CTC2 and Fast Algebraic Attacks on Block Ciphers Revisited

Nicolas T. Courtois

University College of London,
Gower Street, London, UK,
n.courtois@ucl.ac.uk

Abstract. The cipher CTC (Courtois Toy Cipher) described in [4] has been designed to demonstrate that it is possible to break on a PC a block cipher with good diffusion and very small number of known (or chosen) plaintexts. It has however never been designed to withstand all known attacks on block ciphers and Dunkelman and Keller have shown [13] that a few bits of the key can be recovered by Linear Cryptanalysis (LC) – which cannot however compromise the security of a large key. This weakness can easily be avoided: in this paper we give a specification of CTC2, a tweaked version of CTC. The new cipher is MUCH more secure than CTC against LC and the key scheduling of CTC has been extended to use any key size, independently from the block size. Otherwise, there is little difference between CTC and CTC2. We will show that up to 10 rounds of CTC2 can be broken by simple algebraic attacks.

Key Words: algebraic attacks on block ciphers, AES, Rijndael, Serpent, multivariate quadratic equations, MQ problem, overdefined systems of multivariate equations, XL algorithm, XSL algorithm, Gröbner bases, solving systems of sparse multivariate polynomial equations.

1 Introduction

Claude Shannon, the father of information security as a science, has once advised that, breaking a good cipher should require "as much work as solving a system of simultaneous equations in a large number of unknowns", see [20]. This is an important and very explicit recommandation, yet it was ignored and nearly forgotten for more than 50 years. The public research in symmetric cryptography have concentrated on local and statistical aspects of ciphers, and have overlooked the natural "global" approach of the problem. The secret key is defined as a solution of a system of algebraic equations that describes the whole cipher. This system should not be too simple, otherwise somebody might be able solve it...

Algebraic cryptanalysis of block ciphers took long, long time to take off. For example, we can quote [19, 14, 16] for early work on algebraic cryptanalysis of DES, and a recent paper [5] shows how to break 6 rounds of DES given only 1 known plaintext. The first example of a full-round industrial block cipher broken in practice by algebraic attacks has been found in February 2007, see [6]. In this paper we are interested in breaking a toy cipher, the only purpose of which is to study cryptanalysis of block ciphers.

2 The Description of CTC2

CTC2 means for Courtois Toy Cipher, Version 2. It is quite similar to Serpent, except that it is much simpler, and the key schedule is a simple permutation of key bits, like in DES. It is exactly the same as CTC described in [4] except for the equation (5) below that was modified very slightly, and the key scheduling have been generalised and improved.

1. The S-box is the following permutation on $s = 3$ bits that has been chosen as a random non-linear permutation: $\{7, 6, 0, 4, 2, 5, 1, 3\}$. We will number its bits as follows: the input of the S-box is: $4 \cdot x_3 + 2 \cdot x_2 + x_1$, while the output is $4 \cdot y_3 + 2 \cdot y_2 + y_1$.
2. This S-box gives $r = 14$ fully quadratic equations with $t = 22$ terms, i.e. equations of the type:

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

   To be more precise, these equations are exactly:

   $$\begin{cases} 0 = x_1x_2 + y_1 + x_3 + x_2 + x_1 + 1 \\ 0 = x_1x_3 + y_2 + x_2 + 1 \\ 0 = x_1y_1 + y_2 + x_2 + 1 \\ 0 = x_1y_2 + y_2 + y_1 + x_3 \\ 0 = x_2x_3 + y_3 + y_2 + y_1 + x_2 + x_1 + 1 \\ 0 = x_2y_1 + y_3 + y_2 + y_1 + x_2 + x_1 + 1 \\ 0 = x_2y_2 + x_1y_3 + x_1 \\ 0 = x_2y_3 + x_1y_3 + y_1 + x_3 + x_2 + 1 \\ 0 = x_3y_1 + x_1y_3 + y_3 + y_1 \\ 0 = x_3y_2 + y_3 + y_1 + x_3 + x_1 \\ 0 = x_3y_3 + x_1y_3 + y_2 + x_2 + x_1 + 1 \\ 0 = y_1y_2 + y_3 + x_1 \\ 0 = y_1y_3 + y_3 + y_2 + x_2 + x_1 + 1 \\ 0 = y_2y_3 + y_3 + y_2 + y_1 + x_3 + x_1 \end{cases} \tag{1}$$

   One should note that there are many other ways of writing the equations that will also lead to interesting attacks.
3. The number of rounds is N_r.
4. Let $B = 1, 2, \ldots, 128$ be the number of S-boxes in each round. There are $B * s$ bits in each round. We number them $0..Bs - 1$, and we have in order 0 being x_1 of the first S-box, then we have x_2, x_3 of the first S-box, then x_1, x_2, x_3 of the second S-box (if any), etc.
5. The key size is equal to the block size and has H_k bits, with $H_k = 1, 2, 3, \ldots$. The key is: $k = (k_0, \ldots, k_{H_k-1})$. In comparison, in the older specification CTC one had $H_k = B * s$. We note that when $H_k = B * s$ and also when $H_k \leq B * s$, one known plaintext will be (on average) sufficient to determine (more or less uniquely) the secret key. When $H_k \leq 2 * B * s$ we will need to plaintexts etc.

6. Each round i consists of the XOR with the derived key K_{i-1}, a parallel application of the B S-boxes, and then of a linear diffusion layer D is applied (this replaces the simple permutation of wires used in [9]).
   For the last round an additional derived key K_{N_r} is XORed (as in AES).
7. We denote $X_{i\ j}$, for $i = 1..N_r$, $j = 0..Bs-1$, the *inputs* of the $i-th$ round after the XOR with the derived key.
8. We denote $Z_{i\ j}$, for $i = 1..N_r$, $j = 0..Bs-1$, the *outputs* of the $i-th$ round before the XOR with the next derived key.

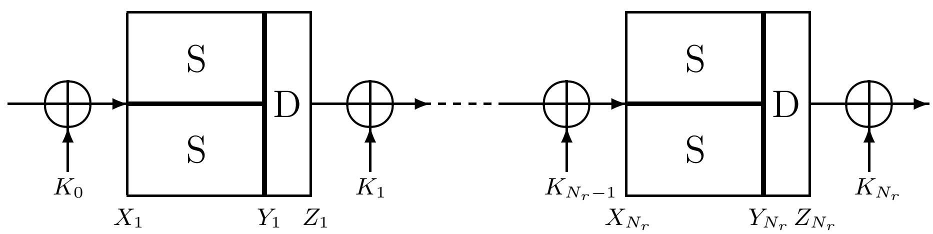

Fig. 1. A toy cipher with $B = 2$ S-boxes per round

9. I order to have uniform notations, we may also denote the plaintext by Z_0 and the ciphertext by X_{N_r+1}. These should not considered as variable names, but as abbreviations that denote (known) constant values.
10. There is no S-boxes in the key schedule and the derived key in round i, K_i is obtained from the global secret key bits $k = (k_1, \ldots, k_{H_k})$ as follows (with $i = 0, 1, \ldots, N_r$):
$$K_{i\ j} \stackrel{def}{=} k_{(j+i\cdot 509 \bmod H_k)}. \tag{2}$$
    In comparison, the older version CTC from [4] used the following:
$$K_{i\ j} \stackrel{def}{=} K_{0\ (j+i \bmod Bs)}. \tag{3}$$
11. With all these notations, the linear equations from the key schedule are as follows:
$$X_{i+1\ j} = Z_{i\ j} \oplus K_{i\ j} \quad \text{for all } i = 0..N_r. \tag{4}$$
12. The diffusion part D of the cipher is defined as follows:
$$\begin{cases} Z_{i\ (j\cdot 1987+257 \bmod Bs)} = & Y_{i\ j} \oplus Y_{i\ (j+137 \bmod Bs)} \oplus Y_{i\ (j+274 \bmod Bs)} \\ & \text{if } j = 257 \bmod Bs \\ Z_{i\ (j\cdot 1987+257 \bmod Bs)} = & Y_{i\ j} \oplus Y_{i\ (j+137 \bmod Bs)} \\ & \text{otherwise} \end{cases} \tag{5}$$
    In comparison, the older version CTC from [4] uses the following:
$$\begin{cases} Z_{i\ (j\cdot 1987+257 \bmod Bs)} = & Y_{i\ j} \\ & \text{if } j = 0 \\ Z_{i\ (j\cdot 1987+257 \bmod Bs)} = & Y_{i\ j} \oplus Y_{i\ (j+137 \bmod Bs)} \\ & \text{otherwise} \end{cases} \tag{6}$$

3 Cryptanalytic Results on CTC2

We did some trials in which we applied to CTC2 the same attacks that were applied to CTC in [4], as well as the conversion to SAT described in [3] and the MiniSat 2.0. algorithm [15].

3.1 How to Generate Equations

A windows executable program CTC2.exe can be downloaded from `www.cryptosystem.net/aes/toyciphers.html`. This program is public domain, though the source code is not currently available. The command line options of this program are:

```
CTC2.exe B Nr /insX [/fixY] [/cp] [/keyZ].
-B=number of S-boxes per round,
-Nr=number of rounds,
-/insX means use X plaintext/ciphertext pairs,
-/fixY means fix Y first key bits to their values to reduce the key space,
-with the option /cp the plaintexts are chosen to be consecutive,
it is in fact a counter mode,
-the option /keyZ modifies the size of the key, default is Z=B*3
which is equal to the block size.
```

The program generates two files in the current directory. One is the set of equations, the second is the solution (key). The solution is useful for verification and can be also used to design clever guess-then-algebraic attacks.

3.2 Achievement Metrics

Depending on the block size and the key size that remains to be found, we estimate the time of one encryption with CTC2 to be roughly between 2^7 and 2^9 CPU clocks. As a consequence, in one second one try between 2^{22} and 2^{24} keys on a 2 GHz CPU. Thus, in 256 seconds one can try between 2^{30} and 2^{32} keys and in one hour one can try between about 2^{34} and 2^{36} keys.

To conclude, in this paper we estimate that if we recover 36-bits of the key in less than one hour, or similarly if we compute 32-bits of the key in less than 256 seconds, the attack is faster than exhaustive search. These are rough estimates.

3.3 Results with Key Size Equal to Block Size

The results are very similar as for CTC, i.e. the same instances are broken within comparable time.

Example of a Successful Attack N$^{\text{O}}$ 1. For 40 S-boxes per round, 5 rounds, 2 chosen plaintext-ciphertext pairs that differ by only one bit, 120-bit key, and 85 first key bits fixed to their values, the remaining 35 bits of the key are found in less than 100 s with the simple ElimLin method described in [4]. This is clearly faster than exhaustive search, as explained in the previous section.

Example of a Successful Attack N$^{\mathbf{O}}$ 2. For 85 S-boxes per round, 6 rounds, 4 chosen plaintext-ciphertext pairs with consecutive plaintexts in the counter mode, 255-bit key, and 212 first key bits fixed to their values, the remaining 43 bits of the key are found in 110 hours with ElimLin method [4]. This is faster than exhaustive search that, following Section 3.2 and given the large size of this cipher, should take about 512 hours.

3.4 Results with Key Size $\neq$ Block Size

When the key size is larger than the block size, one can break 7 and more rounds of CTC2. In this case, the cost of guessing all key bits in the first round is less than the exhaustive search. Still, it is not so easy to break more than 6 rounds.

3.5 Results with Larger Key and Very Small Block Sizes

For 2 S-boxes per round, the block size is 6 bits. There are are $(2^6)! \approx 2^{296}$ permutations on 6 bits. Therefore it makes perfect cryptographic sense to build a cipher with 6-bit block size and key of up to 256 bits and beyond.

If the key size is n bits, we need at least $\lceil n/6 \rceil$ known (or chosen) plaintexts to be able to uniquely determine the key. Then any attack faster than exhaustive search breaks the cipher, and the attack is even more interesting if it uses very small number of known or chosen plaintexts, the minimum being $\lceil n/6 \rceil$.

Example of a Successful Attack N$^{\mathbf{O}}$ 3. For 2 S-boxes per round, 7 rounds, 6 known (not chosen) plaintext-ciphertext pairs, and 32-bit key, the key is recovered in 123 s with "conversion to MiniSat" method, see [5, 3, 15] This is (only very slightly) faster than brute force.

3.6 Results with Key Size $\neq$ Block Size and Larger Block Sizes

Example of a Successful Attack N$^{\mathbf{O}}$ 4. For 12 S-boxes per round, (36-bit blocks), 10 rounds, 1 known plaintext-ciphertext pairs, 256-bit key and 224 first key bits assumed to be known and fixed to their values, the remaining 32 bits of the key are found in 2.5 s with "conversion to MiniSat" method, see [5]. This is about 100 times faster than brute force.

Example of a Successful Attack N$^{\mathbf{O}}$ 5. For 32 S-boxes per round, (96-bit blocks), 7 rounds, 8 plaintext-ciphertext pairs with plaintexts that are consecutive as in the counter mode (it is a chosen-plaintext attack), 256-bit key and 220 first key bits assumed to be known and fixed to their values, the remaining 36 bits of the key are found in 1400 s with the ElimLin method described in [4]. This is about twice faster than brute force.

Example of a Successful Attack N$^{\mathbf{O}}$ 6. For 64 S-boxes per round, (192-bit blocks), 7 rounds, only 1 known plaintext, 256-bit key and 220 first key bits assumed to be known and fixed to their values, the remaining 36 bits of the key are found in 250 s with the ElimLin method described in [4]. This is substantially faster than brute force.

Example of a Successful Attack N^O 7. For 32 S-boxes per round, (96-bit blocks), 10 rounds, 1 known plaintext-ciphertext pairs, 256-bit key and 224 first key bits assumed to be known and fixed to their values, the remaining 32 bits of the key are found in 29 s with "conversion to MiniSat" method, see [5]. This is about 10 times faster than brute force.

4 Open Problems and Further Research

At present time, when the key size equals block size, nobody is able to break 7 rounds of CTC2. Only attacks that require a very small number (say at most 1024) of known or chosen plaintexts should be taken into consideration.

When the key size is longer than the block size, we can currently break up to 10 rounds of CTC2. More can probably be done for instances with artificially large key size.

All the attacks in this paper can be considered as simple and trivial, once they have been discovered, but in fact it seems that nobody has anticipated their existence, and a lot of energy were spend working in totally different directions. The goal for further research is to see if better results can be obtained. We encourage other researchers to try to solve the equations of CTC2 by their favorite method.

5 Conclusion

CTC2 is a simple toy cipher that allows to study algebraic attacks on block ciphers. Most researchers believe that algebraic attacks on block ciphers should use sophisticated Gröbner bases algorithms that work at high degree and should avoid reduction to 0. Recently however, Courtois has shown that among different systems of equations that can be written in algebraic cryptanalysis, some are much easier to solve than expected. Many systems are solved at degree 2, by ElimLin that is much simpler than any existing Gröbner bases algorithm, or by SAT solvers. The extreme sparsity and specific structure of equations derived from block ciphers, allows to manipulate and solve in practice much larger systems of equations than expected. For example, up to 10 rounds of CTC2 can be broken at present time, and here we only tried the most obvious attacks.

The attacks described in this paper can be treated as a reference point for future work on algebraic cryptanalysis. In order to encourage competition between researchers that work on systems that solve large sparse systems of multivariate equations, we have made our tool to generate equations publicly available.

It is a pity that we understand so little about cryptanalysis of block ciphers. Papers like this one, are frequently judged as of relatively small interest. Yet, arguably until now, there was no sensible metric of achievement in cryptanalysis, and certain fundamental questions regarding what is easy and what is hard to achieve were misunderstood. It appears that the whole research in symmetric cryptography is heavily distorted: impractical attacks (that require unrealistic quantities of known plaintexts) occupy a dominant position, while important practical attacks (maybe computation-intensive but using a handful of known/chosen plaintexts) are being spectacularly neglected and disregarded.

Currently, since nobody cannot break more than 10 rounds of CTC2 with algebraic attacks, it may be very difficult to ever hope to break modern block ciphers such as AES that have 10 substantially more complex rounds. However one should understand why is it so: this is because the designers of such ciphers have considerably increased the number of rounds, in order to avoid the aforementioned classical cryptographic attacks, that are all pure theory and can only be executed in fictional scenarios. With this (excessively large) number of rounds, we conjecture for the time being, that ciphers such as AES should also be secure[1] against algebraic cryptanalysis, which maybe finally happens by accident.

From the research perspective however, if we want to learn anything about cryptanalysis, we really need to study the science of breaking ciphers from the start and in realistic conditions: when the number of rounds is not excessive and the plaintext material is scarce. **It is amazing to see that nobody has addressed such fundamental questions before.** A lot remains to be done.

[1] Still, this is to some extent impossible to know because only algebraic attacks with very small computing power, mostly at degree 2, and with poor handling of sparsity have been studied so far. These attacks are still very new, important improvements can still be found.

References

1. Ross Anderson, Eli Biham and Lars Knudsen: *Serpent: A Proposal for the Advanced Encryption Standard.* Available from `http://www.cl.cam.ac.uk/~rja14/serpent.html`
2. C. Berbain, H. Gilbert, and J. Patarin: *QUAD: A Practical Stream Cipher with Provable Security,* In Eurocrypt 2005, LCNS, Springer, 2005.
3. 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/`.
4. Nicolas T. Courtois *How Fast can be Algebraic Attacks on Block Ciphers?* Available at `http://eprint.iacr.org/2006/168/`.
5. Nicolas T. Courtois and Gregory V. Bard: *Algebraic Cryptanalysis of the Data Encryption Standard,* Available at `http://eprint.iacr.org/2006/402/`.
6. Nicolas Courtois, Gregory V. Bard: *Algebraic and Slide Attacks on KeeLoq,* preprint, `eprint.iacr.org/2007/062/`.
7. 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.
8. Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Asiacrypt 2002, LNCS 2501, pp.267-287, Springer.
9. Nicolas Courtois and Josef Pieprzyk: *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations,* Available at `http://eprint.iacr.org/2002/044/`.
10. Joan Daemen, Vincent Rijmen: *AES proposal: Rijndael,* The latest revised version of the proposal is available on the internet, `http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf`
11. Joan Daemen, Vincent Rijmen: *The Design of Rijndael. AES - The Advanced Encryption Standard,* Springer-Verlag, Berlin 2002. ISBN 3-540-42580-2.
12. Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, Springer, pp. 266-281.
13. Orr Dunkelman and Nathan Keller: *Linear Cryptanalysis of CTC,* Available at `http://eprint.iacr.org/2006/250/`.
14. L. Marraro, and F. Massacci. *Towards the Formal Verification of Ciphers: Logical Cryptanalysis of DES, Proc. Third LICS Workshop on Formal Methods and Security Protocols, Federated Logic Conferences (FLOC-99).* 1999.
15. 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/`
16. F. Massacci. *Using Walk-SAT and Rel-SAT for cryptographic key search, Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI'99).* 1999.
17. Jacques Patarin: *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*; Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
18. RSA Security RC5 Challenges status, see `http://www.rsa.com/rsalabs/node.asp?id=2103`.
19. 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.
20. Claude Elwood Shannon: *Communication theory of secrecy systems,* Bell System Technical Journal 28 (1949), see in particular page 704.