

# Shamir’s “cube attack”: A Remake of AIDA, The Algebraic IV Differential Attack

Michael Vielhaber

Hochschule Bremerhaven, FB2, An der Karlstadt 8, D-27568 Bremerhaven, Germany  
and

Instituto de Matemáticas, Universidad Austral de Chile, Casilla 567, Valdivia, Chile

vielhaber@gmail.com, 23.02.2009

**Abstract.** We show that the so-called “cube attack” by Dinur and Shamir is nothing but a restatement of the Algebraic IV Differential Attack by the present author, published one year earlier in a paper known to Dinur and Shamir — a stout plagiarism.

## 1 Introduction

On October 28, 2007 the Algebraic IV Differential Attack, or AIDA for short, was made public on the IACR eprint server as [eprint.iacr.org/2007/413](http://eprint.iacr.org/2007/413), see [1].

On September 13, 2008 (after a talk at CRYPTO ’08 in August 2008 — a plagiarism as *IACR Invited Lecture!*) Dinur and Shamir posted the paper “Cube Attacks on Tweakable Black Box Polynomials” [2] on the same medium as [eprint.iacr.org/2008/385](http://eprint.iacr.org/2008/385).

We show here that the two attacks are *identical*, that Dinur and Shamir *must* have known about this identity, and that [2] only gives some implementation details and certain generalizations, some more useful, some certainly not, but otherwise builds on AIDA *exactly* as described in [1], even again using TRIVIUM as the main example.

## 2 What is AIDA?

When simulating a cryptographic (Boolean) function, we effectively evaluate one position of its truth table (or DNF, Disjunctive Normal Form).

Some cryptographic functions (for example the output of TRIVIUM prior to its full setup length, not visible in normal operation), have a sparse part in its ANF, Algebraic Normal Form. If this part turns out to be linear in the key bits, it will allow to cut in half the necessary search space, for each such sparse equation found.

AIDA is the application of the Inclusion-Exclusion-Principle to evaluate this sparse ANF term via  $2^{|I|}$  DNF evaluations, where  $|I|$  is the number of bits of the initialisation vector (IV bits  $IV_k, 1 \leq k \leq \#IV$ ), set in the sparse term (constant key  $\underline{K}$ ):

$$f(IV_1, \dots, IV_{\#IV}, \underline{K}) = \bigvee_{I \subset \{1, 2, \dots, \#IV\}} d_I(\underline{K}) \bigwedge_{i \in I} IV_i \bigwedge_{i \notin I} \overline{IV}_i (DNF) = \bigoplus_{I \subset \{1, 2, \dots, \#IV\}} a_I(\underline{K}) \bigwedge_{i \in I} IV_i (ANF)$$

with  $d_I(\underline{K}), a_I(\underline{K}) \in \mathbb{F}_2$  and  $a_I(\underline{K}) = \bigoplus_{J \subset I} d_J(\underline{K})$  by the Inclusion-Exclusion-Principle.

### 3 Are both attacks one and the same?

They are indeed, as we will see step by step. Since Dinur and Shamir reinvented every piece of notation, although describing the exactly same situation, here comes sort of a translating dictionary. We will indicate notations and terms introduced in my AIDA paper [1] by underlining and a superscript <sup>[1]</sup>. The corresponding notations and terms that were used by Dinur and Shamir in [2] will be indicated by <sup>[2]</sup>.

**The observable information** from the cryptosystem (at some time step  $t$ ) is named  $OUT(t)$ <sup>[1]</sup> (or  $OUT(t) : IV_I$ <sup>[1]</sup>, if the initialisation vector  $IV$  is specified), and  $p(x_1, \dots, x_n)$ <sup>[2]</sup>, respectively.

**The IV bits spanning the hypercube** are those in  $I$ <sup>[1]</sup> =  $\{i_1, \dots, i_n\} (\subset \{1, \dots, 80\})$  in the case of TRIVIUM), and  $I$ <sup>[2]</sup>. Its product (logical *and*) is called  $IV_I = \bigwedge_{i \in I} IV_i$ <sup>[1]</sup>, and  $t_I = \bigwedge_{i \in I} x_i$ <sup>[2]</sup>, respectively. Observe, that  $n$ <sup>[1]</sup> =  $|I| \leq \#IV < n$ <sup>[2]</sup> =  $\#IV + \#K$ .

**The ANF with IV term  $IV_I$ <sup>[1]</sup> =  $t_I$ <sup>[2]</sup> factored out** is  $OUT(t) = (IV_{i_1} \wedge \dots \wedge IV_{i_n}) \wedge K_k \oplus Z$ <sup>[1]</sup> =  $p(x_1, \dots, x_n) = t_I \cdot p_{S(I)} + q(x_1, \dots, x_n)$ <sup>[2]</sup>, where the expected linear or quadratic term in the key bits is  $K_k$ <sup>[1]</sup> =  $p_{S(I)}^{[2]}$ , or in the more general case  $[\bigoplus_{k=1}^n X_k]$ <sup>[1]</sup> =  $p_{S(I)}^{[2]}$ , and where the large part of the ANF that is avoided through AIDA, all those  $2^{\#IV} - 1$  terms with the IV part different from  $IV_I$ <sup>[1]</sup> =  $t_I$ <sup>[2]</sup> is  $Z$ <sup>[1]</sup> =  $q(x_1, \dots, x_n)$ <sup>[2]</sup>.

**The sum over the  $2^{|I|}$  hypercube values** is

$$\underline{t : \langle i_1, \dots, i_n \rangle^{[1]} = \bigoplus_{J \subset I = \{i_1, \dots, i_n\}} OUT(t) : IV_J^{[1]} = p_I^{[2]} \triangleq \sum_{v \in C_I} p|_v^{[2]},}$$

respectively, as the overall sum of  $2^{|I|}$  observed simulations.

### 4 Is the scientific content the same?

The main (mathematical) results are Proposition 3 of [1] and Theorem 1 of [2]:

**Proposition 3**<sup>[1]</sup>

If  $OUT(t) = (IV_{i_1} \wedge \dots \wedge IV_{i_n}) \wedge K_k \oplus Z$ , then  $K_k = \bigoplus_{J \subset I} OUT(t) : IV_J$ .

**Theorem 1**<sup>[2]</sup>

For any polynomial  $p$  and subset of variables  $I$ ,  $p_I \equiv p_{S(I)} \pmod{2}$ .

From the previous section, we have the identities  $OUT(t)$ <sup>[1]</sup> =  $p$ <sup>[2]</sup>,  $K_k$ <sup>[1]</sup> =  $p_{S(I)}^{[2]}$ , and  $\bigoplus_{J \subset I} OUT(t) : IV_J$ <sup>[1]</sup> =  $p_I^{[2]}$ , which shows the formal equivalence.

The mathematical equivalence of both statements follows in more detail from

$$\begin{aligned} & \underline{K_k^{[1]} = p_{S(I)}^{[2]}, \text{ and } p_I^{[2]} = \sum_{v \in C_I} p|_v^{[2]} = \sum_{v \in C_I} p(x_1, \dots, x_n)|_{(x_{i_1}=v_1, \dots, x_{i_{|I|}}=v_{|I|})} =} \\ & = \underline{\sum_{J \subset I} p(x_1, \dots, x_n)|_{(x_{i_j}=1, i_j \in J; x_{i_j}=0, i_j \notin J)} = \sum_{J \subset I} OUT(t) : IV_J = \bigoplus_{J \subset I} OUT(t) : IV_J^{[1]}.} \end{aligned}$$

Dinur and Shamir’s Theorem 2 is a special case of their Theorem 1, where in part 1 all key bits are set to zero, and in part 2 all key bits but  $x_j$  are set to zero.

## 5 Which phases does the attack consist of?

There are 4 phases:

Phase 1: Search for linear terms in the key bits.

This is the largest, most time-consuming phase of all. There are theoretically  $2^{\#IV}$  (*e.g.* in the case of TRIVIUM,  $2^{80}$ ) possible hypercubes to be considered that is “infinite” time to check them all. Dinur and Shamir report “several weeks” in [2, p. 17, l. 20] to find some equations. This is Phase A in [1], which has not been taken into account in the complexity formula in [2].

Phase 2: Given (up to)  $N := \#K$  linear equations, we have to sum over the corresponding  $N$  hypercubes of dimensions  $d_1, \dots, d_N$ , requiring  $\sum_{k=1}^N 2^{d_k}$  simulations of the cipher under attack.

This step can take a few days with the  $d_k$ ’s approaching 40 (or more for higher dimensions) and is the other phase with relevant runtime. This is Phase B in [1], the order of magnitude given in [2] is  $2^{d-1} \cdot N$ .

Phase 3: Given  $N$  equations in  $N$  key bits, the  $N \times N$  0/1-matrix can be inverted by Gaussian elimination in  $N^3$  bit operations.

Even for  $N = 256$  and thus  $N^3 = 2^{24}$  bit operations, this would take under a second on a PC. This phase therefore has been omitted in [1], because it is negligible. It appears in Section 4.1 in [2].

Phase 4: Having the inverted matrix from Phase 3, the results from Phase 2 can be plugged in to yield the key bits in  $N^2 \leq 2^{16}$  bit operations. This phase has also been omitted in [1] due to irrelevance. It adds the term  $N^2$  to the overall complexity in [2].

### Notes

**1.** Dinur and Shamir completely ignore the most time-consuming part, Phase 1, in their asymptotic formula  $2^{d-1}N + N^2$ , but emphasize the irrelevant speedup of Phase 4 by precomputation (Phase 3).

Phase 1 is expressly excluded: in [2, p. 1] (asymptotic formula); in “does not need adaptations” [2, p. 4, l. 11] (we *do* need to repeat the search phase 1 all over for a new cipher); and finally in [2, p. 6, l. –6] “a *given* black box master polynomial and a maxterm  $t_I$  in it” the maxterm is *given* by whom? This is unsound!

To cite Dinur and Shamir [2, p.7]:

*“Since the preprocessing phase has to be executed only once for each cryptosystem whereas the online phase has to be executed once for each key, some cryptanalytic attacks ‘cheat’ by allowing extremely expensive operations during an unbounded preprocessing phase which make the whole attack impractical.”* Well observed . . .

**2.** In actual attacks, it turns out that the 0/1-matrix is extremely sparse (both in [1] and [2]). In Section 5 of [3], we will develop a model, which shows, why this sparsity is indeed to be expected. Therefore, Phases 3 and 4 (and Lemma 1 of [2]) are even more irrelevant.

## 6 Is there evidence that Dinur and Shamir knew they were “copy”ing from AIDA?

Unfortunately, yes indeed, considering the following paragraph about the attack [2, p. 17, l. 4f]: We have put three layers here ...

“*the citation itself from [2], describing my results in [1]*”,  
{the same topic, but about the results of Dinur and Shamir in [2]},  
[an explanation in terms of [2]].

“*Vielhaber [27] recovered 47 {Dinur and Shamir recover 53} key bits of Trivium with 576 {735} initialization rounds in negligible time. The key bits were recovered after some small IV special subsets were found [those  $IV_I$  leading to a linear equation in the key bits, ‘maxterms’ in [2] notation], each one with the following property: the result of summing up some keystream bit produced by assigning a special subset all possible IV values [summing over the corresponding hypercube], while keeping the other IV bits fixed [in fact, fixed to zero], is equal to some key bit or the sum of two key bits [applying their Theorem 1 or my Proposition 3]*”.

Well enough, this correctly describes the cube attack — however Dinur and Shamir actually use this last paragraph to describe AIDA in their Section 7.2.

Hard to believe that they still did not stumble over the fact that AIDA and the “cube attack” [2] are one and the same thing.

In order to veil the plagiaristic nature of [2], Dinur and Shamir try to put [1] into the “statistical attacks” corner [2, p. 4, l. 9]. Thereafter, they conveniently compare the “cube attack” (to its advantage) with a certain statistical attack instead of with the closer (identical!) and thus more appropriate AIDA. Note that [1] directly and *precisely* gives key bits, and not just statistical approximations. Compare our table on page 5 of [1] with theirs at the end of [2]: Same format, same results — just a larger setup length.

## 7 Is the cube attack “more general” than AIDA?

Dinur and Shamir comment on AIDA with these words [2, p. 17, l. 7f.]:

“[The result] is equal to some key bit or to the sum of two key bits. Note that this is a *very special* case of our cube attack, and it is not clear why the author imposed this *unnecessary restriction*.” (emphasis added)

In fact, the actual results in [1] lead to equations with one or at most two linear terms (and no nonlinear ones) in the key bits. However, the same applies to the results of Dinur and Shamir themselves (see their Tables 1–3 in [2]): Table 1 includes 41, 15, and 6 (linear!) equations with 1, 2, and 3 key bits, respectively, Table 2 has 36, 9, and 6, respectively, such linear equations, and Table 3 consists entirely of entries with a single key bit. Section 5 of [3] explains, why this is to be expected and is not at all an “unnecessary restriction”.

On the other hand, we clearly state in [1, p. 4] (Phase A) that ANF entries of the form  $IV_I[\oplus_{k=1}^n X_k]$  that is  $n$  key bits linearly added, or even using  $X_k = K_{k_1} \wedge K_{k_2}$  that is quadratic terms, may have to be processed as well. We repeat this generalization to more bits or (slightly) higher order terms in [1, Section 7].

While [1] mainly treats TRIVIUM as target, AIDA is described as a general “attack against low-complexity algorithms” [1, p. 6]. By the way, the only results given in [2] are also just against reduced-round versions of TRIVIUM.

So there is nothing “more general” or “less restricted” in [2], as the authors claim.

## 8 What else does [2] deliver?

**Linearity tests:** Section 4.2 contains the Blum-Luby-Rubinfeld linearity test, which is indeed a useful means to speed up Phase 1 (more useful speedup measures will be introduced in [3]). This is of course only just a technical implementation detail and by no means reason to pretend novelty for the attack itself.

**Trivialities:** As has been said already, everything to do with Phases 3 and 4 (Lemma 1, Gaussian elimination and so forth) does not speed up the attack, which is dominated by Phases A and B (Phases 1 and 2).

**Irrelevancies:** Sections 4.1 and 6 on “ $d$ -random polynomials” and a corresponding “crypto” device (10000 key bits, but *linear* feedback, and an output combiner of degree only 16) is a nice theory, but does not enhance the actual applicability of the attack.

## Conclusion

The “cube attack” by Dinur and Shamir is exactly the Algebraic IV Differential Attack, as posted on the IACR eprint server one year earlier, see [1].

The authors of the cube attack have failed to recognize this priority. As a result, the “cube attack” paper is a plagiarism.

It remains only to appeal to the scientific professionalism and honesty of the cryptographic/cryptologic community to not tolerate this case of intellectual property theft and henceforth refer to the attack exclusively as Algebraic IV Differential Attack, or AIDA for short.

## References

- [1] THE ORIGINAL: Michael Vielhaber, “Breaking ONE.FIVUM by AIDA an Algebraic IV Differential Attack” **28 Oct 2007** <http://eprint.iacr.org/2007/413>
- [2] THE COPY: Itai Dinur, Adi Shamir, “Cube attacks on tweakable black box polynomials” **13 Sep 2008** <http://eprint.iacr.org/2008/385>
- [3] Michael Vielhaber, “The Algebraic IV Differential Attack: AIDA attacking the full Trivium” (to appear)