, the transpose of a uniformly randomly chosen invertible binary matrix is still a uniformly randomly chosen invertible binary matrix. Thus we can use equation 4 to calculate the bounds for good linear characteristics as well.

5.3 Boomerang attacks

In boomerang attacks [Wag99], good partial differential characteristics that cover only part of the cipher can be combined to attack ciphers that might be immune to standard differential cryptanalysis. In these attacks, two differential characteristics are combined, one that covers the first part of the cipher and another that covers the second part. If both have about the same probability, the complexity corresponds roughly to the inverse of the product of the square of each of their probabilities [Wag99]. Thus to calculate the number of rounds sufficient to ensure that no good boomerang exists, we determine the number of rounds after which we cannot separate the cipher into two parts and find a differential for each such that the product of their probabilities is less than $2^{-d/2}$.

5.4 Higher order attacks

To prevent higher-order attacks such as [Knu94] and [DS09] ideally we would like to be able to make a statement such as

[9] We should mention here that this bound requires the linear layers are once chosen randomly and then fixed. Malicious instantiation of the linear layers can clearly introduce strong differential characteristics.

> After r rounds there is no output bit and no input subspace of dimension d such that the derivative of the polynomial representation of the output bit with respect to this subspace is the zero-polynomial.

As it seems very difficult to determine whether such a statement holds, we approximate it in the following.

First we determine a bound on the number of rounds needed for terms of degree d to appear. As we will see later, an upper bound for the degree of the encryption function dependent on the number of rounds can be stated recursively as follows. If the degree after r rounds is d then the degree after $r+1$ rounds is at most

$$\min\left(2d, m+d, \frac{n}{2}+\frac{d}{2}\right). \tag{5}$$

We believe that this bound is tight (or at least close to tight) in LowMC in the following way: If the bound suggests that after r rounds the degree is at most d, the polynomial representation of any output bit will contain at least one term of degree d. While clearly there are terms of degree d that can impossibly be realized, we believe that the large number of possible ways to construct terms of degree d and the random nature of the linear layer will cause at least one to be realized.

Now one problem remains. If we know that a term of degree d exists, we can still take the derivative with respect to one variable that is not in the term to ensure that it vanishes. Thus we need to prepend a certain number of rounds that ensures that the term of degree d depends on all input bits. Or in other words, that all terms of order d are realized in the polynomial of the output bit with some probability.

As such an estimate for the number of rounds which need to be prepended, we use the number of rounds needed for one input bit to influence all output bits. We approximate this as follows. An average input bit activates $\frac{7}{8}$ of all m S-boxes. Thus after one round the dimension of the linear space that this bit influences is on average $\frac{7m}{8}$. As the linear layer is random, we can assume that this bound grows linearly, such that after $\left\lceil \frac{8n}{7m} \right\rceil$ rounds, we expect that on average all bits are influenced. Prepending this number of rounds should give a good estimate for the number of rounds needed so that no higer-order derivatives of our chosen degree give us a zero-polynomial in any output bit.

Estimating the growth of the degree Due to its small width of only 3-bit, the degree of the Sbox in its algebraic representation is only two. Since in one round the Sboxes are applied in parallel and since the affine layer does not change the algebraic degree, the algebraic degree of one round is two as well. As a low degree could be used as a lever for a higher-order attack, let us take a look at how the algebraic degree of LowMC develops over several rounds.

Clearly the algebraic degree of the cipher after r rounds is bounded from above by 2^r. It is furthermore generally bounded from above by $n-1$ since the cipher is a permutation. A second upper bound, that is better suited and certainly more realistic for the later rounds, was found by Boura, Canteaut, and De Cannière [BCC11]. In our case it is stated as following: If the cipher has degree d_r after r rounds, the degree after round $r+1$ is at most $\frac{n}{2}+\frac{d_r}{2}$. Differing from [BCC11], in LowMC the Sbox layer only partially consists of Sboxes and partially of the identity mapping. This must be accounted for and requires a third bound: If the cipher has degree d_r after r rounds, the degree after round $r+1$ is at most $m+d_r$. A proof of this can be found in Secion E. This can be summarized as follows:

Lemma 1. *If the algebraic degree of LowMC with m Sboxes and length l of the identity part in the Sbox layer is d_r after r rounds, the degree in round $r+1$ is at most*

$$\min\left(2d_r, m + d_r, \frac{n}{2} + \frac{d_r}{2}\right) \tag{6}$$

where $n = 3m + l$ is the block width of LowMC.

Combining these three bounds, we can easily calculate lower bounds for the number of rounds r needed for different parameter sets l and m of LowMC to reach a degree that is at least as large as the allowed data complexity d minus 1. The results of this for LowMC's parameters are displayed in Table 4.

Table 4. For the different sets of LowMC parameters, bounds are given for the number of rounds for which no good differential or linear characteristics exist (r_{stat}), to avoid good boomerangs (r_{bmrg}), and the number of rounds needed to have a sufficiently high algebraic degree (r_{deg}). The bounds were calculated using Equations 4 and 6.

Sboxes	blocksize	data complexity	r_{stat}	r_{bmrg}	r_{deg}
49	256	64	5	6	6
63	256	128	5	6	7

5.5 Interpolation attack

The bound derived in this section is based on the attack on LowMC by Dinur, Liu, Meier and Wang [DLMW15].

In an interpolation attack [JK97], we determine the polynomial representation of a state bit. This is done by combining knowledge about restrictions of this polynomial with a sufficient number of evaluations of this polynomial. More concretely, let us suppose we can determine the value of some intermediate bit in LowMC by choosing the right plaintexts. Let us furthermore assume that the polynomial representation of this bit in the ciphertext bits can only contain a restricted set of terms. When we now regard the coefficients of these terms as the unknown variables, a pair of plaintext and ciphertext will correspond to a linear equation in these variables. Using sufficiently many plaintext-ciphertext pairs, we can create enough linear independent equations to be able to solve the system of equations and hence deduce the polynomial. As the coefficients usually depend on the key bits, we gain knowledge about the key bits.

Let us now estimate the time complexity of this attack. When we consider the intermediate state bits before the last round, we see that the only quadratic terms that appear are those produced by the S-boxes. As any S-box output bit contains only quadratic term, there are thus only $3m$ quadratic terms available.[10] For rounds farther away from the last round, we determine the number of terms of a given degree by considering all combinations of terms of the previous round that, when multiplied, give terms of that degree.

We thus receive an estimate of the number of terms of each degree for a given number of rounds before the last round. But for the correct estimate, we also need to consider the number of possible coefficients for these terms. We note that the maximal degree in the key bits of a coefficient depends on the degree of its term. For example, let us consider the polynomial of a bit

[10] This is when we represent the polynomials as polynomials over the standard base, that is the bits of the ciphertext. This is intuitively the best choice of base for an attack.

before the second to last round. The maximal degree of the terms is then 4. But the coefficients cannot depend on the key bits. For the terms of the degree 3, the coefficients can only be linear in the key bits. And the terms of degree 2 can only depend quadratically on the key bits. As this limits the number of possible coefficients and thus the number of equations needed for an attack, we estimate the number of equations needed as follows: for each possible term degree, we determine how many possible terms and how many possible coefficients for these terms there are and take the minimum of both.

Lastly we need to note that the complexity of the interpolation attacks depends on solving a linear system of equations in the variables. We estimate thus that a number of $\frac{k}{2.3}$ variables is sufficient to make the attack more complex than bruteforcing the k key bits since the best methods for solving a system of linear equations (which have a large overhead) have a limit exponent of 2.3.

5.6 Experimental Cryptanalysis

We proved that no good differential or linear characteristic can cover sufficiently many rounds to be usable as an attack vector in LowMC. This does not exclude though the possibility of good differentials or linear hulls for which a large number of characteristics combine. Given the highly diffusive, random linear layers, this seems very unlikely.

Likewise we were able to find lower bounds on the number of rounds needed for the algebraic degree of LowMC to be sufficiently high. Even though this is state-of-the art also for traditional designs to date, this gives us no guarantee that it will indeed be high. Unfortunately it is not possible to directly calculate the algebraic degree for any large block size.

To nevertheless strengthen our confidence in the design, we numerically examined the properties of small-scale versions of LowMC. In Table 5, we find the results for a 24-bit wide version with 4 Sboxes. For testing its resistance against differential cryptanalysis, we calculated the full codebook under 100 randomly chosen keys and used the distribution of differences to estimate the probabilities of the differentials. To reduce the computational complexity, we restricted the search space to differentials with one active bit in the input difference.

It can clearly be seen that the probability of differentials quickly saturates to values too low to allow an attack. Clearly, the bound calculated with equation 4 (p_{stat} in the table) overestimates the probability of good characteristics. Even though we were not able to search the whole space of differentials there is little reason to assume that there are other differentials that fare considerably better. It is important to note that the number of impossible differentials goes to 0 after only few rounds. Thus impossible differentials cannot be used to attack any relevant number of rounds. At the same time this assures the absence of any truncated differentials of probability 1.

The minimal algebraic degree[11] is tight for this version when compared with the theoretic upper bound as determined with equation 6.

For the version with only 2 Sboxes the degree deviates for some round numbers from the theoretic bound which is not too surprising considering the size of the cipher. It can beexpected that with increasing block size – and with the thereby quadraticallyincreasing random matrices – the bound is more likely to hold.

5.7 Determining the number of rounds needed for security

The design goal of LowMC is to offer a cipher optimized for schemes whose performance critically depends on the ANDdepth, the number of ANDs or the number of ANDs per bit. We thus tried

[11] That is the minimum of the algebraic degrees of the 24 output bit when written as Boolean functions.

Table 5. Experimental results of full codebook encryption over 100 random keys for a set of small parameters are given. p_{best} and p_{worst} are the best and the worst approximate differential probability of any differential with one active bit in the input difference. n_{imposs} is the number of impossible differentials with one active bit in the input difference. $\deg_{\text{exp}}$ is the minimal algebraic degree in any of the output bits. $\deg_{\text{theor}}$ is the upper bound for the algebraic degree as determined from equation 6. p_{stat} is the probability that a differential or linear characteristic of probability at least 2^{-12} exists (see eq. 4).

$n = 24$, $m = 4$, $k = 12$, $d = 12$

Rounds	p_{best}	p_{worst}	n_{imposs}	$\deg_{\text{exp}}$	$\deg_{\text{theor}}$	p_{stat}
2	$2^{-8.64}$	0	$2^{28.58}$	4	4	-
3	$2^{-12.64}$	0	$2^{28.00}$	8	8	-
4	$2^{-14.64}$	0	$2^{4.25}$	12	12	-
5	$2^{-18.60}$	$2^{-26.06}$	0	16	16	-
6	$2^{-20.49}$	$2^{-25.84}$	0	20	20	-
7	$2^{-23.03}$	$2^{-25.74}$	0	22	22	-
8	$2^{-23.06}$	$2^{-25.74}$	0	23	23	-
10	-	-	-	-	-	$2^{-5.91}$
11	-	-	-	-	-	$2^{-16.00}$
12	-	-	-	-	-	$2^{-26.28}$
19	-	-	-	-	-	$2^{-101.5}$

$n = 24$, $m = 2$, $k = 12$, $d = 12$

Rounds	p_{best}	p_{worst}	n_{imposs}	$\deg_{\text{exp}}$	$\deg_{\text{theor}}$	p_{stat}
4	$2^{-8.64}$	0	$2^{28.55}$	6	8	-
5	$2^{-12.62}$	0	$2^{28.17}$	10	10	-
6	$2^{-12.64}$	0	$2^{24.93}$	10	12	-
7	$2^{-14.64}$	0	$2^{4.75}$	14	14	-
8	$2^{-16.63}$	$2^{-26.47}$	0	14	16	-
9	$2^{-16.64}$	$2^{-26.06}$	0	16	18	-
10	$2^{-20.34}$	$2^{-25.84}$	0	18	20	-
11	$2^{-20.50}$	$2^{-25.84}$	0	22	22	-
12	$2^{-22.94}$	$2^{-26.06}$	0	22	23	-
20	-	-	-	-	-	$2^{-5.91}$
21	-	-	-	-	-	$2^{-10.93}$
22	-	-	-	-	-	$2^{-16.00}$
38	-	-	-	-	-	$2^{-101.5}$

to be as close to the number of rounds needed for security as possible. Based on our discussion of different attack vectors above, our recommendation for the number of rounds is hence the following:

$$r \geq \max(r_{\text{stat}}, r_{\text{bmrg}}, r_{\text{deg}} + r_{\text{diff}}) + r_{\text{outer}}$$

where r_{stat} is a bound for statistical attack vectors such as differentials and linear characteristics as discussed in Section 5.1, r_{bmrng} is the bound for boomerang attacks as discussed in Section 5.3, r_{deg} indicates the number of rounds needed for the cipher to have sufficient degree and where r_{diff} denotes the number of rounds needed for full diffusion as discussed in Section 5.4. Values of these for the parameters of LowMC can be found in Table 4. For the number of rounds which can be peeled off at the beginning and end of the cipher by key guessing and other strategies, we use the ad-hoc formular $r_{\text{outer}} = r_{\text{interpol}}$ where r_{interpol} denotes number of rounds that can be peeled off by an interpolation attack as discussed in Section 5.5.

6 Comparison of Implementations

In the following we report on experiments when evaluating LowMC with MPC protocols in Section 6.1 and with FHE in Section 6.2. The performance of both implementations is independent of the specific choice of the random matrices and vectors used in LowMC (cf. Section 3.3) as we do not use any optimizations that are based on their specific structure.

In both the FHE and MPC settings, for more efficient matrix multiplication, we use a method that is generically better than a naive approach: the "method of the four Russians" [ABH10]. This method reduces the complexity of the matrix-vector product from $O(n^2)$ to $O(n^2/log(n))$, i.e. it's an asymptotically faster algorithm and is also fast in practice for the dimensions we face in LowMC. Asymptotically faster methods like the Strassen-Winograd method method make no sense however, for the dimensions we are considering.

It turns out that considering design-optimizations of the linear layer by introducing structure and thereby lowering the density of the matricies and in turn reducing the number of XOR computations will not improve performance of all these implementations. On the contrary,

as the application of the security analysis suggests, the number of rounds would need to be increased in such a case.

6.1 MPC Setting

As an example for both classes of MPC protocols described in Section 2.1 we use the GMW protocol [GMW87] in the semi-honest setting. As described in [CHK+12], this protocol can be partitioned into 1) a *setup phase* with a constant number of rounds and communication linear in the MC of the circuit (2κ bits per AND gate for κ-bit security), and 2) an *online phase* whose round complexity is linear in the ANDdepth of the circuit. Hence, we expect that the setup time grows linearly in the MC while the online time grows mostly with increasing ANDdepth when network latency is high.

Benchmark Settings. For our MPC experiments we compare LowMC against other ciphers with a comparable level of security. We compare LowMC with the two standardized ciphers Present and AES and also with the NSA cipher Simon which previously had the lowest number of ANDs per encrypted bit (cf. Table 2). More specifically, for lightweight security with at least $\kappa = 80$ bit security we compare LowMC with 80 bit keys against Present with 80 bit key (using the Sbox of [CHM11]) and Simon with 96 bit keys (the Simon specification does not include a variant with 80 bit keys); for long-term security with $\kappa = 128$ bit security we compare LowMC with 128 bit keys against AES-128 (using the Sbox of [BP12]) and Simon with 128 bit key; we set the security parameters for the underlying MPC protocol to $\kappa = 80$ bit for lightweight security and to $\kappa = 128$ bit for long-term security. We exclude the key schedule and directly input the pre-computed round keys. We use the GMW implementation that is available in the ABY-framework [DSZ15] which uses the efficient oblivious transfer extensions of [ALSZ13][12]. We run our MPC experiments on two desktop PCs, each equipped with an Intel Haswell i7-4770K CPU with 3.5 GHz and 16GB of RAM, that are connected by Gigabit LAN. To see the impact of the reduced ANDdepth in the online phase, we measured the times in a LAN scenario (0.2 ms latency) and also a trans-atlantic WAN scenario (50 ms latency) which we simulated using the Linux command `tc`.

In our first experiment depicted in Table 6 we encrypt a single block, whereas in our second experiment depicted in Table 7 we encrypt multiple blocks in parallel to encrypt 12.8 Mbit of data.

Single-Block Results. From our single-block experiments in Table 6 we see that the communication of LowMC is factor 1.6 lower than the next best cipher for lightweight security and factor 6 for long-term security. In terms of total runtime, for lightweight security LowMC is slower than Present and Simon by factor 3 in the LAN setting but outperforms both by factor 3 to 7 in the WAN setting. For long-term security, LowMC is slower than AES and Simon by factor 4 in the LAN setting, but again outperforms both in the WAN setting by factor 3 to 4. These results can be explained by the high number of XOR gates of LowMC compared to AES, which impact the run-time higher than the communication for the AND gates. In the WAN setting, the higher ANDdepth of AES outweighs the local overhead of the XOR gates for LowMC, yielding a faster run-time for LowMC.

[12] Our MPC implementations of the benchmarked block-ciphers are available online at `http://encrypto.de/code/LowMC`.

Lightweight Security						
Cipher	Present		Simon		LowMC	
Communication [kB]	39		26		**16**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	**0.001**	**0.05**	**0.001**	**0.05**	**0.001**	**0.05**
Online [s]	0.008	9.31	**0.005**	4.21	0.033	**1.23**
Total [s]	0.009	9.36	**0.006**	4.26	0.034	**1.29**
Long-Term Security						
Cipher	AES		Simon		LowMC	
Communication [kB]	170		136		**23**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	0.003	**0.05**	0.003	**0.05**	**0.002**	**0.05**
Online [s]	**0.006**	4,01	0.009	6.81	0.038	**1.44**
Total [s]	**0.009**	4.06	0.012	6.86	0.040	**1.49**

Table 6. GMW benchmarking results for single block. Best in class marked in bold.

Multi-Block Results. From our multi-block experiments in Table 7 we see that LowMC needs less communication than all other ciphers: at least factor 6 for lightweight security and factor 12 for long-term security. Also the total runtime of LowMC is the lowest among all ciphers, ranging from factor 3 when compared to Simon for lightweight security to factor 8 when compared to AES for long-term security.

Lightweight Security						
Cipher	Present		Simon		LowMC	
Comm. [GB]	7.4		5.0		**0.8**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	68.08	83.89	45.16	55.49	**14.6**	**18.85**
Online [s]	0.91	10.00	**0.59**	5.82	0.91	**1.29**
Total [s]	68.99	93.89	45.75	61.31	**15.5**	**20.14**
Long-Term Security						
Cipher	AES		Simon		LowMC	
Comm. [GB]	16		13		**1.1**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	101.63	198.53	73.42	87.13	**22.09**	**29.21**
Online [s]	1.29	4.27	**0.95**	7.71	1.58	**1.95**
Total [s]	102.85	202.85	74.38	94.84	**23.70**	**25.29**

Table 7. GMW benchmarking results for multiple blocks to encrypt 12.8 Mbit of data. Best in class marked in bold.

Summary of the Results. To summarize our MPC experiments, the benefits of LowMC w.r.t. the *online time* depend on the network latency: over the low-latency LAN network existing ciphers achieve faster online runtimes than LowMC, whereas in the higher latency WAN network LowMC achieves the fastest online runtime. W.r.t. the *total runtime*, LowMC's benefit in the single-block application again depends on the latency (comparable or slightly less efficient over LAN, but more efficient over WAN), whereas in the multi-block application LowMC significantly improves over existing ciphers by factor 4 to 8. For secure computation protocols with security against malicious adversaries, the benefit of using LowMC would be even more significant, since there the costs per AND gate are at least an order of magnitude higher than in the semi-honest GMW protocol, cf. [NNOB12, LOS14].

6.2 FHE Setting

We implemented LowMC using the homomorphic encryption library HELib [HS13,HS14], which implements the BGV homomorphic encryption scheme [BGV11] and which was also used to

evaluate AES-128 [GHS12a,GHS12b]. Our implementation represents each plaintext, ciphertext and key bits as individual HE ciphertexts on which XOR and AND operations are performed. Due to the nature of the BGV system this means that we can evaluate many such instances in parallel, typically a few hundred. We found this representation to be more efficient than our other "compact" implementation which packs these bits into the slots of HE ciphertexts.

In the homomorphic encryption setting the number of AND gates is not the main determinant of complexity. Instead, the ANDdepth of the circuit largely determines the cost of XOR and AND, where AND is more expensive than XOR. However, due to the high number of XORs in LowMC, the cost of the linear layer is not negligible. In our implementation we use the "method of the four Russians" [ABH10] to reduce the number of HE ciphertext additions from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^2/\log(n))$.

In our experiments we chose the depth for the homomorphic encryption scheme such that the "base level" of fresh ciphertexts is at least the number of rounds, i.e. we consume one level per round. Our implementation also does not precompute round keys in advance, but deriving round keys is considered part of the evaluation (cost).

We consider LowMC instances for Present-80 and AES-128 like security. We always choose a homomorphic encryption security level of 80 for compatibility with [GHS12b]. Our results are given in Table 8. Our implementation is available at `https://bitbucket.org/malb/lowmc-helib`.

d	m	r	n	#blocks	t_{setup}	t_{eval}	t_{sbox}	t_{key}	t_{block}	t_{bit}
128	63	14	256	504	6.7	854.0	605.6	2.2	1.6945	0.0066
64	31	12	128	600	11.8	217.7	166.9	0.6	0.3629	0.0028

Table 8. LowMC (commit `f6a086e`) in HElib [HS13] (commit `be9d3785e`) on Intel i7-4850HQ CPU @ 2.30GHz; d is the allowed data complexity, m is the number of Sboxes, n is the blocksize, r is the number of rounds, # blocks is the number of blocks computed in parallel, t_{setup} is the total setup time, t_{eval} is the total running time of the encryption, t_{sbox} the total time spent in the S-Box layer, t_{key} the total time spent in the key schedule, $t_{block} = t_{eval}/\#\text{blocks}$ and $t_{bit} = t_{block}/n$. All times are in seconds.

For comparison with previous results in the literature we reproduce those results in Table 9 which demonstrates the benefit of a dedicated block cipher for homomorphic evaluation.

d	ANDdepth	#blocks	t_{eval}	t_{block}	t_{bit}	Cipher	Reference	Key Schedule
128	40	120	3m	1.5s	0.0119s	AES-128	[GHS12b]	excluded
128	40	2048	31h	55s	0.2580s	AES-128	[DHS14]	excluded
128	40	1	22m	22m	10.313s	AES-128	[MS13]	excluded
128	40	12	2h47m	14m	6.562s	AES-128	[MS13]	excluded
128	14	504	14.2m	1.7s	0.0066s	LowMC	this work	included
64	24	1024	57m	3.3s	0.0520s	PRINCE	[DSES14b]	excluded
64	12	600	3.6m	0.36s	0.0028s	LowMC	this work	included

Table 9. Comparison of various block cipher evaluations in the literature and this work; Notation as in Table 8. Memory requirements are not listed as they are usually not provided in the literature. The first row is based on experimental data obtained on the same machine and the same instance of HELib as in Table 8.

7 Conclusions, lessons learned, and open problems

We proposed block ciphers with an small number of AND gates and an shallow AND depth, demonstrated the soundness of our design through experimental evidence and provided a se-

curity analysis of these constructions. Of course, as with any other block cipher, more security analysis is needed to firmly establish the security provided by this new design. Furthermore, with the proposal of the LowMC family, we bring together the areas of symmetric cryptographic design and analysis research with new developments around MPC and FHE. Finally, in contrast to current folklore belief, in some implementation scenarios, we identified practical cases where "free XORs" can no longer be considered free and where local computations in an MPC protocol represent a considerable bottleneck.

To finish, we highlight a number of open problems related to the LowMC family of ciphers in the areas of cryptanalysis, design, and implementation. We start with cryptanalysis, as analyzing such an extreme corner of the design space for a symmetric cipher is an interesting endeavor in itself.

- In Eq(4) we give an approach to calculate, for any number of Sboxes per round, block size, and security level, a number of rounds for which we believe and our analysis suggests that LowMC is secure. We on purpose do not add an additional security margin. Are there attack vectors for which those r rounds are not enough? Or is perhaps the opposite true: is it possible to reduce the number of rounds in LowMC, which in turn would further reduce MC and ANDdepth?
- Security of LowMC, or variants of it, in other use-cases for which we do not claim security, i.e. against related-key attacks, or its use in a key-less settings like block-cipher-based compression or hash functions, or sponge-based hash functions.

There are also various open questions related to the design of LowMC-like ciphers:

- Affine layer (1): Can we add more structure into the linear layers in order to reduce the necessary computational effort in those cases where the number of AND gates is no longer the bottleneck? Do such approaches beat applying asymptotically faster linear algebra techniques for applying linear layers as done in Section 6? As we argue in the paper, simply lowering the density of the matrices by several factors of two will not be enough to change performance at all.
- Affine layer (2): For breaking up symmetries it seems to suffice to have sufficiently distinct linear layers in every round, i.e. distinct random matrices `Lmatrix[i]`. We however chose to add distinct round constants, also because the cost of doing so is very low, even though we don't know of an attack. An interesting question hence is: Could LowMC without the ConstantAddition(i) step be secure, perhaps equally so?
- Affine layer (3): Can we add more structure into the linear layer in order to improve the security? Our strategy to use random linear layers gives good *expected* properties, but a small probability for unlucky choices remains. For that reason we introduced an analysis technique in the paper that can quantify this risk, and set the threshold for our concrete instantiations to 2^{-100}. It will be interesting to study what techniques for constructing concrete affine layers (using e.g. linear codes as it is usually done) can achieve. The advantage would be to *remove that chance of being unlucky*, but we conjecture it to be unlikely that it will be possible to e.g. at the same time get close to our bounds on the probability of linear or differential characteristics.
- Sbox-layer: It would be interesting to study Sboxes (larger than 3 bits) with low ANDdepth and Sboxes with low multiplicative complexity.

Currently, the multiplicative complexity and ANDdepth of various cipher constructions is poorly understood. For example, it would be interesting to find efficient algorithms along the lines of [BMP13] for the various ciphers in the literature. While our choice for GF(2) is well motivated, there are scenarios where larger fields might be beneficial. What designs minimize

MC and multiplicative depth under such constraints? In a follow-up work MiMC[AGR+16, GRR+16] was proposed that aims to minimize the number of multiplications in GF(p). Finding designs in GF(p) that minimize multiplicative depth is still an interesting option problem.

Acknowledgements. We thank Dmitry Khovratovich for pointing out to us that an earlier version of our parameter sets for LowMC instantiations was too optimistic. We thank Orr Dunkelman for helpful clarifications on the cipher KATAN. We thank Gaëtan Leurent and François-Xavier Standaert for helpful clarifications regarding the ciphers Fantomas and Robin.

The work of Albrecht was supported by EPSRC grant EP/L018543/1 "Multilinear Maps in Cryptography". The work of co-authors from TU Darmstadt was supported by the European Union's 7th Framework Program (FP7/2007-2013) under grant agreement n. 609611 (PRACTICE), by the DFG as part of project E3 within the CRC 1119 CROSSING, by the German Federal Ministry of Education and Research (BMBF) within EC SPRIDE, and by the Hessian LOEWE excellence initiative within CASED.

References

ABH10. Martin R. Albrecht, Gregory V. Bard, and William Hart. Algorithm 898: Efficient multiplication of dense matrices over GF(2). *ACM Transactions on Mathematical Software*, 37(1), 2010.

AGR+16. Martin Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. Cryptology ePrint Archive, Report 2016/492, 2016. `http://eprint.iacr.org/`.

AIK06. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

AL07. Yonatan Aumann and Yehuda Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. In *Theory of Cryptography Conference (TCC)*, volume 4392 of *LNCS*, pages 137–156. Springer, 2007.

ALSZ13. Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *Computer and Communications Security (CCS)*, pages 535–548. ACM, 2013. Code: `http://github.com/encryptogroup/OTExtension`.

ARS+15. Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *Advances in Cryptology – EUROCRYPT*, volume 9056 of *LNCS*, pages 430–454. Springer, 2015.

BC86. Gilles Brassard and Claude Crépeau. Zero-knowledge simulation of Boolean circuits. In *Advances in Cryptology – CRYPTO*, volume 263 of *LNCS*, pages 223–233. Sringer, 1986.

BCC11. Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and Luffa. In *Fast Software Encryption (FSE)*, volume 6733 of *LNCS*, pages 252–269. Springer, 2011.

BCDa15. Luís T. A. N. Brandão, Nicolas Christin, George Danezis, and anonymous. Toward Mending Two Nation-Scale Brokered Identification Systems. *PoPETs*, 2015(2):135–155, 2015.

BCG+12. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, 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 *Advances in Cryptology – ASIACRYPT*, volume 7658 of *LNCS*, pages 208–225. Springer, 2012.

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 *Advances in Cryptology – EUROCRYPT*, volume 9056 of *LNCS*, pages 315–342. Springer, 2015.

BDJ+10. Stéphane Badel, Nilay Dagtekin, Jorge Nakahara Jr., Khaled Ouafi, Nicolas Reffé, Pouyan Sepehrdad, Petr Susil, and Serge Vaudenay. ARMADILLO: A multi-purpose cryptographic primitive dedicated to hardware. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 6225 of *LNCS*, pages 398–412. Springer, 2010.

BDK02. Eli Biham, Orr Dunkelman, and Nathan Keller. Enhancing differential-linear cryptanalysis. In *Advances in Cryptology – ASIACRYPT*, volume 2501 of *LNCS*, pages 254–266. Springer, 2002.

BDP00. Joan Boyar, Ivan Damgård, and René Peralta. Short non-interactive cryptographic proofs. *Journal of Cryptology*, 13(4):449–472, 12 2000.

BGV11. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:111, 2011.

BHKR13. Mihir Bellare, Viet Tung Hoang, Sriram Keelveedhi, and Phillip Rogaway. Efficient garbling from a fixed-key blockcipher. In *IEEE Symposium on Security and Privacy*, pages 478–492. IEEE, 2013.

BMP13. Joan Boyar, Philip Matthews, and René Peralta. Logic minimization techniques with applications to cryptology. *Journal of Cryptology*, 26(2):280–312, 2013.

BMvT78. Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

BP12. Joan Boyar and René Peralta. A small depth-16 circuit for the AES S-box. In *Information Security and Privacy Conference (SEC)*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 287–298. Springer, 2012.

BPP00. Joan Boyar, René Peralta, and Denis Pochuev. On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$. *Theoretical Computer Science*, 235(1):43–57, 2000.

BS91. Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

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`.

BW99. Alex Biryukov and David Wagner. Slide Attacks. In *Fast Software Encryption (FSE)*, volume 1636 of *LNCS*, pages 245–259. Springer, 1999.

BW00. Alex Biryukov and David Wagner. Advanced Slide Attacks. In *Advances in Cryptology – EUROCRYPT*, volume 1807 of *LNCS*, pages 589–606. Springer, 2000.

Can05. David Canright. A very compact S-box for AES. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 3659 of *LNCS*, pages 441–455. Springer, 2005.

CCF+15. Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. Cryptology ePrint Archive, Report 2015/113, to appear in the Proceedings of FSE 2016, 2015. `http://eprint.iacr.org/`.

CDK09. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 5747 of *LNCS*, pages 272–288. Springer, 2009.

CE85. David Chaum and Jan-Hendrik Evertse. Crytanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers. In *Advances in Cryptology – CRYPTO*, volume 218 of *LNCS*, pages 192–211. Springer, 1985.

CHK+12. Seung G. Choi, Kyung-Wook Hwang, Jonathan Katz, Tal Malkin, and Dan Rubenstein. Secure multi-party computation of Boolean circuits with applications to privacy in on-line marketplaces. In *Cryptographers' Track at the RSA Conference (CT-RSA)*, volume 7178 of *LNCS*, pages 416–432. Springer, 2012. Code: `http://www.ee.columbia.edu/%7ekwhwang/projects/gmw.html`.

CHM11. Nicolas T. Courtois, Daniel Hulme, and Theodosis Mourouzis. Solving circuit optimisation problems in cryptography and cryptanalysis. Cryptology ePrint Archive, Report 2011/475, 2011. `http://eprint.iacr.org/2011/475`.

DBL08. *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *LNCS*. Springer, 2008.

DEM16. Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Higher-order cryptanalysis of LowMC. In *Information Security and Cryptology (ICISC)*, volume 9558 of *LNCS*, pages 87–101. Springer, 2016.

DH77. Whitfield Diffie and Martin E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, June 1977.

DHS14. Yarkin Doroz, Yin Hu, and Berk Sunar. Homomorphic AES evaluation using NTRU. Cryptology ePrint Archive, Report 2014/039, 2014. `http://eprint.iacr.org/2014/039`.

DK10. Ivan Damgård and Marcel Keller. Secure multiparty AES. In *Financial Cryptography (FC)*, volume 6052 of *LNCS*, pages 367–374. Springer, 2010.

DLMW15. Itai Dinur, Yunwen Liu, Willi Meier, and Qingju Wang. Optimized interpolation attacks on LowMC. In *Advances in Cryptology – ASIACRYPT*, volume 9453 of *LNCS*, pages 535–560. Springer, 2015.

DLT14. Ivan Damgård, Rasmus Lauritsen, and Tomas Toft. An empirical study and some improvements of the MiniMac protocol for secure computation. In *Security and Cryptography for Networks (SCN)*, volume 8642 of *LNCS*, pages 398–415. Springer, 2014.

DP08. Christophe De Cannière and Bart Preneel. Trivium. In *New Stream Cipher Designs - The eSTREAM Finalists* [DBL08], pages 244–266.

DPVAR00. Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: Noekeon. In *First Open NESSIE Workshop*, 2000.

DS09. Itai Dinur and Adi Shamir. Cube Attacks on Tweakable Black Box Polynomials. In *Advances in Cryptology – EUROCRYPT*, volume 5479 of *LNCS*, pages 278–299. Springer, 2009.

DSES14a. Yarkin Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar. Toward practical homomorphic evaluation of block ciphers using Prince. In *Financial Cryptography and Data Security (FC) Workshops, BITCOIN and WAHC*, volume 8438 of *LNCS*, pages 208–220. Springer, 2014.

DSES14b. Yarkın Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar. Toward practical homomorphic evaluation of block ciphers using Prince. Cryptology ePrint Archive, Report 2014/233, 2014. `http://eprint.iacr.org/2014/233`, presented at Workshop on Applied Homomorphic Cryptography and Encrypted Computing (WAHC'14).

DSZ15. Daniel Demmler, Thomas Schneider, and Michael Zohner. Aby - a framework for efficient mixed-protocol secure two-party computation. In *Network and Distributed System Security (NDSS)*. The Internet Society, 2015. Code: `https://github.com/encryptogroup/ABY`.

DZ13. Ivan Damgård and Sarah Zakarias. Constant-overhead secure computation of boolean circuits using preprocessing. In *Theory of Cryptology Conference (TCC)*, volume 7785 of *LNCS*, pages 621–641. Springer, 2013.

FIPR05. Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography Conference (TCC)*, volume 3378 of *LNCS*, pages 303–324. Springer, 2005.

FJN+13. Tore K. Frederiksen, Thomas P. Jakobsen, Jesper B. Nielsen, Peter S. Nordholt, and Claudio Orlandi. MiniLEGO: Efficient secure two-party computation from general assumptions. In *Advances in Cryptology – EUROCRYPT*, volume 7881 of *LNCS*, pages 537–556. Springer, 2013.

FJN14. Tore K. Frederiksen, Thomas P. Jakobsen, and Jesper B. Nielsen. Faster maliciously secure two-party computation using the GPU. In *Security and Cryptography for Networks (SCN)*, volume 8642 of *LNCS*, pages 358–379. Springer, 2014.

FLS+10. Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The Skein Hash Function Family. Submission to NIST (Round 3), 2010.

FN13. Tore K. Frederiksen and Jesper B. Nielsen. Fast and maliciously secure two-party computation using the GPU. In *Applied Cryptography and Network Security (ACNS)*, volume 7954 of *LNCS*, pages 339–356. Springer, 2013.

GGNPS13. Benoît Gérard, Vincent Grosso, María Naya-Plasencia, and François-Xavier Standaert. Block ciphers that are easier to mask: How far can we go? In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 8086 of *LNCS*, pages 383–399. Springer, 2013.

GGP10. Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology – CRYPTO*, volume 6223 of *LNCS*, pages 465–482. Springer, 2010.

GHS12a. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology – CRYPTO*, volume 7417 of *LNCS*, pages 850–867. Springer, 2012.

GHS12b. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. Cryptology ePrint Archive, Report 2012/099, 2012. `http://eprint.iacr.org/`.

GLSV14. Vicente Grosso, Gaëtan Leurent, François-Xavier Standaert, and Kerem Varici. LS-designs: Bitslice encryption for efficient masked software implementations. In *Fast Software Encryption (FSE)*, volume 8540 of *LNCS*, pages 18–37. Springer, 2014.

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Symposium on Theory of Computing (STOC)*, pages 218–229. ACM, 1987.

GNPW13. Jian Guo, Ivica Nikolic, Thomas Peyrin, and Lei Wang. Cryptanalysis of Zorro. Cryptology ePrint Archive, Report 2013/713, 2013. `http://eprint.iacr.org/2013/713`.

GRR+16. Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart. Mpc-friendly symmetric key primitives. Cryptology ePrint Archive, Report 2016/542, 2016. `http://eprint.iacr.org/2016/542`.

HEKM11. Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security*. USENIX, 2011.

HJMM08. Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The Grain Family of Stream Ciphers. In *New Stream Cipher Designs - The eSTREAM Finalists* [DBL08], pages 179–190.

HKE13. Yan Huang, Jonathan Katz, and David Evans. Efficient secure two-party computation using symmetric cut-and-choose. In *Advances in Cryptology – CRYPTO*, volume 8043 of *LNCS*, pages 18–35. Springer, 2013.

HKK+14. Yan Huang, Jonathan Katz, Vladimir Kolesnikov, Ranjit Kumaresan, and Alex J. Malozemoff. Amortizing garbled circuits. In *Advances in Cryptology – CRYPTO*, volume 8617 of *LNCS*, pages 458–475. Springer, 2014.

HL08. Carmit Hazay and Yehuda Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *Theory of Cryptography Conference (TCC)*, volume 4948 of *LNCS*, pages 155–175. Springer, 2008.

HS13. Shai Halevi and Victor Shoup. Design and implementation of a homomorphic-encryption library. `https://github.com/shaih/HElib/`, 2013.

HS14. Shai Halevi and Victor Shoup. Algorithms in HElib. In *Advances in Cryptology – CRYPTO*, volume 8616 of *LNCS*, pages 554–571. Springer, 2014.

JK97. Thomas Jakobsen and Lars R. Knudsen. The interpolation attack on block ciphers. In *Fast Software Encryption (FSE)*, volume 1267 of *LNCS*, pages 28–40. Springer, 1997.

JKO13. Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi. Zero-knowledge using garbled circuits: How to prove non-algebraic statements efficiently. In *Computer and Communications Security (CCS)*, pages 955–966. ACM, 2013.

KLPR10. Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. PRINTcipher: A block cipher for IC-printing. In *Cryptographic Hardware and Embedded Systems (CHES)*, volume 6225 of *LNCS*, pages 16–32. Springer, 2010.

Knu94. Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption (FSE)*, volume 1008 of *LNCS*, pages 196–211. Springer, 1994.

KS08. Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5126 of *LNCS*, pages 486–498. Springer, 2008.

KSS12. Benjamin Kreuter, Abhi Shelat, and Chih-Hao Shen. Billion-gate secure computation with malicious adversaries. In *USENIX Security*. USENIX, 2012.

LH94. Susan K. Langford and Martin E. Hellman. Differential-Linear Cryptanalysis. In *Advances in Cryptology – CRYPTO*, volume 839 of *LNCS*, pages 17–25. Springer, 1994.

Lin13. Yehuda Lindell. Fast cut-and-choose based protocols for malicious and covert adversaries. In *Advances in Cryptology – CRYPTO*, volume 8043 of *LNCS*, pages 1–17. Springer, 2013.

LN14. Tancrède Lepoint and Michael Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In *Progress in Cryptology – AFRICACRYPT*, volume 8469 of *LNCS*, pages 318–335. Springer, 2014.

LOS14. Enrique Larraia, Emmanuela Orsini, and Nigel P. Smart. Dishonest majority multi-party computation for binary circuits. In *Advances in Cryptology – CRYPTO*, volume 8617 of *LNCS*, pages 495–512. Springer, 2014.

LP07. Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology – EUROCRYPT*, volume 4515 of *LNCS*, pages 52–78. Springer, 2007.

LP11. Yehuda Lindell and Benny Pinkas. Secure two-party computation via cut-and-choose oblivious transfer. In *Theory of Cryptography Conference (TCC)*, volume 6597 of *LNCS*, pages 329–346. Springer, 2011.

LPS08. Yehuda Lindell, Benny Pinkas, and Nigel P. Smart. Implementing two-party computation efficiently with security against malicious adversaries. In *Security and Cryptography for Networks (SCN)*, volume 5229 of *LNCS*, pages 2–20. Springer, 2008.

LR14. Yehuda Lindell and Ben Riva. Cut-and-choose Yao-based secure computation in the online/offline and batch settings. In *Advances in Cryptology – CRYPTO*, volume 8617 of *LNCS*, pages 476–494. Springer, 2014.

LTW13. Sven Laur, Riivo Talviste, and Jan Willemson. From oblivious AES to efficient and secure database join in the multiparty setting. In *Applied Cryptography and Network Security (ACNS)*, volume 7954 of *LNCS*, pages 84–101. Springer, 2013.

Mat93. Mitsuru Matsui. Linear Cryptoanalysis Method for DES Cipher. In *Advances in Cryptology – EUROCRYPT*, volume 765 of *LNCS*, pages 386–397. Springer, 1993.

MJSC16. Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *Advances in Cryptology – EUROCRYPT*, volume 9665 of *LNCS*, pages 311–343. Springer, 2016.

MNPS04. Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay — a secure two-party computation system. In *USENIX Security*, pages 287–302. USENIX, 2004.

MS94. Willi Meier and Othmar Staffelbach. The self-shrinking generator. In *Advances in Cryptology – EUROCRYPT*, volume 950 of *LNCS*, pages 205–214. Springer, 1994.

MS13. Silvia Mella and Ruggero Susella. On the homomorphic computation of symmetric cryptographic primitives. In *Cryptography and Coding*, volume 8308 of *LNCS*, pages 28–44. Springer, 2013.

MV12. Eric Miles and Emanuele Viola. Substitution-Permutation Networks, Pseudorandom Functions, and Natural Proofs. In *Advances in Cryptology – CRYPTO*, volume 7417 of *LNCS*, pages 68–85. Springer, 2012.

NLV11. Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Cloud Computing Security Workshop (CCSW)*, pages 113–124. ACM, 2011.

NNOB12. Jesper B. Nielsen, Peter S. Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In *Advances in Cryptology – CRYPTO*, volume 7417 of *LNCS*, pages 681–700. Springer, 2012.

PRC12. Gilles Piret, Thomas Roche, and Claude Carlet. PICARO - a block cipher allowing efficient higher-order side-channel resistance. In *Applied Cryptography and Network Security (ACNS)*, volume 7341 of *LNCS*, pages 311–328. Springer, 2012.

PSSW09. Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure two-party computation is practical. In *Advances in Cryptology – ASIACRYPT*, volume 5912 of *LNCS*, pages 250–267. Springer, 2009.

RASA14. Shahram Rasoolzadeh, Zahra Ahmadian, Mahmood Salmasizadeh, and Mohammad Reza Aref. Total Break of Zorro using Linear and Differential Attacks. Cryptology ePrint Archive, Report 2014/220, 2014. `http://eprint.iacr.org/2014/220`.

Sol14. Hadi Soleimany. Probabilistic slide cryptanalysis and its applications to LED-64 and Zorro. In *Fast Software Encryption (FSE)*, volume 8540 of *LNCS*, pages 373–389. Springer, 2014.

SS11. Abhi Shelat and Chih-Hao Shen. Two-output secure computation with malicious adversaries. In *Advances in Cryptology – EUROCRYPT*, volume 6632 of *LNCS*, pages 386–405. Springer, 2011.

SS13. Abhi Shelat and Chih-Hao Shen. Fast two-party secure computation with minimal assumptions. In *Computer and Communications Security (CCS)*, pages 523–534. ACM, 2013.

SZ13. Thomas Schneider and Michael Zohner. GMW vs. Yao? Efficient secure two-party computation with low depth circuits. In *Financial Cryptography (FC)*, LNCS, pages 275–292. Springer, 2013.

TS. Stefan Tillich and Nigel Smart. Circuits of basic functions suitable for MPC and FHE. `http://www.cs.bris.ac.uk/Research/CryptographySecurity/MPC/`.

Wag99. David Wagner. The Boomerang Attack. In *Fast Software Encryption (FSE)*, volume 1636 of *LNCS*, pages 156–170. Springer, 1999.

WWGY13. Yanfeng Wang, Wenling Wu, Zhiyuan Guo, and Xiaoli Yu. Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro. Cryptology ePrint Archive, Report 2013/775, 2013. `http://eprint.iacr.org/2013/775`.

Yao86. Andrew C.-C. Yao. How to generate and exchange secrets. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE, 1986.

A LowMC version 0

In the first version of this paper, we proposed a number of parameter sets for LowMC instantiations that turned out to be too optimistic. The formula for deriving the number of rounds needed for security did not yet contain a bound against boomerang attacks which can pose a strong threat due to the partial non-linear layer. We thank Dmitry Khovratovich for pointing that out to us. Apart from this the formula was identical to the one of version 1.

You can find the parameters which we suggested for LowMC version 0 in Table 10.

Table 10. Parameter sets of LowMC version 0 instantiations. LowMC-S are shallow instances minimizing AND-depth. LowMC-W are wide instances minimizing the ANDs per encrypted bit.

sboxes m	keysize k	blocksize n	data d	rounds r	ANDdepth	ANDs per bit	
21	80	128	64	13	13	6.4	LowMC-S with 80-bit PRESENT-like security
21	80	1024	64	28	28	1.72	LowMC-W with 80-bit PRESENT-like security
42	128	256	128	13	13	6.4	LowMC-S with AES-128-like security
42	128	1024	128	19	19	2.33	LowMC-W with AES-128-like security
42	192	256	128	14	14	6.9	LowMC-S with AES-192-like security
42	192	1024	128	20	20	2.46	LowMC-W with AES-192-like security
42	256	256	128	15	15	7.4	LowMC-S with AES-256-like security
42	256	1024	128	21	21	2.58	LowMC-W with AES-256-like security

B LowMC version 1

Version 1 of LowMC was published at EUROCRYPT 2015. While the formula for calculating the rounds was improved compared to version 0, it was still not sufficient. In particular, the bound used for calculating the security against higher-order attacks was insufficient. This has been exploited in two attacks [DLMW15, DEM16]. This has been amended in this version by taking the number of rounds needed for full diffusion into account. Furthermore the number of outer rounds was independent of the key size. The formula used to be

$$r \geq \max(r_{\text{stat}}, r_{\text{bmrg}}, r_{\text{deg}}) + r_{\text{outer}}$$

where $r_{\text{outer}} = r_{\text{stat}}$.

The suggested parameters for LowMC version 1 can be found in Table 11.

Table 11. Parameter sets of LowMC version 1 instantiations. The first set has PRESENT-like security parameters, the second set has AES-like security parameters.

blocksize n	sboxes m	keysize k	data d	rounds r	ANDdepth	ANDs per bit
256	49	80	64	11	11	6.3
256	63	128	128	12	12	8.86

C Table representation of LowMC Sbox

The representation of the Sbox in algebraic normal form is given as

$$(a \oplus bc, a \oplus b \oplus ac, a \oplus b \oplus c \oplus ab)$$

(see also Figure 2). Alternatively it can be given in table form as following:

```
ABC DEF
000 000
001 001
010 011
011 110
100 111
101 100
110 101
111 010
```

D Other attack vectors

Slide attacks and their variants are ruled-out by the choice of random i.e., high Hamming-weight round constants and distinct linear layers [BW00, BW99, Sol14]. Meet-in-the-middle (MITM) attacks and their variants [DH77, CE85] should be thwarted by the larger number of round-key additions. It is straightforward to transfer the proof of the absence good boomerangs to to differential-linear structures [LH94, BDK02]. Integral attacks [LH94] can be formulated as higher-order differential attacks. As shown in Section 5.4, we do not expect such attacks to exist.

E A lower bound for the algebraic degree of an SPN

The following theorem and proof lends heavily from Boura, Canteaut, and De Cannière [BCC11].

Theorem 1. *Let F be a function that corresponds to the parallel application of m balanced d-bit Sboxes and an identity function of width l. Thus F is from $\mathbb{F}_2^{dm+l}$ into $\mathbb{F}_2^{dm+l}$. Let δ_k be the maximal degree of the product of any k output bits of the Sbox. Then for any function G from $\mathbb{F}^{md+l}$ into $\mathbb{F}_2^k$, we have*

$$\deg(G \circ F) \leq \beta m + deg(G)$$

where

$$\beta = \max_{1 \leq i \leq d} (\delta_i - i) \ .$$

Proof. Let π be the product of j output bits of F. Let x_i, $1 \leq i \leq d$ denote the number of Sboxes that give exactly i bits to the product π and let x_0 be the number of bits taken from identity. We then have

$$\deg(\pi) \leq \max_{(x_0,\ldots,x_d)} \sum_{i=1}^{d} \delta_i x_i + x_0 = \sum_{i=1}^{d} \delta_i x_i + j - \sum_{i=1}^{d} i x_i = \sum_{i=1}^{d} (\delta_i - i) x_i + j \leq \beta m + j$$

as $\sum_{i=1}^{d} i x_i + x_0 = j$ and $\sum_{i=1}^{d} x_i \leq m$. □

F Sbox representation of PRESENT with low depth

Let $S(A, B, C, D) = (E, F, G, H)$ be the Sbox of PRESENT. We used Mathematica's `BooleanConvert` function to convert the Sbox of PRESENT into Algebraic Normal Form (ANF). This yields the following Sbox representation which can be represented by 8 AND gates and has ANDdepth 2:

$$\begin{aligned}
E &= A \oplus C \oplus D \oplus BC \oplus ABD \oplus ACD \oplus BCD \oplus 1 \\
F &= A \oplus B \oplus AC \oplus AD \oplus CD \oplus ABD \oplus ACD \oplus 1 \\
G &= A \oplus C \oplus AB \oplus AC \oplus ABD \oplus ACD \oplus BCD \\
H &= A \oplus B \oplus D \oplus BC.
\end{aligned}$$

G Specification

Here we give detailed specifications of the operations of LowMC and its instantiation. A reference implementation of LowMC including functionality to generate instances as described above and a program to compute the number of rounds can be found at `https://github.com/tyti/lowmc`.

Let $\mathbf{w}$ be an n-bit word. We denote by $\mathbf{w}_i$ the $(i+1)$th least significant bit i.e., $\mathbf{w}_0$ is the least significant bit, $\mathbf{w}_1$ is the second least significant bit and so forth. By $\mathbf{w}_{i:i+2}$ we denote the 3-bit substring of $\mathbf{w}$ that starts with $\mathbf{w}_i$ and ends with $\mathbf{w}_{i+2}$. With a superscript we index quantities that are round-dependent e.g., $\textsc{KeyAddition}^t$ is the round key addition in round t.

As before n denotes the block width in bits, m denotes the number of Sboxes in the Sbox layer. The number of rounds is denoted as r, the key length as k. Bold lower letters denote bit strings i.e. elements of $\mathbb{F}_2^n$ or $\mathbb{F}_2^k$, bold capital letters denote matrices over $\mathbb{F}_2$.

The state $\mathbf{s}$ is thus an n-bit quantity, the key $\mathbf{k}$ a k-bit quantity. The round keys $\mathbf{rk}^t$, indexed with $0 \leq t \leq r$, are n-bit quantities. The round constants $\mathbf{b}^t$, indexed with $0 < t \leq r$, are likewise n-bit quantities. The linear layer matrices $\mathbf{LM}^t$, indexed with $0 < t \leq r$, are invertible $n \times n$ matrices over $\mathbb{F}_2$. The round key generation matrices $\mathbf{KM}^t$, indexed with $0 \leq t \leq r$, are $n \times k$ matrices over $\mathbb{F}_2$ of rank $\min(n, k)$.

Encryption

The encryption is now defined through the following functions. Some of the functions depend on the round t, some definitions are given bitwise, depending on bit i. For the definition of the Sbox, see the table in Appendix C. Note though that the bits are counted from the right there.

$$\begin{aligned}
\textsc{Encrypt}(\mathbf{s}) &= \textsc{LowMCRound}^r \circ \cdots \circ \textsc{LowMCRound}^1 \circ \textsc{KeyAddition}^0(\mathbf{s}) \\
\textsc{LowMCRound}^t(\mathbf{s}) &= \textsc{KeyAddition}^t \circ \textsc{ConstantAddition}^t \circ \textsc{LinearLayer}^t \circ \textsc{SboxLayer}(\mathbf{s}) \\
\textsc{SboxLayer}_i(\mathbf{s}) &= \begin{cases} \text{Sbox}(\mathbf{s}_{3p:3p+2})_q \text{ where } p = \lfloor i/3 \rfloor, \;\; q = i \bmod 3 & \text{if } i < 3m \\ \mathbf{s}_i & \text{if } i \geq 3m \end{cases} \\
\textsc{LinearLayer}_i^t(\mathbf{s}) &= \bigoplus_{0 \leq j < n} \mathbf{LM}_{i,j}^t \cdot \mathbf{s}_j \\
\textsc{ConstantAddition}^t(\mathbf{s}) &= \mathbf{s} \oplus \mathbf{b}^t \\
\textsc{KeyAddition}^t(\mathbf{s}) &= \mathbf{s} \oplus \mathbf{rk}^t
\end{aligned}$$

Key schedule

The round keys $\mathbf{rk}^t$ are derived from the main key $\mathbf{k}$ via multiplication with the binary matrices $\mathbf{KM}^t$:

$$\mathbf{rk}_i^t = \bigoplus_{0 \leq j < k} \mathbf{KM}_{i,j}^t \cdot \mathbf{k}_j$$

Instantiation

To use the cipher, the matrices $\mathbf{LM}^t$, used in the linear layer, the round constants $\mathbf{b}^t$ and the matrices $\mathbf{KM}^t$, used in the round key generation, must be specified.

In principal any sufficiently random instantiation method is suitable for the instantiation of the matrices and the constants. As a middle ground between accountability, randomness and ease of implementation, we chose the LFSR from the Grain cipher [HJMM08] as self-shrinking generator [MS94]. The LFSR generates a stream of random bits a_i via the following recursion relation

$$a_{i+80} = a_{i+62} \oplus a_{i+51} \oplus a_{i+38} \oplus a_{i+23} \oplus a_{i+13} \oplus a_i.$$

The initial 80 bits a_0 to a_{79} are all set to 1. The first 160 bits generated from this starting state are discarded, as is done in Grain as well. Thus the first bit that is used is a_{240}. To use the LFSR as a self-shrinking generator, the output bits of the LFST are evaluated in pairs. When the first bit is equal to 1, the second bit is produced. When the first bit is 0 though, the second bit is just discarded.

This stream of bits is then used to generate the $\mathbf{LM}^t$, the $\mathbf{b}^t$ and the $\mathbf{KM}^t$.

We begin by creating the $\mathbf{LM}^t$. Let $s = 240$. We now fill the matrix with the bits from the bit stream row by row: $\mathbf{LM}^0_{i,j} = a_{s+ni+j}$. Now we check whether the $\mathbf{LM}^0$, we generated this way, is invertible. If it is not, we discard it and redo the same procedure with the next bits from the bit stream. Otherwise we continue to create the next matrix.

Once we have generated all $\mathbf{LM}^t$, we create the round constants $\mathbf{b}^t$. Let a_{s-1} be the last used bit from the bit stream. We then set $\mathbf{b}^0_i = a_{s+i}$ and continue likewise until all round constants are set.

Finally the matrices $\mathbf{KM}^t$ for the key generation are created. Let again a_{s-1} be the last used bit from the bit stream. We then set $\mathbf{KM}^0_{i,j} = a_{s+ni+j}$. If the rank of this matrix is not equal to $\min(n, k)$, we discard it and redo the same procedure with the next bits from the bit stream. Otherwise we continue to create the next matrix in the same manner.