Ciphers for MPC and FHE

Martin R. Albrecht[1(✉)], Christian Rechberger[2], Thomas Schneider[3], Tyge Tiessen[2], and Michael Zohner[3]

[1] Information Security Group, Royal Holloway, University of London, London, UK
martinralbrecht@googlemail.com
[2] Technical University of Denmark, Copenhagen, Denmark
{crec,tyti}@dtu.dk
[3] TU Darmstadt, Darmstadt, Germany
{thomas.schneider,michael.zohner}@ec-spride.de

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 by far. 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 Introduction

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

E. Oswald and M. Fischlin (Eds.): EUROCRYPT 2015, Part I, LNCS 9056, pp. 430–454, 2015.
DOI: 10.1007/978-3-662-46800-5_17

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 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].

While our design approach is not specific to block ciphers but can be equally applied to e.g. hash functions, in this work, we propose block ciphers that are specifically designed for application in MPC and similar contexts. 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 tender 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 [DSES14] it was observed that using ciphers with low ANDdepth is of central importance for efficient evaluations within homomorphic encryption schemes. Therefore, the authors of [DSES14] 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.

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". 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 by far with respect to these metrics. In terms of ANDdepth, the closest competitor is PRINCE. 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

– Low ANDdepth, and low MC, which positively impacts the latency and throughput of the FHE, MPC, or ZK evaluation of the cipher.

- Partial Sbox layer.
- Security arguments against large classes of statistical attacks like differential attacks, similar to other state-of-the-art designs are given in Section 5. 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,BODD+14], 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.
- In contrast to 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.
- We explicitly de-couple the security claim of a block cipher from the block size.

2 Schemes

In this section we list several schemesfor MPC, FHE, and ZK that benefit from evaluating our cipher. We give a list of example applications for LowMC in the full version of the paper.

2.1 Multi-Party Computation (MPC)

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.

2.2 Fully Homomorphic Encryption (FHE)

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.

2.3 Zero-Knowledge Proof of Knowledge (ZK)

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].

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[1]. The number of rounds needed to reach the security claims is then derived from these parameters.

To reduce the MC, 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], we will show that security is viable. To reach security in spite of a low MC, 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.

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:

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

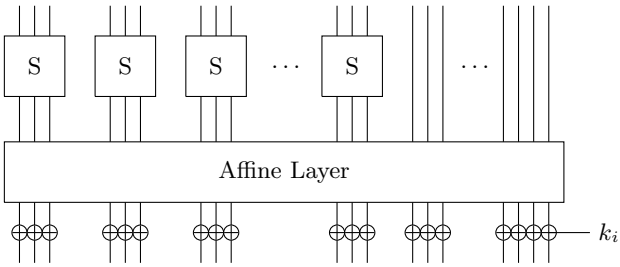

Fig. 1. Depiction of one round of encryption with LowMC

- Maximum differential probability: 2^{-2}
- Maximum linear probability: 2^{-2}
- Simple circuit description involving MC = 3 AND gates, with ANDdepth=1
- Each of the 8 non-zero component functions has algebraic degree 2

The Sbox is specified in 2, and coincides with the Sbox used for PRINTcipher [KLPR10]. Other representations of the Sbox can be found in the full version of this paper.

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.

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.

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 $\min(n, k)$.

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

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

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),state)
  end
  ciphertext = state
```

3.2 Parameters

Our security analysis against differential, linear, higher-order, meet-in-the-middle, algebraic, and slide attacks suggests that, except with negligible probability, any uniformly randomly chosen set of matrices leads to a secure construction for the parameters given in Table 1. For a larger selection of parameters bundled with security bounds, see the full version of this paper.

Table 1. Parameter sets of LowMC instantiations. One 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

3.3 Instantiation of LowMC

To maximize the amount of diffusion done 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] and [MS94]) as a source of random bits. The exact procedure can be found in the full version of this paper.

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.

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 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 10 AND gates and ANDdepth 2, cf. [DSES14]. 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 number of rounds r equals ANDdepth, and is hence a crucial factor to minimize. For this we evaluate the security of the construction against an array of known attack vectors. Below we especially discuss differential, linear and high-order attacks, as their analysis is a relevant technical contribution in itself. For a short discussion of other attack vectors, we refer to the full version of this paper.

We aim to prove the LowMC designs secure against classes of known attacks. However, due to the choice of random linear layers it is not immediately clear how to bound the probability of differential or linear characteristics. This is something we will investigate and resolve in Section 5.1. Due to the extremely simple description of the Sbox, higher order [Knu94] and cube attacks [DS09] that exploit a relatively slow growth in the algebraic degree appear to be the most promising attack vector, and are studied in Section 5.4. The quality of these bounds is tested on small versions of LowMC. This all will allow us to formulate in Section 5.6 a relatively simple expression for deriving a lower bound for the number of rounds given other parameters like the desired security level in terms of time and data, and block size.

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.

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	**12**	8.85	full version
Lightweight security						
PrintCipher-96	160	96	96	96	96	full version
PrintCipher-48	80	48	48	48	48	full version
Present	80 or 128	64	64	62 (93)	62 (31)	full version ([CHM11])
Simon	96	64	64	42	21	[BSS+13]
Simon	64	32	32	32	16	[BSS+13]
Prince	128	64	64	24	30	[DSES14]
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	11	**6.31**	full version

5.1 Differential Characteristics

In differential attacks, 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" difference 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. The basic idea behind the proof is the following. We calculate for each possible good differential characteristic 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. We then show that the sum of these probabilities, which is an upper bound for the probability that any good characteristic exists, is negligible.

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 a difference that activates 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

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

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 probability larger than 0 is

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

Let $(\alpha_0, \alpha_1, \ldots, \alpha_r)$ now be a given characteristic over r rounds where the differences α_i are at the end 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 probability larger than 0 in a random instantiation of LowMC as

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

Summing now over all possible characteristics over r rounds that activate at most d Sboxes, we can calculate an upper bound for the probability that there exists an r-round characteristic with d or fewer active Sboxes as

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

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 no good differentials are present except for 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. 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 linear probability of the 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 half of the cipher and another that covers the second half. If both have about the same probability, the complexity corresponds roughly to the inverse of the fourth power of this probability [Wag99]. Thus to calculate the number of rounds sufficient to make sure that no boomerang exists, we calculate the number of rounds after which differential characteristics of probability $2^{-d/4}$ exist only with negligible probability and then double this number.

5.4 Higher Order Attacks

Due to its small size, 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 high-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 et al. [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 Boura et al. [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 the full version of this paper. 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*

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

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.

5.5 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

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 5.

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

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[2] is tight for this version when compared with the theoretic upper bound as determined with equation 5. More experimental cryptanalysis can be found in the full version of this paper.

5.6 Fixing the Number of Rounds

We base our recommendation for the number of rounds on the following:

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

[2] 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 5. p_{stat} is the probability that a differential or linear characteristic of probability at least 2^{-12} exists (see eq. 4).

(a) $n = 24$, $m = 4$, $k = 12$, $d = 12$

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	-	-	-	-	-	$2^{-101.5}$

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, and where r_{deg} indicates the number of rounds needed for the cipher to have sufficient degree 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 $r_{\text{outer}} = r_{\text{stat}}$.

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 precomputed round keys. We use the GMW implementation that is available in the ABY-framework [DSZ15] which uses the efficient oblivious transfer extensions of [ALSZ13][3]. 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.

[3] Our MPC implementations of the benchmarked block-ciphers are available online as part of the ABY-framework https://github.com/encryptogroup/ABY.

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

Lightweight Security						
Cipher	Present		Simon		LowMC	
Communication [kB]	39		**26**		51	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	0.003	0.21	**0.002**	0.21	**0.002**	**0.14**
Online [s]	**0.05**	13.86	**0.05**	5.34	0.06	**1.46**
Total [s]	**0.05**	14.07	**0.05**	5.45	0.06	**1.61**
Long-Term Security						
Cipher	AES		Simon		LowMC	
Communication [kB]	170		136		**72**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	0.01	0.27	0.009	0.23	**0.002**	**0.15**
Online [s]	**0.04**	4.08	0.05	6.95	0.07	**1.87**
Total [s]	**0.05**	4.35	0.06	7.18	0.07	**2.02**

Single-Block Results. From our single-block experiments in Table 6 we see that the communication of LowMC is higher by factor 2 compared to the lightweight security ciphers but lower by factor 2 compared to the long-term security ciphers. In terms of total runtime, for lightweight security LowMC performs similar to Present and Simon in the LAN setting and outperforms both by factor 3 to 9 in the WAN setting. For long-term security AES is slightly faster than LowMC in the LAN setting, but slower than LowMC in the WAN setting by factor 2. 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.

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 2 for lightweight security and factor 4 for long-term security. Also the total runtime of LowMC is the lowest among all ciphers, ranging from factor 6 when compared to Simon for lightweight security to factor 9 when compared to AES for long-term security.

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 comparable or even slightly 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 6 to 9. For secure computation protocols with security against malicious adversaries, the benefit of using LowMC would be even more significant, since

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

Lightweight Security						
Cipher	Present		Simon		LowMC	
Comm. [GB]	7.4		5.0		**2.5**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	214.17	453.89	268.93	568.35	**43.33**	**138.63**
Online [s]	2.71	34.35	3.29	37.06	**2.02**	**17.12**
Total [s]	216.88	488.24	272.22	605.41	**45.36**	**155.75**

Long-Term Security						
Cipher	AES		Simon		LowMC	
Comm. [GB]	16		13		**3.5**	
Runtime	LAN	WAN	LAN	WAN	LAN	WAN
Setup [s]	553.41	914.27	444.30	727.48	**62.01**	**193.90**
Online [s]	2.50	33.52	2.97	34.42	**2.36**	**21.11**
Total [s]	555.91	947.79	447.27	761.90	**64.37**	**215.01**

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.

Table 8. LowMC (commit `f6a086e`) in HElib [HS13] (commit `e9d3785e`) 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 in seconds, t_{sbox} the total time spent in the S-Box layer in seconds, t_{key} the total time spent in the key schedule in seconds, $t_{block} = t_{eval}/\#\text{blocks}$ and $t_{bit} = t_{block}/n$. The rows marked as "main" contain the main parameter proposals. The rows marked as "perf", "cons" or "smll" contain alternative parameter sets being conservative, performance oriented or relatively small respectively.

d	m	r	n	#blocks	t_{setup}	t_{eval}	t_{sbox}	t_{key}	t_{block}	t_{bit}	Memory	Comment
128	63	12	256	600	11.6	506.1	353.2	1.6	0.8434	0.0033	1.58GB	main
128	86	11	512	600	11.7	847.6	451.5	3.2	1.4127	0.0028	2.62GB	perf
128	86	12	512	600	11.7	893.9	480.1	3.2	1.4898	0.0029	2.62GB	cons
64	49	11	256	600	11.0	383.0	206.3	0.9	0.6383	0.0025	1.52GB	main
64	49	10	256	600	11.5	305.6	255.6	1.1	0.5093	0.0020	1.37GB	perf
64	34	11	128	600	13.0	260.7	204.0	0.7	0.4345	0.0034	1.08GB	smll

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.

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.

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	12	600	8m	0.8s	0.0033s	LowMC	this work	included
64	24	1024	57m	3.3s	0.0520s	PRINCE	[DSES14]	excluded
64	11	600	6.4m	0.64s	0.0025s	LowMC	this work	included

7 Conclusions, Lessons Learned, and Open Problems

We proposed block ciphers with an extremely small number of AND gates and an extremely shallow AND depth, demonstrated the soundness of our design through experimental evidence and provided a security 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 familiy, 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. Is it possible to reduce the number of rounds in LowMC further, which in turn would further reduce MC and ANDdepth? Analyzing such an extreme corner of the design space for a symmetric cipher is an interesting endavor in itself. 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.

Currently, the MC 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 including the recent lightweight cipher proposals 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 ANDdepth under such constraints?

Acknowledgments. We thank Dmitry Khovratovich for pointing to us out that an earlier version of our parameter sets for LowMC instantiations are too optimistic. See the full version of this paper for details. 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 7$^{\text{th}}$ 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] Albrecht, M.R., Bard, G.V., Hart, W.: Algorithm 898: Efficient multiplication of dense matrices over GF(2). ACM Transactions on Mathematical Software **37**(1) (2010)

[AIK06] Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. SIAM Journal on Computing **36**(4), 845–888 (2006)

[AL07] Aumann, Y., Lindell, Y.: Security against covert adversaries: efficient protocols for realistic adversaries. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 137–156. Springer, Heidelberg (2007)

[ALSZ13] Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer and extensions for faster secure computation. In: Computer and Communications Security (CCS), pp. 535–548. ACM (2013). Code: http://github.com/MichaelZohner/OTExtension

[BC86] Brassard, G., Crépeau, C.: Zero-knowledge simulation of boolean circuits. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 223–233. Springer, Heidelberg (1987)

[BCC11] Boura, C., Canteaut, A., De Cannière, C.: Higher-order differential properties of KECCAK and *Luffa*. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 252–269. Springer, Heidelberg (2011)

[BCG+12] Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)

[BDP00] Boyar, J., Damgård, I., Peralta, R.: Short non-interactive cryptographic proofs. Journal of Cryptology **13**(4), 449–472 (2000)

[BGV11] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: Fully homomorphic encryption without bootstrapping. Electronic Colloquium on Computational Complexity (ECCC) **18**, 111 (2011)

[BMP13] Boyar, J., Matthews, P., Peralta, R.: Logic minimization techniques with applications to cryptology. Journal of Cryptology **26**(2), 280–312 (2013)

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

[BODD+14] Bar-On, A., Dinur, I., Dunkelman, O., Lallemand, V., Keller, N., Tsaban, B.: Cryptanalysis of SP Networks with Partial Non-Linear Layers. In: Cryptology ePrint Archive, Report 2014/228 (2014). http://eprint.iacr.org/2014/228

[BP12] Boyar, J., Peralta, R.: A small depth-16 circuit for the AES S-box. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012. IFIP AICT, vol. 376, pp. 287–298. Springer, Heidelberg (2012)

[BPP00] Boyar, J., Peralta, R., Pochuev, D.: On the multiplicative complexity of Boolean functions over the basis $(\wedge, \oplus, 1)$. Theoretical Computer Science **235**(1), 43–57 (2000)

[BSS+13] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. In: Cryptology ePrint Archive, Report 2013/404 (2013). http://eprint.iacr.org/2013/404

[Can05] Canright, D.: A very compact S-box for AES. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 441–455. Springer, Heidelberg (2005)

[CDK09] De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)

[CHK+12] Choi, S.G., Hwang, K.-W., Katz, J., Malkin, T., Rubenstein, D.: Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 416–432. Springer, Heidelberg (2012). http://www.ee.columbia.edu/~kwhwang/projects/gmw.html

[CHM11] Courtois, N.T., Hulme, D., Mourouzis, T.: Solving circuit optimisation problems in cryptography and cryptanalysis. In: Cryptology ePrint Archive, Report 2011/475 (2011). http://eprint.iacr.org/2011/475

[DHS14] Doroz, Y., Hu, Y., Sunar, B.: Homomorphic AES evaluation using NTRU. In: Cryptology ePrint Archive, Report 2014/039 (2014). http://eprint.iacr.org/2014/039

[DLT14] Damgård, I., Lauritsen, R., Toft, T.: An empirical study and some improvements of the minimac protocol for secure computation. In: Abdalla, M., De Prisco, R. (eds.) SCN 2014. LNCS, vol. 8642, pp. 398–415. Springer, Heidelberg (2014)

[DPVAR00] Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: Nessie proposal: Noekeon. In: First Open NESSIE Workshop (2000)

[DS09] Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009)

[DSES14] Doröz, Y., Shahverdi, A., Eisenbarth, T., Sunar, B.: Toward practical homomorphic evaluation of block ciphers using Prince. In: Cryptology ePrint Archive, Report 2014/233 (2014), presented at Workshop on Applied Homomorphic Cryptography and Encrypted Computing (WAHC 2014). http://eprint.iacr.org/2014/233

[DSZ15] Demmler, D., Schneider, T., Zohner, M.: Aby - a framework for efficient mixed-protocol secure two-party computation. In: Network and Distributed System Security, NDSS 2015. The Internet Society (2015). Code: https://github.com/encryptogroup/ABY

[DZ13] Damgård, I., Zakarias, S.: Constant-overhead secure computation of boolean circuits using preprocessing. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 621–641. Springer, Heidelberg (2013)

[FJN+13] Frederiksen, T.K., Jakobsen, T.P., Nielsen, J.B., Nordholt, P.S., Orlandi, C.: MiniLEGO: efficient secure two-party computation from general assumptions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 537–556. Springer, Heidelberg (2013)

[FJN14] Frederiksen, T.K., Jakobsen, T.P., Nielsen, J.B.: Faster maliciously secure two-party computation using the GPU. In: Abdalla, M., De Prisco, R. (eds.) SCN 2014. LNCS, vol. 8642, pp. 358–379. Springer, Heidelberg (2014)

[FLS+10] Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein Hash Function Family. Submission to NIST (Round 3) (2010)

[FN13] Frederiksen, T.K., Nielsen, J.B.: Fast and maliciously secure two-party computation using the GPU. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 339–356. Springer, Heidelberg (2013)

[GGNPS13] Gérard, B., Grosso, V., Naya-Plasencia, M., Standaert, F.-X.: Block ciphers that are easier to mask: how far can we go? In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 383–399. Springer, Heidelberg (2013)

[GHS12a] Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012)

[GHS12b] Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the aes circuit. In: Cryptology ePrint Archive, Report 2012/099 (2012). http://eprint.iacr.org/2012/099

[GLSV14] Grosso, V., Leurent, G., Standaert, F.-X., Varici, K.: LS-designs: Bitslice encryption for efficient masked software implementations. In: FSE 2014. LNCS, vol. 8540. Springer, Heidelberg (2015)

[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Symposium on Theory of Computing (STOC), pp. 218–229. ACM (1987)

[GNPW13] Guo, J., Nikolic, I., Peyrin, T., Wang, L.: Cryptanalysis of Zorro. In: Cryptology ePrint Archive, Report 2013/713 (2013). http://eprint.iacr.org/2013/713

[HEKM11] Huang, Y., Evans, D., Katz, J., Malka, L.: Faster secure two-party computation using garbled circuits. In: USENIX Security. USENIX (2011)

[HJMM08] Hell, M., Johansson, T., Maximov, A., Meier, W.: The grain family of stream ciphers. In: Robshaw, M., Billet, O. (eds.) New Stream Cipher Designs. LNCS, vol. 4986, pp. 179–190. Springer, Heidelberg (2008)

[HKE13] Huang, Y., Katz, J., Evans, D.: Efficient secure two-party computation using symmetric cut-and-choose. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 18–35. Springer, Heidelberg (2013)

[HKK+14] Huang, Y., Katz, J., Kolesnikov, V., Kumaresan, R., Malozemoff, A.J.: Amortizing garbled circuits. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 458–475. Springer, Heidelberg (2014)

[HS13] Halevi, S., Shoup, V.: Design and implementation of a homomorphic-encryption library (2013). https://github.com/shaih/HElib/

[HS14] Halevi, S., Shoup, V.: Algorithms in HElib. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 554–571. Springer, Heidelberg (2014)

[JKO13] Jawurek, M., Kerschbaum, F., Orlandi, C.: Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In: Computer and Communications Security (CCS), pp. 955–966. ACM (2013)

[KLPR10] Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: a block cipher for IC-printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)

[Knu94] Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)

[KS08] Kolesnikov, V., Schneider, T.: Improved garbled circuit: free XOR gates and applications. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 486–498. Springer, Heidelberg (2008)

[KSS12] Kreuter, B., Shelat, A., Shen, C.-H.: Billion-gate secure computation with malicious adversaries. In: USENIX Security. USENIX (2012)

[Lin13] Lindell, Y.: Fast cut-and-choose based protocols for malicious and covert adversaries. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 1–17. Springer, Heidelberg (2013)

[LOS14] Larraia, E., Orsini, E., Smart, N.P.: Dishonest majority multi-party computation for binary circuits. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 495–512. Springer, Heidelberg (2014)

[LP07] Lindell, Y., Pinkas, B.: An efficient protocol for secure two-party computation in the presence of malicious adversaries. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 52–78. Springer, Heidelberg (2007)

[LP11] Lindell, Y., Pinkas, B.: Secure two-party computation via cut-and-choose oblivious transfer. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 329–346. Springer, Heidelberg (2011)

[LPS08] Lindell, Y., Pinkas, B., Smart, N.P.: Implementing two-party computation efficiently with security against malicious adversaries. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 2–20. Springer, Heidelberg (2008)

[LR14] Lindell, Y., Riva, B.: Cut-and-choose yao-based secure computation in the online/offline and batch settings. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 476–494. Springer, Heidelberg (2014)

[Mat93] Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)

[MNPS04] Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay – a secure two-party computation system. In: USENIX Security, pp. 287–302. USENIX (2004)

[MS94] Meier, W., Staffelbach, O.: The self-shrinking generator. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 205–214. Springer, Heidelberg (1995)

[MS13] Mella, S., Susella, R.: On the homomorphic computation of symmetric cryptographic primitives. In: Stam, M. (ed.) IMACC 2013. LNCS, vol. 8308, pp. 28–44. Springer, Heidelberg (2013)

[NNOB12] Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 681–700. Springer, Heidelberg (2012)

[PRC12] Piret, G., Roche, T., Carlet, C.: PICARO – a block cipher allowing efficient higher-order side-channel resistance. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 311–328. Springer, Heidelberg (2012)

[PSSW09] Pinkas, B., Schneider, T., Smart, N.P., Williams, S.C.: Secure two-party computation is practical. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 250–267. Springer, Heidelberg (2009)

[RASA14] Rasoolzadeh, S., Ahmadian, Z., Salmasizadeh, M., Aref, M.R.: Total Break of Zorro using Linear and Differential Attacks. In: Cryptology ePrint Archive, Report 2014/220 (2014). http://eprint.iacr.org/2014/220

[SS11] Shelat, A., Shen, C.-H.: Two-Output Secure Computation with Malicious Adversaries. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 386–405. Springer, Heidelberg (2011)

[SS13] Shelat, A., Shen, C.-H.: Fast two-party secure computation with minimal assumptions. In: Computer and Communications Security (CCS), pp. 523–534. ACM (2013)

[SZ13] Schneider, T., Zohner, M.: GMW vs. Yao? Efficient secure two-party computation with low depth circuits. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 275–292. Springer, Heidelberg (2013)

[TS] Tillich, S., Smart, N.: Circuits of basic functions suitable for MPC and FHE. http://www.cs.bris.ac.uk/Research/CryptographySecurity/MPC/

[Wag99] Wagner, D.: The boomerang attack. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (1999)

[WWGY13] Wang, Y., Wu, W., Guo, Z., Yu, X.: Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro. In: Cryptology ePrint Archive, Report 2013/775 (2013). http://eprint.iacr.org/2013/775

[Yao86] Yao, A.C.-C.: How to generate and exchange secrets. In: IEEE Symposium on Foundations of Computer Science (FOCS), pp. 162–167. IEEE (1986)