Cryptanalysis of LEDAcrypt

Daniel Apon[1], ***Ray Perlner***[1], Angela Robinson[1], Paolo Santini[2]

1: NIST

2: Università Politecnica delle Marche, Florida Atlantic University

Significance

- We present an attack on the QC-LDPC-McEliece construction of [Baldi et al. 2007]
- This construction was the basis of the second-round NIST PQC candidate, LEDAcrypt
- Prior to our attack this construction had a nearly 12-year history without a major break
- Our attack was a major factor in the non-selection of LEDAcrypt for the third round of the NIST PQC process
  - In response, the LEDAcrypt team published an updated spec which avoided the attack
  - NIST ultimately decided that this updated spec represented too large a tweak and made LEDAcrypt too similar to its competitor BIKE

  (BIKE is based on the QC-MDPC-McEliece scheme of [Misoczki et al. 2012])

LEDAcrypt Overview

- Conceptually very similar to QC-MDPC McEliece/Niederreiter
  - Private key is a sparse binary quasicyclic parity check matrix:
    $$L = (L_0 \quad \dots \quad L_{n_0-1})$$
  - Public key is a systematic form quasicyclic parity check matrix for the same code:
    $$M = {L_{n_0-1}}^{-1} L$$
  - Cyclic blocks are of dimension p and can be treated as polynomials in
    $$F_2[x]/\langle x^p - 1 \rangle$$
  - Recovering any row of L from M is sufficient to break the scheme
- Unique feature of unpatched LEDAcrypt:
  - Private key factors into two sparser matrices H and Q:

$$L = HQ = (H_0 \quad \dots \quad H_{n_0-1}) \begin{pmatrix} Q_{0,0} & \cdots & Q_{0,n_0-1} \\ \vdots & \ddots & \vdots \\ Q_{n_0-1,0} & \cdots & Q_{n_0-1,n_0-1} \end{pmatrix}$$

LEDAcrypt Parameters

- n_0: Number of cyclic blocks in H, L, and M
- p: Dimension of cyclic blocks
- d_v: Row Hamming weight of each block of H
- $m = (m_0, m_1, \ldots m_{n_0-1})$: Row weights of blocks of Q arranged like, e.g.:

$$\begin{pmatrix} m_0 & m_1 & m_2 & m_3 \\ m_3 & m_0 & m_1 & m_2 \\ m_2 & m_3 & m_0 & m_1 \\ m_1 & m_2 & m_3 & m_0 \end{pmatrix}$$

- t: Errors corrected by L, in decrypt/decaps (irrelevant for our attack)

LEDAcrypt Parameters (2nd Round, CPA)

NIST Cat.	n_0	p	t	d_v	m	errors out of decodes
1	2	$14,939$	136	11	$[4,3]$	14 out of $1.2 \cdot 10^9$
	3	$7,853$	86	9	$[4,3,2]$	0 out of $1 \cdot 10^9$
	4	$7,547$	69	13	$[2,2,2,1]$	0 out of $1 \cdot 10^9$
3	2	$25,693$	199	13	$[5,3]$	2 out of $1 \cdot 10^9$
	3	$16,067$	127	11	$[4,4,3]$	0 out of $1 \cdot 10^9$
	4	$14,341$	101	15	$[3,2,2,2]$	0 out of $1 \cdot 10^9$
5	2	$36,877$	267	11	$[7,6]$	0 out of $1 \cdot 10^9$
	3	$27,437$	169	15	$[4,4,3]$	0 out of $1 \cdot 10^9$
	4	$22,691$	134	13	$[4,3,3,3]$	0 out of $1 \cdot 10^9$

Summary of Attacks

- Weak key attack (All parameter sets)
  - A class of keys produced by LEDAcrypt's keygen with probability 2^{-x}, that can be recovered by an attack requiring the equivalent of 2^{y} AES operations
  - Considered an attack if $x + y$ less than the security parameter λ
  - E.g.
    - For category 5 CPA parameters with $n_0 = 2$ (most effective relative to claimed security level), $x = 47.72; y = 49.22; x + y = 96.94$
    - For category 5 CCA parameters with $n_0 = 2$ $x = 57.50; y = 52.54; x + y = 110.04$
    - For category 1 CPA parameters with $n_0 = 4$ (least effective relative to claimed security level), we expect $x \approx 40; y \approx 50$

Summary of Attacks Cont.

- Average case attack (Asymptotic)
  - Can be considered an extension of the weak key attack with x<<1
  - Difficult to estimate concrete advantage over standard attacks
  - we suspect it is significant already for claimed category 5 parameters with $n_0 = 2$.

Key Recovery for MDPC Codes
Information Set Decoding

- Basic idea: Guess p bits of low weight row of L
  - Note that the rows of L are in the row space of M
  - Linearly solve for the rest of the row
  - The p bits we guess are called the "*information set*"
- More detailed procedure:
  - Permute columns of M resulting in M' $= MP = (A|B)$.
    - Hope first p bits of a row of LP are (1, 0, …, 0).
    - If so, the row of LP is the top row of $A^{-1}M$'
    - More advanced ISD algorithms e.g. Stern, Leon, MMT, BJMM, MO… reduce complexity somewhat by trying multiple guesses for the first p bits of a row of LP
    - Asymptotic complexity where LP has row weight w : $\left(\frac{1}{p}\right)\left(\frac{n_0}{n_0-1}\right)^w$

Using LEDAcrypt's Product Structure Basic Idea

- Parameters of LEDAcrypt are set based on treating the code defined by M as an MDPC code and running the ISD attack on the previous slide
  - Attack complexity is essentially the inverse probability of guessing a randomly chosen p bits of a row of L
- Idea: Choose the bits to guess non-randomly

Using LEDAcrypt's Product Structure
Choosing the Bits to Guess

- Want to find p bits of a row of L that are more likely than average to be (almost) entirely zero
- Equivalently: Want (almost) all the nonzero bits of the row of L to be in the remaining $(n_0-1)p$ bits
- Define those $(n_0-1)p$ bits as the support of a module in $(\mathbb{Z}[x]/\langle x^p-1\rangle)^{n_0}$ given by

$$L' = H'Q' = \begin{pmatrix} H'_0 & \cdots & H'_{n_0-1} \end{pmatrix} \begin{pmatrix} Q'_{0,0} & \cdots & Q'_{0,n_0-1} \\ \vdots & \ddots & \vdots \\ Q'_{n_0-1,0} & \cdots & Q'_{n_0-1,n_0-1} \end{pmatrix}$$

- **If the supports of H' and Q' contain the supports of H and Q respectively, then all the nonzero bits of the support of L are contained in the support of L'**

Contiguous Nonzero Coefficients

- The attack is not very good unless H' and Q' are chosen carefully
  - We want a significant fraction of the bits of $H'Q'$ to be zero so we can guess that L has the same zero bits
  - But generally a product of two polynomials has quadratically more nonzero coefficients than the starting polynomials, which would make H' and Q' quite sparse
  - This would make it very unlikely that the supports of H and Q are contained in H' and Q' respectively
- In contrast, if two polynomials are chosen with large numbers of consecutive coefficients,
  - e.g. $1 + x + x^2 + \cdots + x^{a-1}$ and $1 + x + x^2 + \cdots + x^{b-1}$,
  - the product only has only $a + b - 1$ nonzero coefficients
  - We will use polynomials like this in our attacks

Example: Weakest Keys (Category 5, $n_0 = 2$)

- $p = 36877; d_v = 11; m = (7, \quad 6)$

- Choose $H'_i = Q'_{j,k} = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor}$
- Probability that each nonzero bit of H_i, $Q_{j,k}$ is contained in support of H'_i, $Q'_{j,k}$ as appropriate is ~1/4.
- The total number of nonzero bits in these polynomials is
$$11 \cdot 2 + 7 \cdot 2 + 6 \cdot 2 = 48$$
- So we might guess that a single iteration of ISD with this information set would recover 1 in $4^{48} = 2^{96}$ private keys
- But wait, there's more!

Equivalent Keys

- Many choices for the private key components, H and Q will produce the same public key M

- **In particular**

$$\boldsymbol{H_0, H_1, Q_{0,0}, Q_{0,1}, Q_{1,0}, Q_{1,1}}$$

And

$$\boldsymbol{x^{\alpha} H_0, x^{\beta} H_1, x^{\gamma-\alpha} Q_{0,0}, x^{\gamma-\alpha} Q_{0,1}, x^{\gamma-\beta} Q_{1,0}, x^{\gamma-\beta} Q_{1,1}}$$

Are valid private keys with the same public key for any integers α, β, γ!

  - If any equivalent private key has support within support H', Q', that key can be recovered
  - Doesn't help as much as you might think, since small changes in α, β, γ don't usually change whether support of H', Q' contains support of H, Q
  - Nonetheless, this consideration brings number of keys broken by single information set up to about 1 in 2^{80}

- But wait, there's more!

Equivalent Choices of H' and Q'

- We generated our information set by taking

$$H'_i = Q'_{j,k} = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor}$$

- But we'd get the same information set by taking

$$H'_0 = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor + a}$$

$$H'_1 = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor + b}$$

$$Q'_{0,0} = Q'_{0,1} = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor - a}$$

$$Q'_{1,0} = Q'_{1,1} = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor - b}$$

- This consideration brings the number of keys broken by a single iteration of ISD up to 1 in $2^{72.8}$
- But wait, there's more!

Advanced Information Set Decoding

- ISD does not require that we only guess zeroes
  - In fact it requires that we don't
  - Advanced information set decoding algorithms e.g. Stern, MMT, BJMM, MO can tolerate up to about 6 nonzero bits in the information set without increasing the cost of an iteration

- Can be modeled by letting the support of H,Q be contained in higher-weight polynomials like:

$$H'_i = Q'_{j,k} = 1 + x + x^2 + \cdots x^{\left\lfloor \frac{p}{4} \right\rfloor + \varepsilon}$$

- If so, we expect nonzero bits in H and Q to be distributed like this:

within support of $L' = H'Q'$

Advanced Information Set Decoding Cont.

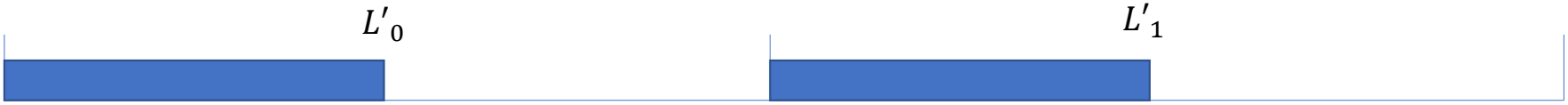
- We expect nonzero bits in H and Q to be distributed like this:

within support of $L' = H'Q'$

- As long as no more than 6 nonzero bits are outside the middle $\frac{p}{2}$ bits of the support, we can recover the key
- This consideration brings the number of keys broken by a single iteration of ISD up to 1 in $2^{62.66}$

How Many Equally Good (and Independent) Information Sets?

- Our information set is defined by the support of $L' = H'Q'$
- We can graph the support we've been using as:



- Two things we can change:
  - The relative offset of the two blocks
  - The ring representation in which nonzero coefficients are consecutive

Changing the Offset

- Results in an L' that looks like:

- Note that shifting both blocks the same amount just gives an equivalent key
- Shifting by a small amount doesn't change much
- There are about $m_0 + m_1 + d_v = 24$ independent choices of offsets

Ring Representations

- There is a large family of Hamming weight preserving ring isomorphisms for $\mathbb{Z}[x]/\langle x^p - 1\rangle$ given by $f(x) \to f(x^k)$
- We can try polynomials which have consecutive nonzero coefficients in the image under one of these isomorphisms, and everything still works
- E.g. We can have $H'_i = Q'_{j,k} = 1 + x^k + x^{2k} + \cdots x^{k\left\lfloor \frac{p}{4} \right\rfloor}$
- Choices of k between 1 and $\frac{p-1}{2}$ result in mostly independent information sets
  - (k and $-k$ result in equivalent information sets)

Rejection Sampling Considerations

- Our calculation above assumes any H and Q with the correct weights results in a valid key
- In fact, the key generation procedure for unpatched LEDAcrypt, rejects any L which is not full weight
  - We estimate 67.4% of the weakest keys are rejected
  - While only 39.2% of all keys are rejected
- This results in ~1 bit of security gained against our attack

- Thanks: Corbin McNeil for analyzing this consideration

Putting it All Together

- We have about $2^{18.72}$ (mostly) independent ways to recover about 1 in 2^{64} private keys for the cost of a single 36877×36877 matrix inversion
  - These recover at least 1 in $2^{47.74}$ private keys total
  - Assume they cost about $2^{30.5}$ AES operations
- So for about $2^{30.5+18.72} = 2^{49.22}$ AES operations, we can recover 1 in $2^{47.74}$ keys

Considerations for $n_0 > 2$

- Naïvely applying the previous approach to cases where $n_0 > 2$ requires constraints on the support of $n_0 + {n_0}^2$ polynomials in the private key
- Attack works better when we only try to guess the support of 2 blocks of L at a time
- E.g. We can try to find low weight codewords in the row space of

$$(M_0 \quad M_1)$$

- Then we only need to worry about $3\ n_0$ polynomials, i.e. $H_i, Q_{j,0}, Q_{k,1}$
- Net effect: Increasing n_0 still makes the attack less effective, but not as much as one might naïvely think

Less Weak Keys

- The previous example concerns only the weakest possible keys
- We can use more complicated information set patterns to mount a higher complexity attack on a larger class of somewhat-less-weak keys
  - Generally the support of each block H' may be divided into d'_v nonconsecutive stretches of consecutive coefficients
  - And the support of each block of Q' may be divided into m'_i nonconsecutive stretches of consecutive coefficients
  - We can use one ring representation for H'_i, and $Q'_{i,j}$ and a different ring representation for H'_k, and $Q'_{k,l}$
- For attack parameters around $d'_v = 6; m'_i = (5,5)$, we think we can recover nearly all of the keys for LEDAcrypt (CPA, Category 5, $n_0 = 2$) for something like 2^{248} classical AES operations
  - (Note: Not rigorous and not in paper; aiming for a slight overestimate)

Asymptotics

- For MDPC codes, the complexity of key recovery on a key of size k is exponential in $\tilde{O}(k^{1/2})$
- Assuming H and Q are similarly sparse, our attack runs in $\tilde{O}(k^{1/4})$
- That said, simply enumerating H and Q also runs in $\tilde{O}(k^{1/4})$
  - Considered in submission but concrete complexity was too high to affect parameters

Conclusion

- Our attack shows that LEDAcrypt's product structure is a security problem both asymptotically and concretely
- Attacks to find the weakest class of keys are close to practical for all parameter sets
- The fact that weak key attacks grade smoothly into more expensive attacks on successively larger classes of keys makes security analysis very difficult
  - Except when the product structure is trivial (i.e. Q is an identity matrix)