Block-Anti-Circulant Unbalanced Oil and Vinegar

Alan Szepieniec[1,2] and Bart Preneel[1]
alan@nervos.org

[1]KU Leuven and [2]Nervos Foundation

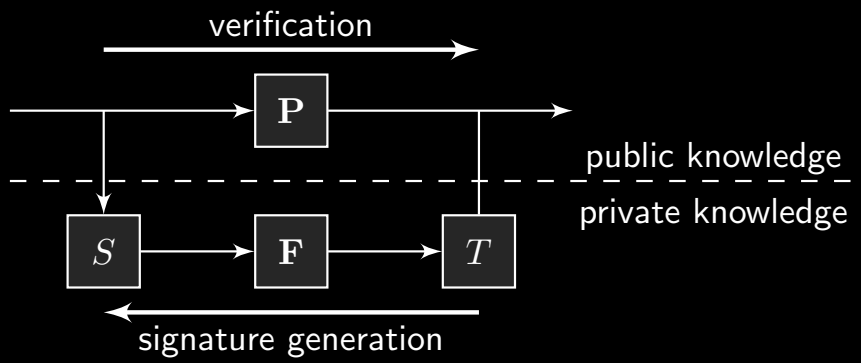
2019-08-15

- Unbalanced Oil and Vinegar
  - Construction
  - Petzoldt's Compression Trick
  - Field Lifting
- Block-Anti-Circulant
  - Definition
  - Block-Anti-Circulant UOV
  - Implementation Aspects
- Security
  - Direct Attack
  - Kipnis-Shamir Attack
  - UOV Reconciliation Attack
  - Subfield Differential Attack
- Performance
  - Sizes
  - Speed
  - Implementation

Unbalanced Oil and Vinegar

Unbalanced Oil and Vinegar

- post-quantum signature scheme
- ~~1997~~ 1999
- multivariate quadratic



Verify($\mathbf{P}$, d, $\mathbf{s}$):
$\mathbf{P}(\mathbf{s}) \stackrel{?}{=} \mathsf{H}(d)$

- NIST PQC: Rainbow and LUOV

public keys	signatures	
humongous	tiny	basic scheme
huge	tiny	Petzoldt's trick
acceptable	acceptable	field lifting (LUOV)
slightly smaller still	slightly smaller still	⟵ this work

UOV: Construction

$$n = o + v \quad \text{and} \quad m = o$$

- Key Generation:
  - sample $S \in \mathbb{F}_q^{n \times n}$ and $F^{(i)} \in \mathbb{F}_q^{n \times n}$ for $i = 0, \ldots, m-1$

$$F^{(i)} = \left(\right) \qquad S = \left(\right)$$

  - compute $pk = (P^{(i)})_{i=0}^{m-1}$ with $P^{(i)} = S^{\mathsf{T}} F^{(i)} S$

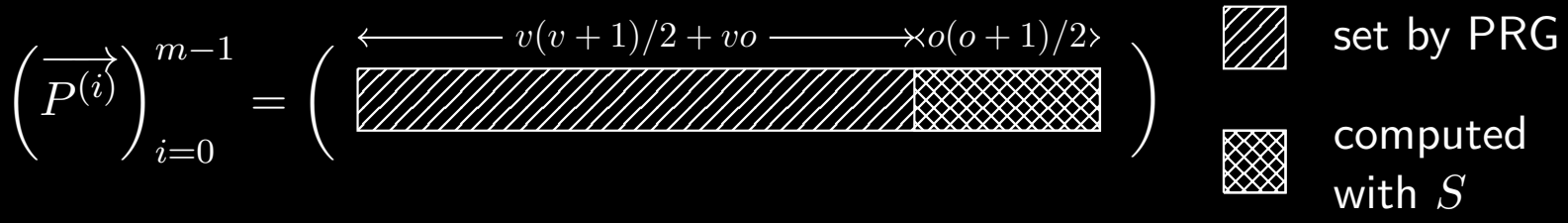
- Verify Signature $\mathbf{s} \in \mathbb{F}_q^n$ on document $d \in \{0,1\}^*$:
  - evaluate and test:

$$\left(\mathbf{s}^{\mathsf{T}} P^{(i)} \mathbf{s} \right)_{i=0}^{m-1} \stackrel{?}{=} \mathsf{H}(d)$$

UOV: Petzoldt's Compression Technique

- observation:
  - action of S on quadratic polynomials is *linear*
  - $F^{(i)}$ live in dimension $v(v+1)/2+ov$ subspace of $\mathbb{F}_q^{n\times n}$
  - therefore, $P^{(i)}$ must live in subspace of the same dimension
- idea:
  - generate first $v(v+1)/2+ov$ coefficients from PRG
  - compute remaining $o(o+1)/2$ coefficients such that $\left(S, \left(F^{(i)}\right)_{i=0}^{m-1}\right)$ is a valid secret key
  - repeat for each polynomial
- shrinks public key from $mn(n+1)/2$ field elements to $mo(o+1)/2$ field elements plus seed

- increasing v is (almost) free



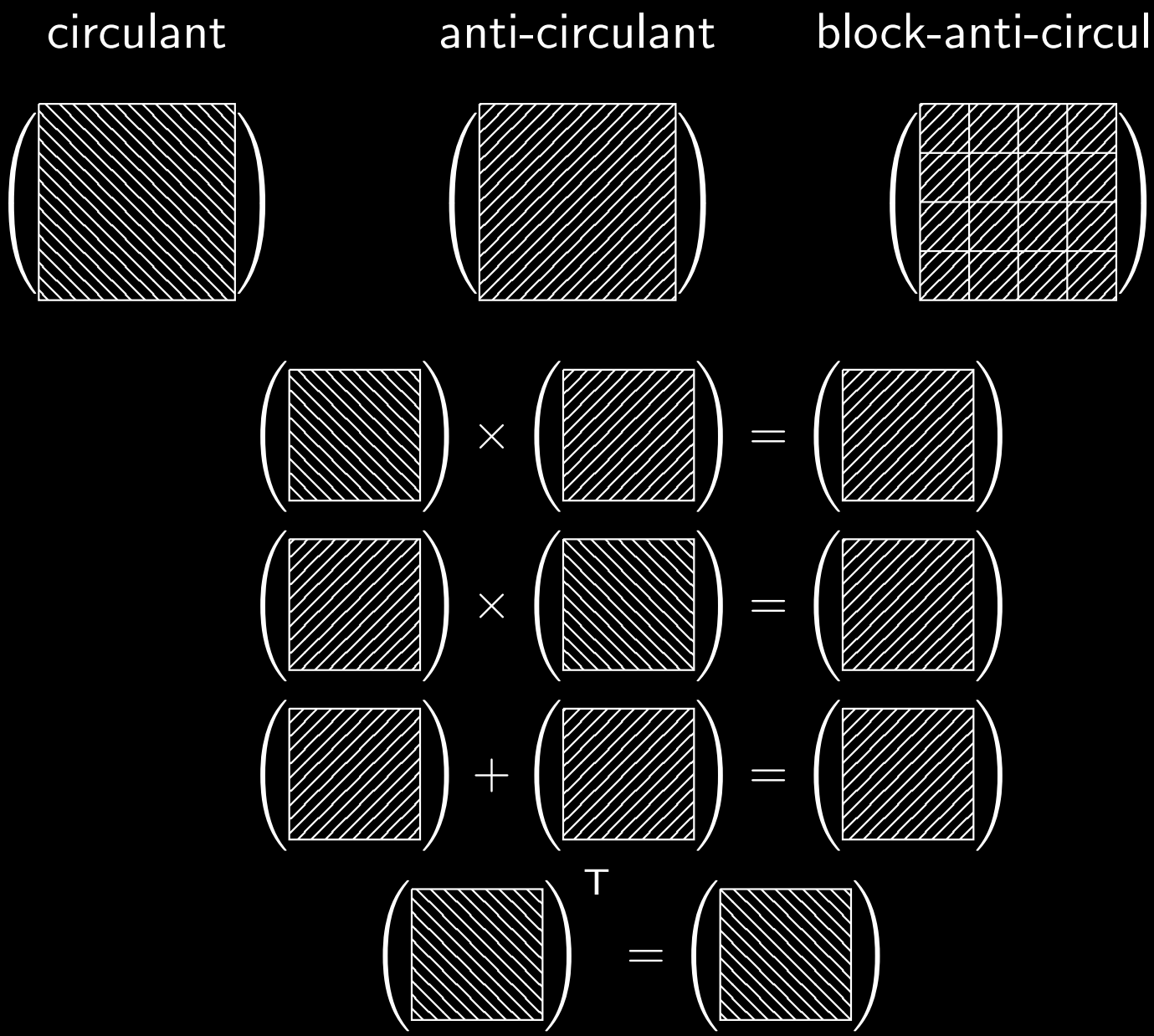
Lifted UOV

- observation:
  - complexity of direct attack increases with q
  - but: larger q means larger public key
  - ... or does it?
- idea:
  - define public key over small base field ($\mathbb{F}_2$)
    $\Longrightarrow$ small public key
  - compute signatures over large extension field ($\mathbb{F}_{2^r}$)
    $\Longrightarrow$ at the expense of larger signatures

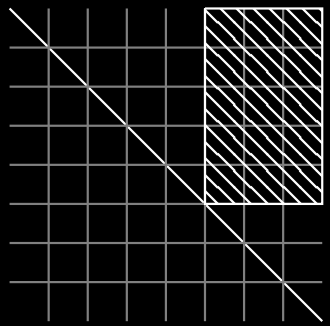
$\longrightarrow$ LUOV NIST submission

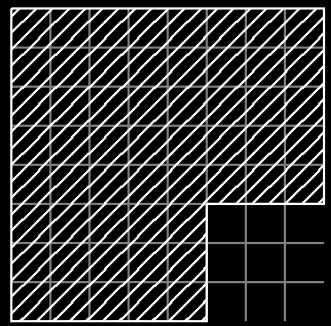
Block-Anti-Circulant Matrices

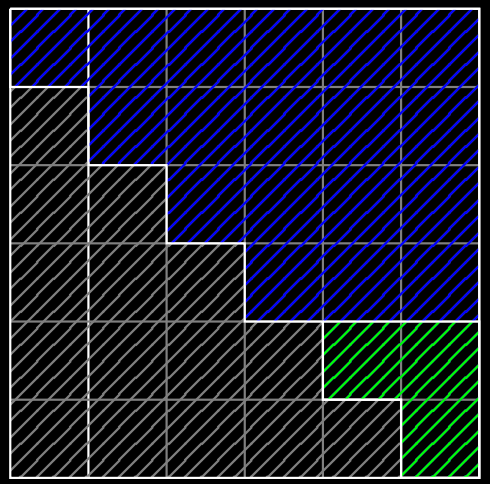
Circulant and Anti-Circulant



Block-Anti-Circulant UOV

$$S = \left(\right.$$  $$\left. \right)$$

$$F^{(i)} = \left(\right.$$  $$\left. \right)$$

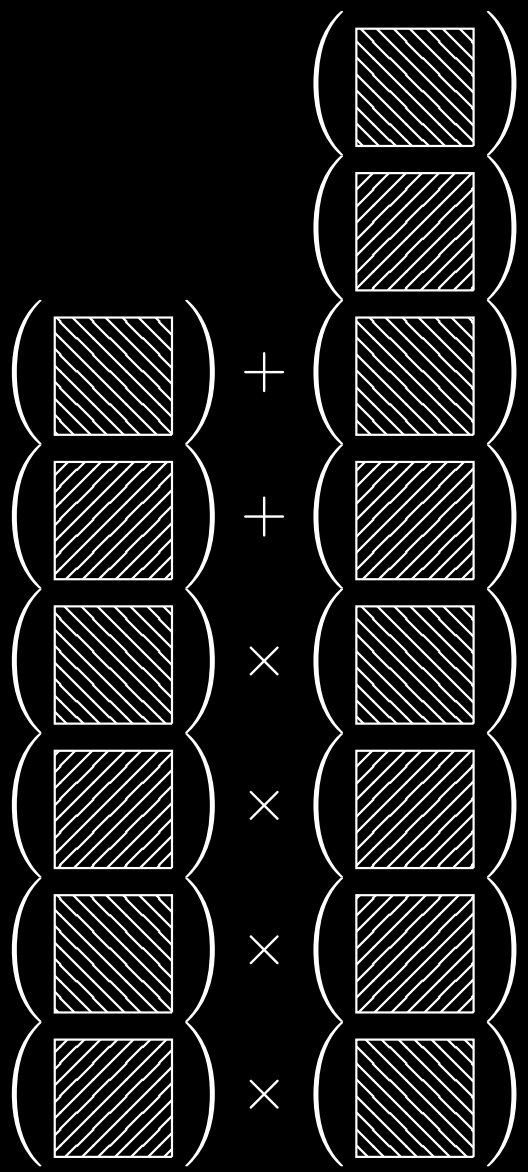
$$P^{(i)} = \left(\right.$$  $$\left. \right)$$

 first row set by PRG

 first row computed using S

 inferred from symmetry

Implementing Block-Anti-Circulant Matrices (1)



$\leftrightarrow \quad \mathbb{F}_q[x]/\langle x^\ell - 1 \rangle$

$\leftrightarrow \quad \mathbb{F}_q[x]/\langle x^\ell - 1 \rangle$???

$\leftrightarrow$ element-wise addition

$\leftrightarrow$ element-wise addition

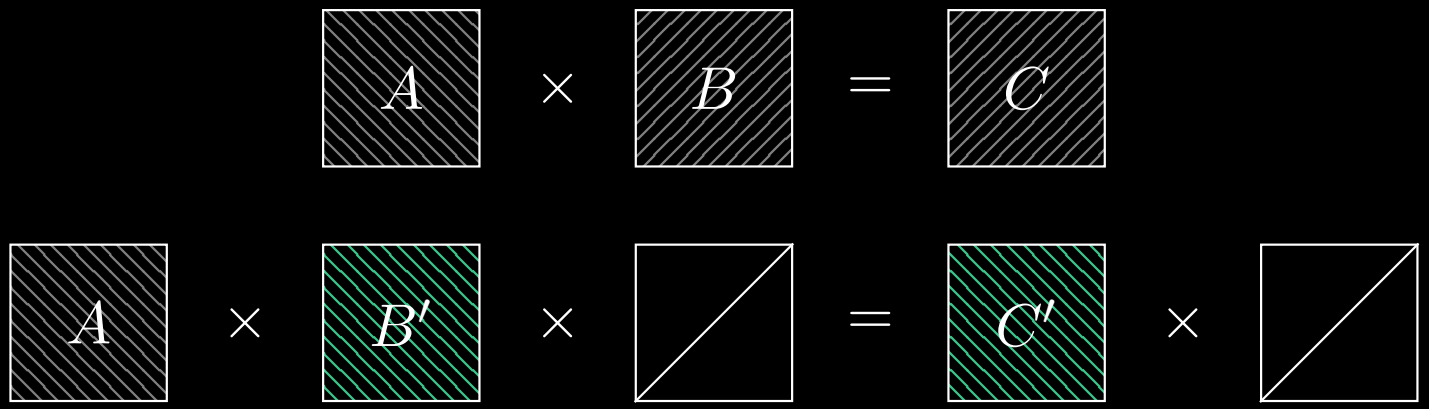
$\leftrightarrow \quad A(x) \times B(x) \bmod x^\ell - 1$

$\leftrightarrow$ never happens

$\leftrightarrow$???

$\leftrightarrow$???

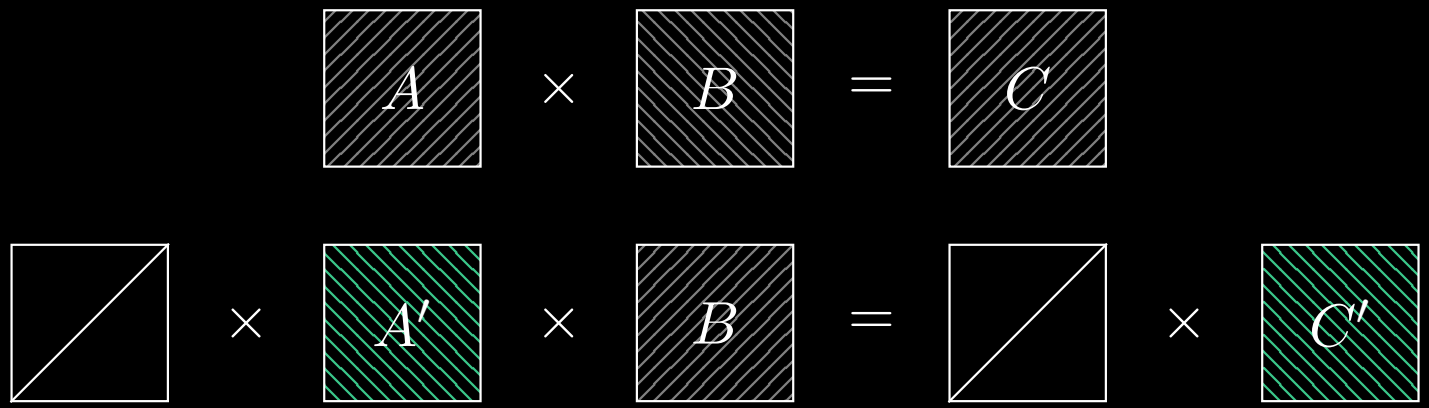
Circulant Times Anti-Circulant




```
A * flip(B) = flip(C)
```

$$A(x) \times x^{\ell} B(x^{-1}) = x^{\ell} C(x^{-1}) \bmod x^{\ell} - 1$$

Anti-Circulant Times Circulant



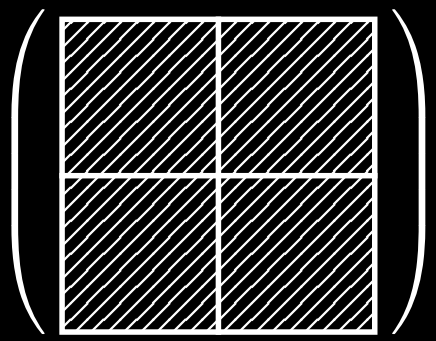

```
shl1(A) * B = shl1(C)
```

$$x^{-1}A(x) \times B(x) = x^{-1}C(x) \bmod x^{\ell} - 1$$

$$A(x) \times B(x) = C(x) \bmod x^{\ell} - 1$$

only because the matrix is identified with its first row!

Implementing Block-Anti-Circulant Matrices (2)

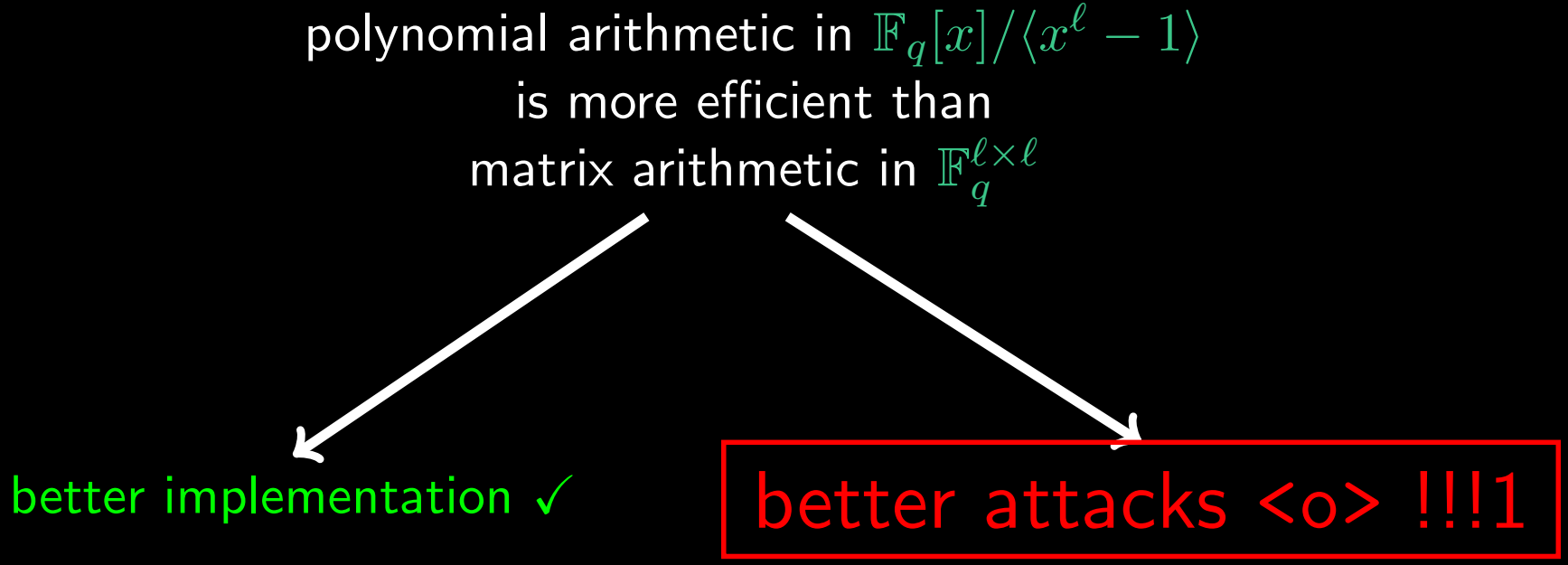


$\leftrightarrow \quad \texttt{matrix} <\mathbb{F}_q[x]/\langle x^\ell - 1\rangle>$

polynomial arithmetic in $\mathbb{F}_q[x]/\langle x^\ell - 1\rangle$
is more efficient than
matrix arithmetic in $\mathbb{F}_q^{\ell\times\ell}$

Security

Structured Matrices



Conservative Security Assumption

circulant matrices $\leftrightarrow \mathbb{F}_q[x]/\langle x^\ell - 1\rangle = \frac{\mathbb{F}_q[x]}{\langle f_0(x)\rangle} \oplus \frac{\mathbb{F}_q[x]}{\langle f_1(x)\rangle} \oplus \cdots$

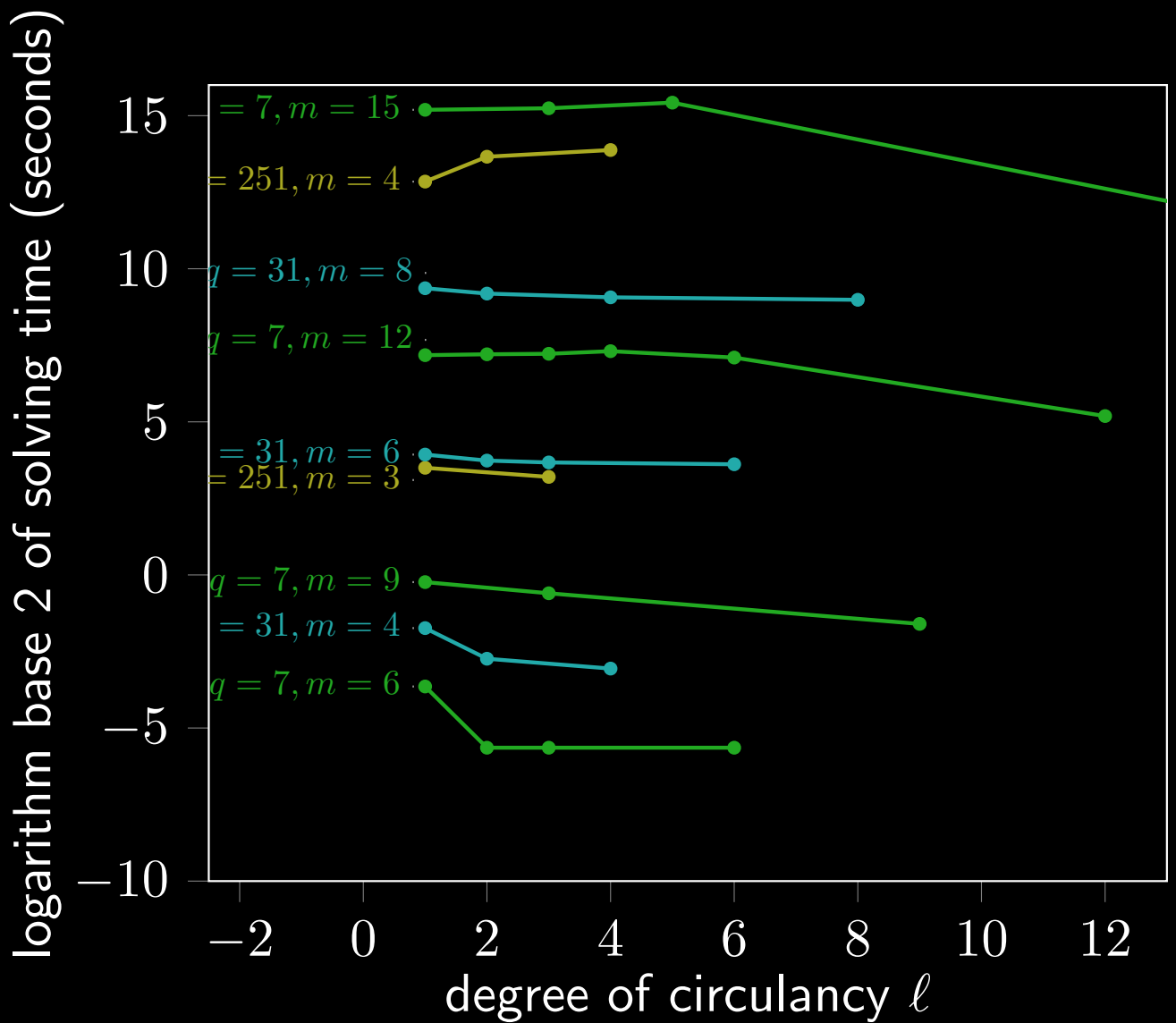
- conservative security estimate:
  - treat every block as one variable over $\mathbb{F}_q[x]/\langle f_j(x)\rangle$
    - where $f_j(x)$ is the largest-degree irreducible factor of $x^\ell - 1$
  - o oil variables $\longrightarrow o/\ell = O$ oil variables
  - v vinegar variables $\longrightarrow v/\ell = V$ vinegar variables
  - $N = n/\ell = O + V$ in total
  - m equations remains m equations

Direct Attack

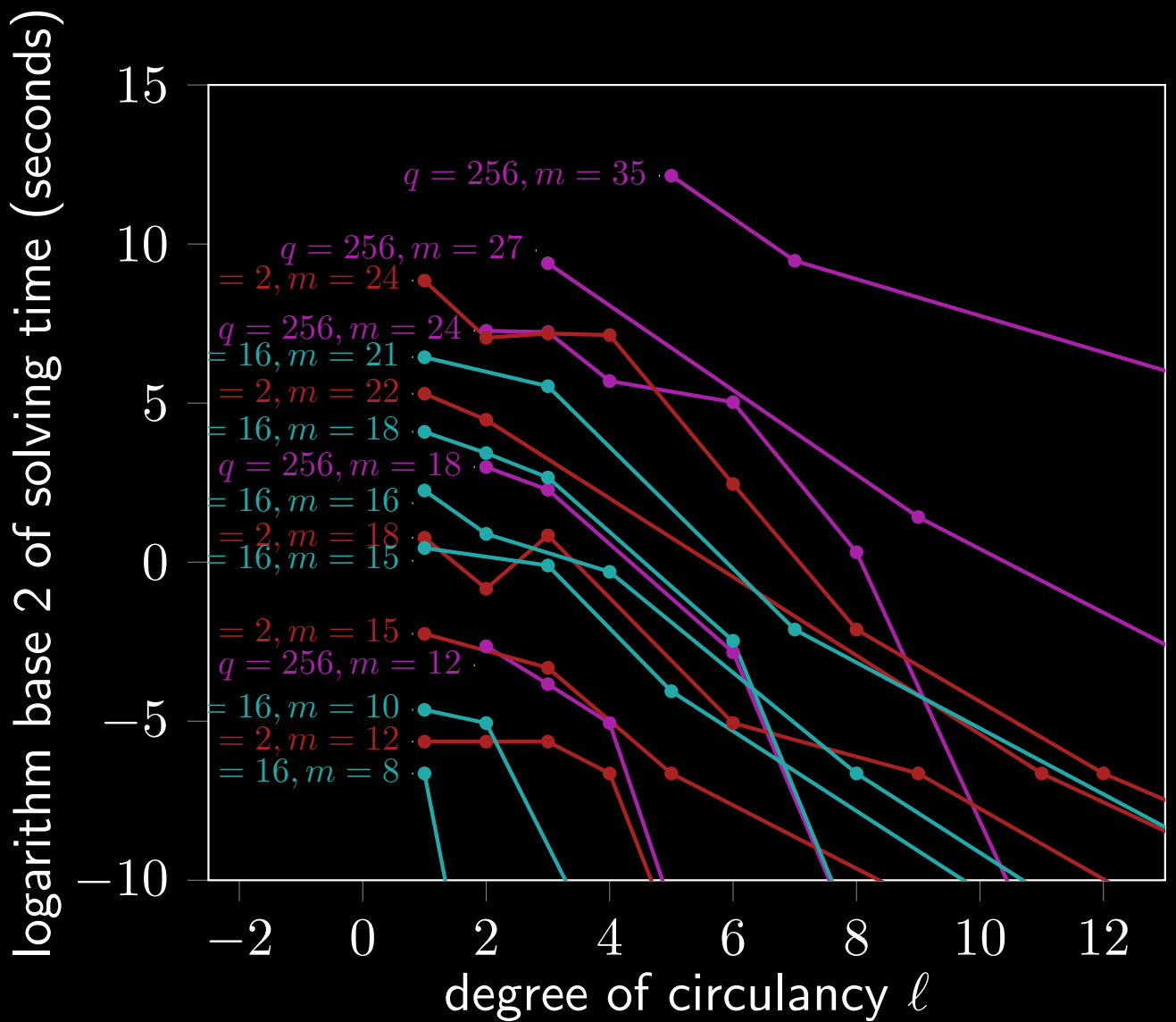
- solve $\mathbf{P}(\mathbf{x}) = \mathsf{H}(d)$ for $\mathbf{x}$ algebraically

- still m equations ... and $N = n/\ell > m$ ...
- $\Longrightarrow$ no speed-up expected

Direct Attack: Odd Characteristic



Direct Attack: Characteristic Two



Kipnis-Shamir

Standard UOV

- $P_i P_j^{-1} =$

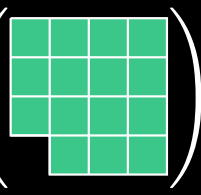
$$S^{\mathsf{T}} \left(\right) S^{\mathsf{T}-1}$$

- complexity $O(q^{v-o})$

Block-Anti-Circulant UOV

- $P_i P_j^{-1} =$

$$S^{\mathsf{T}} \left(\right) S^{\mathsf{T}-1}$$



- complexity $O(q^{V-O})$

- Petzoldt's compression trick: increasing v is (almost) free

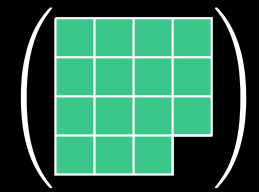
UOV Reconciliation

Standard UOV

- Solve algebraically for S^{-1}:
  $\forall i \,.\, {S^{-1}}^{\mathsf{T}} P_i S^{-1} =$
  
- m equations in v variables
- with unique solution

  $\longrightarrow$ complexity of system solving for v equations and v variables

Block-Anti-Circulant UOV

- Solve algebraically for S^{-1}:
  $\forall i \,.\, {S^{-1}}^{\mathsf{T}} P_i S^{-1} =$
  
- m equations in V variables
- with unique solution

  $\longrightarrow$ complexity of system solving for V equations and V variables

- Petzoldt's compression trick: increasing v is (almost) free

Subfield Differential Attack

- important new attack on field lifting
  - proposed parameters need updating
- field lifting:
  - public key: $\mathbf{P}(\mathbf{x}) \in (\mathbb{F}_2[\mathbf{x}])^m$
  - signature equation: $\mathbf{P}(\mathbf{s}) = \mathsf{H}(d) \in {\mathbb{F}_{2^r}}^m$ for some $\mathbf{s} \in {\mathbb{F}_{2^r}}^n$

- differential subfield attack:
  - select intermediate field $\mathbb{F}_{2^d}$ such that $\mathbb{F}_2 \subset \mathbb{F}_{2^d} \subset \mathbb{F}_{2^r}$
  - interpret $\mathbb{F}_{2^r} \cong \mathbb{F}_{2^d}[x]/\langle f(x) \rangle$ with $\deg(f(x)) = s = r/d$
  - choose $\mathbf{z} \xleftarrow{\$} (\mathbb{F}_{2^d}[x]/\langle f(x) \rangle)^n$
  - with probability $\mathrm{P} \approx 1 - e^{-2^{dn-rm}}$ there is a $\boldsymbol{\delta} \in {\mathbb{F}_{2^d}}^n$ such that $\mathbf{P}(\mathbf{z} + \boldsymbol{\delta}) = \mathsf{H}(d)$
  - so solve $\mathbf{P}(\mathbf{z} + \boldsymbol{\delta}) = \mathsf{H}(d)$ for $\boldsymbol{\delta} \in {\mathbb{F}_{2^d}}^n$
    - n variables, $m \times s$ equations ...
    - ... of which $m \times (s-1)$ are linear $\Leftarrow \mathbf{P}(\mathbf{x}) \in (\mathbb{F}_2[\mathbf{x}])^m$
    - after elimination: $n + m - ms$ variables, m equations
- simple fix: choose r prime

Performance

Public Key and Signature Sizes

scheme	parameters	$\|pk\|$	$\|sig\|$	sec. lvl.
Plain	$q = 256, v = 106, m = o = 53$	658.36 kB	159 bytes	128.85
PCT	$q = 256, v = 106, m = o = 53$	74.07 kB	159 bytes	128.85
LUOV	$q = 2, v = 296, m = o = 40, r = 68$	4.00 kB	2.79 kB	128.17
bacuov	$q = 3, V = 49, O = 7, \ell = 7, r = 12$	2.34 kB	1.14 kB	129.32
Plain	$q = 256, v = 164, m = o = 82$	2.38 MB	246 bytes	191.89
PCT	$q = 256, v = 164, m = o = 82$	272.5 kB	246 bytes	191.89
LUOV	$q = 2, v = 444, m = o = 60, r = 84$	13.40 kB	5.16 kB	190.00
bacuov	$q = 3, V = 76, O = 10, \ell = 7, r = 18$	6.58 kB	2.65 kB	192.08
Plain	$q = 256, v = 224, m = o = 112$	6.05 MB	336 bytes	256.50
PCT	$q = 256, v = 224, m = o = 112$	692.13 kB	336 bytes	256.50
LUOV	$q = 2, v = 600, m = o = 82, r = 90$	34.06 kB	7.49 kB	256.13
bacuov	$q = 3, V = 104, O = 14, \ell = 7, r = 11$	17.59 kB	2.22 kB	256.68

Speed

scheme	parameters	sec. lvl.	Keygen	Sign	Verify
LUOV	$q = 2$, $r = 48$ $m = 43$, $v = 222$	NIST II	14 795 kc 6.165 ms	54 831 kc 22.846 ms	35 748 kc 14.895 ms
bacuov	$q = 3$, $V = 56$ $O = 8$, $\ell = 7$, $r = 12$	NIST II	682 436 kc 284.323 ms	332 498 kc 138.527 ms	593 693 kc 247.355 ms
bacuov	$q = 7$, $V = 60$ $O = 5$, $\ell = 13$, $r = 5$	NIST II	2 685 130 kc 1118.648 ms	611 928 kc 254.956 ms	1 265 715 kc 527.364 ms
LUOV	$q = 2$, $r = 64$ $m = 61$, $v = 302$	NIST IV	40 039 kc 16.683 ms	163 638 kc 68.182 ms	90 331 kc 37.638 ms
bacuov	$q = 3$, $V = 84$ $O = 11$, $\ell = 7$, $r = 16$	NIST IV	2 354 156 kc 980.803 ms	1 402 365 kc 584.275 ms	2 452 899 kc 1021.961 ms
bacuov	$q = 7$, $V = 76$ $O = 7$, $\ell = 11$, $r = 16$	NIST IV	4 801 991 kc 2000.665 ms	2 260 414 kc 941.775 ms	3 740 856 kc 1558.526 ms
LUOV	$q = 2$, $r = 80$ $m = 76$, $v = 363$	NIST V	176 100 kc 73.374 ms	527 341 kc 219.723 ms	248 874 kc 103.696 ms
bacuov	$q = 3$, $V = 104$ $O = 14$, $\ell = 7$, $r = 16$	NIST V	5 096 796 kc 2123.658 ms	2 804 193 kc 1168.403 ms	4 758 999 kc 1982.899 ms
bacuov	$q = 7$, $V = 97$ $O = 9$, $\ell = 11$, $r = 8$	NIST V	12 019 482 kc 5007.974 ms	2 399 406 kc 999.722 ms	5 401 021 kc 2250.364 ms

C Implementation

- https://www.github.com/aszepieniec/bacuov/
- ```
  gcc -o bench *.c -O3 -DGF_PRIME_MODULUS=7
  -DGFP_NUMBITS=3 -DEXTENSION_DEGREE=5
  -DDEFINING_POLYNOMIAL="\x03\x01\x00\x00\x00"
  -DDEGREE_OF_CIRCULANCY=13 -DBACUOV_PARAM_O=5
  -DBACUOV_PARAM_V=60
  ```
- lots of optimizations possible
  ```

Thank you!

alan@nervos.org

https://nervos.org/

https://github.com/nervosnetwork/