Algebraic Cryptanalysis of the Data Encryption Standard

Nicolas T. Courtois[1] and Gregory V. Bard[2]

[1]University College of London, Gower Street, London, UK,
n.courtois@ucl.ac.uk
[2]Fordham University, NY, USA
Gregory.Bard@ieee.org

Abstract. In spite of growing importance of AES, the Data Encryption Standard is by no means obsolete. DES has never been broken from the practical point of view. The triple DES is believed very secure, is widely used, especially in the financial sector, and should remain so for many many years to come. In addition, some doubts have been risen whether its replacement AES is secure, given the extreme level of "algebraic vulnerability" of the AES S-boxes (their low I/O degree and exceptionally large number of quadratic I/O equations).
Is DES secure from the point of view of algebraic cryptanalysis, a new very fast-growing area of research? We do not really hope to break it, but just to advance the field of cryptanalysis. At a first glance, DES seems to be a very poor target — as there is (apparently) no strong algebraic structure of any kind in DES. However in [14] it was shown that "small" S-boxes always have a low I/O degree (cubic for DES as we show below). In addition, due to their low gate count requirements, by introducing additional variables, we can always get an extremely sparse system of quadratic equations.
To assess the algebraic vulnerabilities is the easy part, that may appear unproductive. In this paper we demonstrate that in this way, several interesting attacks on a real-life "industrial" block cipher can be found. One of our attack is the fastest known algebraic attack on 6 rounds of DES. Yet, it requires only **one single known plaintext** (instead of a very large quantity) which is quite interesting in itself.
Though (on a PC) we recover the key for only six rounds, in a much weaker sense we can also attack 12 rounds of DES. These results are very interesting because DES is known to be a very robust cipher, and our methods are very generic. They can be applied to DES with modified S-boxes and potentially other reduced-round block ciphers.

Key Words: block ciphers, algebraic cryptanalysis, DES, s^5DES, AES, solving overdefined and sparse systems of multivariate equations, ElimLin algorithm, Gröbner bases, logical cryptanalysis, SAT solvers.

1 Introduction

According to Shannon, breaking a good cipher should require "as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type" (see [42]). For example, the problem of key recovery in AES given one known plaintext can be written as solving a system of 4000 multivariate quadratic equations, see [13, 14]. In general, this problem (called the MQ problem) is NP-hard, and solving this particular system remains a very ambitious goal. Nevertheless, there is a growing body of positive results: systems of equations that arise in the cryptanalysis of block, stream and public-key encryption schemes, turn out to be — for some specific reason — efficiently solvable, see [37, 12, 11, 23, 26, 16, 20, 18, 19], to quote only some major results. Yet the potential of efficiently solving certain multivariate systems of equations with special properties is still underestimated in scientific community. For example, in 2002, Courtois and Pieprzyk, have conjectured that sparse systems of equations are in general much easier to solve than dense systems of the same size. In 2006, Courtois Bard and Jefferson have discovered that SAT solvers, but also known Gröbner bases algorithms such as F4, can in fact solve efficiently very sparse systems of multivariate quadratic equations (dense MQ is a known NP-hard problem) [1]. To the best of our knowledge no researcher have so far demonstrated such working attacks on systems of equations of comparable size. In this paper use very similar methods, but instead of randomly generated sparse systems, we use systems of equations derived from a real-life block cipher. As a result, much larger systems can be solved in practice.

The rest of the paper is organized as follows: In the next section we study several methods of writing equations for DES. In Section 3 we summarise our attacks, explain in details important previous and related work, and give a complete description of a couple of (best to date) attacks we did perform. In Section 4 we compare algebraic cryptanalysis of DES to AES, and algebraic cryptanalysis to differential and linear cryptanalysis. In Section 5 we show one example in which our attacks can solve systems of equations that have many solutions more easily. Then comes the conclusion.

2 Algebraic Vulnerabilities of DES S-boxes

Unlike AES, there is no special algebraic structure in DES S-boxes that makes them particularly vulnerable. In most of this work, we treat them exactly as any other S-box of the same size. (These attacks should therefore also work on DES with any modified set of S-boxes and also give few examples for s^5DES, a clone of DES designed to be more resistant known cryptanalytic attacks [28]).

The S-boxes in DES have $n = 6$ inputs and $m = 4$ outputs. There are many ways in which one can write I/O equations for these S-boxes. The speed and the success of the algebraic attack will greatly depend on how this is done. In our work we consider the following three classes of equations that, heuristically, seem to be relevant to algebraic cryptanalysis:

- Class 1. Low-degree multivariate I/O relations (cf. definition below),
- Class 2. I/O equations with a small number monomials (can be of high or of low degree),
- Class 3. Equations of very low degree (between 1 and 2), low non-linearity and extreme sparsity that one can obtain by adding additional variables.

We have tried several types of equations falling in one of the above categories, as well as a number of their combinations. We have computed and tested all the equations we consider in this paper, (and some others), and they can be obtained on demand from the authors.

Very little is known about what approach would make an algebraic attack efficient and why. In our simulations, though the last Class number 3 seems to be the best choice, all the three do in fact give solvable systems of equations for several rounds of DES, in spite of the fact that some of them are substantially larger in size. We anticipate that better methods for writing DES as a system of equations should be proposed in the future, and we consider the question of finding the "best" representation as an important research topic in itself.

2.1 Low-degree Multivariate I/O Relations

The following notion plays an essential role in algebraic attacks on LFSR-based stream ciphers, see [16, 8] as well as for a couple of (weak) block ciphers [27, 20].

Definition 1 (The I/O degree, [8, 2]). *Consider a function $f : GF(2)^n \rightarrow GF(2)^m$, $f(x) = y$, with $x = (x_0, \ldots, x_{n-1})$, $y = (y_0, \ldots, y_{m-1})$.*
The I/O degree of f is the smallest degree of the algebraic relation

$$g(x_0, \ldots, x_{n-1}; y_0, \ldots, y_{m-1}) = 0$$

that holds with certainty for every pair (x, y) such that $y = f(x)$.

The minimum number (and frequently the exact number) of equations of some type that do exist for one S-box can be obtained by applying the following theorem:

Theorem 1 (Courtois [14, 16, 19]). *For any $n \times m$ S-box, $F : (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$, and for any subset $\mathcal{T}$ of t out of 2^{m+n} possible monomials in the x_i and y_j, if $t > 2^n$, there are at least $t - 2^n$ linearly independent I/O equations (algebraic relations) involving (only) monomials in $\mathcal{T}$, and that hold with probability 1, i.e. for every (x, y) such that $y = F(x)$.*

Proof (sketch). All the monomials can be rewritten as a function of n variables and their Algebraic Normal Form (ANF) belong to a linear space of dimension 2^n. If their number is bigger than the dimension, there will be at least $t - 2^n$ linear dependencies between these ANF, and the same linear dependencies will also hold for the original monomials. $\square$

Example of Application of Theorem 1 to DES

For example, we can consider the equations of the following type:

$$\sum \alpha_{ijk} x_i y_j y_k + \sum \beta_{ijk} x_i x_j y_k + \sum \gamma_{ij} x_i y_j + \sum \delta_i x_i + \sum \epsilon_i y_i + \eta = 0$$

These equations are of degree 3. The total number of monomials that arise in these equations is $t = 1 + nm + n + m + n \cdot m(m-1)/2 + m \cdot n(n-1)/2 = 131$. By straightforward application of Theorem 1 we get:

Corollary 1. *For any 6×4 S-box (not only a DES S-box) the number of linearly independent equations r of this type is at least:*

$$r \geq t - 2^n = 67.$$

Table 1. The Real Number of Equations Observed for Different S-boxes

DES S-box	1	2	3	4	5	6	7	8
$r =$	67	67	67	67	68	68	67	67

Thus, for any 6×4 S-box (not only a DES S-box) there are at least $r \geq t - 2^n = 67$ such equations. In practice, for DES S-boxes, we get sometimes 67, sometimes 68:
Remark: Apparently the S-boxes of DES behave more or less as random S-boxes of the same size, however with this type of equations it seems that we can still "distinguish" them from random. It appears that, for all types of I/O low degree equations, with t approximatively above 131, the difference between the DES S-boxes and random S-boxes is no longer visible. This is an important remark because it means that (at least with these and similar types of equations) we do not expect algebraic attacks to be actually more efficient on DES itself compared to versions with modified or random S-boxes. And in general, the attacks should also work for many other ciphers.

Fully Cubic Equations

We also consider fully cubic equations in the 10 variables x_i and y_i. We have

$$t = 1 + (n+m) + (n+m)(n+m-1)/2 + (n+m)(n+m-1)(n+m-2)/6 = 176,$$

and thus $r \geq t - 2^n = 112$. Computer simulations give exactly 112 for all the 8 S-boxes of DES. Here we can no longer see any difference between DES and random S-boxes. The same numbers are obtained for s^5DES [28].
Remark: one fully functional example of equations for the full 16-round DES based on these (cubic) polynomials can be downloaded from [7]).

Sparse Cubic Equations

We can observe that not all cubic equations we found are dense. In this section we give the number of very sparse cubic equations that we found for different DES S-boxes. These equations have up to 8 monomials. We expect that more sparse cubic equations can be found. It is possible that replacing our 112 equations by a smaller but particularly sparse subsystem of equations (if it uniquely defines the S-box) gives better results in some of our attacks. Currently we do not know a convincing example.

Table 2. The Number of Sparse Cubic Found for Different S-boxes

DES S-box	1	2	3	4	5	6	7	8
$HW \leq 6, r =$	2	4	6	1	15	1	0	2
$HW \leq 8, r =$	3	8	11	13	17	8	1	7

2.2 Quadratic Equations

Though no theorem guarantees their existence, for certain S-boxes, there are also quadratic equations for DES S-boxes. Their number is not very large and they cannot alone be used to mount an algebraic attack. In comparison, for s^5DES [28], there is more quadratic equations, but the number remains quite small and they do not allow to uniquely define the S-boxes.

Table 3. Quadratic Equations Observed for Different S-boxes

S-box	1	2	3	4	5	6	7	8
DES	1	0	0	5	1	0	0	0
s^5DES	3	3	3	4	3	3	3	3

I/O Equations of Degree 4

We have $t = 386$, $r \geq t - 2^n = 322$. We obtain exactly this many for each S-box.

Remark. For all above mentioned types of low-degree equations, it is possible to delete some equations, for example taking every second equation. This leads to systems that are less over-determined and should give worse results in Gröbner basis attacks. However in some SAT attacks it seems to give slightly better results.

2.3 Relations with a Very Small Number Monomials

These equations were first proposed and studied in [17]. First, we study equations that can be of arbitrary degree but that contain only one monomial. These are called monomial equations in [17]. For example $x_1x_2x_5y_3y_4 = 0$. One should note that we count 1 as a monomial and the equation $x_1x_2x_5y_3y_4 = 1$ would be counted as a binomial equation. We have also studied and computed equations with 3 and 4 monomials.

Since linear combinations may ruin the sparsity of equations (that's our focus), all these equations do not have to be linearly independent. Still, from our count of binomial equations we exclude those that are trivial because they are linear combinations of simpler, monomial equations. Similarly, from our count of trinomial equations we exclude equations that would be a XOR of one monomial and one binomial equation, etc. The number of equations with 4 monomials is getting already quite large, however it is possible to select among these a smaller subset of equations that will also have a substantially lower degree (e.g 4 instead of maximum 10). We have decided to limit the sum of the degrees of the 4 monomials to 15 which also forces the degree to be ≤ 4 and to have some monomials of degree 3. For example, for DES S-box S1, we have the following equation $0 = x[1]x[5]x[32] + x[1]x[2]x[5] + x[1]x[3]x[4]x[5] + x[1]x[5]y[31]$, Here, the bits are numbered not according to their position in the S-box, but from 1 to 32, according to their position in the whole round function of DES. The sum of degrees in this equation is $3 + 3 + 4 + 3 = 13$.

In the following table we give the numbers of equations of each type we found for DES, and compare with results obtained for several randomly generated S-boxes of the same size.

Table 4. Equations that Contain a Small Number of Monomials in DES

	random S-box	DES S-box 1	2	3	4	5	6	7	8
1 monomial	$0 - 463$	170	140	179	145	207	154	153	173
2 monomials	$233 - 524$	360	385	322	362	303	345	379	329
3 monomials	$1 - 112$	123	125	56	66	74	115	81	99
4 monomials	$1880 - 6106$	716	608	771	567	484	543	750	448
4 m; $\sum$ deg ≤ 15	$250 - 1053$	87	73	104	57	86	104	94	75

Remark 1. We observe that for a random S-box, the number of equations of different types is rather strongly variable, On the contrary, all the DES S-boxes give quite similar results and clearly these equations are a good method to distinguish the DES S-boxes from a random function. We note also that monomial equations have a curious property that, for a random S-box, it is not totally unusual to have 0 such equations.
Remark 2. When equations of this type are used alone to describe DES (especially with a single plaintext/ciphertext pair), and the key is computed by an algebraic attack, they typically will *not* uniquely define the solution to the system. This is because typically, when all $y_i = 0$ and regardless the value of x, these equations will all be satisfied (!). Though in some cases (by miracle or by chance) we still were able to recover the right key by our attacks, we advocate the usage of these equations in conjunction with some other equations that permit the removal of spurious solutions to systems of equations. Our experience shows that in some cases, mixing these equations with more traditional I/O equations of degree 3 gives a faster attacks than with cubic equations alone.

2.4 Equations with Additional Variables

Equations Related to Efficient Hardware Implementation. By adding up to 52 additional variables per S-box and per round, it is possible to dramatically reduce the size of equations, increase their sparsity and decrease their non-linearity. All equations will have either 0 or 1 nonlinear monomial. There are many different methods to achieve this, and ours is directly derived from the low-gate count non-standard representation of DES that has been developed by Matthew Kwan, see [29]. These are our "favorite equations" so far, and one example of system of equations that contains all these (exact) equations we use can be downloaded from [7]). In practice, we have observed a speedup factor between 2 and 20 compared to the same attack done with the sets with 112 cubic equations per S-box.

Quadratic Representations with a Minimum Number of Added Variables. In the previous version, we add as much as up to 52 additional variables. One can do much better and it is possible to see that, due to the size of the DES S-boxes, by adding just one variable the degree of the equations collapses from 3 to 2. More generally we have:

Theorem 2. *For every S-box with 6 input bits and 4 output bits, if we add* **any** *additional variable that is defined by an arbitrary Boolean function of 6 input bits, the number of linearly independent quadratic equations of degree 2 with these* $4+6+1 = 11$ *variables is at least 3.*

Proof. Indeed, following a reasoning as in Theorem 1, with 11 variables instead of 10, there are $\binom{11}{2} + 11 + 1 = 67$ quadratic monomials, while the number of cases is 64. Therefore there are at least 3 quadratic equations.
We do not know a satisfactory choice for the additional variable to be added. More research about quadratic representation of DES S-boxes is needed.

3 Our Attacks on DES

3.1 Summary

From our equations on the S-boxes, it is easy to write a system of multivariate equations that describe the whole cipher. This system will be of degree 2, 3, 4 or more, depending on which equations we use for the S-boxes. This system should have a unique solution (if it is not the case one should either fix some variables or use some extra equations). Examples of such systems of equations can be downloaded from [7]. Interestingly, though almost all researchers in cryptography we know believe that there is no method whatsoever capable of solving (in practice) such systems of equations, we have discovered two totally different families of methods (that are of very different nature) and that both work quite well.

1. The first is a particularly simple elimination algorithm called ElimLin, which can be see as a very simplified version of known Gröbner bases algorithms. It is fully described in Section 3.3.
2. The second is a simple and straightforward ANF to CNF conversion method following [1]. For each monomial in the equations we add a dummy variable, and CNF equations that relate it logically to variables and sub-monomials. To encode long XORs we use additional dummy variables and obtain shorter XORs. When the conversion is done, we obtain a large SAT problem, on which we run MiniSat 2.0, a very efficient and one of the latest SAT solvers, that is freely available on the internet with source code [34].

These methods can also be combined as follows: first we derive additional equations (not always sparse) by ElimLin (or by using other methods such as F5 [23]), then we add these new equations with the initial (very sparse) equations, then we run the conversion and then MiniSat.

3.2 Related Work and What's New

In the past, many researchers quite naturally wondered if DES could be broken by solving a system of Boolean equations, see for example [43, 25] and Section 4.3.2. of [22]. The idea was known as a method of "formal" coding. Unhappily, most people worked with a "functional" approach to describing S-boxes and whole rounds of the cipher. This is a very strong limitation that overlooks a wide range of attacks, see [8, 14, 37, 27]. Nevertheless, at Crypto'85 Chaum and Evertse looked at bits (and their linear combinations) inside the DES encryption, that do not depend on some key bits (or their linear combinations), see [4]. If a bit can be found that computed in the forward direction from the plaintext, and computed from the ciphertext in the backwards direction, this bit gives an equation that does not depend on some key bits. Such equations can be used to speed-up the exhaustive search and for 6 rounds of DES, an attack 2^2 times faster than brute force is reported. This can be seen as the first algebraic attack on a reduced version of DES (our best attack will be faster). The modern concept of algebraic cryptanalysis using arbitrary algebraic relations, see [8, 14, 37, 27] is much richer in possibilities and working attacks. Our results should be compared with previous work on solving very large systems of multivariate equations and to previous successful attacks on general block ciphers with no special/algebraic properties such as in [4]. None of our solving methods is completely new. The use of Gröbner bases for solving systems of equations derived from a cipher has become very popular since [14], yet no convincing attacks on block ciphers were reported so far. The use of SAT solvers to break 3 rounds of DES have previously been shown to be feasible by Massacci and Marraro [32]. The authors of [32] call it "logical cryptanalysis" to emphasise the "automated reasoning" view. We consider this to be a part of "algebraic cryptanalysis" especially that we do not write SAT systems directly, but first write multivariate low-degree equations, then work on general-purpose conversion. We also consider that the methods of abstract algebra include and go beyond classical logic and reasoning. Unlike as in [32], our method — write equations, convert and solve — is very general and applicable to any block or stream cipher. It has an interesting property that the equations can be combined with any set of "additional" equations that are typically derived in Gröbner bases-like and related algorithms. SAT solvers may then be also used as a tool to complete any algebraic attack that does not work (sufficiently well) by itself, and this could be interesting because SAT solvers make heuristic guesses based on non-algebraic (but rather statistical) observations.

Unhappily, the 3 rounds of DES broken in [32] are arguably very weak, even with one single plaintext, and the authors report "an abrupt jump in complexity" at 4 rounds. Maybe for this reason the result remained almost unnoticed in the cryptographic community. In this paper we break twice as many rounds as anyone could ever break given such small quantity of known plaintexts (after this paper has been written, Raddum and Semaev have proposed yet another approach of this type, that unfortunately works only for up to 4 rounds of DES, and runs out of memory for 5 rounds, see [38, 39]). Our attacks on DES are the first to be be faster than the (older) algebraic attack on 6 round of DES [4]. Our methods are also clearly of much broader applicability.

The immediate contribution of this paper is to show that some very sparse systems of multivariate low-degree equations over smal finite fields derived from block ciphers can be solved in a matter of seconds on a PC. This by both our conversion to SAT, as well as techniques in the line of Gröbner bases (in fact we only worked with extremely simple monomial elimination tools that were however highly optimised in terms of memory management, and the order of operations was rearranged to conserve sparsity). One can wonder to what extent the systems we are solving here are special (i.e. weak)? It is very hard to know what exactly makes systems efficiently solvable, but it appears that sparsity alone will make systems efficiently solvable, both by SAT solvers and classical Gröbner bases methods, see [1]. One may notice that in the past, a reduction from the MQ problem to SAT, has been used to show that MQ was NP-hard. Now it is being used to solve very large instances of MQ that were believed intractable to handle.

3.3 Examples of Working Attacks — Fast Algebraic Attacks on Block Ciphers

We start with a very simple yet remarkable algebraic attack that we call ElimLin. The ElimLin function works as follows: we take the initial system (that is of degree 2 or 3) and look if there are linear equations in the linear span of the equations. If so we can eliminate several variables, by simple substitution by a linear expression. Then, quite surprisingly, new linear equations can be obtained, and this can go on for many, many iterations. This process is repeated until no more linear equations can be found. The order of variables is such that the variables that appear in the smallest number of equations are eliminated first, which helps to preserve sparsity, while key variables are eliminated at last.

ElimLin alone gives very good results, given its extreme simplicity. We write a system of 112 fully cubic equations per S-box following Section 2.1, for 4 full rounds of DES, and for one known plaintext. We fix first 19 key bits to their real values. 37 remain to be determined. The time to compute 2^{36} times 4 rounds of DES on our 1.6 GHz Centrino CPU can be estimated to be about 8000 seconds. Instead, ElimLin takes only 8 seconds to find the correct solution. Attacks on 5 rounds can still be (marginally) faster than brute force. For example, with 3 known plaintexts and 23 variables fixed, we compute the key in 173 seconds, compared to about 540 s that would be needed by a brute force attack.

With eliminate ElimLin we did not go very far, but still we do two more rounds than in [31]. We observed that strictly better results (in terms of feasibility) can be obtained with XL algorithm and the so called T' method [13, 14, 21], or algorithms such as F4 or F5, however we do not report any results with these, as they do not really go much further, and we feel that our implementation of these still needs improvement, and the T' method is not optimal (in particular it computes the same equations many times). We have also tried ready packages such as MAGMA [30] and Singular [40], and found that these systematically run out of memory on our examples due to (apparently) lack of adequate support for sparse elimination on large systems, and this even on some simple examples we could solve completely with ElimLin in less than 1 hour.

Comparison to s^5DES. The same attacks work on s^5DES and the attack on 5 rounds with 3 chosen plaintexts is about 8 times faster. This might be due to the fact that for s^5DES, a large subset of equations we use here are in fact of degree 2, see Section 2.2.

3.4 Examples of Working Attacks — Attacks with Conversion to SAT

With a very simple early version of our ANF to CNF converter, we write a system of quadratic equations with additional variables as described in section 2.4. We do it for full 6 rounds of DES, fix 20 key variables (it does not really matter which) and do the conversion that takes few seconds. Then with the latest version of MiniSat 2.0. with pre-conditioning we compute the key in 68 seconds while the exhaustive search

would take about 4000 s. The complexity to recover full 56-bit key by this attack is about 2^{48} applications of reduced DES (feasible in practice).
Remark 1: We have tried if either MAGMA [30] or Singular [40] could solve this system of equations that we solve in 68 s. Both crash with out of memory message after allocating nearly 2 Gbytes. The memory usage reported by MiniSat is 9 Mbytes.
Remark 2: Unhappily, we cannot apply this attack to s^5DES, because it is based on special gate-efficient representation developed for DES [29], and no such representation of s^5DES is known.

4 Algebraic Cryptanalysis: the Great Challenge

4.1 Can Large Systems of Very Sparse Low-Degree Equations Be Solved Efficiently?

In our (best) system of equations in section 3.4 above, we have 2900 variables, 3056 equations and 4331 monomials. [1] The system is very sparse and compact, it has on average less than 1 non-linear monomial per equation. It is solved in 68 seconds.
We believe to be the first to show that such large systems of equations generated from a real-life cipher structure can be efficiently solvable. Obviously, not every system with similar parameters is efficiently solvable, and clearly the security of DES (as probably for any other cipher) against our attacks does quickly increase with the number of rounds.
Comparison to AES. Nevertheless, the following question can be asked, can we hope to break, say 6 rounds of AES by using SAT solvers? In comparison to ours, the binary system of equations proposed by Courtois and Pieprzyk in [14] has 4000 equations and 1600 variables: it is in fact overdefined and may seem easier to solve. Very unhappily, this system has substantially more monomials, about $137 \cdot 200 = 27400$, much more than a few thousands.[2]

4.2 Algebraic vs. Linear and Differential Cryptanalysis

Our vision of cryptanalysis changes each time a new cipher is considered, and each time we discover a new powerful attack. In the past DES has been thoroughly cryptanalysed by linear and differential cryptanalyses for up to 16 rounds. In this context our results may appear quite insignificant. We believe that, on the contrary our results are very good and are interesting for several reasons.
First, we can recover the key given one single known plaintext. A tiny amount of data needed by the attacker is maybe the most striking feature of algebraic cryptanalysis. This unprecedented quality of algebraic attacks has simply no equivalent

[1] Some equations are linear and if we eliminated them, we would have 1298 variables, 1326 equations and 10369 monomials. It would become less sparse (15 monomials per equation on average) but still very sparse. We don't do this, it makes the attack run slower.

[2] Another system of equations that describes the whole AES have been proposed by Murphy and Robshaw [36], and it contains on average less than one non-linear monomial per equation. This is very similar to ours, however their system is over $GF(256)$, not over $GF(2)$.

in any known cryptographic attack. It is precisely the reason why algebraic attacks are potentially very devastating, and this however immature and inefficient they are today. For example, from one single MAC computed by an EMV bank card with a chip that is printed on a customer receipt, one would recover the key of the card, and from this single key, the master key of the issuing bank that could be used to make false bank cards. Luckily, there is no reason to believe that this could happen in a foreseeable future.

Nevertheless, we contend that it is inappropriate to compare algebraic cryptanalysis with linear and differential cryptanalysis and claim it is slower. In a certain way, linear and differential cryptanalysis became the reference as a by-product of our incapacity to ever find any attack on DES, that would be better than exhaustive search in a realistic setting. Algebraic cryptanalysis, while still not very powerful and unable to break full DES, does slowly emerge as more or less the only branch of cryptanalysis that may work in real life (very few known plaintexts are available). We suggest that attacks that require only a very small number of known plaintexts should be considered as a research topic of its own right. They should mainly be compared just to other attacks of this type. Moreover, if we can agree that for DES algebraic cryptanalysis is currently no match compared to classical attacks, we may as well argue that actually none of these attacks are of practical importance. Both represent the current state of research in cryptology, and yet it is the algebraic cryptanalysis that is new and can still be improved a lot. (It will already improve just by using better SAT solvers and more powerful computers. For some systems we have observed a speed-up of a factor 8 between MiniSat version 1.4 and 2.0.)

One should also note that, the situation that we have for DES could be very different for AES. Since AES is, by design, very strong against differential and linear cryptanalysis, the number of rounds is accordingly quite small in AES, and the threat is indeed that some form of algebraic cryptanalysis could give better results for this cipher (comparatively to linear and differential attacks). However, since the initial attack proposal [13, 14], it seems that no visible progress is being made in this direction. Our feeling that, before attacking AES, we need to learn much more about algebraic cryptanalysis, and try it on many other ciphers. This was the main motivation of the present paper.

5 Algebraic Cryptanalysis As a Tool for Studying Ciphers

Algebraic (and logical) cryptanalysis is not only a tool for key recovery with unprecedented capabilities. It can also be used to solve many other problems that arise in cipher design such as detecting weaknesses, special properties, weak keys, finding collisions, second pre-images, long-range impossible differentials etc. In the past, these tasks were done manually by a cryptanalyst. In the very near future, they will be automated. We provide one example.

5.1 A Special-Property Finder on Full 12 Rounds of DES

Let `‘0123456789ABCDEF’` be a fixed DES key (one that we did not choose and has no special properties). We want to find an "educational" example of differential cryptanalysis for the first 12 rounds of DES with difference `(‘00196000’,‘00000000’)`, that comes from the best existing differential characteristic for DES, see [15]. It is known that this difference is reproduced after two rounds with probability exactly 2^{-8}, regardless the value of the key. The naive method to find a sample plaintext for which this difference holds throughout the whole computation is exhaustive search. For 10 consecutive rounds it requires about 2^{41} reduced DES computations and we estimate that it would take about 4 days on our laptop. For 12 consecutive rounds it requires about 2^{49} reduced DES computations which would last for about 3 years.
An algebraic approach to this problem is obvious: we can write this problem as a system of equations that has many solutions (we expect approximatively 2^{24} and 2^{16}, respectively). We have tried this approach. By using our (last) quadratic and very sparse representation of the S-box, and by converting it to SAT, we have managed to find a desired solution. For 10 rounds this is particularly easy, we do it in 50 seconds while in addition fixing 6 additional variables to values chosen by us (many different solutions can thus be obtained). For 12 rounds it is harder to do, and the solution was found in 6 hours (instead of 3 years). For example, one can verify that the plaintext `‘4385AF6C49362B58’` is a solution to this problem for 12 rounds and the key `‘0123456789ABCDEF’`.
Thus we are able to find a special property of 12 rounds of DES within a time much much smaller than the inverse of the probability of this property. This is a nice and unexpected result with unclear ramifications. The system of equations is very similar that in key recovery attacks, yet due to the multiplicity of solutions, it is much easier to solve and we could do it on a laptop PC for as many as 12 rounds. It is not a key recovery attack, but could be treated as a weak "certificational" algebraic attack on 12 rounds of DES.

5.2 Discussion

This attack is open to interpretation and discussion. How do we perceive and interpret a cryptographic attack greatly depends on how it compares with other attacks. This perception may change when we discover new attacks (for example, it would change if somebody have found another attack that would achieve a similar speed-up). Here is our current interpretation of this result. We encourage other researchers to challenge this interpretation.
DES with 12-rounds can be treated and used as a pseudo-random permutation generator. We have found a new weakness of this generator and this w.r.t. attackers disposing of a very low computing power (e.g. only 50 seconds on a PC for 10 rounds).
Thus far it was known that DES had a particular property w.r.t. differential cryptanalysis that happens with a small probability and that can be detected when treating it as a "black box". In a "glass box" scenario, when the key and the algorithm is known to the attacker, plaintexts that have these properties can be detected and generated

much faster. DES with 12 rounds cannot be treated as a "black box" or "random oracle" or "random cipher". We expect that our attack works for every possible DES key.

From differential cryptanalysis (a basic and naive application of it, could be improved) we already knew that DES with 12 rounds cannot be treated as a "black box" by an adversary that can do 2^{49} queries to the oracle. We also knew that it cannot be treated as a "black box" when the adversary can carefully choose the key — this is because DES has some weak keys (and there is also the complementation property). Here we learn that it cannot be treated as a "black box" when the key is random, and known to the adversary: the adversary can do more things than just implement 12 rounds of DES and experiment with it. The adversary does not have to be very powerful. He doesn't need to make 2^{49} queries to the oracle, and he needs a rather small computing power that is no match for computing answers for these 2^{49} queries. To summarize, DES with up to 12 rounds is not a very good permutation generator even against adversaries with very limited computing power.

5.3 Future Research

We believe that many other results of this kind can be obtained by our (and similar) methods. In particular, it appears that SAT solvers are particularly efficient in solving problems that have many solutions as demonstrated in recent work on hash function cryptanalysis with SAT solvers [35]. In general, we expect that it should be also possible to break certain hash functions and MACs by algebraic and/or logical attacks similar to ours, or in combination with other methods. It should be also possible to answer many questions such as: given a block cipher, is there a choice of a subset of key bits to be fixed such that there will be a differential true with probability 1 for 4 rounds. In some cases the attack would just consist of running ready computer packages designed to efficiently solve SAT instances or/and systems of multivariate equations and may require very little human intervention.

6 Conclusion

In this paper we show that in many interesting cases, it is possible to solve in practice very large systems of multivariate equations with more than 1000 unknowns derived from a contemporary block cipher such as DES. Several methods were considered, and our best key-recovery attack allows one to break in practice, up to 6 complete rounds of DES and given only 1 known plaintext (!). Very few attacks on ciphers that work given such a low quantity of plaintext material are known. At the same time, our approach is extremely general. It is clearly possible to use it to find algebraic attacks of this type in an automated way starting from the very description of a symmetric cipher, and without the need to find any strong property or particular weakness. This opens new avenues of research which is rich in possibilities (many different algebraic representations of the S-boxes) and in which experimentation is an essential ingredient.

Until now, direct attempts to attack block ciphers with Gröbner bases have given very poor results. In 2006 Courtois proposed a general strategy for "fast" algebraic attacks on block ciphers [9]. We need to avoid methods such as Gröbner bases that expand systems of equations to a larger degree (e.g. 4 or 5) and then solve them. Instead, we need to find methods to produce systems of equations that, though may be much larger in size, can be nevertheless much easier to solve and by much simpler techniques, without time and memory-consuming expansion. Here, linear algebra and known elimination techniques need to be complemented with heuristics that take advantage and (to some degree) preserve sparsity. Then, for attacks such as [9] and in the present paper, it appears that current Gröbner bases techniques are no match compared much simpler techniques such as ElimLin.

For DES (and also for KeeLoq, see [10]) it appears that the fastest algebraic attacks currently known the fastest algebraic attacks are those obtained with modern SAT solvers. Our specific approach is to write problems algebraically and work on conversion. This allows methods from both families to be combined in many ways. By just the few simple working examples we give in this paper, we have considerably enlarged the family of algebraic cryptanalytic methods available to the researchers.

It should be noted that we ignore why some systems of equations are efficiently solvable. We just demonstrate that they are. It is certainly an important topic for further research to understand why these attacks actually work, but it would be wrong to believe that only attacks that are well understood should be studied in cryptology. This is because the number of possible algebraic attacks that can be envisaged is very large: one finite DES S-box can be described by a system of algebraic equations in an infinite number of ways, and the attacks that should be studied in priority are the fastest ones, not the ones for which a nice mathematical theory already exists such as Gröbner bases. Moreover, if one does not experiment, or if one only studies attacks that are faster than linear and differential cryptanalysis some important attacks on block ciphers will never be discovered.

Another interesting contribution of this paper is to point out that while the performance of algebraic elimination methods is usually greatly degraded when the system of equations has many solutions, SAT solvers in fact do benefit from it. This potential remains largely unexplored, and may lead to interesting results in cryptanalysis of hash functions and MACs. As an illustration we computed a special property of 12 full rounds of DES.

References

1. Gregory V. Bard, Nicolas T. Courtois and Chris Jefferson: *Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers,* Available at `http://eprint.iacr.org/2007/024/`.
2. Daniel Augot, Alex Biryukov, Anne Canteaut, Carlos Cid (co-editor), Nicolas Courtois, Christophe De Cannière, Henri Gilbert (co-editor), Cédric Lauradoux, Matthew Parker, Bart Preneel, Matt Robshaw, and Yannick Seurin: *AES Security Report,* D.STVL.2 report, IST-2002-507932 ECRYPT European Network of Excellence in Cryptology public deliverable. Written on 31 September 2005, revised on 30 January 2006. Avaible at `www.ecrypt.eu.org/documents/D.STVL.2-1.0.pdf`.
3. Eli Biham and Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems,* In Journal of Cryptology, vol. 4, pp. 3-72, IACR, 1991.
4. David Chaum and Jan-Hendrik Evertse, Cryptanalysis of DES with a Reduced Number of Rounds, In Crypto'85, pp 192-211, LNCS 218, Springer.
5. Anne Tardy-Corfdir, Henri Gilbert: *A Known Plaintext Attack of FEAL-4 and FEAL-6,* Crypto'91, LNCS 576, Springer, pp. 172-181, 1992.
6. Don Coppersmith, *The development of DES, Invited Talk, Crypto'2000, August 2000.*
7. *Nicolas Courtois: Examples of equations generated for experiments with algebraic cryptanalysis of DES,* `http://www.cryptosystem.net/aes/toyciphers.html`.
8. *Nicolas Courtois: General Principles of Algebraic Attacks and New Design Criteria for Components of Symmetric Ciphers,* in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. 67-83, Springer, 2005.
9. Nicolas T. Courtois *How Fast can be Algebraic Attacks on Block Ciphers?* In online proceedings of Dagstuhl Seminar 07021, *Symmetric Cryptography* 07-12 January 2007, E. Biham, H. Handschuh, S. Lucks, V. Rijmen (Eds.), `http://drops.dagstuhl.de/portals/index.php?semnr=07021`, ISSN 1862 - 4405, 2007. Also available from `http://eprint.iacr.org/2006/168/`.
10. Nicolas Courtois, Gregory V. Bard, David Wagner: *Algebraic and Slide Attacks on KeeLoq,* preprint, `eprint.iacr.org/2007/062/`.
11. Nicolas Courtois, Adi Shamir, Jacques Patarin, Alexander Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, In Advances in Cryptology, Eurocrypt'2000, LNCS 1807, Springer, pp. 392-407.
12. Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, Springer, pp. 266-281.
13. Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, Asiacrypt 2002, LNCS 2501, pp.267-287, Springer.
14. Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Available at `http://eprint.iacr.org/2002/044/`.
15. Nicolas Courtois: *The Best Differential Characteristics and Subtleties of the Biham-Shamir Attacks on DES*, On `eprint.iacr.org/2005/202`.
16. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer. A long, extended version of this paper is available from `www.nicolascourtois.me.uk`.
17. Nicolas Courtois, Guilhem Castagnos and Louis Goubin: *What do DES S-boxes Say to Each Other ?* Available on `eprint.iacr.org/2003/184/`.
18. Nicolas Courtois: *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback,* Crypto 2003, LNCS 2729, pp: 177-194, Springer.
19. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS, to appear in Springer in early 2005. Extended version available on `http://eprint.iacr.org/2003/125/`.
20. Nicolas Courtois: *The Inverse S-box, Non-linear Polynomial Relations and Cryptanalysis of Block Ciphers,* in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. 170-188, Springer, 2005.
21. Nicolas Courtois and Jacques Patarin, *About the XL Algorithm over* $GF(2)$*,* Cryptographers' Track RSA 2003, LNCS 2612, pages 141-157 ,Springer 2003.

22. Marc Davio, Yvo Desmedt, Marc Fosseprez, René Govaerts, Jan Hulsbosch, Patrik Neutjens, Philippe Piret, Jean-Jacques Quisquater, Joos Vandewalle, Pascal Wouters: *Analytical Characteristics of the DES.* In David Chaum editor, Crypto 1983, pp. 171-202, Plenum Press, New York, 1984.
23. Jean-Charles Faugère: *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5),* Workshop on Applications of Commutative Algebra, Catania, Italy, 3-6 April 2002, ACM Press.
24. *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication (FIPS PUB) 46-3, National Bureau of Standards, Gaithersburg, MD (1999). Available from `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`
25. J. Hulsbosch: *Analyse van de zwakheden van het DES-algoritme door middel van formele codering,* Master thesis, K. U. Leuven, Belgium, 1982.
26. Antoine Joux, Jean-Charles Faugère: *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS 2729, pp. 44-60, Springer.
27. Thomas Jakobsen: *Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree,* Crypto 98, LNCS 1462, Springer, pp. 212-222, 1998.
28. Kwangjo Kim, Sangjin Lee, Sangjoon Park, Daiki Lee: *Securing DES S-boxes against Three Robust Cryptanalysis,* SAC'95, pp.145-157, LNCS 2595, Springer.
29. Matthew Kwan: *Reducing the Gate Count of Bitslice DES,* the full paper is available from `http://eprint.iacr.org/2000/051` and the equations we have used are available at `http://www.darkside.com.au/bitslice/nonstd.c`.
30. MAGMA, High performance software for Algebra, Number Theory, and Geometry, — a large commercial software package: `http://magma.maths.usyd.edu.au/`
31. Fabio Massacci: *Using Walk-SAT and Rel-SAT for Cryptographic Key Search,* In International Joint Conference on Artifical Intelligence IJCAI'99, pp. 290-295, 1999.
32. Fabio Massacci and Laura Marraro: *Logical cryptanalysis as a SAT-problem: Encoding and analysis of the U.SS. Data Encryption Standard,* In Journal of Automated Reasoning, vol 24, pp. 165-203. 2000. Essentially the same paper appears in the proceedings of SAT-2000 conference, Highlights of Satisfiability Research at the Year 2000, J. Gent, H. van Maaren, and T. Walsh, Eds, pp. 343-376, IOS Press, Amsterdam, 2000.
33. Mitsuru Matsui: *Linear Cryptanalysis Method for DES Cipher,* Eurocrypt'93, LNCS 765, Springer, pp. 386-397, 1993.
34. MiniSat 2.0. An open-source SAT solver package, by Niklas Eén, Niklas Sörensson, available from `http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/`
35. Ilya Mironov and Lintao Zhang *Applications of SAT Solvers to Cryptanalysis of Hash Functions,* In Proc. Theory and Applications of Satisfiability Testing, SAT 2006, pp. 102-115, 2006. Also available at `http://eprint.iacr.org/2006/254`.
36. Sean Murphy, Matt Robshaw: *Essential Algebraic Structure within the AES,* Crypto 2002, LNCS 2442, Springer.
37. Jacques Patarin: *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*; Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
38. Haavard Raddum and Igor Semaev: New Technique for Solving Sparse Equation Systems, ECRYPT STVL website, January 16th 2006, see also `eprint.iacr.org/2006/475/`
39. Harvard Raddum and Igor Semaev: *Solving MRHS linear equations,* accepted for presentation at ECRYPT Tools for Cryptanalysis workshop, Kraków, Poland, September 2425, 2007.
40. Singular: A Free Computer Algebra System for polynomial computations. `http://www.singular.uni-kl.de/`
41. Adi Shamir: *On the security of DES,* Crypto'85, LNCS 218, Springer, pages 280-281.
42. Claude Elwood Shannon: *Communication theory of secrecy systems,* Bell System Technical Journal 28 (1949), see in particular page 704.
43. I. Schaumuller-Bichl: *Cryptanalysis of the Data Encryption Standard by the Method of Formal Coding,* In Cryptography, Proc. Burg Feuerstein 1982, LNCS 149, T. Beth editor, Springer-Verlag, 1983.