Efficient Implementations of Rainbow and UOV using AVX2

1Kyung-Ah Shim, 1Sangyub Lee, and 2Namhun Koo

1National Institute for Mathematical Sciences, Republic of Korea
2Institute for Mathematical Sciences, Ewha Womans University, Republic of Korea

CHES 2022
19 September, 2022
MQ-PKC (preliminaries)
Implementations of UOV and Rainbow
Our Efficient Implementations of UOV and Rainbow
  • Block Matrix Inversion
  • Precomputation
Conclusion
**Key Generation**

- \( P = S \circ F \circ T \): public key
  1. \( F \): easily invertible quadratic map
  2. \( S, T \): invertible affine(or linear) maps
- \((F, S, T)\): secret key

**Figure:** Structure of MQ Signature Scheme
MQ-PKC (2/5)

Signature Generation or Decryption

For given message $M$ and hash value $h = H(M)$, compute $s$ with $h = P(s)$ so that $s = (T^{-1} \circ F^{-1} \circ S^{-1})(h)$. Then $s$ is a signature for message $M$.

Signature Verification or Encryption

Compute $P(s) = h'$. If $h = h'$ accept it. If not, reject it.

Figure: Structure of MQ Signature Scheme
How to construct easily invertible $\mathcal{F}$ : Single Field Type - OV map

- **Notations**
  - $x_1, \ldots, x_v$ : Vinegar variables ($v$ : the number of Vinegar variables)
  - $x_{v+1}, \ldots, x_{v+o}$ : Oil variables ($o$ : the number of Oil variables, the number of equations in $\mathcal{F}$)
  - $n = v + o$ : the number of variables in $\mathcal{F}$

- **Each component function** $\mathcal{F}^{(k)}$ of $\mathcal{F} = (\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots, \mathcal{F}^{(o)})$ is of the form

$$
\mathcal{F}^{(k)}(x_1, \ldots, x_n) = \sum_{i=1}^{v} \sum_{j=i}^{v} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{v} \sum_{j=v+1}^{n} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \gamma^{(k)}
$$

where $\alpha_{ij}^{(k)}, \beta_i^{(k)}, \gamma^{(k)} \in \mathbb{F}_q$. 
How to construct easily invertible $\mathcal{F}$: Single Field Type - OV map

- **How to invert $\mathcal{F}$**

  - Choose Vinegar variables at random and plug them into each $\mathcal{F}^{(k)}$

$$
\sum_{i=1}^{v} \sum_{j=i}^{v} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{v} \sum_{j=v+1}^{n} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{v} \beta_{i}^{(k)} x_i + \sum_{i=v+1}^{n} \beta_{i}^{(k)} x_i + \gamma^{(k)}
$$

Then the red parts are converted to constants in above equation.

- since there are no quadratic terms with oil variables in each $\mathcal{F}^{(k)}$ with $1 \leq k \leq o$, we obtain a linear equation with Oil variables $x_{v+1}, \cdots, x_{n}$ for each $\mathcal{F}^{(k)}$.

- In other words, we obtain a linear equation system with $o$ equations with $o$ variables which can be easily solvable.

- If this linear equation system is not solvable, choose Vinegar variables and try again.

- Note that the obtained linear equation system would be solvable with very high probability.

---

MQ signature schemes using OV map

- UOV (Unbalanced Oil and Vinegar) - "unbalanced’ means $v > o$.
- Rainbow - multi-layered UOV : a Finalist of NIST PQC standardization
- There are many other variants of UOV and Rainbow

An important remark about Rainbow

- Recently Beullens proposed a simple attack on Rainbow[Beu22] : an equivalent key of Rainbow with security level 1 can be recovered in 53 hours by a laptop.
- Because of this attack, Rainbow team announced that they replace the Rainbow level 1 parameters with their level 3 parameters and level 3 with level 5 parameters.

Our parameter selection

- **UOV**
  - Recently, Beullens proposed the intersection attack [Beu21] which is suitable when $v < 2^o$. As a result, UOV parameters frequently used in the past, for example $(o, v) = (44, 59)$ for Security Level I, are vulnerable under the intersection attack.
  - So we suggested new UOV parameters considering the complexity of the intersection attack (see a table in next slide).
  - Very recently, the Rainbow team (collaborated with Beullens) suggested new UOV parameters after the simple attack had been proposed. Note that their parameters are slightly different from our parameters.

- **Rainbow**
  - Our implementation of Rainbow is based on source codes of Rainbow team which are submitted for NIST PQC Standardization Round 3. So our parameters in our paper are same with NIST PQC Round 3.
  - In this presentation, we apply parameter changes of Rainbow due to the attacks proposed in [Beu22].

## Implementations of UOV and Rainbow (2/4)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Security Category</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>λ (Gates)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UOV</td>
<td>(o, v)</td>
<td>(46, 70)</td>
<td>(72, 109)</td>
<td>(96, 144)</td>
<td></td>
</tr>
<tr>
<td>Direct Attack</td>
<td></td>
<td>144.05</td>
<td>212.05</td>
<td>274.847</td>
<td></td>
</tr>
<tr>
<td>Intersection Attack</td>
<td></td>
<td>166.87</td>
<td>236.36</td>
<td>291.501</td>
<td></td>
</tr>
<tr>
<td>Rainbow</td>
<td>λ (Gates)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(v, o₁, o₂)</td>
<td>(68, 32, 48)</td>
<td>(96, 36, 64)</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Direct Attack</td>
<td></td>
<td>234</td>
<td>285</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Intersection Attack</td>
<td></td>
<td>177</td>
<td>226</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Suggested Parameters of UOV/Rainbow at Three SCs.

Note that we select $q = 256$ for all our parameters above.
Detail of our implementations

- Similarly with Rainbow, we apply $T = \begin{pmatrix} I & T' \\ 0 & I \end{pmatrix}$ for UOV.

- Intel(R) Core(TM) i9-10900X CPU running at the constant clock frequency of 3.70GHz.

- Each result is an average of 10,000 measurements for each function using the C programming language with GNU GCC version 10.1.0 compiler on Centos 7.9.2009. Hyperthreading and Turbo Boost are switched off.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>SC</th>
<th>I</th>
<th>III</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOV</td>
<td>KeyGen.</td>
<td>29,077,126</td>
<td>98,870,925</td>
<td>161,016,435</td>
</tr>
<tr>
<td></td>
<td>Sign</td>
<td>201,834</td>
<td>707,959</td>
<td>1,486,775</td>
</tr>
<tr>
<td></td>
<td>Verify</td>
<td>125,312</td>
<td>222,012</td>
<td>485,344</td>
</tr>
<tr>
<td>Rainbow</td>
<td>KeyGen.</td>
<td>65,099,975</td>
<td>214,977,689</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Sign</td>
<td>322,799</td>
<td>807,309</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Verify</td>
<td>151,466</td>
<td>395,259</td>
<td>—</td>
</tr>
</tbody>
</table>
Implementations of UOV and Rainbow (4/4)

Detail of our Implementations on Signing process

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Layer</th>
<th>Operations</th>
<th>I</th>
<th>III</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOV</td>
<td>1</td>
<td>Vinegar Value Substitutions</td>
<td>58.09 %</td>
<td>48.37 %</td>
<td>56.60 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Computation of $LS_V^{-1}$</td>
<td>36.38 %</td>
<td>47.98 %</td>
<td>41.71 %</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Etc.</td>
<td>5.53 %</td>
<td>3.65 %</td>
<td>1.69 %</td>
</tr>
<tr>
<td>Rainbow</td>
<td>1</td>
<td>Vinegar Value Substitutions</td>
<td>18.58 %</td>
<td>34.37 %</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Computation of $LS_{V,1}^{-1}$</td>
<td>11.37 %</td>
<td>8.03 %</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Etc.</td>
<td>1.26 %</td>
<td>0.68 %</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Vinegar Value Substitutions</td>
<td>39.54 %</td>
<td>29.07 %</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Computation of $LS_{V,2}^{-1}$</td>
<td>25.38 %</td>
<td>25.34 %</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Etc.</td>
<td>3.88 %</td>
<td>2.51 %</td>
<td>—</td>
</tr>
</tbody>
</table>

Run-time of UOV/Rainbow signing is mostly dominated by the two operations.

- Substitution of Vinegar Values into the Central Polynomials
- Solving Obtained Linear System
Efficient Implementations of UOV and Rainbow

The most dominated part of signing of UOV and Rainbow

- **Substitution of Vinegar Values into the Central Polynomials** - computing the coefficient matrix \( LS_V \) and constant term of the linear system which we will obtain.

- **Solving Obtained Linear System** - we require to compute the inverse matrix \( LS_V^{-1} \) of the coefficient matrix obtained above.

Key idea of our efficient implementations of UOV and Rainbow

- **Block Matrix Inversion** - we replace an inversion of \( m \times m \) matrix into two inversions of \( m/2 \times m/2 \) matrices when \( m \) is even.

- **Precomputation** - the above two parts can be precomputed, and then signing process can be significantly improved.
Theorem 1

Let $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ be a matrix partitioned into $2 \times 2$ blocks.

(i) Assume $A$ is nonsingular. Then the matrix $R$ is invertible if and only if the Schur complement $(D - CA^{-1}B)$ of $A$ is invertible and

$$R^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{pmatrix}.$$ 

(ii) Assume $D$ is nonsingular. Then the matrix $R$ is invertible if and only if the Schur complement $(A - BD^{-1}C)$ is invertible and

$$R^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{pmatrix}.$$ 

⇒ It requires two inversions and six matrix multiplications of the half-sized matrices.
Theorem 2

For a nonsingular $k \times k$ matrix $R$ in the above, $R^{-1} \cdot \alpha$ requires two inversions, two matrix multiplications of the half-sized block matrices and four block matrix-vector products, where $k$ is even and $\alpha = (\alpha_1, \cdots, \alpha_{k/2})^T$.

**Sketch of Proof.** A nonsingular square matrix $R$ of $2 \times 2$ blocks is represented by the LDU decomposition of block matrices based on the Schur complement as

\[
R = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & O \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & O \\ 0 & D - CA^{-1}B \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix} = L \cdot D_{Sc} \cdot U.
\]

So, we have $R^{-1} = U^{-1} \cdot D_{Sc}^{-1} \cdot L^{-1}$. 
Repeated BMI

- An inversion of $m \times m$ matrix can be replaced by 2 inversions of $m/2 \times m/2$ matrices. In a similar manner, each of 2 inversions of $m/2 \times m/2$ matrices can be replaced by 2 inversions of $m/4 \times m/4$ matrices if $m$ is a multiple of 4.

- Like this, for $k = 2^l \cdot k'$, we can apply the BMI $l$ times. We define the number of these iterations of the BMI as a depth. We cannot expect that $l$ iterations will always be effective, because $2^l$ inversions of $k/2^l \times k/2^l$ matrices are required.
<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>GE</th>
<th>BMI (Depth 1)</th>
<th>BMI (Depth 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>94,033</td>
<td>72,302</td>
<td>—</td>
</tr>
<tr>
<td>48</td>
<td>101,498</td>
<td>75,136</td>
<td>72,479</td>
</tr>
<tr>
<td>50</td>
<td>142,243</td>
<td>82,768</td>
<td>—</td>
</tr>
<tr>
<td>56</td>
<td>175,195</td>
<td>103,357</td>
<td>99,445</td>
</tr>
<tr>
<td>64</td>
<td>225,081</td>
<td>87,150</td>
<td>70,091</td>
</tr>
<tr>
<td>68</td>
<td>322,315</td>
<td>187,947</td>
<td>170,788</td>
</tr>
<tr>
<td>72</td>
<td>355,173</td>
<td>208,355</td>
<td>190,480</td>
</tr>
<tr>
<td>96</td>
<td>713,462</td>
<td>252,538</td>
<td>226,627</td>
</tr>
<tr>
<td>100</td>
<td>923,489</td>
<td>453,441</td>
<td>391,747</td>
</tr>
</tbody>
</table>

Table: Gaussian Elimination (GE) vs. Block Matrix Inversion Technique in CPU Cycles.

- The larger the size, the greater the performance improvement.
- Especially excellent improvements in the case of 64 and 96 are due to the fact that the multiples of 32 are optimal parameters which are suitable for the AVX2 vectorization.
Appling BMI to UOV and Rainbow Signing

After obtaining \( LS_V \) from the Vinegar value substitution, we set

\[
LS_V = \begin{pmatrix} A & B \\ C & D \end{pmatrix}
\]

and apply the BMI on \( LS_V \).

If \( A \) or \([ D - CA^{-1}B ]\) is not invertible then we choose another Vinegar values.

Note that the probability that the matrices are invertible is

\[
\left( 1 - \frac{1}{q} \right)^2 \approx 99.22\%.
\]

Details of our proposed algorithm applying BMI on the signing of Rainbow and UOV are given in Algorithm 7 and 8 in our paper, respectively.
The below table describes our implementation results on our proposed BMI method in CPU Cycles.

Compared to UOV implemented with Gaussian elimination, by using the BMI with the depth 1, we obtain speedups of 12.36%, 20.41%, and 32.42% at the three security categories, respectively.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>SC</th>
<th>I</th>
<th>III</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOV</td>
<td>G.E.</td>
<td>201,834</td>
<td>707,959</td>
<td>1,486,775</td>
</tr>
<tr>
<td></td>
<td>BMI (Depth 1)</td>
<td>176,884</td>
<td>563,519</td>
<td>1,004,704</td>
</tr>
<tr>
<td></td>
<td>BMI (Depth 2)</td>
<td>—</td>
<td>535,660</td>
<td>981,351</td>
</tr>
<tr>
<td>Rainbow</td>
<td>G.E.</td>
<td>322,799</td>
<td>807,309</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>BMI (Depth 1)</td>
<td>270,731</td>
<td>650,400</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>BMI (Depth 2)</td>
<td>271,986</td>
<td>639,965</td>
<td>—</td>
</tr>
</tbody>
</table>

Table: Implementation Results of BMI on the Intel at Three SCs, in CPU cycles.
Precomputation (1/3)

The general idea of using an offline/online phase was first introduced by Even, Goldreich, and Micali [EGM90].

Offline/Online Signing of UOV.

- **Offline phase**
  - After choosing random Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$, substitute $s_V$ into $o$ equations $\mathcal{F}(k)$ $(1 \leq k \leq o)$ to get the linear system $LS_V$ of $o$ equations and $o$ unknowns and a constant vector $c_V = (c_1, \cdots, c_m)$.
  - Compute $LS_V^{-1}$. If $LS_V$ is not invertible then go back to the first step.
  - Store $< s_V, c_V, LS_V^{-1} >$ as the precomputed values.

- **Online phase**
  - Choose a random salt $r$ and compute $h = \mathcal{H}(\mathcal{H}(m) || r)$ for a message $m$.
  - From $< s_V = (s_1, \cdots, s_V), c_V = (c_1, \cdots, c_m), LS_V^{-1} >$, compute $LS_V^{-1} \cdot h_V^T = \alpha$, where $h_V = (h_1 - c_1, \cdots, h_m - c_m)$ and $h = (h_1, \cdots, h_m)$.
  - Compute $T^{-1} \cdot (S_V, \alpha)^T = \sigma$ and output $\tau = (\sigma, r)$ as a signature on $m$.

Precomputation (2/3)

Offline/Online Signing of Rainbow

- The offline phase of the first layer of Rainbow is similar with UOV.
- But in the second layer precomputation is limited - Some Vinegar variables $x_{v+1}, \cdots, x_{v+o_1}$ of the second layer are determined depending on the (hashed) message $h$.
- So precomputable values are:
  - $LS_{V,1}^{-1}$ - the inverse of the coefficient matrix $LS_{V,1}$ of the linear system obtained in the first layer
  - $C_{V,1}$ - a vector of constant terms in $\left(\mathbb{F}(1)(s_V), \cdots, \mathbb{F}(o_1)(s_V)\right)$
  - $\left(\mathbb{F}(o_1+1)(s_V), \cdots, \mathbb{F}(o_1+o_2)(s_V)\right)$ - linear terms and constant terms when $s_V$ is substituted into central polynomials in second layer
Precomputation (3/3)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Security Category</th>
<th>Unit</th>
<th>I</th>
<th>III</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>UOV</td>
<td>Sign w/o Precomp.</td>
<td>cycle</td>
<td>201 834</td>
<td>707 959</td>
<td>1 486 775</td>
</tr>
<tr>
<td></td>
<td>Precomp. (offline)</td>
<td>189 224</td>
<td>690 586</td>
<td>1 460 168</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sign w/ Precomp. (online)</td>
<td>cycle</td>
<td>11 788</td>
<td>19 439</td>
<td>23 133</td>
</tr>
<tr>
<td></td>
<td>Total (offline + online)</td>
<td>201 012</td>
<td>710 025</td>
<td>1 483 301</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Precomp. Memory Cost per Sig.</td>
<td>byte</td>
<td>2 256</td>
<td>5 402</td>
<td>9 504</td>
</tr>
<tr>
<td>Rainbow</td>
<td>Sign w/o Precomp.</td>
<td>cycle</td>
<td>68 203</td>
<td>322 799</td>
<td>807 309</td>
</tr>
<tr>
<td></td>
<td>Precomp. (offline)</td>
<td>37 212</td>
<td>173 204</td>
<td>508 890</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sign w/ Precomp. (online)</td>
<td>cycle</td>
<td>31 973</td>
<td>142 179</td>
<td>278 511</td>
</tr>
<tr>
<td></td>
<td>Total (offline + online)</td>
<td>69 185</td>
<td>315 383</td>
<td>787 401</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Precomp. Memory Cost per Sig.</td>
<td>byte</td>
<td>2 152</td>
<td>4 792</td>
<td>5 648</td>
</tr>
</tbody>
</table>
Resilence against Leakage or Reuse of Precomputed Values

**Store** $< s_V, c_V, LS_V^{-1} >$ **Securely.** The precomputed values $< s_V, c_V, LS_V^{-1} >$ should be stored securely.

**Theorem 3**

If $(n + 1)$ tuples $< m^{(i)}, \tau^{(i)}, s_V^{(i)}, c_V^{(i)}, LS_V^{(i)-1} >$ are given such that the $n \times n$ matrix $(\sigma^{(1)^T}, \sigma^{(2)^T}, \ldots, \sigma^{(n)^T})$ is invertible then the secret key of UOV is completely recovered in polynomial-time.

**Theorem 4**

If $(n + 1)$ tuples $< m^{(i)}, \tau^{(i)}, s_V^{(i)}, c_V^{(i)}, (LS_V^{(i)})^{-1} >$ are given such that the $n \times n$ matrix $(\sigma^{(1)^T}, \sigma^{(2)^T}, \ldots, \sigma^{(n)^T})$ is invertible then an equivalent key of Rainbow is completely recovered in polynomial-time.
Resilience against Leakage or Reuse of Precomputed Values

Do not Reuse $< s_V, c_V, Ls_{V}^{-1} >$. The precomputed value $< s_V, c_V, Ls_{V}^{-1} >$ should not be reused in signing.

Theorem 5 [SK20]
If $(m + 1)$ signatures generated by the reused Vinegar values are given then
- the equivalent key of UOV is completely recovered in polynomial time,
- the complexity of the KRAs using good keys on Rainbow is determined by solving a multivariate system of $m$ quadratic equations with $o_1$ variables.

Theorem 6
If $(o_2 + 1)$ signatures generated by reusing the precomputed values then an equivalent key of Rainbow is recovered in polynomial-time with high probability.

Conclusion

We presented two efficient implementation methods to improving signing of UOV and Rainbow, and gave implementation results on our methods.

- The Block Matrix Inversion - improves the process to solve the linear system
  - 10-40% faster than Gaussian elimination

- Precomputation - improves the process substituting Vinegar values and solving the linear system
  - For UOV, about 17, 36, and 64 times improvements for Security Level 1, 3, and 5, respectively.
This is the end of my presentation

Thank you for your attention.