Algebraic Cryptanalysis of a Small-Scale Version of Stream Cipher LEX

Vesselin Velichkov Vincent Rijmen Bart Preneel
Katholieke Universiteit Leuven
Dept. ESAT/SCD, Computer Security and Industrial Cryptography Group
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium
{vesselin.velichkov, vincent.rijmen, bart.preneel}@esat.kuleuven.be

Abstract

In this paper * we analyse with respect to algebraic attacks a small-scale version of the stream cipher Lex. We base it on a small-scale version of the block cipher AES with 16-bit state and 16-bit key. We represent the small-scale Lex and its key schedule in two alternative ways: as a system of cubic boolean equations and as a system of quadratic boolean equations. We use Gröbner bases to solve the two systems for different number of rounds and sizes of the leak. We obtain the best results for the quadratic representation of the cipher. For this case we are able to recover the secret key in time less than 2 minutes by solving a system of 374 quadratic boolean equations in 208 unknowns resulting from 5 rounds of the cipher.

1 Introduction

Lex is a 128-bit key stream cipher proposed by Alex Biryukov in [1]. Lex was selected for phase 3 of the eSTREAM competition [13]. It was not chosen for the eSTREAM portfolio. Lex is based on the notion of "leak extraction" which is defined in [1].

The motivation for the current work is the following citation from the design of Lex [1, Section 3.3]: "Applicability of these [algebraic attacks] to LEX is to be carefully investigated. If one could write a non-linear equation in terms of the outputs and the key - that could lead to an attack.".

Three attacks on Lex have been published so far [2, 3, 4]. None of them exploits the algebraic structure of the cipher. With the presented work we try to make a small step towards filling this gap.

The paper is organized as follows. In Section 2 we give an overview of the existing attacks on Lex. In Section 3 we give a short description of stream cipher Lex. In Section 4 we propose Lex(2,2,4) - a small-scale version of Lex. In Section 5 we represent Lex(2,2,4) and its key schedule as a system of cubic and quadratic boolean equations. In Section 6 we describe a modification of the Gröbner bases attack algorithm presented in [7] which we apply for a key recovery attack on Lex(2,2,4). In Section 7 we give information on the experimental setting in which we perform our experiments. In Section 8 we describe our results and in Section 9 we conclude.

*This work was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The first author is funded by a DBOF fellowship of K.U.Leuven.

2 Previous work

In [2] Wu and Preneel presented a slide attack on the original version of Lex [1]. The attack exploits the initialization phase of the cipher. It requires 500×320 bits of key stream, each generated from $2^{60.8}$ different IVs under the same key. As a result of finding three collisions in the output key stream, 96 bits of the key can be recovered. The remaining 32 bits of the key are recovered by exhaustive search. Subsequently Lex was tweaked to resist the attack by using a full AES encryption during initialization (instead of the modified version of AES used before). No change was made to the stream generation.

In [3] Johansson it. al present an attack on Lex in which it is possible to decrypt some ciphertext without recovering the key. The attack requires $2^{65.66}$ key stream bits produced by one IV and the first approx. 320 bits from $2^{65.66}$ other IVs. This attack is not applicable to the tweaked version of Lex [1], where the maximum number of IVs used under the same key is required to be 2^{32}.

The most recent attack on Lex is the one proposed by Dunkelman and Keller [4]. The attack identifies special states in two AES encryptions which satisfy a certain difference pattern. The secret key is retrieved in time of 2^{112} operations using $2^{36.3}$ bytes of key stream produced by the same key. The attack is applicable also to the tweaked version of Lex.

3 Lex

In this Section we give a short overview of stream cipher Lex. From now on whenever we refer to Lex we shall mean its 128-bit version - Lex-128.

Lex is based on the block cipher AES [5]. It has 128-bit key and 128-bit IV. During initialization the key of Lex is expanded into 11 round keys by a standard AES key schedule. Next the IV is encrypted with AES-128, the first round key is *exor*-ed with the output and the result becomes the input to the first round of Lex. The input to every round of Lex is transformed to the output by the AES round transformation circularly using the first 10 of the 11 round keys. After every round, four bytes of the output (the "leaks") are extracted as four bytes of the key stream produced by Lex. At odd rounds the four bytes of the leak are extracted at positions $(0,0),(0,2),(2,0),(2,2)$; at even rounds the four bytes of the leak are extracted at positions $(0,1),(0,3),(2,1),(2,3)$. The output of every round is fed as the input to the next round. The operation of Lex is shown in Figure 1.

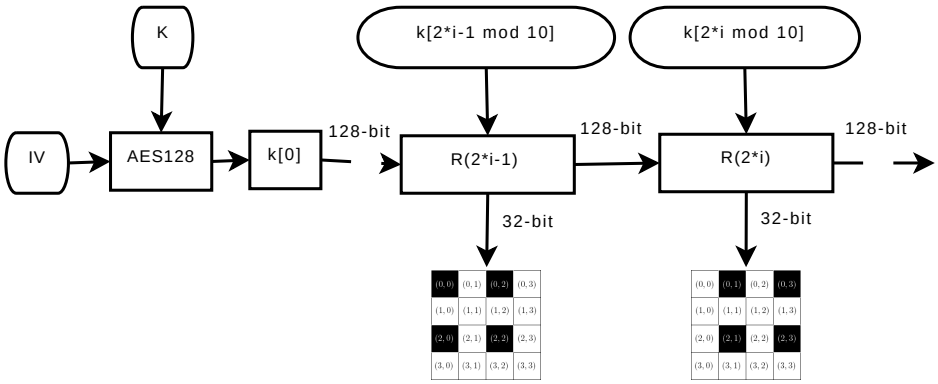

Figure 1: Lex: R is the round transformation of AES-128, $i \geq 1$

4 Lex(2,2,4)

With Lex(2,2,4) we designate a small-scale version of stream cipher Lex based on the block cipher SR(10,2,2,4). SR(10,2,2,4) is one of the small-scale versions of AES proposed in [6]. Lex(2,2,4) has a state of 2×2 words of size 4 bits each. Thus Lex(2,2,4) has 16-bit state and 16-bit key. At every round Lex(2,2,4) leaks 4 bits, which is $\frac{1}{4}$-th of the whole state as is also the case for Lex. At odd rounds the byte of the leak is extracted at position $(0, 0)$; at even rounds the byte of the leak is extracted at position $(0, 1)$. The operation of Lex(2,2,4) is identical to the one of Lex and is shown in Figure 2.

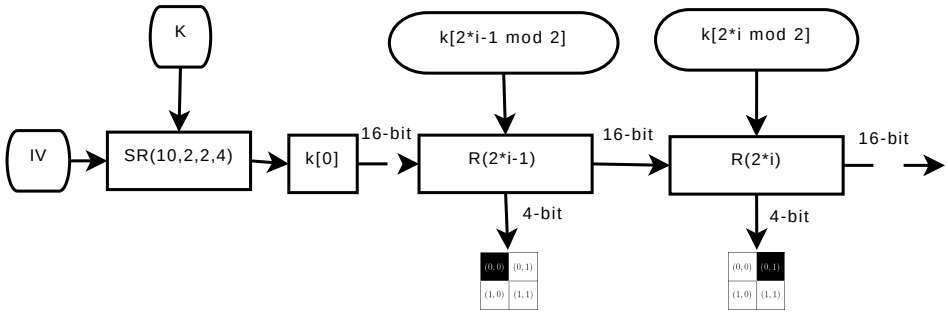

Figure 2: Lex(2,2,4): R is the round transformation of SR(10,2,2,4) , $i \geq 1$

5 Constructing Equations for Lex(2,2,4)

In this Section we describe a representation of Lex(2,2,4) and its key schedule as a system of boolean equations. The polynomials composing the equations are in the ring of boolean polynomials $GF(2^4) \equiv GF(2)[z]/ < z^4 + z + 1 >$. We use two alternative representations of Lex(2,2,4): as a system of cubic equations and as a system of quadratic equations. We base our second representation on the quadratic equations representation of Rijndael described in [10]. The information in this section is summarized in Table 1.

5.1 Cubic Equations

The system of cubic equations representing Lex(2,2,4) is composed of two sets of equations: cipher equations and key schedule equations. The variables composing the cipher equations for one round are the input and output bits of the round. Note that the output bits of one round are also the input bits for the next round. Thus for one round there are 32 variables. For every additional round 16 new variables are added. The variables composing the key schedule equations are the bits of the round keys. For two round keys there are 32 key variables. A complete system of cubic equations for one round and two round keys is composed of 36 equations in 64 variables (see Table 1).

5.2 Quadratic Equations

In [10] the cipher Rijndael is represented as a system of multivariate quadratic equations. This representation uses the fact that each Rijndael S-box is completely defined

by a system of 23 quadratic equations [9]. In a similar way we construct a system of multivariate quadratic equations representing one round of Lex(2,2,4) and its key schedule for two round keys. We describe each 4-bit S-box of Lex(2,2,4) as a system of 11 quadratic equations. Similarly to the cubic case, the complete system of quadratic equations representing Lex(2,2,4) is composed of two sets of equations: cipher equations and key schedule equations. The variables of the cipher equations are the input and output bits of the S-boxes. There are four 4-bit S-boxes in one round of Lex(2,2,4). The cipher equations for one round include the input and output bits of the four S-boxes of the round plus the input bits to the four S-boxes of the next round. Thus there are 48 variables in the cipher equations for one round. The variables of the key schedule equations are the bits of the initial key and the input and output bits of the S-boxes in the key schedule. There are two 4-bit S-boxes in the key schedule for two round keys (where the second key is derived from the first). Thus there are 32 variables in the key schedule equations for two round keys. A complete system of quadratic equations for one round and two round keys is composed of 94 equations in 80 variables (see Table 1).

	Lex(2,2,4)	**Cubic**	**Quadratic**
Cipher	Variables	32	48
	Linear eqs.	4	20
	Nonlinear eqs.	16	44
Key schedule	Variables	32	32
	Linear eqs.	8	8
	Nonlinear eqs.	8	22
Total	Variables	64	80
	Equations	36	94

Table 1: Equations for one round of Lex(2,2,4) and two keys

6 Key recovery attack on Lex(2,2,4) using Gröbner bases

In [7] is presented a general Gröbner bases attack algorithm. It is applied for key recovery attacks on instances of block ciphers FLURRY and CURRY. This algorithm is also discussed in [8]. In this Section we adapt the algorithm from [7] for the case of Lex(2,2,4). We describe the modified algorithm next.

1. Set up a polynomial system $\mathcal{E} = \{e_i = 0\}$ for R rounds of Lex(2,2,4), starting from round 1 (so that we can use the bits of the leak after round 0). The equations $\{e_i = 0\}$ are obtained for each round as discussed in Section 5. The system $\mathcal{E}$ consists of cipher equations, key schedule equations and leak equations.

2. Set the number of bits L which are guessed at the output of every round (it is possible that $L = 0$). In this way the total number of leaked bits per round becomes $4 + L$. Because we have constructed $\mathcal{E}$ starting from round 1,for R rounds we have $(R + 1)$ leaks of size $(4 + L)$ bits each. The total number of guessed bits for R rounds is $(R + 1)L$.

3. For all possible $2^{(R+1)L}$ values of all guessed bits do:

3a. Let the current value of the guessed bits to be $l_0, l_1, \ldots, l_{(R+1)(L-1)}$. Compose the system $\mathcal{D} = \{d_i = 0\}$ of $(R+1)L$ additional linear equations arising from the guessed bits:

$$\begin{aligned} x_0^{(0)} + l_0 &= 0 \\ x_1^{(0)} + l_1 &= 0 \\ &\ldots \\ x_{L-1}^{(R)} + l_{(R+1)(L-1)} &= 0, \end{aligned}$$

where $x_i^{(r)}, 0 \leq r \leq R, 0 \leq i \leq L-1$ are the variables corresponding to the guessed bits from the leak after round r. Let $\mathcal{I}$ be the ideal generated by the set of polynomials $\mathcal{P} = (\bigcup_i e_i) \cup (\bigcup_i d_i)$. Following the terminology of [7] we call $\mathcal{I}$ the key recovery ideal.

3b. Compute the dimension $dim(\mathcal{I})$ of $\mathcal{I}$. If $dim(\mathcal{I}) = 0$ (a finite number of solutions exist) then do:

3bi. Compute a degree-reverse lexicographic Gröbner basis G of $\mathcal{I}$.

3bii. Compute the variety V of G. V contains the solutions to the system $\mathcal{E} \cup \mathcal{D}$, including the key bits. Store the key bits of the solutions in a list T of possible key candidates.

4. For all entries t_i in T do: use t_i as a key for Lex(2,2,4) and produce output for $r > R$ rounds. Compare the outputs from the last $r - R$ rounds with the output for the same rounds produced by Lex(2,2,4) under the secret key. If the outputs match, then t_i is the secret key - store it in k and go to next step.

5. Return k and terminate.

7 Experimental setting

We have performed our experiments using the open-source computer algebra system Sage [11] on a machine with 2.2 GHz CPU AMD Opteron(tm) Processor 275 and 4 GB RAM with OS GNU/Linux. For computation of Gröbner bases in the boolean polynomial ring Sage uses the open-source library PolyBoRi [12].

8 Results

For different number of rounds and sizes of the leak we construct a system of equations as discussed in Section 5. We solve the system using the algorithm described in

Section 6. The results from our experiments are shown in Table 2 and Table 3. Each row of Tables 2 and 3 gives information on constructing and solving the system of equations resulting from Lex(2,2,4) for a given number of rounds and a given size of the leak after each round. The fact that the system resulting from a certain size of the leak cannot be solved is indicated by the abbreviation "n/a" (not available) in the last three columns of the tables. The rest of the information in the tables is the following:

R. Number of rounds for which equations are generated: between 1 and 6. We chose to limit our experiments to the first 6 rounds because at round 5 (for the quadratic case) we are already able to recover the key without guessing any bits from the leaks.
Leak. The number of bits which are leaked after every round. Four bits of every leak are known by design. The remaining bits of the leak (if any) are guessed.
Guess. Total number of guessed bits for the specified number of rounds and size of the leak.
Eqs. The number of equations in one system resulting for the specific number of rounds and leaks.
Var. The number of variables participating in the equations in one system.
Odef. A measure of the extent to which the algebraic system is over-defined. This value is obtained by dividing the number of equations by the number of variables.
Sol. Number of solutions to the given system.
Gb. The time (in seconds) necessary for the computation of the Gröbner basis of the polynomials composing the given system.
Variety. The time (in seconds) necessary for the computation of the algebraic variety of the given system.

From the data in Table 2 and Table 3 it can be seen that the best result is obtained for the quadratic representation of Lex(2,2,4) (Table 3) for 5 rounds and 4-bit leak (bold row). In this case we solve a system of 374 equations in 208 variables. We obtain one solution which contains the bits of the secret key. The times necessary for the computation of the Gröbner basis and the variety are 1.024 sec and 85.01 sec respectively. Given the fact that those are the two most computationally expensive operations we can estimate the total timing of the attack to be less than 2 minutes.

9 Conclusion

In this paper we presented algebraic cryptanalysis of a small-scale version of the stream cipher Lex [1]. We were able to recover the secret key of the small-scale cipher in time less than 2 minutes by solving a system of 374 quadratic boolean equations in 208 unknowns resulting from 5 rounds of the cipher. Although mathematically successful, this result cannot be classified as an attack in the cryptographic sense. The reason is that in our experimental environment exhaustive search on 2^{16} values was measured to take 0.248 seconds. Nevertheless our results indicate that our approach may be successful when applied to the full-scale version of LEX, in which case exhaustive key search is not a trivial task. This is a possible direction for future work.

10 Acknowledgements

The authors would like to thank Martin Albrecht for proofreading and insightful comments on the preliminary draft as well as for his extremely helpful suggestions for improving the efficiency of the Gröbner basis computation in Sage.

r	leak	guess	eqs	var	odef	sol	gb, *sec*	variety, *sec*
1	16	24	64	64	1.000	1	0.144	0.35
	15	22	62	64	0.969	4	0.144	0.60
	14	20	60	64	0.937	n/a	n/a	n/a
2	12	24	84	80	1.050	1	0.200	0.730
	11	21	81	80	1.012	1	0.212	0.730
	10	18	78	80	0.975	5	0.228	1.460
	9	15	75	80	0.938	n/a	n/a	n/a
3	11	28	108	96	1.125	1	0.260	1.450
	10	24	104	96	1.083	1	0.276	1.450
	9	20	100	96	1.041	1	0.256	1.480
	8	16	96	96	1	n/a	n/a	n/a
4	10	30	130	112	1.160	1	0.336	2.880
	9	25	125	112	1.116	1	0.328	2.800
	8	20	120	112	1.071	1	0.328	2.810
	7	15	115	112	1.027	n/a	n/a	n/a
5	9	30	150	128	1.171	1	0.424	8.52
	8	24	144	128	1.125	1	0.436	10.55
	7	18	138	128	1.078	1	0.412	10.63
	6	12	132	128	1.031	n/a	n/a	n/a
6	9	35	175	144	1.215	1	0.512	18.71
	8	28	168	144	1.166	1	0.508	19.08
	7	21	161	144	1.118	1	0.536	19.28
	6	14	154	144	1.069	n/a	n/a	n/a

Table 2: Cubic equations for Lex(2,2,4)

References

[1] Alex Biryukov, "The Design of a Stream Cipher LEX," Selected Areas in Cryptography 2006:67-75

[2] Bart Preneel, Hongjun Wu, "Resynchronization Attacks on WG and LEX," FSE 2006:422-432

[3] Englund, Hell, Johansson, "A Note on Distinguishing Attacks," Information Theory for Wireless Networks, 2007 IEEE Information Theory Workshop, Publication Date: 1-6 July 2007, pages: 1-4

[4] Orr Dunkelman, Nathan Keller, "A New Attack on the LEX Stream Cipher," ASIACRYPT 2008: 539-556

[5] Joan Daemen, Vincent Rijmen, "The Design of Rijndael," AES - The Advanced Encryption Standard, Springer-Verlag, 2002

[6] Carlos Cid, Sean Murphy, Matthew J. B. Robshaw, "Small Scale Variants of the AES," FSE 2005, 145-162

[7] Johannes Buchmann, Andrei Pyshkin, Ralf-Philipp Weinmann, "Block Ciphers Sensitive to Groebner Basis Attacks," CT-RSA 2006, 313-331

r	leak	guess	eqs	var	odef	sol	gb, *sec*	variety, *sec*
1	8	8	114	80	1.425	256	0.256	41.18
	7	6	111	80	1.387	n/a	n/a	n/a
2	8	12	190	112	1.696	1	0.364	2.79
	7	9	185	112	1.651	1	0.352	2.77
	6	6	180	112	1.607	4	0.360	3.68
	5	5	175	112	1.562	n/a	n/a	n/a
3	8	16	266	144	1.847	1	0.592	15.14
	7	12	259	144	1.799	1	0.572	14.93
	6	8	252	144	1.750	1	0.564	15.09
	5	4	245	144	1.701	2	0.524	15.38
	4	0	238	144	1.653	n/a	n/a	n/a
4	8	20	342	176	1.943	1	0.784	39.77
	7	15	333	176	1.892	1	0.768	39.99
	6	10	324	176	1.841	1	0.744	40.05
	5	5	315	176	1.789	1	0.740	40.12
	4	0	306	176	1.739	n/a	n/a	n/a
5	8	24	418	208	2.009	1	1.012	84.67
	7	18	407	208	1.957	1	1.004	84.42
	6	12	396	208	1.904	1	1.032	84.27
	5	6	385	208	1.851	1	1.024	84.43
	4	**0**	**374**	**208**	**1.798**	**1**	**1.024**	**85.01**
6	8	28	494	240	2.058	1	1.364	168.92
	7	21	481	240	2.004	1	1.400	157.73
	6	14	468	240	1.950	1	1.312	157.49
	5	7	455	240	1.895	1	1.300	157.36
	4	0	442	240	1.841	1	1.316	157.72

Table 3: Quadratic equations for Lex(2,2,4)

[8] Martin Albrecht, "Algebraic Attacks on the Courtois Toy Cipher," Journal Cryptologia, Volume 32, Issue 3 (July 2008), pages 220-276, ISSN,0161-1194

[9] Nicolas Courtois, Josef Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations," ASIACRYPT 2002, 267-287

[10] Alex Biryukov, Christophe De Canniere, "Block Ciphers and Systems of Quadratic Equations" FSE 2003, 274-289

[11] Stein, William, "*Sage, Open Source Mathematical Software (Version 3.1.4)*," The Sage Group, 2008, `http,//www.sagemath.org`.

[12] M. Brickenstein, A. Dreyer, "PolyBoRi: A framework for Groebner basis computations with Boolean polynomials," Electronic Proceedings of the MEGA 2007 - Effective Methods in Algebraic Geometry, Strobl, Austria, June 2007

[13] eSTREAM, ECRYPT Stream Cipher Project, http://www.ecrypt.eu.org/stream/

[14] Bruno Buchberger, "An Algorithmical Criterion for the Solvability of Algebraic Systems of Equations," Aequationes Mathematicae, vol. 4, no. 3, 1970, pp. 374-383.