Predicting Outcomes of ElimLin Attack on Lightweight Block Cipher Simon

Nicolas T. Courtois[1], Pouyan Sepehrdad[2], Guangyan Song[1] and Iason Papapanagiotakis-Bousy[1]
[1]*Computer Science Department, University College London, Gower street, WC1E 6BT, London, U.K.*
[2]*Product Security Team, Qualcomm Inc, San Diego, CA, U.S.A.*

Keywords: Cryptanalysis, Finite Fields, Polynomial Equations, Block Ciphers, NP-hard Problems, MQ Problem, Phase Transitions, XL Algorithm, Gröbner Bases, ElimLin, Prediction, Time Series.

Abstract: There are two major families in cryptanalytic attacks on symmetric ciphers: statistical attacks and algebraic attacks. In this position paper we argue that algebraic cryptanalysis has not yet been developed properly due to the weakness of the theory which has substantial difficulty to prove most basic results on the number of linearly independent equations in algebraic attacks. Consequently most authors present a restricted range of attacks which are shown experimentally to work with their computer but refrain from claiming results which would work on a larger computer but have not yet been tested. For example in recent 2015 work of Raddum we discover that (experimentally) ElimLin attack breaks up to 16 rounds of Simon block cipher however it is hard to know what happens for 17 rounds. In this paper we argue that one CAN predict and model the behavior of such attacks and evaluate complexity of the attacks which we cannot yet execute. To the best of our knowledge this has never been done before.

1 INTRODUCTION - ElimLin

ElimLin has been invented in 2006/2007 (N. Courtois, 2007; Courtois, 2007b). It is typically viewed as a super-simplified Gröbner basis calculation technique which nevertheless is a stand-alone powerful cryptanalytic attack: it can break ciphers and recover the key just by running some software without any human intervention. The study of ElimLin is interesting (N. Courtois, 2012; P. Susil, 2016) precisely because it is simpler to understand than more complex polynomial algebra techniques (N. Courtois, 2000; N. Courtois, 2003; J.-M. Chen and Yang, 2004; M. Bardet, 2004; T.J. Hodges, 2015).

1.1 Fundamental Problems

It is very disappointing to study the current Gröbner basis theory which only provides simple unconditional answers on the complexity of algebraic attacks in some restricted cases and under assumptions which systems of equations which we really are facing do not satisfy (N. Courtois, 2003; J.-M. Chen and Yang, 2004; M. Bardet, 2004; T.J. Hodges, 2015). In general it is important to note that extremely few things are known in general about upper bounds on complexity of algorithms. This is simply a major difficulty in computer science in general: This applies in particular to most scientific statements which could guarantee security of industrial cryptographic systems. Almost never researchers in cryptography attempt to prove such statements. The nature of statements is simply such that they cannot be proven, they can only be disproved, cf. (Nash, 2012; Courtois, 2008). We can derive this for example from Karl Popper philosophy of science. More importantly this impossibility to prove have been explicitly stated and affirmed specifically for the hardness of problems in cryptanalysis of ciphers by US mathematician John Nash inside his recently declassified letter to the NSA written as early as 1955 (Nash, 2012). We see that there are good fundamental reasons to work run extensive experimentations and extrapolate rather than try to achieve an arguably impossible task to prove that our attack will work for larger parameters.

1.2 ElimLin Basics

ElimLin is a curious sort of attack, cf. slide 126 in (Courtois, 2016a). It can be described informally as an iteration of 2 simple steps:

1. Find linear equations in the linear span.

Courtois, N., Sepehrdad, P., Song, G. and Papapanagiotakis-Bousy, I.
Predicting Outcomes of ElimLin Attack on Lightweight Block Cipher Simon.
DOI: 10.5220/0005999504650470
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT*, pages 465-470
ISBN: 978-989-758-196-0

2. Eliminate some variables, and try 1. again.

ElimLin is a software cryptanalytic attack which allows one to recover the secret key of many block ciphers (Courtois, 2007b; Courtois and Debraize, 2008) and more recently in (N. Courtois and Susil, 2014; Raddum, 2015; P. Susil, 2016).

The main characteristic of ElimLin is that it quietly dissolves and makes disappear non-linear equations and generates linear equations. This algorithm basically makes progressively disappear the main and 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.

2 ElimLin ON LIGHTWEIGHT CIPHERS

A major difficulty with ElimLin is that so far it has been successful only for relatively simple lightweight ciphers. For more complex ciphers it seems to do things which are relatively trivial, e.g. equations generated do NOT penetrate deeply inside the cipher, or very slowly, cf. slide 153 in (Courtois, 2016a).

We are going right now to make some definite progress in the direction of distinguishing between trivial and non-trivial behavior for ElimLin. This is NOT only about penetrating deeper inside the cipher. Previous experience shows that ElimLin only starts to work at a certain threshold. Before this threshold, again nothing non-trivial can be observed even though slow penetration occurs. This is not really apparent in any of the current works or is lost in vast quantities of data generated in computer simulations. It will be more clearly visible in this paper. In this paper we define a new criterion which shows that it is possible to see that there exist two very different and easily distinguishable patterns in ElimLin. Either the attack follows one pattern, and does nothing trivial, or it follows another pattern and it is very clearly doing well.

2.1 Phase Transitions

It is known that many NP-hard problems are subject to "phase transition", with certain parameters that problem is hard, and then will rather abruptly transition from "hard" to "easy to solve". This what we observe with ElimLin. Let K be the number of Plaintext/Ciphertext (P/C) pairs used in an ElimLin attack. In this paper we are going to discover that at a certain threshold the number of NEW linear and linearly independent equations generated at various stages of the attack can follow one curve, and then **switch** to another curve with a different asymptotic growth rate.

Conjecture 2.1 Consider a system of multivariate equations derived from a block cipher written following one of the two basic strategies described in (N. Courtois, 2007; Courtois, 2016a). Consider a simple known plaintext attack with K Plaintext/ciphertext (P/C) pairs. Consider a case such that the cipher is eventually broken by ElimLin, cf. (Courtois, 2007b; Courtois, 2007a; Courtois and Debraize, 2008; Courtois, 2016c; Raddum, 2015). The number of new and linearly independent linear equations generated by ElimLin algorithm goes through several distinct stages St0-St3:

St0 Initially it grows linearly with K, and for certain individual stages of the attack is simply equal to 0 and does NOT grow, cf. our later r_i notation in Section 3.

St1 Then it switches to another curve where it grows faster than linearly in K.

St2 This until it reaches a saturation stage where the cipher is completely broken by ElimLin. Here we have a very rapid phase transition cf. Section refBigPictureUpAndDown where the number of equations r_i generated at one stage rebecomes 0 simply because an earlier stage of the attack reaches a certain threshold where combinatorial explosion in additional equations generated makes it complete the whole attack and not requite the next stage to be executed].

One (old) example from 2007 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 (Courtois, 2016a) and which originally comes from (Courtois and Debraize, 2008).

3 OUR EXPERIMENTAL SETUP AND NOTATION

More examples can be easily obtained using a basic software setup which we use at UCL to run a hands-on student lab session on algebraic cryptanalysis of block ciphers (Courtois, 2016c), which is part of GA18 course on cryptanalysis taught at UCL. One example could be easily obtained for the CTC2 cipher, cf. (Courtois, 2007b; Courtois, 2007a; Courtois, 2016c). A more "modern" example can be generated by using the equations generator for Simon block cipher developed by Guangyan Song and UCL student Ilyas Azeem in 2015-6, the complete source code of which is available at github, cf. (Courtois, 2016c;

Courtois, 2016b). The ElimLin is executed using using one of our implementations of ElimLin (Courtois, 2016c; Courtois, 2016b) which has the nice particularity to display on screen the number of linear equations generated at each stage/iteration of the algorithm.

We recall the two main and only steps of ElimLin:

1 Find r_i linear equations in the linear span, $i = 0,1,2,3,\ldots$.

2a If $r_i > 0$ eliminate some r_i variables, increment i and try again Step 1.

2b Algorithm terminates when $r_i = 0$ for some i.

In addition, in this paper, and by convention we are going to define a step $i = 0$ which is different than above, but the same which is implemented in a common implementation of ElimLin (Courtois, 2016b). We will assume that r_0 will be the number of linear equations which already appear in the equations, without executing any linear algebra. This is a convention which allows researchers to distinguish more easily between a "misleading" starting number of variables (which is sometimes artificially inflated due to methods used for equation generation and formal coding) and the "real" or intrinsic number of variables which is there prior to execution or ElimLin.

Definition 3[ElimLin progress indicators] Let V_{start} or simply V if there is no confusion be the initial number of variables. We define by r_i the number of linear/affine equations over GF(2) generated at each stage of the algorithm where by convention r_0 is the number of linear/affine equations over GF(2) already present. We define

$$V^i_{broken} = \sum_{j=0}^{i} r_j \qquad V_{broken} = \sum_{i=0}^{\infty} r_i \tag{1}$$

where by convention $r_i = 0$ if algorithm has reached $V^i_{broken} = V$ at an earlier stage. We define also

$$V^i_{broken} = V - \sum_{j=0}^{i} r_j \tag{2}$$

and accordingly let $V_{unbroken} = V - \sum_{i=0}^{\infty} r_i$. Overall we will say that the algorithm terminates if $V_{broken} = V$ and $V_{unbroken} = 0$ and we deliberately ignore the fact that some variables could be subject to a brute force step, cf. FXL method in (N. Courtois, 2003). Overall our goal is to achieve for a certain i that

$$\frac{V^i_{broken}}{V_{start}} = 1 \qquad \text{or} \qquad \frac{V^i_{unbroken}}{V_{start}} = 0 \tag{3}$$

4 HOW TO PREDICT THE SUCCESS OF ElimLin

We start by a simple example which shows that accurate prediction is feasible.

4.1 On Growth Rate in ElimLin

On the figure below we show the number of linear equations generated at stage 4 [counting from 0] of the ElimLin algorithm for 8 rounds of Simon block cipher. We should note that nothing remarkable happens at earlier stages 0,1,2,3, the growth is linear and therefore very easy to predict. We have then asked Microsoft Excel to produce a polynomial interpolation for this data series.

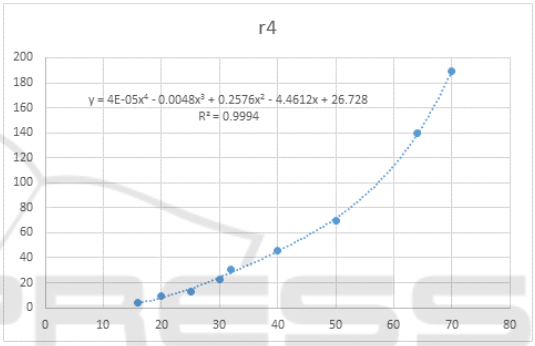

Figure 1: Number of linearly independent equations generated at stage 4 of the ElimLin algorithm for 8 rounds of Simon 64/128 obtained with the exact software setup of (Courtois, 2016c).

4.2 Analysis of Our Prediction

First property we need to note is that the polynomial prediction seems to be the right tool here. This is justified by the fact that the normalized squared error value R^2 is very high and close to the theoretical maximum of 1, and also that high degree coefficients are substantially smaller (in absolute values) that lower degree polynomial coefficients.

Secondly we observe that ElimLin eventually breaks the cipher here because the number of newly generated linear equations grows faster than the number of variables which grows linearly with K. Eventually we obtain a sufficient number of linear equations which makes ElimLin compute all the variables and obtain the 128-bit secret key together with all the intermediate variables. This was previously observed in other cases where ElimLin was applied cf. (Courtois, 2016a; Courtois and Debraize, 2008; Courtois, 2007a; N. Courtois, 2007).

4.3 Limitations vs. Improvements

We do not know any counter-example which would contradict our asymptotic super-linear growth rule. For 8 rounds of Simon cipher we observed that the growth remains strictly linear for equations computed at stage 3, however some equations are only computed at stage 4 and this occurs quite early starting from $K = 1$. However Simon is a particularly simple cipher. For more complex ciphers, the ElimLin algorithm could have serious problems to enter this behavior or start working and exhibit any sort of non-trivial behavior. However when ElimLin starts to work for some cipher and produces new equations which depend on the plaintext or ciphertext data in a non-trivial way, steady progress is observed. Except maybe if there isn't enough data available. A certain type of counter-example could occur for some ciphers with small blocks which would maybe not be able to allow K to be large enough for ElimLin to terminate.

Improvements. We stress that this powerful phenomenon of combinatorial explosion which eventually leads to ElimLin breaking the cipher occurs already for a basic straightforward application of ElimLin in a known-plaintext attack (KPA) scenario. Several methods to make ElimLin work better by using well-chosen P/C pairs. One method which has been used ever since ElimLin was invented (Courtois, 2007b; Courtois, 2007a; N. Courtois, 2007) is to exploit a CPA (chosen-plaintext attacks) with plaintext which differ by extremely few bits, for example in a counter mode. More recently new methods have been proposed which allow to generate additional linear equations not automatically discovered by ElimLin (N. Courtois and Susil, 2014) and (P. Susil, 2016).

In this paper we again focus on very simple variants of ElimLin in order to show that **high accuracy** predictions (e.g. with $R^2 \approx 1$) are possible to achieve. This success should encourage and set up standards for further research on more realistic and more powerful but also more complex applications of ElimLin, cf. (N. Courtois and Susil, 2014; P. Susil, 2016; Raddum, 2015).

5 THE BIG PICTURE - LONG TERM PREDICTIONS

As already explained in Conjecture 2.1 page 2, we expect that there are at least 3 distinct stages in ElimLin algorithm. With these notations until now we have been only looking at stage St1 of the attack. In this section we look at the big picture and the main phase transition ST1-2 which is the most fundamental one because we have really here a transition from "hard" to "easy" and some sort of combinatorial explosion in the number of generated equations at one stage which for even higher K makes this stage i of the attack unnecessary because the attack will terminate at an earlier stage $i-1$. This is best seen by looking at how the value of $V_{unbroken}$ evolves with growing K.

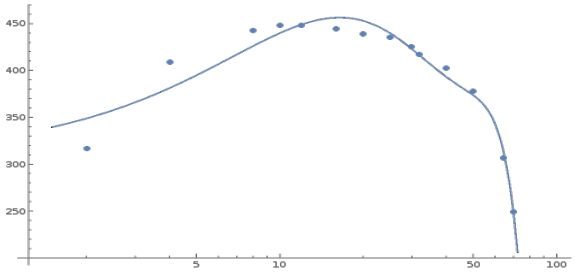

Figure 2: Number of variables when ElimLin terminates $V_{unbroken}$ for 8 rounds of Simon 64/128 obtained with the exact software setup of (Courtois, 2016c).

This curve seems a lot harder to predict than until now. For example we have obtained the following polynomial predictor

$$V_{unbroken}(K) = -\frac{1.35}{10^4}K^4 + \frac{2.07}{10^2}K^3 - 1.12 \cdot K^2 + 22.4 \cdot K + 308.25 \quad (4)$$

for which we have $R^2 = 0.9863$. Substantially better accuracy can be achieved by focusing on another quantity and looking at a wider range or prediction methods. We have tried many different predictors obtained with Microsoft Excel and WolframAlpha cloud online service www.wolframcloud.com which offers higher precision and more advanced functionalities. In general focusing on the ratio $V_{unbroken}/V_{start}$ gives substantially better results, as V_{start} grows linearly with K and our primary interest in this paper is a deviation from that linear growth which is the default (trivial) behavior. The curves become then more regular and more intelligible, cf. later Figure 4. For example we obtained $R^2 = 0.9941$ with

$$\frac{V_{unbroken}(K)}{V_{start}} = 1.07 \cdot x^{-0.485} - 0.127 \quad (5)$$

Then we obtained $R^2 = 0.9989$ with a polynomial of degree 6:

$$\frac{V_{unbroken}(K)}{V_{start}} = -\frac{2}{10^{10}}K^6 - \frac{5}{10^8}K^5 + \frac{5}{10^6}K^4 - \frac{3}{10^4}K^3 + \frac{7.5}{10^3}K^2 - 0.112K + 0.8083 \quad (6)$$

Finally we have obtained an even better result $R^2 = 0.9998$ with

$$\frac{V_{unbroken}(K)}{V_{start}} = -\frac{9.57}{10^3}(\ln K)^4 + \frac{1.00}{10}(\ln K)^3 - \frac{3.19}{10}(\ln K)^2 - \frac{1.35}{10}\ln K + 0.65. \quad (7)$$

Goals. Overall in security of block ciphers the main goal for further development of prediction techniques will be first to predict the moment when the curve on Figure 2 starts falling. This is the main phase transition from "hard" to "easy" in any ElimLin attack.

5.1 Are Reliable Extrapolations Possible?

So far we have only worked with data from simulations and all our predictions are highly accurate just because we had quite a few data points and have deployed sufficient effort to compute predictors with standard techniques. The fact that **several methods** give remarkably good results should however be a warning sign. This together with insufficient amount of data will be a major difficulty when researchers in the future are going to try to build predictive models to extrapolate and compute the complexity of simulations which they have not done. In an extrapolation scenario it is easy to see that even though different methods used will give excessively good results on the actual data in a certain interval, they could diverge and could fail to predict the curve on a larger interval. Even in this case these predictions can be used to determine the possible or plausible range of results which one can expect to achieve.

A real solution for this difficulty would be to have an "à priori" idea about the nature of the curve and its growth. We had that in Section 4.1 and the polynomial approximation has larger degree coefficients which have decreased rapidly confirming our "à priori" idea on super-linear growth approximated by a polynomial. Here for predicting $V_{unbroken}(K)$ which is essentially proportionnal ot sum of the r_i we also expect a polynomial prediction and we obtained $R^2 = 0.9989$. This is also a somewhat consistent predictor with rapidly decreasing coefficients of higher degree. If so, shouldn't we use a lower degree predictor?

5.2 Are Low Degree Polynomial Extrapolations Possible?

If we look at our later Figure 4 or again how quickly lower degree coefficients decrease in predictors above, we could be tempted to use a linear or low degree predictor. Our experience shows that cutting small higher degree terms in these predictors is disastrous and does not lead to quality predictions, not even on smaller intervals. For example we can cut the beginning of current series and restrict to polynomials of degree 2 for $\frac{V_{unbroken}(K)}{V_{start}}$ and we obtain a predictor of $\frac{V_{unbroken}(K)}{V_{start}} \approx \frac{6}{10^5}K^2 - \frac{7.7}{10^3}K + 0.26$ which is neither very consistent with more accurate predictors computed above nor with the data trend at most places.

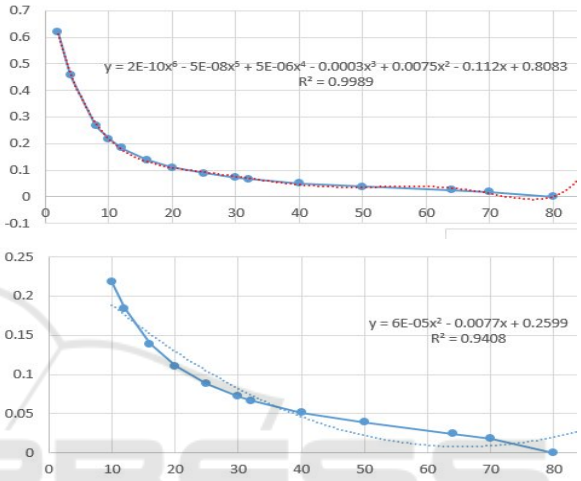

Figure 3: Good vs. bad predictors: decreasing the degree of a polynomial prediction for the same Simon 64/128 8 rounds.

The latter graph is definitely not a predictor we would recommend!

6 ON COMPARATIVE STRENGTH OF CIPHERS

Another major application of ElimLin techniques and possible extrapolations is to make comparison between different ciphers. For example current research indicates that Simon is broken for up to 16 rounds (Raddum, 2015) and CTC2 for up to 7 rounds, both using ElimLin algorithm. Is it possible to claim that one round of Simon is substantially weaker than one round of CT2? We would agree with this claim. On the figure below we show how the quantity $\frac{V_{unbroken}(K)}{V_{start}}$ decreases for 7 rounds for each cipher. We conclude that both ciphers are broken for some K value but with ElimLin we still have a long way to go before the attack terminates resulting in much higher data complexity and running times. This shows that there exist ciphers fundamentally weaker than CTC2 even

though CTC2 was designed to be quite vulnerable to algebraic attacks (Courtois, 2007b; Courtois, 2007a).

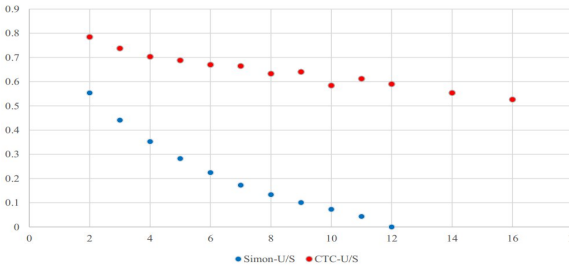

Figure 4: The value of $\frac{V_{unbroken}(K)}{V_{start}}$ for 7 rounds of Simon 64/128 (blue) compared to toy cipher CTC2 with 7 rounds and 8 S-boxes per round and 10/24 key bits known (red).

This per-round weakness of Simon is expected to be compensated by a larger number of rounds.

7 CONCLUSION

In this position paper we propose to predict the success of the number of equations generated in algebraic attacks on the recently very popular NSA block cipher Simon. We argue that such predictions are **necessary**. This is due the fundamental weakness of Gröbner basis theory (T.J. Hodges, 2015) and in complexity theory in general (Nash, 2012). This is very important for ciphers such as Simon which is currently a candidate for ISO standardisation. In absence of more powerful computers we have no other choice than to extrapolate in order to simply evaluate the security of ciphers against already known attacks.

In this paper we show that very highly accurate predictions are possible. However in absence of an "à priori" model of how the curve should behave they do not remove some major incertitudes. The fact that highly accurate predictions can be obtained by **several** very different methods indicates that such predictions are not exact science and must be manipulated with precaution. Simple low degree predictions will not work cf. Figure 3 Higher degree polynomial approximations seem to work very well. In future works we are going to analyse more complex application scenarios of ElimLin (N. Courtois and Susil, 2014; N. Courtois and Susil, 2014; Raddum, 2015; P. Susil, 2016) and produce quality extrapolations.

REFERENCES

Courtois, N. (2000-2016b). Algebraic cryptanalysis software,. http://www.cryptosystem.net/aes/tools.html.

Courtois, N. (2007a). Ctc2 and fast algebraic attacks on block ciphers revisited. In *eprint*. eprint.iacr.org/2007/152/.

Courtois, N. (2007b). How fast can be algebraic attacks on block ciphers? In *Dagstuhl Seminar 07021, Symmetric Cryptography*. dagstuhl.de.

Courtois, N. (2008). New frontier in symmetric cryptanalysis. In *Invited talk at Indocrypt 2008*. http://www.nicolascourtois.com/papers/front_indocrypt08.pdf.

Courtois, N. (2016a). Algebraic attacks vs. design of block and stream ciphers. In *slides used in UCL GA18 course "Cryptanalysis", University College London*. http://www.nicolascourtois.com/papers/algat_all_teach_2015.pdf.

Courtois, N. (2016c). Software and algebraic cryptanalysis lab,. In *University College London*. http://www.nicolascourtois.com/papers/ga18/AC_Lab1_ElimLin_Simon_CTC2.pdf.

Courtois, N. and Debraize, B. (2008). Specific s-box criteria in algebraic attacks on block ciphers with several known plaintexts. In *WEWoRC 2007, pp 100-113*. Springer.

J.-M. Chen, N. C. and Yang, B.-Y. (2004). On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis. In *ICICS'04, pp. 401-413*. Springer.

M. Bardet, J.-C. Faugère], B. S. (2004). On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *ICPSS, Proceedings of International Conference on Polynomial System Solving*.

N. Courtois, T. Mourouzis, G. S. P. S. and Susil, P. (2014). Combined algebraic and truncated differential cryptanalysis on reduced-round simon,. In *proc. of SECRYPT 2014*. INSTICC.

N. Courtois, G. V. B. (2007). Algebraic cryptanalysis of the data encryption standard,. In *IMA Cryptography and Coding, pp. 152-169*. Springer.

N. Courtois, J. P. (2003). About the xl algorithm over gf(2). In *CT-RSA 2003, pp. 141-157*. Springer.

N. Courtois, A. Shamir, J. P. A. K. (2000). Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Eurocrypt'2000, pp. 392-407*. Springer.

N. Courtois, P. Sepherdad, P. S. S. V. (2012). Elimlin algorithm revisited. In *FSE 2012*. Springer.

Nash, J. (2012). letter to the nsa. In *declassified material from 1955*. www.nsa.gov.

P. Susil, P. Sepehrdad, S. V. N. C. (2016). On selection of samples in algebraic attacks and a new technique to find hidden low degree equations. In *Int. J. Inf. Sec. vol. 15 iss. 1, pp. 51-65*. Springer.

Raddum, H. (2015). Algebraic analysis of the simon block cipher family. In *LatinCrypt 2015, pp. 157-169*. Springer.

T.J. Hodges, S.D. Molina, J. S. (2015). On the existence of semi-regular sequences. In *DIMACS Workshop on the Mathematics of Post-Quantum Cryptography*. arxiv.org/1412.7865.