From: Moody, Dustin (Fed)
To: [redacted]
Subject: Fw: BIKE slides
Date: Friday, February 23, 2018 10:14:17 AM
Attachments: BIKE.pptx

From: Perlner, Ray (Fed)
Sent: Friday, February 23, 2018 10:02 AM
To: Moody, Dustin (Fed)
Subject: BIKE slides

BIKE

(Bit-Flipping Key Exchange)

Presented by Ray Perlner

High Level Summary

- Variants of McEliece/ Neiderreiter based on Quasi-Cyclic MDPC codes
  - Non-algebraic codes like MDPC codes look good for key reduction with quasi cyclic structure
    - (unlike algebraic codes e.g. those used in DAGS and BigQuake)
  - Performance is competitive with lattice-based schemes, but attack complexity seems easier to analyze.
  - Has somewhat high dec. failure rate (< 10^{-7}); targeting IND-CPA.
- Three versions
  - BIKE-1: McEliece KEM: Optimized for speed of KeyGen
  - BIKE-2: Niederreiter KEM: Optimized for PK, ciphertext size.
  - BIKE-3: patented LWE-like “Ouroboros” key exchange.
    - Uses modified “noisy syndrome” decoder.
    - Slightly different security assumption (probably.)

Some Coding Theory

- Generator matrix (Systematic form)
  - $n \times k$

$$G = [I_k \mid C]$$

- Parity Check matrix (Systematic form)
  - $n \times (n-k)$

$$H = [-C^T \mid I_{n-k}]$$

- Definining feature: $HG^T = 0$
- Codewords x may either be defined as
  - n-bit vectors that can be expressed as $x = mG$ for k-bit m
  - Solutions to $Hx^T = 0$

- Syndrome: $s = H(mG + e)^T = H(e^T)$
  - Mapping s to minimal weight e is sometimes easy but NP hard in general.
- McEliece Encryption: $mG + e$ is ciphertext, m is plaintext.
- Niederreiter Encryption: s is ciphertext, e is plaintext.
  - Note: Both "McEliece" and Niederreiter KEMs for BIKE use Hash(e) as shared secret.

MDPC (Moderate Density Parity Check) Codes *(special case where n = 2k)*

- Secret ***sparse*** parity check matrix:

$$H = (\,H_0|H_1)$$

- Public parity check
  - Random Row mixing (BIKE-1): $H_{pub1} = RH = (\,RH_0|RH_1)$
  - Systematic form (BIKE-2): $H_{pub2} = H_1^{-1}H = \ (\,H_1^{-1}H_0\big|I)$

- Public Generator Matrix (Systematic Form)
  - $G_{pub} = (I|(H_1^{-1}H_0)^T)$

- NOTE: $HG_{pub}{}^T = H_{pub1}\,G_{pub}{}^T = H_{pub2}\,G_{pub}{}^T = 0.$
  - So all are the same code.

Decoding MDPC codes (The Bit-Flip Algorithm)

- Want to find low weight e such that $He^T = s$

Algorithm 1 `Bit Flipping Algorithm`

Require: $H \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$
Ensure: $eH^T = s$

$e \leftarrow 0$
$s' \leftarrow s$
while $s' \neq 0$ **do**
  $\tau \leftarrow$ threshold $\in [0,1]$, found according to some predefined rule
  for $j = 0, \ldots, n-1$ **do**
    if $|h_j \star s'| \geq \tau\,|h_j|$ **then**
      $e_j \leftarrow e_j + 1 \mod 2$
  $s' \leftarrow s - eH^T$
return e

h_j denotes the j-th column of H, as a row vector, '$\star$' denotes the component-wise product of vectors, and $|h_j \star s|$ is the number of unchecked parity equations involving j.

Decoding MDPC codes with noisy syndrome (used in BIKE-3)

- Want to find low weight e, e’ such that $He^T + e'^T = s$

Algorithm 2 `Extended Bit Flipping Algorithm`

Require: $H \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$, integer $u \geq 0$

Ensure: $\left|s - eH^T\right| \leq u$

$e \leftarrow 0$

$s' \leftarrow s$

while $|s'| > u$ **do**

    $\tau \leftarrow$ threshold $\in [0, 1]$, found according to some predefined rule // *whatever that means*

    for $j = 0, \ldots, n-1$ **do**

        if $|h_j \star s'| \geq \tau \, |h_j|$ **then**

            $e_j \leftarrow e_j + 1 \mod 2$

    $s' \leftarrow s - eH^T$

return e

Quasi-Cyclic structure

- Use $n = 2k$, where k is prime and $x^k - 1$ is $(x - 1)$ times a primitive polynomial mod 2.
- Represent $k \times k = (n - k) \times (n - k)$ blocks as polynomials in the ring $\mathrm{GF2}[x]/x^k - 1$.
  - Now block multiplication commutes.
  - And blocks only require k bit representation.
  - They look like this:

$$\begin{pmatrix} a & b & c & d & e & f \\ f & a & b & c & d & e \\ e & f & a & b & c & d \\ d & e & f & a & b & c \\ c & d & e & f & a & b \\ b & c & d & e & f & a \end{pmatrix}$$

BIKE 1-3 Summary Table
(Switching to their notation for variable names.)

- m and g are random polynomials in $\mathrm{GF2[x]}/(x^r - 1)$
- e_0 and e_1 are polynomials in the same ring with hamming weights summing to t. e, when present has Hamming weight $t/2$.

Comparison between BIKE versions. For ease of comparison, we provide a summary of the three schemes in Table 2 below.

	BIKE-1	BIKE-2	BIKE-3
SK	(h_0, h_1) with $\lvert h_0\rvert = \lvert h_1\rvert = w/2$		
PK	$(f_0, f_1) \leftarrow (gh_1, gh_0)$	$(f_0, f_1) \leftarrow (1, h_1 h_0^{-1})$	$(f_0, f_1) \leftarrow (h_1 + gh_0, g)$
Enc	$(c_0, c_1) \leftarrow (mf_0 + e_0, mf_1 + e_1)$	$c \leftarrow e_0 + e_1 f_1$	$(c_0, c_1) \leftarrow (e + e_1 f_0, e_0 + e_1 f_1)$
	$K \leftarrow \mathbf{K}(e_0, e_1)$		
Dec	$s \leftarrow c_0 h_0 + c_1 h_1$; $u \leftarrow 0$	$s \leftarrow ch_0$; $u \leftarrow 0$	$s \leftarrow c_0 + c_1 h_0$; $u \leftarrow t/2$
	$(e'_0, e'_1) \leftarrow \mathtt{Decode}(s, h_0, h_1, u)$		
	$K \leftarrow \mathbf{K}(e'_0, e'_1)$		

Table 2: Algorithm Comparison

- If you do out the math $s = e_0 h_0 + e_1 h_1$ (for BIKE-1,2) and $s = e_0 h_0 + e_1 h_1 + e$ for (BIKE-3)

BIKE Parameters

- Polynomials are over ring $\mathrm{GF2[x]}/(x^r - 1)$
- $n = 2r$ is the number of bits in the error vector (e_0, e_1)
- t is the Hamming weight of the error vector.
- w is the row weight of the MDPC code (h_0, h_1)

	BIKE-1 and BIKE-2				BIKE-3			
Security	n	r	w	t	n	r	w	t
Level 1	20,326	10,163	142	134	22,054	11,027	134	154
Level 3	39,706	19,853	206	199	43,366	21,683	198	226
Level 5	65,498	32,749	274	264	72,262	36,131	266	300

Table 3: Suggested Parameters.

Performance

(Note: Jacob's numbers look similar, although consistently larger by a factor of ~2.)

BIKE-1

Quantity	Size	Level 1	Level 3	Level 5
Private key	$w \cdot \lceil \log_2(r) \rceil$	2,130	2,296	4,384
Public key	n	20,326	43,786	65,498
Ciphertext	n	20,326	43,786	65,498

Table 4: Private Key, Public Key and Ciphertext Size in Bits.

Operation	Level 1	Level 3	Level 5
Key Generation	730,025	1,709,921	2,986,647
Encapsulation	689,193	1,850,425	3,023,816
Decapsulation	2,901,203	7,666,855	17,483,906

Table 6: Latency Performance in Number of Cycles.

BIKE-2

Quantity	Size	Level 1	Level 3	Level 5
Private key	$w \cdot \lceil \log_2(r) \rceil$	2,130	3,296	4,384
Public key	r	10,163	21,893	32,749
Ciphertext	r	10,163	21,893	32,749

Table 7: Private Key, Public Key and Ciphertext Size in Bits.

Operation	Level 1	Level 3	Level 5
Key Generation	6,383,408	22,205,901	58,806,046
Encapsulation	281,755	710,970	1,201,161
Decapsulation	2,674,115	7,114,241	16,385,956

Table 9: Latency Performance in Number of Cycles.

BIKE-3

Quantity	Size	Level 1	Level 3	Level 5
Private key	$w \cdot \lceil \log_2(r) \rceil$	2,010	3,168	4,522
Public key	n	22,054	43,366	72,262
Ciphertext	n	22,054	43,366	72,262

Table 10: Private Key, Public Key and Ciphertext Size in Bits.

Operation	Level 1	Level 3	Level 5
Key Generation	433,258	1,100,372	2,300,332
Encapsulation	575,237	1,460,866	3,257,675
Decapsulation	3,437,956	7,732,167	18,047,493

Table 12: Latency Performance in Number of Cycles.

BIKE-2 Batch Key Generation

- Assumes polynomial inversion is more expensive than polynomial multiplication
- Generate polynomials x, y, z …
- Compute $tmp^{-1} = (x \cdot y \cdot z \cdot \cdots)^{-1}$
- To get e.g. x^{-1} compute $x^{-1} = tmp^{-1} \cdot y \cdot z \cdot \cdots$.

Operation	Reference	Batch	Gain (%)
Level 1	6,383,408	1,647,843	74.18%
Level 3	22,205,901	4,590,452	79.32%
Level 5	58,806,046	9,296,144	84.19%

Table 13: Reference Versus Batch Key Generation (in cycles, for $N = 100$).

Known attacks: Information Set Decoding

- Basic idea Guess k-bits of low weight codeword/ error vector and use linear algebra to find the rest.
  - Find error vector:
    - Permute columns of G resulting in $G' = GP = (A|B)$.
    - Hope first k bits of eP are zero.
    - If so, can multiply first k bits of $(mG + e)P$ by A^{-1} to recover m
    - Asymptotic complexity: $\left(\frac{n}{n-k}\right)^{t}$
  - Find MDPC private key:
    - Permute columns of H_{pub} resulting in $H' = H_{pub} = (A|B)$.
    - Hope first k bits of a row of HP are (1, 0, …, 0).
    - If so, the row of HP is the top row of $A^{-1} H'$
    - Asymptotic complexity: $\left(\frac{n}{n-k}\right)^{w}$
- Complications
  - Fancier versions of ISD: Stern's algorithm, MMT, BJMM etc.
    - Same asymptotic complexity as t/n and w/n go to zero. (Note for MDPC: $t \approx w \approx \sqrt{n}$)
  - k target rows in parity check matrix: Improves key recovery complexity to $\frac{1}{k}\left(\frac{n}{n-k}\right)^{w}$.
  - Ring structure plus Decoding One Out of Many (DOOM) improves error finding complexity to $\frac{1}{\sqrt{k}}\left(\frac{n}{n-k}\right)^{t}$.
  - Grover's algorithm gives near full square root speedup

Known attacks: Reaction Attacks

- Guo, Johannson, Stankovsky [GJS 2016] show how to recover private key from statistical analysis of decryption failures.
- This attack does not affect the claimed security of BIKE, since it is recommended for ephemeral-ephemeral use only, and only claims IND-CPA security.

Choice of r

- Polynomials are over ring $\mathrm{GF2[x]}/(x^r - 1)$
- Recall that r is chosen so that $\frac{x^r-1}{x-1}$ is irreducible mod 2.
- Why?
- Possible reasons:
  - It's easy to tell whether a polynomial is invertible (only requires odd hamming weight strictly less than r)
  - Might be worried about folding attacks like [Hauteville, Tillich 2015] on LRPC codes.

Security Proof

- Submission gives an attempted security proof
  - Basic assumptions:
    - QC - MDPC codes in systematic form look random.
    - Syndromes from random QC codes and low weight error vectors look random.
  - Won’t go into detail, but I think there are errors in the proof
    - Claims BIKE-3 and BIKE-1 have same assumptions (I think it BIKE-1 should have same assumptions as BIKE-2).
    - A little less clear about distinction between search and decision than I’d like
    - Since $\mathrm{GF2[x]}/(x^r - 1)$ factors as $\mathrm{GF2[x]}/(x-1) \otimes \mathrm{GF2[x]}/(x^{r-1} + \cdots + 1)$, parity of syndromes/ codes is often predictable. (Pointed out on forum.)
  - Nonetheless, for what it’s worth, I think something like the attempted proof can be correctly stated/ proved.

Similar submissions

- Straight up knock off
  - QC-MDPC-KEM
- Pretty much the same problem
  - HQC (If BIKE is NTRU, this is RingLWE)
- Similar problem; probably harder to analyze
  - LEDApkc/LEDAkem

- Basically the same scheme, but Rank metric
  - LAKE/Locker, Ouroboros-R
- Basically the same scheme, but Euclidean metric
  - NTRUxxx

Advantages and limitations

- Advantages
  - All known IND-CPA attacks are well-understood information set decoding type attacks.
    - ISD has been known for 45 years and improvements have left asymptotic complexity the same.
    - Compares favorably with lattice attacks (stability) and Rank-Metric attacks (newness)
  - Relatively small key sizes (10,000 to 65,000 bits)
  - Reasonably fast for all operations.
    - Except for BIKE2 keygen without batching, operations look like they take less than a millisecond on a good processor for 128 bit security.
- Limitations
  - High Decryption failure rate
  - Does not provide IND-CCA security
  - Security proof could use improvement/clarification
  - Key/Message sizes are slightly larger than some (ring/ cyclic) lattice and rank schemes.
  - Vague possibility there might be something to exploit in ring structure.