Two Philosophies For Solving Non-Linear Equations in Algebraic Cryptanalysis

Nicolas T. Courtois

University College London, Computer Science, Room 6.18. Gower Street, WC1E 6BT, London, UK
n.courtois@ucl.ac.uk

Abstract. Algebraic Cryptanalysis [36] is concerned with solving systems of particular multivariate non-linear equations derived from various cryptanalysis problems. Many different methods for solving such problems have been proposed in cryptanalytic literature: XL and XSL method, Gröbner bases, SAT solvers, and few other. In this paper we survey these methods and point out that the main working principle in all these is essentially the same. One quantity grows faster than another quantity which leads to a phase transition and the problem becomes efficiently solvable. We illustrate this with examples from both symmetric and asymmetric cryptanalysis. Exact analysis can be quite difficult with complex redundancies and additional equations which help the attacker. In this paper we point out that there exist a second (more) general way of formulating algebraic attacks which did NOT so far have a comparable success in cryptographic literature. Past experience shows that the algebraic degree and/or time complexity of cryptanalysis problems could be reduced by a variety of dedicated **coding** techniques which involve redundancy with addition of new variables. This opens numerous new possibilities for the attackers and leads to interesting optimization problems where the existence of additional equations may be somewhat deliberately engineered by the attacker. For example we show examples of I/O relations where introduction of extra variables allows to substantially reduce the degree and the complexity of polynomial equations.

Key Words: algebraic cryptanalysis, finite fields, overdefined systems of equations, NP-hard problems, phase transitions, XL algorithm, Gröbner bases, XSL algorithm, degree falls, mutants, ElimLin, error correcting codes, algebraic codes, elliptic curves, ECDL problem, Semaev polynomials, block ciphers, Simon cipher

1 Two Approaches To Solving Non-Linear Equations

There are two major philosophies in algebraic cryptanalysis and for the general problem of solving large system of non-linear polynomial/algebraic equations.

1. Either we expand the number of monomials.
2. Or we expand the number of variables

Let us also recall what is the main working principle in both types of techniques: we make two values grow, yet one grows faster. We will see several examples of this in this paper. It allows one to understand why both families of techniques may and will in many cases work.

1. When we add **new monomials**, we grow both the number of monomials T and the number of new equations R. For any system of equations with a certain R/T we can easily improve the R/T ratio by increasing the degree. Then R grows **faster** because there are several ways to obtain the same monomial. The number of monomials does typically NOT[1] grow as fast as the number of new equations.
2. When we add **new variables** we also grow both the number of monomials T and can generate more equations R. Here it is maybe less obvious that R can also grow **faster** and sometimes even asymptotically faster than T, well, at least and limited to[2] a certain interval. One example of this can be found in Section 12.3 in [24] another in [11].

1.1 Historical Developments

Both types of methods already existed and both philosophies worked quite well in their own (somewhat disjoint) space in algebraic cryptanalysis of DES in [5]. More generally, both sorts have also been studied for solving systems of polynomial equations over finite finite fields at Eurocrypt 2000 [6].

It is important to see that techniques of type 1. which expand monomials are nowadays standard, well studied, fully automated by software and do not[3] require a lot of attention. The second family has not been sufficiently studied.

1.2 Difficulties with Family 2 Techniques

There have been some negative results on techniques of type 2. At Eurocrypt 2000 [6] the authors consider the general problem of solving arbitrary quadratic or low degree equations over a finite fields. We have then re-linearization technique which adds new variables (type 2) and the XL algorithm which generates

[1] Identical monomials are generated many times, for example $x_1x_2x_3$ will be obtained 3 times, when multiplying x_1x_3 by x_2, etc. Cf. also slide 80 of [20].

[2] A situation where R grows faster than T permanently must be an illusion. Let $F \leq R$ be the number of linearly independent equations. These equations belong to the linear space of dimension T. Thus $F \leq T$ and very frequently $F \leq T - 1$, cf. [7].

[3] Such software methods are sometimes called "plug and pray" attacks, cf. [24] and the main point in this paper and in [24] is that we would like to develop a richer galaxy of attacks where the attacker plays a more active role.

new products of variables (type 1). At Eurocrypt 2000 it was concluded that relinearization technique is highly redundant[4] and that XL works better [6]. Then researchers have discovered that at higher degrees XL is also redundant [3, 22, 23, 8, 9, 7, 1] and modern Gröbner basis techniques are precisely about removing even more redundancies in XL, cf. [1, 2].

One reason why the second family has not been sufficiently studied is that gives the code breaker a **very considerable degree of freedom**. This is actually a big a problem: it is not clear **how to even start** to design[5] an attack based on this idea. On the positive side some success was definitely achieved with methods related to hardware implementation and multiplicative complexity S-box optimizations [16, 5, 14]. Until now there were extremely few attempts to invent new non-trivial attacks techniques based on 2nd type methods, maybe with the exception of [24].

It is clear that the 2nd method always somewhat contains the 1st method, new variables can be just monomials, which again however leads to known problems with redundancy cf. [6]. Until now cryptographic literature knows very few convincing attacks of the 2nd type. This with exception of SAT solver attacks, which very clearly greatly benefit from added variables cf. for example [14, 5, 16].

Combination Attacks. It is also important to note that both approaches 1. and 2. can and should be combined. To put it simply, the second approach may make the first approach work better, equations become more overdefined and the so called degree of regularity [2] is expected to decrease, i.e. system is solved by Gröbner basis software or other software at a lower degree, which implies lower running time and less memory, cf. also [11, 24].

We are now going to review several classical methods in Algebraic Cryptanalysis, explain what quantities are expanded, and look at the question of how quickly these numbers grow.

1.3 XL Algorithm, F4, F5 and Variants

The XL algorithm [6] was extensively studied and there exist countless variants of this algorithm. For example if m quadratic equations with n variables over $\mathbb{F}_2$ which is assumed to have one unique solution. The basic XL algorithm consists of multiplying all equations by monomials of degree $D-2$ to create a larger number of R equations of degree D. In general in this and similar algorithms the number of linearly independent equations $F \leq R$ has a simple and totally predictable behavior, see [7, 3]. For larger D the prediction is less accurate and also this is where more sophisticated algorithms such as F5 emerge and can make

[4] Interestingly one could repair the linearization technique by some form of decimation (erasing a subset of equations) where the redundancies are removed.

[5] A related concept is the concept of "Algebraic Complexity Reduction" of [14] which has been a great success in a restricted case of a block ciphers with a lot of high-level self-similarity and which is different and stronger. In [14] the attacker also makes well chosen guesses on special combinations of variables.

a difference. Here we show a simple example at degree 5 taken from [7].

n	24	24	24
m	16	27	32
D	5	5	5
R	37200	62775	74400
T	55455	55455	55455
F	33800	53325	55454

A ready software tool which allows to run such simulations can be found in [10]. For small D this sort of experiments can be predicted with 100% accuracy in practice, cf. [7, 3]. We expect that we have always:

$$\text{For } D = 5, \quad F = \min\left(T - 1, R - (n+1)\binom{m}{2} - (n+1)m\right)$$

We obtain a curve which a collation of two closed formulas with a very neat and abrupt transition. Actually even though no randomly generated or random-looking counter-examples are known, we should remain sceptical if this will be always the case in general, especially at the transition boundary. The crucial object of study in this paper is precisely this "phase transition" phenomenon where we shift from one predictable curve to another equally predictable curve. The very existence of phase transitions indicates the predictions in algebraic cryptanalysis will never be an exact science and that rules can eventually be breached or inexact. However predictions are (badly) needed in order to be able to evaluate the complexity of different attacks.

1.4 XSL Algorithm

The XSL algorithm is inspired by the idea that XL algorithm is essentially a tool for dense equations in which all monomials play a similar role. This is very rarely the case in cryptanalysis. If the equations are sparse, a peculiar method was invented where a phase transition is sought by multiplying only by a selection of monomials for $D = 4$ which is simply using only monomials which are already used, cf. [22, 23].

One example of a data series obtained in an application of the XSL attack to a toy cipher can be found in Appendix C. of [23]. However to the best of our knowledge until now nobody has yet studied if the behavior of XSL attacks can be predicted accurately. We consider the data series from Appendix C. of [23] and used Microsoft Excel to fit a polynomial model for these data which minimizes the least square error. Let K be the number of rounds in this 6-bit toy cipher. We then observe that in this precise attack we have almost exactly (we have $R^2 = 1$ in all cases):

$$\begin{cases} R = 936K^2 - 208K + 144 \\ T = 882K^2 - 147K + 7 \\ T' = 504K + 168 \\ F = 850K^2 - 109K + 6 \end{cases}$$

In this attack, the terminating condition is $F \geq T - T'$, cf. [23] and we conclude that this attack works for up to 16 rounds for this toy cipher.

2 The Algebraiz-ation Challenge

Now in order to make further progress in algebraic cryptanalysis, two interesting questions are as follows:

1. Can we efficiently generate or discover additional[6] equations the existence of which is maybe not expected or less easily predicted? This is frequently the case and it helps to solve equations with substantially lower complexity.
2. Can we solve by algebraic cryptanalysis problems which seem unsolvable or a poor fit for algebraic cryptanalysis? For example, the DES S-box do not have any strong algebraic structure. Yet algebraic coding and algebraic cryptanalysis is possible, cf. [5]. A question which is even (a lot) more difficult is a question of ECDL problem in elliptic curves, cf. [35, 30, 24].

These two research directions seem unrelated at the first sight. In fact there are related at more than one level. Two fundamental definitions are at the heart of the connection:

Definition 2.0.1 (An I/O relation, [5, 19, 17]). Consider a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$, $f(x) = y$, with $x = (x_0, \ldots, x_{n-1})$, $y = (y_0, \ldots, y_{m-1})$.
We call an I/O relation any polynomial

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

which hold with certainty, i.e. for every pair (x, y) such that $y = f(x)$.

This allows to define a very useful notion of:

Definition 2.0.2 (The I/O degree, [5, 19, 17]). Again consider a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$. The I/O degree of f is the smallest possible degree in the linear space of existing polynomial I/O relations as defined above.

And here is our second definition:

Definition 2.0.3 (Multiplicative Complexity (MC) [31, 16]). MC is the minimum number of AND gates which are needed if we allow an unlimited number of NOT and XOR gates.

Both notions of I/O relations and MC lead to showing that some degree or algebra-ization is always possible and inevitable in algebraic cryptanalysis. They lead to specific compact algebraic (or multivariate polynomial) encoding methods for various cryptanalysis problems.

[6] This happens for example in the cryptanalysis of the multivariate public-key cryptosystems with the discovery of so called "implicit equations" [29, 4] which we call "I/O relations" in our Def. 2.0.1. cf. for example [4, 28, 27]. We also have a closely related notion of so called "degree falls" sometimes also called "mutants" which are for example observed in ElimLin attacks [5, 33, 11, 15].

2.1 On Small S-boxes

If in cryptanalysis of HFE we have specific structural reasons why some algebraic polynomial "I/O relations" do exist cf. for [28, 27]. In symmetric cryptanalysis, we do not have a strong internal algebraic structure, we however do have specific structural properties which are consequences of how the cipher is designed.

Any very small S-box (for example up to 4 bits) works with both definitions above: it leads to I/O relations and to relatively small MC, and therefore to a rich universe of possible algebraic attacks. For example we have a generic folklore Courtois Theorem 1 which we will find in many papers [5, 23, 13]:

Theorem 1 (Courtois). *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) $g(x, y)$ involving (only) monomials in $\mathcal{T}$, and that hold with probability 1, i.e. for every (x, y) such that $y = F(x)$.*

Remark. Simon is a recent example of a block cipher with truly exceptionally low MC and an excessively small S-box. Here the S-box is an AND gate, the simplest possible non-linear component. The complexity of Simon is yet substantially lower than with CTC2, DES or GOST for which algebraic cryptanalysis were previously studied and implemented [9, 5, 14]. It should not therefore be surprise that Simon will be our favorite block cipher to study.

3 ElimLin Attacks on Simon

Two recent papers consider the ElimLin attacks on Simon [33, 11]. ElimLin is a remarkably simple algebraic attack which to some extent break any cipher, if not too complex. The study of ElimLin is an excellent case where many interesting things happen simultaneously: degree falls, generation of extra equations the existence of which was not initially expected (like in [4]), phase transitions.

ElimLin is a curious sort of attack, cf. slide 126 in [20]. It can be described informally in 2 simple steps:

1. Find linear equations in the linear span.
2. Eliminate some variables, and iterate (try 1. again).

ElimLin is a stand-alone attack which allows one to recover the secret key of many block ciphers [8, 9, 12, 21] and more recently in [33, 25].

The main characteristic of ElimLin is that it quietly dissolves non-linear equations and generates linear equations. This algorithm basically makes progressively disappear the main and **the** only thing which makes cryptographic schemes not broken by simple linear algebra: non-linearity. It is not clear however why this works and how well the ElimLin attack scales for larger systems of equations. In recent 2015 work of Raddum we discover that (experimentally) ElimLin breaks up to 16 rounds of Simon cipher [33] however it is hard to know exactly what happens for 17 rounds.

3.1 The Overdefined Heuristic

Now ElimLin has something which renders XL, XSL, T' method [20, 22], and many other potentially obsolete at least in block cipher cryptanalysis. Actually none of these methods works very well, because they have been designed to solve cryptanalysis problems with small quantity of data which leads to a large value of the so called "degree of regularity" [2] which is essentially the maximum degree D of polynomials manipulated which we mentioned before for XL (however for a better algorithm this degree can be lower). Our main observation is something which was actually known longer, at least since [6]. The fact is that **overdefined** systems of equations are substantially easier to solve. Moreover, it could be possible for the attacker, to try to design an attack which creates such systems of equations. We can call it an **overdefined strategy**, cf. [17, 24].

One simple method to achieve this is to increase the data complexity in the attack which makes the "degree of regularity" decrease [8, 9]. For example we consider an attack K known plaintexts and study how the complexity of the attack grows with K. Then we discover a fascinating aspect of ElimLin which only recently have attracted some attention [11]. The fact is that the number of equations generated can go through several stages cf. [11] as K grows. Initially there are no non-trivial equations. Then we obtain a curve which grows **faster** than linear in K. Our recent[7] paper show that for example super-linear growth is possible [11]. Then eventually it achieves saturation and grows linearly.

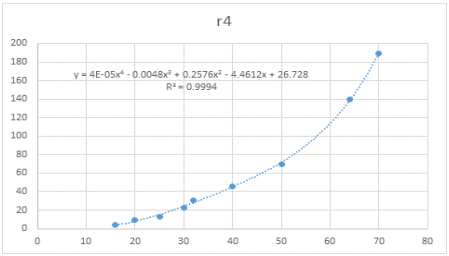

Fig. 1. Number of linearly independent equations generated at stage 4 of the ElimLin algorithm for 8 rounds of Simon 64/128 according to [11].

In a recent PhD thesis we also study enhancements for ElimLin algorithm similar to those studied in [25] and show that the number of equations which could be added to ElimLin can follow a curve which is a collation of not less than 5 distinct intervals where the result seems to be predictable with perfect accuracy, to later switch to another curve. This suggests that we need to remain sceptical about any prediction technique however accurate it may seem.

[7] One (older) example which shows that the number of equations grows faster than linear as a function of the data complexity K in ElimLin can be found at slide 153 in [20].

4 Coding ECC Cryptanalysis Problems

The same quest of trying to construct systems of equations which are very highly overdefined is also what motivates our recent research on coding ECC cryptanalysis problems [24]. We expect to achieve some sort of happy tradeoff between increasing the number of variables and lowering the regularity degree of the equations which are then going to become more efficiently solvable. This is our recent approach which was designed as an alternative to early and more recent attempts [35, 30] to design an index calculus algorithm for the ECDL problem based on so called Semaev Polynomials [35]. Traditionally, ECC relations of type $P1 + P2 = P3$ will be coded by the S3 polynomial which following [34] is

$$S_3(x_1, x_2, x_3) = (x_1 - x_2)^2 x_3^2 - 2\left[(x_1 + x_2)(x_1 x_2 + A) + 2B\right] x_3 + (x_1 x_2 - A)^2 - 4B(x_1 + x_2)$$

This polynomial is of degree 6 and it is already quite complex. Can we do better?

In this paper we do not claim to study these problems in detail. We just recall that the main idea is that the attacker wants to code an ECC cryptanalysis problem by expanding the number of variables and this in order for the "degree of regularity" to decrease, i.e. system is expected to solved more easily by various techniques. We refer to [24] for a more systematic presentation of this attack strategy. In this paper we just show one concrete example on how a redundant set of ECC variables can lead to some unusually simple equations to exist.

4.1 On ECC Codes

The philosophy of adding new variables can be studied in terms of certain types of **ECC codes**.

Definition 4.1.1 (ECC Code).
We call an ECC Code any injective application

$$F : E(\mathbb{F}_p)^L \to E(\mathbb{F}_p)^K$$

which is defined for all except a small number of special EC points.

Remark: We should note that error correcting codes which are defined or constructed using elliptic curves are typically defined as subsets of $\mathbb{F}^K$ where $\mathbb{F}$ is a finite field, cf. for example page 11 in [26]. In this paper and in [24] we find it more convenient to define ECC Codes as a subset of $\mathbb{E}^K$ where $\mathbb{E}$ is an elliptic curve, even though later we will just look at EC coordinates of these points in $\mathbb{F}^K$. We refer to [26] for additional literature pointers about error correcting codes and those which use elliptic curves.

Now we are going to exhibit one ECC property which to the best of our knowledge has not been studied before and which show that redundant sets of variables can lead to substantial simplification in the complexity of systems of polynomial equations. Our paper [24] contains more such properties and explains more in detail the process where the existence of such properties can be seen as an alternative to (or an enhancement to) some recent attempts to solve the ECDL problem in [35, 32, 30], which attempts so far were not a great success and better methods need precisely to be invented.

4.2 D73 - A New Family of Cubic I/O Relations

We have:

Theorem 4.2.1 (D73 Theorem). We consider the following set of variables on EC, a special form of ECC Code with 3 inputs and 7 outputs for any Weierstrass elliptic curve modulo a large P.

$$(P1, P2, P3) \mapsto \begin{matrix} P1 & P2 & P1+P2 \\ P1+P3 & P2+P3 & P1+P2+P3 \\ P3 \end{matrix}$$

Now we consider only the x coordinates of these 7 points. We call them $sx1, sx2, sx12$ for the points in the first line, then $sx13, sx23, sx123$ are for the points the first line, and $sx3$ will be the x coordinate for the last point $P3$. This is summarized on the picture below:

$$\begin{matrix} sx1 & sx2 & sx12 \\ sx13 & sx23 & sx123 \\ sx3 \end{matrix}$$

If all the 7 points are distinct from the ECC neutral element ∞ we have:

```
sx1*sx2*(sx23-sx13) +sx1*sx3*(sx12-sx23) +sx2*sx3*(sx13-sx12)
+sx123[sx1*(sx13-sx12)+sx2*(sx12-sx23)+sx3*(sx23-sx13)] = 0
```

Remark. Our D73 equation is a homogenous polynomial of degree 3. We challenge the reader to discover anything comparable in terms of elegance and simplicity for an ECC Code expansion with a similar expansion factor. The main point in this paper is that having redundant variables could be a good idea. It may allow to greatly simplify polynomial equations and effectively replace Semaev polynomials by some simpler and lower degree polynomials. This we have not demonstrated, we just demonstrate the existence of some simpler polynomials, while in any cryptanalytic attack on the ECDL problem we expect to use a lot more polynomials of other types and for the time being we refrain from making any conclusions about how our discovery might impact the complexity of such methods, as currently such methods are yet very inefficient cf. [30, 24]. An interesting question is for example to construct very highly overdefined encodings of ECDL problem with properties of type $\lim_{K\to\infty} F/T = 1$ and some other "density" properties. Some early attempts to achieve this can be found in [24] which paper also shows that there are some very substantial difficulties to make this sort of approach work.

5 Conclusion

In this paper we compare different known techniques for solving non-linear algebraic equations in algebraic cryptanalysis and show that they all can be seen as a race between two different quantities one of which grows faster. We illustrate this with examples derived from both symmetric and asymmetric cryptanalysis. The crucial question is the possibility to accurately predict the behavior of such attacks and that they will later switch to another curve and a phase transitions will occur. Now the question is can we do better than just contemplate these transitions? In this paper we point out that there exist a second somewhat more general way of formulating algebraic attacks, where the attacker plays a more active role. This is the question of **algebraic coding** which did NOT so far have great success in cryptographic literature, and was frequently ignored as a trivial first step, or rejected in some inefficient/redundant attack methods.

We point out that this problem of finding a "good"[8] algebraic coding was so far poorly studied. Yet it gives the attacker a very considerable degree of freedom, especially if we allow the coding to be redundant. We need to pay more attention to hard combinatorial optimization problems such as finding non-trivial redundant representations leading to important simplifications in algebraic description complexity. The bottom line is that we open the possibilities to invent a number of new "out-of-the-box" attacks with non-trivial à priori coding steps. For example multiplicative complexity and other S-box optimizations lead to some quite competitive attacks on block ciphers [5, 16, 14]. The primary challenge for the future remains how to code and re-code cryptanalysis problems in better ways. The attacker is not merely hoping that some interesting equations exist [29, 4] or will be found by our attack [11], which approach to software cryptanalysis we called "plug and pray" in [24], but how to "engineer" an attack where new "interesting" equations will exist.

For example many authors have tried to develop an index calculus or a point splitting attack on the ECDL problem through the use of so called Semaev/Summation polynomials [35, 30] without great success. In this paper and in [24] we suggest that the degree and complexity of ECC coding problems can be reduced with redundant coding of variables. Better algebraiz-ation through simpler polynomials is however is probably by far not enough to solve hard cryptanalysis problems. More attention needs also be paid to questions such as "densely connected equations topology" cf. Section 6 and 6.7 in [24], and additional "constraints coding" questions, cf. Part VI in [24].

References

1. Magali Bardet, Jean-Charles Faugère and Bruno Salvy, *On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations*, in Proceedings of International Conference on Polynomial System Solving (ICPSS, Paris, France), pp. 71-75, 2004. Also known as Research report RR-5049, INRIA 2003.

[8] See Part 1 on slide 56 and 58 in [20].

2. Ludovic Perret: *Gröbner bases techniques in Cryptography,* http://web.stevens.edu/algebraic/Files/SCPQ/SCPQ-2011-03-30-talk-Perret.pdf
3. Jiun-Ming Chen, Nicolas Courtois and Bo-Yin Yang: *On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis,* In ICICS'04, LNCS 3269, pp. 401-413, Springer, 2004.
4. Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, Springer, pp. 266-281, 2001.
5. Nicolas Courtois, Gregory V. Bard: *Algebraic Cryptanalysis of the Data Encryption Standard,* In Cryptography and Coding, 11-th IMA Conference, pp. 152-169, LNCS 4887, Springer, 2007.
6. 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.
7. Nicolas Courtois, Jacques Patarin: *About the XL Algorithm over $GF(2)$,* Cryptographers' Track RSA 2003, San Francisco, April 13-17 2003, LNCS 2612, pp. 141-157, Springer.
8. 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/.
9. Nicolas Courtois *CTC2 and Fast Algebraic Attacks on Block Ciphers Revisited* Available at http://eprint.iacr.org/2007/152/.
10. Nicolas Courtois: Some algebraic cryptanalysis software http://www.cryptosystem.net/aes/tools.html.
11. Nicolas T. Courtois, Iason Papapanagiotakis-Bousy, Pouyan Sepehrdad and Guangyan Song: *Predicting Outcomes of ElimLin Attack on Lightweight Block Cipher Simon,* In Secrypt 2016 proceedings.
12. Nicolas Courtois and Blandine Debraize: *Specific S-box Criteria in Algebraic Attacks on Block Ciphers with Several Known Plaintexts,* In WEWoRC 2007, LNCS 4945, pp 100-113, Springer, 2008.
13. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS 3506, pp. 3-20, Springer 2005. Extended version available on http://eprint.iacr.org/2003/125/.
14. Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* preprint, 2010-2014, available at http://eprint.iacr.org/2011/626.
15. Nicolas Courtois, Pouyan Sepherdad, Petr Susil and Serge Vaudenay: *ElimLin Algorithm Revisited,* In FSE 2012, LNCS, Springer.
16. Nicolas T. Courtois, Daniel Hulme and Theodosis Mourouzis: *Multiplicative Complexity and Solving Generalized Brent Equations With SAT Solvers,* In COMPUTATION TOOLS 2012, ISBN 978-1-61208-222-6, pp. 22-27, 22 July 2012, best paper award.
17. Nicolas T. Courtois: *New Frontier in Symmetric Cryptanalysis,* Invited talk at Indocrypt 2008, 14-17 December 2008.
    Extended version of slides presented: http://www.nicolascourtois.com/papers/front_indocrypt08.pdf.
18. Nicolas Courtois and Blandine Debraize: *Algebraic Description and Simultaneous Linear Approximations of Addition in Snow 2.0.,* In ICICS 2008, 10th International Conference on Information and Communications Security, 20 - 22 October, 2008, Birmingham, UK. In LNCS 5308, pp. 328-344, Springer, 2008.

19. Nicolas Courtois: *General Principles of Algebraic Attacks and New Design Criteria for Components of Symmetric Ciphers,* in AES 4, LNCS 3373, pp. 67-83, Springer, 2005.
20. Nicolas Courtois: *Algebraic Attacks vs. Design of Block and Stream Ciphers,* slides used in GA18 course Cryptanalysis taught at University College London, 2014-2016, `http://www.nicolascourtois.com/papers/ algat_all_teach_2015.pdf`
21. Nicolas Courtois: *Software and Algebraic Cryptanalysis Lab, a lab used in GA18 course Cryptanalysis taught at University College London, 17 March 2016,* `http://www.nicolascourtois.com/papers/ga18/AC_Lab1_ElimLin_Simon_CTC2.pdf`
22. *Nicolas Courtois and Josef Pieprzyk: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Asiacrypt 2002, LNCS 2501, pp.267-287, Springer.
23. Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Available at `http://eprint.iacr.org/2002/044/`. Contains two different (earlier) versions of the XSL attack, see also [22].
24. Nicolas Courtois: *High Saturation Complete Graph Approach for EC Point Decomposition and ECDL Problem,* preprint July-September 2016, `http://eprint.iacr.org/2016/704.pdf`.
25. Petr Susil, Pouyan Sepehrdad, Serge Vaudenay, Nicolas Courtois: *On selection of samples in algebraic attacks and a new technique to find hidden low degree equations.* in International Journal of Information Security vol. 15 iss. 1, pp. 51-65, Springer, 2016.
26. Lorenz Minder: *Cryptography based on error correcting codes,* PhD thesis 3846 (2007), EPFL, 27 July 2007, at `http://algo.epfl.ch/_media/en/projects/lorenz_thesis.pdf`.
27. Ming-Deh A. Huang, Michiel Kosters, Sze Ling Yeo: *Last Fall Degree, HFE, and Weil Descent Attacks on ECDLP,* In Crypto 2015, LNCS 9215, pp. 581-600, August 2015.
28. Antoine Joux, Jean-Charles Faugère: *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS 2729, pp. 44-60, Springer.
29. Jacques Patarin: *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*; Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
30. Christophe Petit, Michiel Kosters, Ange Messeng: *Algebraic Approaches for the Elliptic Curve Discrete Logarithm Problem over Prime Fields,* In PKC 2016, vol. 2, LNCS 9615, pp. 3-18, Springer, 2016.
31. Joan Boyar, René Peralta: *A New Combinational Logic Minimization Technique with Applications to Cryptology.* In SEA 2010: 178-189.
    An early version was published in 2009 at `http://eprint.iacr.org/2009/191`. It was revised 13 Mar 2010.
32. Christophe Petit and Jean-Jacques Quisquater: *On polynomial systems arising from a Weil descent,* In Asiacrypt 2012, LNCS 7658, pp. 451-466, Springer 2012.
33. Håvard Raddum: *Algebraic Analysis of the Simon Block Cipher Family,* In LatinCrypt 2015, LNCS 9230, pp. 157-169, Springer, 2015, cf. `https://www.simula.no/file/simonpaperrevisedpdf/download`.
34. Igor Semaev: *New algorithm for the discrete logarithm problem on elliptic curves,* Preprint, 10 April 2015, available at `eprint.iacr.org/2015/310/`.
35. Igor Semaev: *Summation polynomials and the discrete logarithm problem on elliptic curves,* Preprint, available at `eprint.iacr.org/2004/031/`.
36. Claude Elwood Shannon: *Communication theory of secrecy systems,* Bell System Technical Journal 28 (1949), see in particular page 704.