Improving Block Cipher Design by Rearranging Internal Operations

Liran Lerman, Jorge Nakahara Jr. and Nikita Veshchikov
Université Libre de Bruxelles (ULB), Dept. d'Informatique, Brussels, Belgium

Keywords: Block Cipher Design, Security And Performance Analysis, Rearranging Internal Operations.

Abstract: This paper discusses the impact of a simple strategy in block cipher design: *rearranging the internal cipher components*. We report on a test case in which we observed a significant upgrade on a cipher's security. We applied this approach in practice and report on an updated design of the IDEA block cipher, in which we swapped all exclusive-or operations for multiplications. The consequences of these modifications are far reaching: there are no more weak multiplicative subkeys (because multiplications are not keyed anymore) and overall diffusion improves sharply in the encryption framework. The *unkeyed multiplication* is novel in itself since it did not exist in IDEA as a primitive operation and it alone guarantees stronger diffusion than the exclusive-or operation. Moreover, our analysis so far indicate that the new cipher resists better than IDEA and AES against old and new attacks such as the recent biclique technique and the combined Biryukov-Demirci meet-in-the-middle attack. Experiments on an 8-bit microcontroller indicate the new design has about the same performance as IDEA. A theoretical analysis also suggests the new design is more resistant to power analysis than IDEA.

1 INTRODUCTION

The main motivation for this research came from a simple question: how the order of internal cipher components affects its security? Our investigations shed some light on (undocumented) design decisions that are not always provided with every announcement of new cryptographic primitives.

Previous work includes reordering the S-boxes in the DES cipher (Matsui, 1995). The conclusions were that some S-box orderings, in fact, would considerably weaken the security against differential and linear cryptanalysis. This means that the order of S-boxes could serve as a potential trapdoor. Therefore, not only the cipher components are relevant for security, but also the order in which they are applied.

As a concrete instantiation, we analysed what happens in the International Data Encryption Algorithm (IDEA) (Lai et al., 1991) block cipher if the exclusive-or and modular multiplication were swapped. This modified design might be of independent theoretical interest.

In (Borisov et al., 2002), Borisov *et al.* described a modified IDEA cipher in which some $\oplus$ operations were swapped with $\boxplus$. This modified cipher was called IDEA-X. The objective was to have an appropriate target for their multiplicative-differential attack, since this attack did not affect IDEA. In (NakaharaJr, 2009), Nakahara studied different reorderings of the four round transformations in AES, but no security threat was detected compared to the original ordering in the AES. IDEA was released before the NIST competition for the Advanced Encryption Standard (AES) (FIPS197, 2001). Even nowadays, there are still novel analyses (Biham et al., 1417; Wei et al., 2012) against IDEA. IDEA provided a formidable and challenging testing ground for all kinds of cryptanalytic techniques, already at a time when DES was the prevailing benchmark. Nowadays, AES is the *de facto* world standard. The recent biclique technique effectively reach the full round versions of the AES, IDEA and PRESENT ciphers (Bogdanov et al., 2011; Khovratovich et al., 2012; Abed et al., 2012) with (time) complexity less than exhaustive key search in the single-key model. Also, Biham *et al.* (Biham et al., 1417) independently attacked the full IDEA using the Biryukov-Demirci relation and a meet-in-the-middle approach. Several other attacks also exploited weaknesses in the key schedule such as (Biryukov et al., 2002; Borst et al., 1997; Daemen et al., 1993; Hawkes, 1998) to attack the encryption framework.

This paper is organized as follows: Sect. 2 lists the main contributions of this research; in Sect. 3 we concretely instantiate our strategy of rearranging internal cipher components. We apply this approach to the IDEA block cipher; Sect. 4 motivates and describes a

Lerman L., Nakahara Jr J. and Veshchikov N..
Improving Block Cipher Design by Rearranging Internal Operations.
DOI: 10.5220/0004498200270038
In *Proceedings of the 10th International Conference on Security and Cryptography* (SECRYPT-2013), pages 27-38
ISBN: 978-989-8565-73-0

new key schedule; Sect. 5 provides security analyses; Sect. 6 concludes the work.

2 CONTRIBUTIONS

The *contributions* of this work are manifold:

- the *focus* of this paper is to assess the consequences of a simple design strategy: how the rearrangement of internal cipher components affects its security (and performance). As an example, we analysed what happens in the IDEA block cipher if we swap all the exclusive-or (denoted $\oplus$) and multiplication (denoted $\odot$) operations. We call the new design IDEA*. This simple modification has not been reported before, which may be of independent interest. In IDEA*, subkeys are no longer a mandatory input to the multiplication operations, meaning that both inputs are variable. IDEA* also uses the unkeyed division operation, denoted $\boxdot$, so that $a \boxdot b = a \odot b^{-1} = a/b$, where $a, b \in \mathrm{GF}(2^{16}+1)$. Therefore, if a table of multiplicative inverses is provided, a division costs one multiplication plus a table look-up. Note that unlike exclusive-or, $a \boxdot b \neq b \boxdot a$, so the order of the operands matters in $\boxdot$. In the rest of this paper, we discuss the many implications of swapping $\oplus$ by $\odot$ in IDEA.
- IDEA* employs the same three algebraic operations of IDEA which means application environments that already use IDEA can adopt IDEA* without major changes in infrastructure. Consequently, IDEA* fits in the same legacy environments as used by IDEA, such as PGP/GPG, digital rights management, video scrambling for pay-TV, internet audio/video distribution, government and corporate IT infrastructure protection.
- *The unkeyed multiplication is a new primitive operation and has a considerable impact:* there are no more weak multiplicative subkeys in IDEA* regardless of the key schedule algorithm. Moreover, wordwise diffusion is stronger with multiplication because of a wrap-around effect in comparison to the bitwise diffusion in exclusive-or. This fact is corroborated in Lai's Low-High algorithm (Lai, 1992) for multiplication in $\mathrm{GF}(2^{16}+1)$. Note that swapping $\oplus$ with $\boxplus$ would not eliminate weak subkeys, as subkeys would still be a mandatory input to $\odot$. This modified version was called IDEA-X by Borisov *et al.* (Borisov et al., 2002). They showed multiplicative differential attacks on IDEA-X, which do not affect IDEA. Likewise, swapping $\boxplus$ for $\odot$ would not work either, because subkeys would still be input to $\odot$ in the the first half-rounds.
- We suggest an updated key schedule for IDEA* with full key diffusion after the third generated subkey, which makes each round equally strong, since the subkeys quickly depend on all bits of the user key. This design was borrowed from (Nakahara.Jr et al., 2003b) and effectively counters meet-in-the-middle (MITM), related-key, slide and advanced slide (among other) attacks. This means that the encryption framework cannot be purposefully weakened due to particular bit patterns in the key. Comparatively, in IDEA, different rounds do not have the same strength because subkey bits do not overlap, and the total key entropy per round can be much lower than 96 bits. In IDEA*, individual key bits cannot be flipped independently without affecting several subkeys at once, thus hindering divide-and-conquer attacks that try to exploit independent subkey bits such as the biclique technique.
- Swapping $\oplus$ for $\odot$ makes differential power analysis theoretically more difficult against IDEA* than IDEA as the former operation is more side-channel resistant than the latter. Additionally, IDEA*'s key schedule counters simple power analysis due to its elaborated structure that do not allow the internal instructions of a physical implementation to be straightforwardly analysed through power traces, for instance.
- Our analyses indicate that IDEA* better resists previous attacks than IDEA, including the recent biclique technique (Sect. 5.4) and the meet-in-the-middle Biryukov-Demirci (Sect. 5.2). In order to have a fair security comparison, we suggest the number of rounds in IDEA* to be 6.5 instead of 8.5 as originally in IDEA, since the number of modular multiplications becomes approximately the same. IDEA* uses six multiplications/divisions per round while IDEA uses four multiplciations per round. This means that 6.5-round IDEA* (with 36 $\odot$'s) shall provide the same strenght as 8.5-round IDEA (34 $\odot$'s).

3 THE IDEA* BLOCK CIPHER

The International Data Encryption Algorithm (IDEA) is a block cipher designed by Lai and Massey (Lai et al., 1991) based on a previous design called PES (Lai and Massey, 1990). IDEA operates on a 64-bit state, uses a 128-bit key and iterates 8.5 rounds (Fig. 1). A main feature of IDEA is the combination of three group operations on 16-bit words: addition in $\mathbb{Z}_{2^{16}}$ ($\boxplus$), exclusive-or (xor) ($\oplus$) and multiplication ($\odot$) in GF($2^{16}+1$) with $0 \equiv 2^{16}$. Mixing incompatible group operations, in the sense that they satisfy neither associativity nor distributivity rules, is responsible for the confusion property (Lai, 1992) in accordance with Shannon's seminal work (Shannon, 1949). This is not a unique feature of IDEA. Ciphers such as RC5 (Menezes et al., 1997) and HIGHT (Hong et al., 2006) use only *A*ddition-*R*otation-*X*or operations, which led to the terminology of ARX designs. In this setting, IDEA could be called an AMX (Addition-Mult-Xor) cipher. IDEA's design follows the Lai-Massey scheme and is not a Feistel nor a Substitution Permutation Network (SPN) scheme and therefore adds diversity to the portfolio of block cipher frameworks.

IDEA* preserves the wordwise structure and the same group operations in IDEA as well as the design philosophy of repeating a strong round structure a small number of times, instead of iterating a weak round function a large number of times. IDEA* also adopted the design feature of never repeating the same group operation two or more times during the encryption/decryption frameworks (Lai, 1992) and there is full (text) diffusion after a single round. Note that complete diffusion in IDEA and IDEA* is achieved in a single round.

The original MA-box (with Multiplication and Addition) in IDEA becomes an AX-box in IDEA* (with Addition and Xor). One full encryption round in IDEA* consists of two half rounds: key-whitening (KW) and AX (Fig. 2). The KW half-round simply adds or xors the j-th subkey of the i-th round $Z_j^{(i)}$, $1 \leq i \leq 6$, $1 \leq j \leq 9$, to each 16-bit word of the input. A text block (a,b,c,d) becomes (A,B,C,D) = $(a \oplus Z_1^{(i)},\ b \boxplus Z_2^{(i)},\ c \boxplus Z_3^{(i)},\ d \oplus Z_4^{(i)})$. Decryption is done by just applying the additive inverse or the xor of the subkeys in the correct order.

The AX half-round contains an AX-box and an almost involutory structure (see Fig. 2 for encryption and Fig. 3 for decryption.). In more detail, the input to the AX-box is $(A \boxdot C,\ B \boxdot D)$. Let (E,F) denote the AX-box output. Then, $F = (((A \odot C^{-1}) \oplus Z_5^{(i)}) \boxplus (B \odot D^{-1})) \oplus Z_6^{(i)}$, and $E = ((A \odot C^{-1}) \oplus Z_5^{(i)}) \boxplus F$. The output of the AX half-round for encryption becomes $(A \odot F,\ C \odot F,\ B \odot E,\ D \odot E)$. For decryption, the AX-box input becomes $(A \odot F \boxdot (C \odot F),\ B \odot E \boxdot (D \odot E))$ = $(A \odot F \odot C^{-1} \odot F^{-1},\ B \odot E \odot D^{-1} \odot E^{-1})$ = $(A \odot C^{-1},\ B \odot D^{-1})$, which is necessary to recreate the same AX-box input as for encryption: (E,F). Decryption proceeds as $(A \odot F \boxdot F,\ B \odot E \boxdot E,\ C \odot F \boxdot F,\ D \odot E \boxdot E)$ = (A,B,C,D). Therefore, the AX half-round is almost its own inverse, except that $\odot$'s are exchanged for $\boxdot$. A novelty in IDEA* is the use of unkeyed $\odot$ i.e. with both operands variable. In IDEA, one operand in every $\odot$ is always a fixed (unknown) subkey, which may weaken the multiplication depending on the subkey value (Daemen et al., 1993; Biryukov et al., 2002). The last half round contains just a key whitening with subkeys $(Z_1^{(7)},\ Z_2^{(7)},\ Z_3^{(7)},\ Z_4^{(7)})$. Notice that $\odot$ has much better diffusion power than $\oplus$ (which is just bitwise). This fact is corroborated by Lai's Low-High algorithm (Lai, 1992) for multiplication in GF$(2^{16}+1)$: let $a,b \in \mathbb{Z}_{2^{16}+1}$, $R = ab \bmod 2^{16}$ and $Q = ab \operatorname{div} 2^{16}$. Then

$$a \odot b = \begin{cases} R - Q, & \text{if } R \geq Q \\ R - Q + 2^{16} + 1, & \text{if } R < Q \end{cases}$$

where R denotes the remainder (”Low” part) and Q denotes the quotient (”High” part) when ab is divided by 2^{16}. It essentially means that the result of $\odot$ depends on all 32 bits of the extended multiplication.

Efficient hardware implementations of IDEA* in terms of speed and area can be performed by using modulo $2^n + 1$ arithmetic for addition and multiplication operations like in IDEA (Zimmernmann, 1999).

4 KEY SCHEDULE OF IDEA*

IDEA* iterates 6.5 rounds and uses six subkeys per round for a total of 40 subkeys. The key schedule of IDEA* is borrowed from the MESH-64 cipher (Nakahara.Jr et al., 2003b). Let c_i denote 16-bit constants defined as follows: $c_0 = 1$ and $c_i = 3 \cdot c_{i-1}$, for $i \geq 1$ with multiplication in GF$(2)[x]/p(x)$, where $p(x) = x^{16} + x^5 + x^3 + x^2 + 1$ is a primitive polynomial. The constant ”3” is represented by the polynomial $x+1$ in GF$(2)[x]/p(x)$. Let a 128-bit key K be partitioned into eight 16-bit words K_j, $-7 \leq j \leq 0$. The elements $K_j \oplus c_{j+7}$ form the eight initial values in the following formula, for $1 \leq i \leq 40$:

$$K_i = ((((((K_{i-8} \boxplus K_{i-7}) \oplus K_{i-6}) \boxplus K_{i-3}) \oplus K_{i-2}) \boxplus K_{i-1}) \lll 7) \oplus c_{i+7}\,. \qquad (1)$$

The j-th subkey of the i-th round, $Z_j^{(i)}$, for $1 \leq j \leq 6$ and $1 \leq i \leq 7$, is just the element $K_{6(i-1)+j}$. For instance, $Z_1^{(1)} = K_1$ and $Z_1^{(2)} = K_6$.

Low-weight differences in the key schedule (1) quickly become unpredictable because of fast key avalanche due to the primitive polynomial $q(x) = x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$ and the inter-leaving of $\boxplus$, fixed bit rotation ($\lll 7$) and $\oplus$, all of which are efficient and lightweight operations. Following equation (1), we find out that $Z_3^{(1)}$ is the first subkey that depends on all eight words of K. All following subkeys also fully depend on K. Thus, complete key diffusion is achieved even faster that text diffusion in the encryption framework. Moreover, subkey bits in IDEA* overlap and depend nonlinearly on each other due to (1), unlike the simple bit permutation mapping subkeys to a user key in IDEA.

Concerning differentials in the key schedule, we have analysed wordwise (xor and subtraction) differences in the key with difference value 8000_x, because it affects only the most significant bit in a word, and thus propagates across $\boxplus$ and $\oplus$ with certainty. But, this difference do not survive for long in (1), soon becoming heavier Hamming-weight differences. Thus, the combined $\lll$, $\oplus$ and $\boxplus$ provide fast key diffusion at low cost and destroy algebraic invariants and difference patterns in subkeys, thwarting related-key attacks (Kelsey et al., 1996; Biham et al., 2008) on IDEA*. These operations, plus the constants c_i, make the key schedule nonlinear and prevent patterns in the key schedule to propagate or to cancel difference patterns in the encryption framework, further countering MITM (Demirci et al., 2003; Biham et al., 1417; Ayaz and Selcuk, 2007), slide and advanced slide attacks (Biryukov and Wagner, 1999).

The existence of weak keys in IDEA demonstrated: (i) how a strong encryption framework can be compromised by a comparatively weak key schedule. Although the number of weak keys in differential and linear settings represents a small fraction of the key space (Daemen et al., 1993) it is still more than in any other block cipher, and even larger than the number of weak and semi-weak keys in DES (Menezes et al., 1997) combined; (ii) IDEA is not suitable as a building block in compression function constructions since the key can be chosen or manipulated by an opponent in hash functions (Nakahara.Jr et al., 2003a; Wei et al., 2012). Actually, (Nakahara.Jr et al., 2003a) demonstrated that weak keys are a persistent problem even if the number of rounds were doubled. To further counter biclique attacks (Khovratovich et al., 2012), simple modifications to the IDEA key schedule as suggested in (Daemen et al., 1993) are not enough.

For decryption, (1) could be run backwards if the last eight subkeys were stored instead of the original user key K.

5 SECURITY ANALYSIS

The design of IDEA* avoids subkeys as inputs in all multiplication operations. Thus, there are no weak keys anymore. This fact concerns differential, linear (Daemen et al., 1993), differential-linear (Hawkes, 1998; Borst et al., 1997) and boomerang (among other) attacks (Biryukov et al., 2002), since distinguishers based on weak keys do not apply to IDEA*. In the context of multiplicative differentials, (Borisov et al., 2002) described attacks on IDEA-X, a variant of IDEA in which $\boxplus$ were substituted by $\oplus$. The weak subkeys are the ones combined via $\oplus$. However, IDEA* has both modular additions and unkeyed multiplications, which effectively counter multiplicative differentials.

There are well-known relations connecting $\odot$ and $\boxplus$, such as (i) $X^* = -X = 2^{16}+1-X = 1-X$ mod 2^{16} that implies $X \boxplus X^* = 1 \Leftrightarrow X \odot (X^*)^{-1} = 0$; and (ii) $X \odot (X^*)^{-1} = 1 \Leftrightarrow X - X^* = 0$. But, these relations are not enough for achieving a comprehensive attack using multiplicative differentials. In (Raddum, 2003), Raddum improved on the attack in (Borisov et al., 2002) using wordwise difference $\delta = fffd_x$. We analysed IDEA* under this xor difference and the 1-round iterative characteristic $(\delta,\ \delta,\ \delta,\ \delta) \rightarrow (\delta,\ \delta,\ \delta,\ \delta)$. The following was computed for $Z \in \{Z_2^{(i)}, Z_3^{(i)}\}$: $\delta \overset{\boxplus Z}{\rightarrow} \delta$ for $Z = 0$ with certainty, and for $Z \in \{0002_x, 8002_x, fffe_x\}$ with probability 2^{-1} (for other subkeys the probability is zero); $(\delta,\ \delta) \overset{\boxdot}{\rightarrow} 0$ with probability $2^{-15.41}$ and $(\delta, 0) \overset{\odot}{\rightarrow} \delta$ with probability $2^{-15.71}$. While for IDEA-X, the 1-round characteristic holds with probability 2^{-4}, for IDEA* it is $2^{-2(15.41)-4(15.71)} = 2^{-93.69}$, without accounting for the penalty due to addition with Z (for some of which the probability drops to zero). Using subtraction difference instead of xor difference, both $(\delta, \delta^{-1}) \overset{\boxdot}{\rightarrow} 0$ and $(\delta, \delta) \overset{\odot}{\rightarrow} 0$ hold with probability $2^{-16.26}$, where $\delta^{-1} = 4000_x$; $\delta \overset{\oplus Z}{\rightarrow} \delta$ with variable probability, for instance, 1 if $Z = 0$, 2^{-1} if $Z = 2$, 2^{-2} if $Z = 8$, but for some subkey values such as 1, 3, 5, 6 and 7, the probability is zero. So, in the best cases, $(\delta, \delta, \delta, \delta) \rightarrow (\delta, \delta, \delta, \delta)$ would hold with probability $2^{-6(16.26)} = 2^{-97.56}$, without accounting for the penalty due to the xor with Z (for some of which the probability drops to zero).

Mod-n attacks are countered in IDEA* just like in IDEA: the combination of the three group operations is enough to destroy invariant relations modulo Fermat primes. The attacks in (Kelsey et al., 1999) apply to ciphers employing addition and bitwise rotation.

5.1 Linear Cryptanalysis

For a linear analysis, *without any weak-key assumption*, we start studying a single unkeyed multiplication. We exhaustively computed linear approximations to $\odot$ (similar results hold for $\boxdot$) for arbitrary, nonzero bit-masks with low Hamming weight. The most relevant results are concerned with bit-masks that affect only the least significant bit (LSB) of a 16-bit word, while the remaining bits are inactive. These approximations are optimal for $\oplus$ and $\boxplus$ since they avoid carry bits. Consequently, this approach allows us to take care of approximations covering all three group operations simultaneously. Let $(\Gamma_1,\Gamma_2)\overset{\odot}{\rightarrow}\Gamma_3$ denote a linear approximation to $X\odot Y=W$, that is, $(X\cdot\Gamma_1)\odot(Y\cdot\Gamma_2)=W\cdot\Gamma_3$. We computed exhaustively all linear approximations involving the LSBs of both the input and output of $\odot$ and the ones with nonzero biases are $(0,0)\overset{\odot}{\rightarrow}0$ with bias 2^{-1}, $(1,1)\overset{\odot}{\rightarrow}1$ with bias $2^{-13.4731}$. We used *bias* as the magnitude of the difference between the probability of the linear relation from 1/2: $|p-1/2|$, following Matsui, and the bias range is [0,1/2]. If one uses the notion *correlation* instead, $c=|2p-1|=2$*bias, then the range becomes [0,1]. Table 1 exhaustively lists non-trivial linear relations with non-zero bias for one full round. In IDEA, these relations hold with bias 2^{-1} under several weak-key assumptions. Note that $(0,0)\overset{\odot}{\rightarrow}1$, $(0,1)\overset{\odot}{\rightarrow}0$, $(1,0)\overset{\odot}{\rightarrow}0$, $(1,0)\overset{\odot}{\rightarrow}1$, $(1,1)\overset{\odot}{\rightarrow}0$ have bias 0. These bias figures corroborate our design decisions in IDEA*, since not all combinations of bit-masks affecting the LSB of the inputs and the output to $\odot$ hold with nonzero bias.

Concatenating 1-round linear relations from Table 1 into 2-round relations leads to bias below 2^{-32}, which makes a linear attack infeasible (Matsui, 1994) since the codebook size is only 2^{64}. But, it is possible to extend it through a KW half-round without decreasing the bias since this half-round contains only $\boxplus$ and $\oplus$. Therefore, the best trade-off consists of 1.5-round relations such as $(0,0,0,\gamma)\rightarrow(0,0,\gamma,0)$ or $(0,\gamma,0,0)\rightarrow(0,0,0,\gamma)$ with two KW and one AX half-round and bias $2^{-25.94}$. A key-recovery attack on top of such a 1.5-round relation would recover subkeys both from an AX half-round before and another AX half-round after the linear relation, for a total of 2.5 rounds. From Sect. 4, there are no savings due to non-overlapping bits between $(Z_5^{(i)},Z_6^{(i)})$ and $(Z_5^{(i+2)},Z_6^{(i+2)})$ two rounds apart. Using the Piling-up Lemma (Matsui, 1994) leads to a data complexity of $8(2^{-25.94})^2=2^{54.88}$ known plaintexts (and memory) and a time complexity of $2^{54.88}(2^{16})^4=2^{118.88}$ 1-round computations. This means $2^{118.88}/2.5\approx2^{117.55}$ 2.5-round computations. Note that not all user key bits were recovered in this case. These results compare favourably with those for IDEA (Daemen et al., 1993) (for which there are linear relations covering the full cipher) and MESH-64 (Nakahara.Jr et al., 2003b) (for which there are linear relations covering four rounds).

LINEAR HULLS. Consider the 1-round linear relation $(0,0,0,\gamma)\rightarrow(0,0,\gamma,0)$ from Table 1 but with $\gamma=2$ i.e. exploiting the second LSB as mask. Taking into account the linear approximations $(2,2)\overset{\odot}{\rightarrow}2$ with bias $2^{-13.496}$; $(3,2)\overset{\boxplus}{\rightarrow}2$, $(2,3)\overset{\boxplus}{\rightarrow}3$, $(3,3)\overset{\boxplus}{\rightarrow}2$, $(2,2)\overset{\boxplus}{\rightarrow}3$ and $(2,2)\overset{\boxplus}{\rightarrow}2$ all with bias 2^{-2}, one can track three separate trails across 1-round IDEA*: one trail uses $(2,3)\overset{\boxplus}{\rightarrow}2$ twice, another uses $(2,2)\overset{\boxplus}{\rightarrow}3$ and $(2,3)\overset{\boxplus}{\rightarrow}2$ and the last one uses $(2,3)\overset{\boxplus}{\rightarrow}3$ and $(3,3)\overset{\boxplus}{\rightarrow}2$ inside the AX-box. All trails have bias $2^{3-2-2-13.49-13.49}=2^{-27.98}$. The combined bias of both linear trails is $\sqrt{3}\cdot2^{-27.98}=2^{-27.19}$ which is lower than that of the 1-round relation for $\gamma=1$. In summary, the trails are few and there is an extra penalty due to the carry bits. For more than one round, unless the number of trails increases well above the drop in the combined bias due to the approximations of $\boxplus$'s inside the AX-box, the overall bias (using K. Nyberg's rule (Nyberg, 1995)) will remain lower than for one-round relations. This means that a potential linear hull effect will not be enough to counter a significant bias drop in the long run due to the penalty paid by carry bits. Even more true since IDEA* has only 6.5 rounds. The same reasoning applies to the other relations in Table 1. Concerning the results in Sect. 5.1, we conclude that 3-round IDEA* is secure against linear cryptanalysis, including linear hulls.

5.2 Biryukov-Demirci Attack

The application of $\odot$ and $\boxdot$ in place of $\oplus$ in IDEA* implies that there is no more high-probability linear relation involving the LSB's of the two middle 16-bit words in a text block, not even across a single round. This is an essential weakness exploited in many attacks on IDEA (Junod, 2005; Biham et al., 1417; Khovratovich et al., 2012; Sun and Lai, 2009).

The Biryukov-Demirci (BD) relation exploits the fact that the two middle 16-bit words in IDEA only uses $\oplus$ and $\boxplus$ to mix intermediate data across the cipher state. Consequently, the LSB of the corresponding plaintext and ciphertext words are related, since there is no carry bits in the LSB position. Let the input to a round be (X_1,X_2,X_3,X_4), its output be (Y_1,Y_2,Y_3,Y_4) and the i-th MA-box output be (s_i,t_i). Then, in IDEA both LSB$(X_2\oplus Z_2^{(i)}\oplus s_i)$ = LSB(Y_3) and LSB$(X_3\oplus Z_3^{(i)}\oplus t_i)$ = LSB(Y_2) hold with certainty. Comparatively, in IDEA*, there are $\odot$'s across all four 16-bit words in every round, instead of $\oplus$, and the BD relation involving (X_2, Y_3) and (X_3, Y_2) become $(X_2\boxplus Z_2^{(i)})\odot s_i=Y_3$ and $(X_3\boxplus Z_3^{(i)})\odot t_i=Y_2$. Note that $\mathrm{LSB}((X_2 \boxplus Z_2^{(i)}) \odot s_i)$ does not equal $\mathrm{LSB}(Y_3)$ anymore, since the $\odot$ operation has a wrap-around effect (Lai's Low-High algorithm (Lai, 1992)) and consequently the LSB of the multiplication does not depend only on the LSBs of its two inputs: $\mathrm{LSB}((X_2 \boxplus Z_2^{(i)}) \odot s_i) \neq \mathrm{LSB}(X_2 \boxplus Z_2^{(i)}) \oplus \mathrm{LSB}(s_i)$. Overall, $\odot$ has much stronger diffusion than $\oplus$ (Sect. 3). If we assume a linear approximation of $\odot$ of the form $(1,1) \overset{\odot}{\rightarrow} 1$ as in Sect. 5.1, then the approximation $\mathrm{LSB}((X_2 \boxplus Z_2^{(i)}) \odot s_i) = \mathrm{LSB}(X_2 \boxplus Z_2^{(i)}) \oplus \mathrm{LSB}(s_i)$ would holds with bias $2^{-13.4731}$. After two rounds the bias becomes $2^{-25.9462}$ and after three rounds the bias becomes $2^{-38.4193}$, which is too low since the codebook size is only 2^{64}.

5.3 Differential Analysis

For differential analysis, we employed both xor differences ($\Delta X = X \oplus X^*$) and subtraction differences ($\Delta X = X - X^*$) involving 16-bit words across a single $\odot$, such as $X \odot Y = W$. Let $\Delta^{\oplus} W = (X \odot Y) \oplus (X \oplus \delta_1) \odot (Y \oplus \delta_2)$ denote the output difference of an unkeyed $\odot$ for $\delta_i \in \{8000_x, 0000_x\}$, $i \in \{1,2\}$. Note that $\Delta^{-} W = (X \odot Y) - (X - \delta_1) \odot (Y - \delta_2)$ behaves exactly like $\Delta^{\oplus} W$ because $X \oplus Y = 8000_x \Leftrightarrow X = Y \oplus 8000_x \Leftrightarrow X = Y \boxplus 8000_x \Leftrightarrow X - Y = 8000_x$. Thus, we denote $\Delta^{\oplus} W$ and $\Delta^{-} W$ simply as ΔW. Note that for multiplicative difference, $X \odot (X^*)^{-1} = 1 \Leftrightarrow X - X^* = 0 \Leftrightarrow X \oplus X^* = 0$ that is, zero xor-difference implies $\odot$ difference equal to one. For xor difference 8000_x there is no equivalent difference value for $\odot$. Thus, the results in Table 2 do not apply for multiplicative differentials.

For (Δ_1, Δ_2) = (0000_x, 8000_x) or (8000_x, 0000_x) the probability that $\Delta W = 8000_x$ is 2^{-15}. For (Δ_1, Δ_2) = (8000_x, 8000_x), the probability that $\Delta W = 8000_x$ is $2^{-14.98}$, and the probability that $\Delta W = 0000_x$ is 2^{-15}. These data also hold for the case $X \odot Y^{-1} = W$. Thus, we can construct Table 2. Note that the minimum number of active $\odot$'s is three, and there are no conditions on subkey values for the difference propagation compared to IDEA. Moreover, these 1-round characteristics hold with much smaller probability than for IDEA under weak-key conditions (Daemen et al., 1993). This fact is a consequence of the cipher design, which placed $\odot$'s in order to mix the AX-box outputs to each 16-bit word in a block at the end of each round (guaranteeing full diffusion in a single round).

Concatenating 1-round characteristics from Table 2 across two rounds results in probability less than or equal to 2^{-120}. Recall that the codebook is only 2^{64}. But, these characteristics can be extended across one KW half-round, since the difference 8000_x propagates for free across $\boxplus$ and $\oplus$. Thus, the best trade-off consists of 1.5-round characteristics such as $(0,0,\delta,0) \rightarrow (\delta,0,0,0)$ or $(0,\delta,0,\delta) \rightarrow (0,0,\delta,\delta)$ with probability 2^{-45}. A key-recovery attack on top of such 1.5-round characteristics would recover subkeys both from an AX half-round before and another AX half-round after the characteristic, for a total of 2.5 rounds. As shown in Sect. 4, there is no overlapping between bits of $(Z_5^{(i)}, Z_6^{(i)})$ and $(Z_5^{(i+2)}, Z_6^{(i+2)})$ two rounds apart. This implies a data complexity proportional to 2^{45} chosen plaintexts (and memory) and a time complexity of $2^{45}(2^{16})^4 = 2^{109}$ 1-round computations. This means $2^{109}/2.5 \approx 2^{107.67}$ 2.5-round computations.

For *truncated differentials*, using either xor or subtraction differences, we adopt the approach in (Borst et al., 1997). For instance, for a single unkeyed $\odot$ such that $X \odot Y = W$, for arbitrary $\Delta X \neq 0$ and $\Delta Y = 0$ the equality $\Delta W = \Delta X$ happens with probability 2^{-15}. For $\Delta X \neq 0$ and $\Delta Y = \Delta X$, $\Delta W = 0$ with probability around 2^{-16} for arbitrary, nonzero ΔX values. Let $A,B,C,D,E,F,G,H,I \in \mathbb{Z}_{2^{16}} - \{0\}$. A 1-round truncated differential for IDEA* can have the form $(A,0,B,0) \overset{2^{-16}}{\rightarrow} (C,0,C,0) \overset{2^{-16} * 2^{-15} * 2^{-15}}{\rightarrow}$ (C, C, 0, 0), where the first part $(A,0,B,0) \overset{2^{-16}}{\rightarrow}$ (C, 0, C, 0) means that A and B differences cause the same difference C after crossing the first KW half-round with probability 2^{-16}. For the AX half-round there are two critical points: (i) the input difference to the leftmost $\boxdot$ has input differences C and C. The resulting difference is the leftmost input to the AX-box, which we expect to be zero, that is, the transition $(C,C) \rightarrow 0$ across a $\boxdot$. This happens with a 2^{-16} chance. The rightmost AX-box input has zero difference. Thus, the input difference to the AX-box is $(0,0)$ and always gives $(0,0)$ output difference; (ii) the double C differences when combined with the zero differences from the AX-box are preserved with a 2^{-15} chance each. So, for a single round the truncated differential $(A,0,B,0) \rightarrow (C,C,0,0)$ holds with probability around 2^{-62}. The corresponding probability of this differential for a random permutation is 2^{-32}, due to the zero output difference words. Therefore, this differential is not useful for distinguishing 1-round IDEA* from a random permutation.

If we let the double C differences turn into arbitrary differences D and E for instance, then the probability increases to 2^{-32}, resulting in the 1-round truncated differential $(A,0,B,0) \rightarrow (D,E,0,0)$. Across the next half-round, D and E will lead to differences, say, F and G and the block difference becomes $(F,G,0,0)$. This means the input difference to the following AX-box becomes (H,I). Then, with probability 2^{-32} the AX-box output difference is (G,F). When this difference is combined with $(F,G,0,0)$ we obtain a difference $(F\odot F, F\odot 0, G\odot G, G\odot 0)$. If we wish $F\odot F$ and $G\odot G$ to lead to zero difference, then it will cost 2^{-16} each and the final probability for the 2-round differential $(A,0,B,0)\rightarrow(D,E,0,0)\rightarrow(0,J,0,K)$ reaches $2^{-64-16-16}=2^{-96}$, where J is the difference coming out of $F\odot 0$, and K from $G\odot 0$. If we do not set conditions on $F\odot 0$ nor on $G\odot 0$, then one will have an output difference of the form (L,J,M,K) with nonzero J, K, L, M. On the one hand the probability increases to 2^{-64}, but on the other hand: (i) crossing the next round would decrease the probability further, and (ii) it would hinder attacks since there are no bit patterns or other filtering conditions on J, K, L and M. Overall, these 1- and 2-round truncated differentials are much shorter than the ones obtained for IDEA and MESH ciphers. Moreover, for a boomerang distinguisher (Biryukov et al., 2002), suppose we use such truncated differentials, say, with two rounds in the encryption direction and one round for the decryption direction, that is, four truncated differentials. This leads to a probability of $(2^{-96})^2\cdot(2^{-64})^2=2^{-320}$, which is too low for a codebook of only 2^{64} texts. Suppose the full codebook is used. Then, 2^{64} texts can provide up to $2^{64}\cdot(2^{64}-1)/2\approx 2^{127}$ text pairs. Even using 1-round truncated differentials in each direction, the probability is already $(2^{-64})^2\cdot(2^{-64})^2=2^{-256}$.

DIFFERENTIAL-LINEAR ATTACKS. For a differential-linear attack (Biham et al., 2005), combining 1-round characteristics from Table 2 with the highest probability 2^{-45} and 1-round relations from Table 1 with the highest bias $2^{-25.94}$, we arrive at differential-linear distinguishers with probability $1/2+2pq^2$, where p is the characteristic probability and q is linear bias. For the concatenation of a single 1-round characteristic such as $(0,0,\delta,0)\rightarrow(\delta,0,0,0)$ and a 1-round relation, such as $(0,0,0,\gamma)\rightarrow(0,0,\gamma,0)$ this probability is $1/2+2\cdot 2^{-45}\cdot(2^{-25.94})^2=1/2+2^{-95.88}$ which makes the attack infeasible since the codebook size of IDEA* is only 2^{64}. Combining the 1-round *truncated differential* $(A,0,B,0)\rightarrow(C,C,0,0)$ with 1-round linear relations in Table 1 such as $(0,0,0,\gamma)\rightarrow(0,0,\gamma,0)$ leads to a combined probability of $1/2+2\cdot 2^{-62}\cdot 2^{-51.88}=1/2+2^{-112.88}$, which is again too low for an attack on 2-round IDEA*.

IMPOSSIBLE DIFFERENTIALS. Impossible-differential distinguishers in IDEA, such as $(a,\ 0,\ a,\ 0)\overset{2.5\ rounds}{\not\rightarrow}(b,\ b,\ 0,\ 0)$ (Biham et al., 1999) with a and b nonzero 16-bit differences, do not apply to IDEA* because differences across $\odot$ and $\boxdot$ behave differently than across $\oplus$. One can have both $(a,b)\overset{\odot}{\rightarrow}0$ and $(a,b)\overset{\odot}{\rightarrow}c$ for nonzero a, b and c with nonzero probability. We verified exhaustively that this probability is close to 2^{-16} independent of the particular values of a, b and c. So far, even for a single round, $(a,0,a,0)\rightarrow(b,b,0,0)$ still holds with nonzero probability (either starting before or after a half-round). We have thus far not yet found alternative impossible differentials for (reduced-round) IDEA* versions.

SQUARE ATTACKS. Concerning square attacks, we follow the terminology of (Daemen et al., 1997). A key-recovery attack on 2-round IDEA* is the following: consider key-dependent λ-sets containing 2^{16} plaintexts of the form $(z_1\oplus i, c, i-z_3, c)$ where c is an arbitrary 16-bit constant (which makes the 2nd and 4th words passive, denoted P), i assumes all possible 16-bit value exactly once (which makes it an active word, denoted A), and z_1,z_3 are guesses for $Z_1^{(1)}$ and $Z_3^{(1)}$, respectively. We choose the two A words, due to the i's, such that they contain the values $\{0,1,2,\ldots,65535\}$ *in the same order*. The objective of this particular λ-set is to bypass the first round of IDEA* and to propagate the λ-set pattern (A,P,A,P). When z_1 and z_3 correctly match $Z_1^{(1)}$ and $Z_3^{(1)}$, the input to the first AX-box will be two P (passive, 16-bit) words. When z_1,z_3 are wrong, the input to the first AX-box will not be (P,P) because the inputs to the leftmost $\boxdot$ will not be (i,i).

This construction implies that for the correct z_1,z_3, the output of the leftmost $\boxdot$ we will be $i\boxdot i^{-1}=1$ for all $i\in\mathbb{Z}_{2^{16}}$. In other words, both inputs to the AX-box will be constants or passive words. The input λ-set to the second round will be (A,A,P,P) and the input to the second AX-box will be (A,A) since they are the $\boxdot$ combination of an active A word and a passive P word. Unfortunately, the output of this AX-box will be $(?,?)$, where '?' denotes a garbled word, with no pattern that allows to distinguish it from a random 32-bit variable (due to the combination of A and P words inside the AX-box). Nonetheless, if we denote the second round output by (x,y,u,v) for any λ-set, and if the values z_1,z_3 were guessed correctly, then $u\boxdot v=u\odot v^{-1}$ and $x\boxdot y=x\odot y^{-1}$, over the λ-set, should both be A (active) words. Otherwise, the z_1,z_3 values were wrong. This key-recovery attack on 2-round IDEA* costs $2^{32}\cdot 2^{16}=2^{48}$ chosen plaintexts, 2^{16} memory and effort $2^{32}/2=2^{31}$ half-round computations in the worst case, which is equivalent to $2^{31}/4=2^{29}$ 2-round computations.

In (Knudsen and Rijmen, 2008), Kudsen and Rijmen presented a new attack setting in which *the key is known by the adversary*, for instance, in the context of a hash function. They studied so called known-key distinguishers based on λ-sets. It is an inside-out approach in which a single A word inside a target cipher is left free to propagate both in the encryption and decryption directions. This approach allows distinguishers up to 7-round AES (such that at least one balanced word survive in the state). For IDEA, known-key distinguishers can reach at most 2.5 rounds, such as $(?,?,A,?) \overset{KW}{\leftarrow} (B,?,A,?) \overset{AX}{\leftarrow} (A,P,P,P) \overset{KW}{\rightarrow} (A,P,P,P) \overset{AX}{\rightarrow} (B,A,?,?) \overset{KW}{\rightarrow} (?,A,?,?)$, where KW and AX denote half-rounds. Similarly, for IDEA*, 2.5-round known-key distinguishers exist such as $(?,?,A,?) \overset{KW}{\leftarrow} (?,?,A,?) \overset{AX}{\leftarrow} (A,P,P,P) \overset{KW}{\rightarrow} (A,P,P,P) \overset{AX}{\rightarrow} (?,A,?,?) \overset{KW}{\rightarrow} (?,A,?,?)$ and $(A,?,?,?) \overset{KW}{\leftarrow} (A,?,?,?) \overset{AX}{\leftarrow} (P,A,P,P) \overset{KW}{\rightarrow} (P,A,P,P) \overset{AX}{\rightarrow} (A,A,?,A) \overset{KW}{\rightarrow} (A,A,?,A)$. These distinguishers indicate that 2.5-round IDEA* may not be an ideal primitive in compression functions constructions in hash functions.

5.4 Biclique Attacks

For the reasons listed below, we argue that the design of IDEA* imposes enough countermeasures against biclique attacks (Khovratovich et al., 2012), which heavily relies on MITM attacks (Demirci et al., 2003) and poor diffusion in the key schedule.

- the Biryukov-Demirci relation discussed in Sect. 5.2 does no hold for IDEA*.
- Sect. 4 detailed *full key diffusion* in the key schedule after (and including) $Z_3^{(1)}$. As a consequence, key bits overlap in every subkey, meaning there are no neutral key bits and thus, related-key differentials (with nonzero difference only in the key and holding with probability 1) needed in bicliques cannot be constructed based on independent subkey bits.
- Sect. 3 detailed *full text diffusion in a single round*. Moreover, there is improved wordwise diffusion provided by $\odot$'s replacing $\oplus$'s across every block in the AX half-rounds. Therefore, the effort for the MITM and biclique constructions becomes equivalent to that of an exhaustive key search, because there is no shortcut that allows to partition the subkeys into independent sets as required in (Khovratovich et al., 2012).

5.5 Side-channel and Performance Analysis

For the sake of practical usability, cryptographic primitives should be carefully designed and implemented in such a way that the internally processed information remains secure.

From a practical point of view, side-channel analysis (SCA) represents a serious threat for the security of cryptographic systems in addition to conventional cryptanalysis. SCA allows an adversary to recover cryptographic keys by analysing critical pieces of information unintentionally leaked through physical means. Power analysis (Kocher et al., 1999) is one of the strongest kinds of SCA. Its underlying assumption says that the instantaneous power consumption of an integrated circuit relates to the executed instructions and processed data. Two widely investigated families of power attacks are the simple and differential power analysis: SPA and DPA (Kocher et al., 1999). Briefly, the former focuses on instruction-related key aspects present in a few power traces, whilst the latter focuses on data-related key aspects present in a typically higher amount of power traces. For a comprehensive explanation we refer to (Mangard et al., 2007).

Power analysis has already been performed on IDEA (e.g. (Lemke et al., 2004; Oswald and Preneel, 2002)). Oswald and Preneel (Oswald and Preneel, 2002) assessed the theoretical vulnerability of IDEA to power analysis. As the key schedule of IDEA is relatively simple due to the straightforward cyclic shifting of the key, it turns out that SPA represents a threat. IDEA* counters SPA theoretically by using a more elaborate key schedule, as shown in Sec. 4. Lemke *et al.* (Lemke et al., 2004) and Pan *et al.* (Pan et al., 2008) realized DPA on each one of the boolean and arithmetic operations ($\odot, \boxplus, \oplus$) used in IDEA. They showed that $\oplus$ is more DPA-resistant than $\boxplus$, which is in turn more DPA-resistant than $\odot$. Nonlinear functions are less robust against DPA than linear functions (Pan et al., 2008; Guilley et al., 2004; Benoît and Peyrin, 2010; Prouff, 2005).

The swapping of $\odot$ and $\oplus$ operations in IDEA* makes it theoretically more DPA-resistant than IDEA. While in IDEA the keys are input to $\odot$ and $\boxplus$, in IDEA* the keys are input to $\oplus$ and $\boxplus$. Moreover, implementing DPA against an unknown implementation of IDEA* is expected to be more time consuming than performing the same attack on e.g. AES, DES, SERPENT, PRESENT or mCrypton, because the number of key hypotheses is 2^{16} for IDEA* and respectively 2^8, 2^6, 2^4, 2^4, 2^4 for the others. Aiming especifically at countering DPA on IDEA, Neiβe and Pulkus (Neiße and Pulkus, 2004) proposed algorithms to protect the cipher's arithmetic and boolean operations by switching masks among the operations. IDEA* may also benefit from this countermeasure. It should be noted however that depending on the designer's resources and constraints DPA may still remain an issue for IDEA* as software countermeasures for boolean operations can be costly. Nevertheless other countermeasures can be applied such as randomizing the order of the operations in each execution, adding noise by executing other instructions in parallel to the encryption/decryption and adding random delays between operations (Mangard et al., 2007).

Concerning efficiency, empirical experiments on IDEA, IDEA* and AES encryption were performed on an 8-bit microcontroller ATmega328P. The same level of optimization was used for all algorithms. These analyses showed that 6.5-round IDEA* is 6% slower than an 8.5-round IDEA (same for AES) thanks to a precomputation of all multiplicative inverses (Fig. 4). Also, our AES implementation is 4% faster than IDEA*. It should be emphasized that increasing the number of rounds in IDEA does not protects it against side-channel attacks (Nakahara.Jr et al., 2003a). Indeed, power analyses are performed on the first (or the last) round independent of the total number of rounds.

6 CONCLUSIONS

This paper analysed a simple design decision: *what is the impact on a cipher's security due to a rearrangement of the internal cipher components?* We observed significant and relevant consequences when we swapped the exclusive-or and multiplication operations in the IDEA cipher.

We called the updated design IDEA*. Our analyses indicate IDEA* effectively counters all previously reported attacks on IDEA, including theoretical power analysis. IDEA* has improved overall diffusion through the use of unkeyed multiplication, a new key schedule and uses 6.5 rounds (compared with 8.5 rounds in IDEA). In summary, the new design counters not only differential (Borst et al., 1997) and linear analysis (Daemen et al., 1993; Biham et al., 2007) but also impossible differentials (Biham et al., 1999), truncated differentials (Knudsen and Rijmen, 1997), boomerang (Wagner, 1999; Biryukov et al., 2002), square (Demirci, 2003; Nakahara.Jr et al., 2002; Biham et al., 2007), differential-linear (Hawkes, 1998; Borst et al., 1997), meet-in-the-middle (Demirci et al., 2003; Ayaz and Selcuk, 2007), multiplicative differentials (Borisov et al., 2002), Biryukov-Demirci (Junod, 2005; Biham et al., 1417; Sun and Lai, 2009), higher-order differential (Biham et al., 2005), related-key (Biham et al., 2008) and mod-n attacks (Kelsey et al., 1999). We have also taken into account recent developments such as biclique analysis (Abed et al., 2012; Khovratovich et al., 2012) that reach the full versions of IDEA, AES and PRESENT. Finally, we focused also on algorithmic countermeasures against power analysis in order to compare IDEA to IDEA*. We showed that IDEA* is theoretically more resistant against power analysis than IDEA. As such, our contributions also improved our understanding of the IDEA cipher in view of old and new atttacks.

In summary: simple changes in a cipher can have significant impacts in it security (and performance). These changes and design decisions are often undocumented, even in new designs, which may lead to suspicion of trapdoors. Only a thorough analysis can give some evidence of the strength of new designs against modern attacks.

As a topic for future work, we suggest to study different permutation of cipher components in other high-profile cryptographic primitives, such as hash functions and stream ciphers. Potential targets include the MESH ciphers (Nakahara.Jr et al., 2003b), which have an Add-Mult-Xor (AMX) design similar to that of IDEA. The point is that in these ciphers, there is a clear asymmetry between the internal operations: addition and xor are lightweight operations (a few CPU cycles) with poor diffusion, while modular multiplication is heavyweight (several CPU cycles) with better diffusion.

ACKNOWLEDGEMENTS

The authors would like to thank INNOVIRIS, the Brussels Institute for Research and Innovation, under the ICT Impulse program CRYPTASC, for sponsoring this work.

REFERENCES

Abed, F., Forler, C., List, E., Lucks, S., and Wenzel, J. (2012). Biclique cryptanalysis of the PRESENT and LED lightweight ciphers. IACR ePrint Archive 2012/591.

Ayaz, E. and Selcuk, A. (2007). Improved DST cryptanalysis of IDEA. In *Selected Areas in Cryptology (SAC)*, LNCS 4356, pages 1–14. Springer.

Benoît, O. and Peyrin, T. (2010). Side-channel analysis of six SHA-3 candidates. In *CHES*, LNCS, pages 140–157. Springer.

Biham, E., Biryukov, A., and Shamir, A. (1999). Miss in the middle attacks on IDEA, Khufu and Khafre. In *Fast Software Encryption (FSE)*, LNCS 1636, pages 124–138. Springer.

Biham, E., Dunkelman, O., and Keller, N. (2005). New combined attacks on block ciphers. In *Fast Software Encryption (FSE)*, LNCS 3557, pages 126–144. Springer.

Biham, E., Dunkelman, O., and Keller, N. (2007). A new attack on 6-round IDEA. In *Fast Software Encryption (FSE)*, LNCS 4593, pages 211–224. Springer.

Biham, E., Dunkelman, O., and Keller, N. (2008). A unified approach to related-key attacks. In *Fast Software Encryption (FSE)*, LNCS 5086, pages 73–96. Springer.

Biham, E., Dunkelman, O., Keller, N., and Shamir, A. (2011/417). New data-efficient attacks on reduced-round IDEA. IACR ePrint 2011/417.

Biryukov, A., Nakahara.Jr, J., Preneel, B., and Vandewalle, J. (2002). New weak-key classes of IDEA. In *Information and Communications Security (ICICS)*, LNCS 2513, pages 315–326. Springer.

Biryukov, A. and Wagner, D. (1999). Slide attacks. In *Fast Software Encryption (FSE)*, LNCS 1636, pages 245–259. Springer.

Bogdanov, A., Khovratovich, D., and Rechberger, C. (2011). Biclique cryptanalysis of the full AES. IACR ePrint archive 2011/449.

Bogdavov, A. and Rechberger, C. (2010). A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. IACR ePrint archive 2010/532.

Borisov, N., Chew, M., Johnson, R., and Wagner, D. (2002). Multiplicative differentials. In *Fast Software Encryption (FSE)*, LNCS 2365, pages 17–33. Springer.

Borst, J., Knudsen, L., and Rijmen, V. (1997). Two attacks on reduced IDEA (extended abstract). In *EUROCRYPT*, LNCS 1233, pages 1–13. Springer.

Daemen, J., Govaerts, R., and Vandewalle, J. (1993). Weak keys for IDEA. In *CRYPTO*, LNCS 773, pages 224–231. Springer.

Daemen, J., Knudsen, L., and Rijmen, V. (1997). The block cipher SQUARE. In *Fast Software Encryption (FSE)*, LNCS 1267, pages 149–165. Springer.

Demirci, H. (2003). Square-like attacks on reduced rounds of IDEA. In *Selected Areas in Cryptography (SAC)*, LNCS 2595, pages 147–159. Springer.

Demirci, H., Selcuk, A., and Türe, E. (2003). A new meet-in-the-middle attack on the IDEA block cipher. In *Selected Areas in Cryptography (SAC)*, LNCS 3006, pages 117–129. Springer.

FIPS197 (2001). Advanced encryption standard (AES). FIPS PUB 197 Federal Information Processing Standard Publication 197, U.S. Department of Commerce.

Guilley, S., Hoogvorst, P., and Pacalet, R. (2004). Differential power analysis model and some results. In Quisquater, J.-J., Paradinas, P., Deswarte, Y., and Kalam, A., editors, *CARDIS*, pages 127–142. Kluwer.

Hawkes, P. (1998). Differential-linear weak key classes of IDEA. In *EUROCRYPT*, LNCS 1403, pages 112–126. Springer.

Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., and Chee, S. (2006). HIGHT: A new block cipher suitable for low-resource device. In Goubin, L. and Matsui, M., editors, *Cryptographic Hardware and Embedded Systems*, LNCS 4249, pages 46–59. Springer.

Isobe, T. (2011). A single-key attack on the full GOST block cipher. In *Fast Software Encryption (FSE)*, LNCS 6733, pages 290–305. Springer.

Joye, M. and Quisquater, J.-J., editors (2004). *Cryptographic Hardware and Embedded Systems - CHES 2004*, LNCS 3156. Springer.

Junod, P. (2005). New attacks against reduced-round versions of IDEA. In *Fast Software Encryption (FSE)*, LNCS 3557, pages 384–397. Springer.

Kelsey, J., Schneier, B., and Wagner, D. (1996). Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER and triple-DES. In *CRYPTO*, LNCS 1109, pages 237–251. Springer.

Kelsey, J., Schneier, B., and Wagner, D. (1999). Mod n cryptanalysis, with applications against RC5P and M6. In *Fast Software Encryption (FSE)*, LNCS 1636, pages 139–155. Springer.

Khovratovich, D., Leurent, G., and Rechberger, C. (2012). Narrow-bicliques: cryptanalysis of full IDEA. In *EUROCRYPT*, LNCS 7237, pages 392–410. Springer.

Knudsen, L. and Rijmen, V. (1997). Truncated differentials of IDEA. Technical report, ESAT-COSIC Tech report 97-1.

Knudsen, L. and Rijmen, V. (2008). Known-key distinguishers for some block ciphers. In *Asiacrypt*, LNCS 4833, pages 315–324. Springer.

Kocher, P., Jaffe, J., and Jun, B. (1999). Differential power analysis. In *CRYPTO*, LNCS, pages 388–397. Springer.

Lai, X. (1992). *On the Design and Security of Block Ciphers*. PhD thesis, ETH no. 9752, Swiss Federal Institute of Technology, Zurich.

Lai, X. and Massey, J. (1990). A proposal for a new block encryption standard. In *EUROCRYPT*, LNCS 473, pages 389–404. Springer.

Lai, X., Massey, J., and Murphy, S. (1991). Markov ciphers and differential cryptanalysis. In *EUROCRYPT*, LNCS 547, pages 17–38. Springer.

Lemke, K., Schramm, K., and Paar, C. (2004). Dpa on n-bit sized boolean and arithmetic operations and its application to IDEA, RC6, and the HMAC-construction. In (Joye and Quisquater, 2004), pages 205–219.

Mangard, S., Oswald, E., and Popp, T. (2007). *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer.

Matsui, M. (1994). Linear cryptanalysis method for DES cipher. In *EUROCRYPT*, LNCS 765, pages 386–397. Springer.

Matsui, M. (1995). On correlation between the order of s-boxes and the strength of DES. In *EUROCRYPT*, LNCS 950, pages 366–375. Springer.

Menezes, A., vanOorschot, P., and Vanstone, S. (1997). *Handbook of Applied Cryptography*. CRC Press.

NakaharaJr, J. (2009). On the order of round components in the AES. *International Journal of Network Security (IJNS)*, 9:44–50.

Nakahara.Jr, J., Preneel, B., and Vandewalle, J. (2002). Square attacks on reduced-round PES and IDEA block ciphers. 23rd Symposium on Information Theory in the Benelux.

Nakahara.Jr, J., Preneel, B., and Vandewalle, J. (2003a). A note on weak-keys of PES, IDEA and some extended variants. In *Information Security Conference (ISC)*, LNCS 2851, pages 269–279. Springer.

Nakahara.Jr, J., Rijmen, V., Preneel, B., and Vandewalle, J. (2003b). The MESH block ciphers. In *Information Security Applications (WISA)*, LNCS 2908, pages 458–473. Springer.

Neiße, O. and Pulkus, J. (2004). Switching blindings with a view towards IDEA. In (Joye and Quisquater, 2004), pages 230–239.

Nyberg, K. (1995). Linear approximation of block ciphers. In *EUROCRYPT*, LNCS 950, pages 439–444. Springer.

Oswald, E. and Preneel, B. (2002). A theoretical evaluation of some NESSIE candidates regarding their susceptibility towards power analysis attacks. Technical report, Katholieke Universiteit Leuven.

Pan, J., denHartog, J., and deVink, E. (2008). An operation-based metric for CPA resistance. In Jajodia, S., Samarati, P., and Cimato, S., editors, *SEC*, volume 278 of *IFIP*, pages 429–443. Springer.

Prouff, E. (2005). DPA attacks and s-boxes. In *Fast Software Encryption (FSE)*, LNCS, pages 424–441. Springer.

Raddum, H. (2003). Cryptanalysis of IDEA-X/2. In *Fast Software Encryption (FSE)*, LNCS 2887, pages 1–8. Springer.

Shannon, C. (1949). Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715.

Sun, X. and Lai, X. (2009). The key-dependent attack on block ciphers. In *ASIACRYPT*, LNCS 5912, pages 19–36. Springer.

Vergos, H., Vassalos, E., and Bakalis, D. (2011). Modulo $2^n + 1$ arithmetic units with embedded diminished-to-normal conversion. In *Digital System Design (DSD), 14th Euromicro Conference*, pages 468–475.

Wagner, D. (1999). The boomerang attack. In *Fast Software Encryption (FSE)*, LNCS 1636, pages 156–170. Springer.

Wei, L., Peyrin, T., Sokolowski, P., Ling, S., Pieprzyk, J., and Wang, H. (2012). On the (in)security of IDEA in various hashing modes. IACR ePrint archive 2012/264.

Zimmernmann, R. (1999). Efficient VLSI implementation of modulo $2^n + 1$ addition and multiplication. In *Computer Arithmetic, 14th IEEE Symposium*, pages 158–167.

APPENDIX

Table 1: 1-round linear relations in IDEA* with $\gamma = 1$.

1-round linear relation	bias	# active $\odot$'s
$(0,0,0,\gamma) \to (0,0,\gamma,0)$	$2^{-25.94}$	2
$(0,0,\gamma,0) \to (\gamma,0,\gamma,\gamma)$	$2^{-63.35}$	5
$(0,0,\gamma,\gamma) \to (\gamma,0,0,\gamma)$	$2^{-38.41}$	3
$(0,\gamma,0,0) \to (0,0,0,\gamma)$	$2^{-25.94}$	2
$(0,\gamma,0,\gamma) \to (0,0,\gamma,\gamma)$	$2^{-25.94}$	2
$(0,\gamma,\gamma,0) \to (\gamma,0,\gamma,0)$	$2^{-38.41}$	3
$(0,\gamma,\gamma,\gamma) \to (\gamma,0,0,0)$	$2^{-38.41}$	3
$(\gamma,0,0,0) \to (0,\gamma,\gamma,\gamma)$	$2^{-63.35}$	5
$(\gamma,0,0,\gamma) \to (0,\gamma,0,\gamma)$	$2^{-38.41}$	3
$(\gamma,0,\gamma,0) \to (\gamma,\gamma,0,0)$	$2^{-25.94}$	2
$(\gamma,0,\gamma,\gamma) \to (\gamma,\gamma,\gamma,0)$	$2^{-50.88}$	4
$(\gamma,\gamma,0,0) \to (0,\gamma,\gamma,0)$	$2^{-38.41}$	3
$(\gamma,\gamma,0,\gamma) \to (0,\gamma,0,0)$	$2^{-38.41}$	3
$(\gamma,\gamma,\gamma,0) \to (\gamma,\gamma,0,\gamma)$	$2^{-50.88}$	4
$(\gamma,\gamma,\gamma,\gamma) \to (\gamma,\gamma,\gamma,\gamma)$	$2^{-50.88}$	4

Table 2: 1-round characteristics in IDEA* using $\oplus$ or $-$ differences and $\delta = 8000_x$.

1-round characteristic	probability	# active $\odot$'s
$(0,0,0,\delta) \to (\delta,\delta,\delta,0)$	2^{-75}	5
$(0,0,\delta,0) \to (\delta,0,0,0)$	2^{-45}	3
$(0,0,\delta,\delta) \to (0,\delta,\delta,0)$	2^{-75}	5
$(0,\delta,0,0) \to (\delta,\delta,0,\delta)$	2^{-75}	5
$(0,\delta,0,\delta) \to (0,0,\delta,\delta)$	2^{-45}	3
$(0,\delta,\delta,0) \to (0,\delta,0,\delta)$	2^{-75}	5
$(0,\delta,\delta,\delta) \to (\delta,0,\delta,\delta)$	2^{-90}	6
$(\delta,0,0,0) \to (0,\delta,0,0)$	2^{-45}	3
$(\delta,0,0,\delta) \to (\delta,0,\delta,0)$	2^{-75}	5
$(\delta,0,\delta,0) \to (\delta,\delta,0,0)$	2^{-45}	3
$(\delta,0,\delta,\delta) \to (0,0,\delta,0)$	2^{-90}	6
$(\delta,\delta,0,0) \to (\delta,0,0,\delta)$	2^{-75}	5
$(\delta,\delta,0,\delta) \to (0,\delta,\delta,\delta)$	2^{-90}	6
$(\delta,\delta,\delta,0) \to (0,0,0,\delta)$	2^{-90}	6
$(\delta,\delta,\delta,\delta) \to (\delta,\delta,\delta,\delta)$	2^{-90}	6

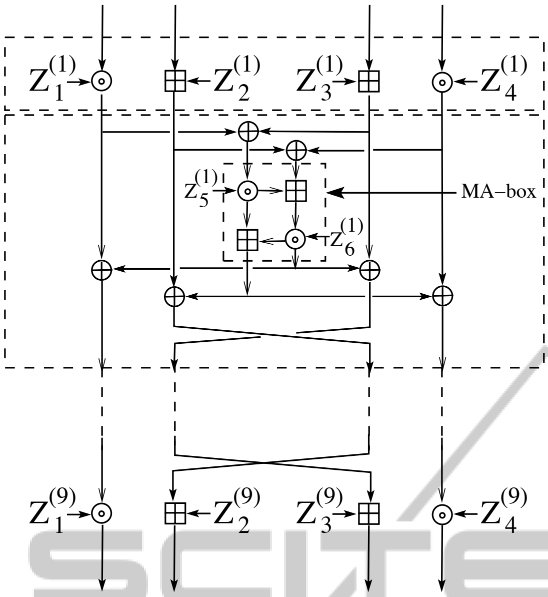

Figure 1: Computational graph of the IDEA cipher.

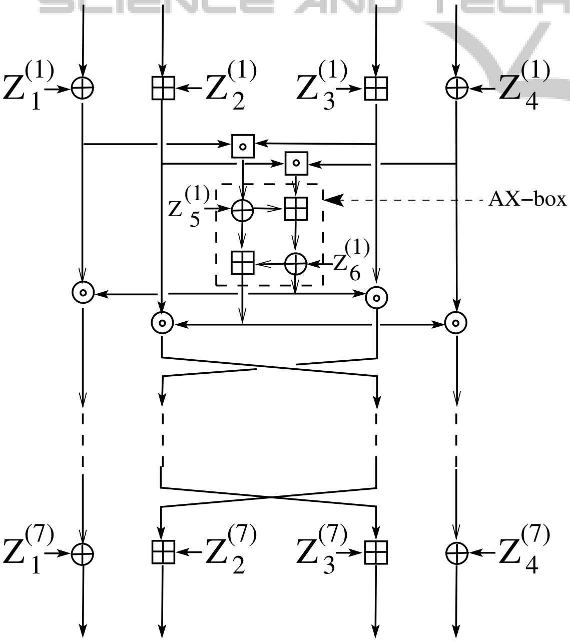

Figure 2: Computational graph of IDEA* for encryption.

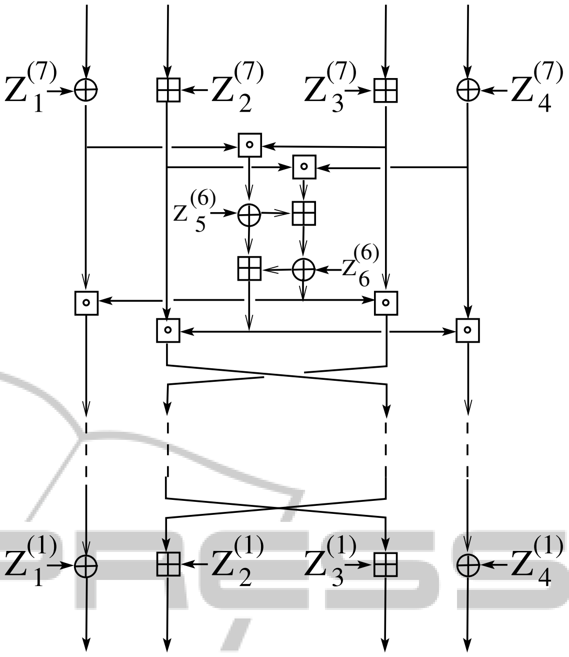

Figure 3: Computational graph of IDEA* for decryption.

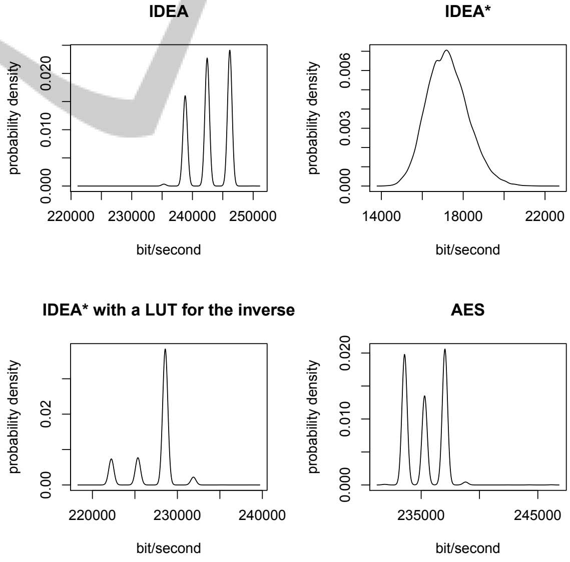

Figure 4: Probability density of the throughput (bit/second) of encryptions on an 8-bit microcontroller ATmega328P of four implementations (20,000 measurements): IDEA, IDEA*, IDEA* with a lookup table for the inverse and AES.