KU LEUVEN

The Lifted UOV signature scheme

Ward Beullens Bart Preneel
Alan Szepieniec Frederik Vercauteren

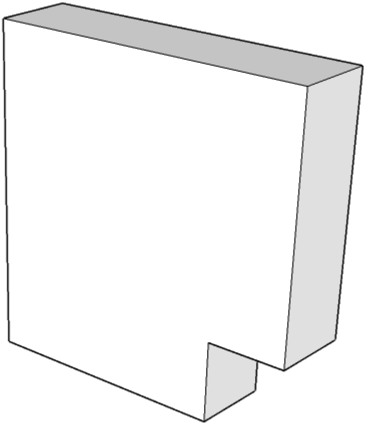
imec COSIC

5 April 2018

The **Lifted Unbalanced Oil and Vinegar** scheme is a variant of the UOV signature scheme.

One of the oldest and best studied multivariate signature schemes is **Unbalanced Oil and Vinegar** (UOV). It is fast and has small signatures, but the **public keys are large**.

We propose a simple adaptation of UOV that has **much smaller public keys**.



1. Unbalanced Oil and Vinegar
2. The main improvement
3. Brief security analysis
4. Some more improvements
5. Conclusion

The UOV signature scheme uses a map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ known as a UOV map.

Partition the n variables into $v = n - m$ vinegar variables $x_1, \cdots, x_v$ and m oil variables $x_{v+1}, \cdots, x_n$. A UOV map consists of m Polynomials of the form

$$f(x) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j} x_i x_j + \sum_{i=1}^{n} \beta_i x_i + \gamma \qquad \alpha_{i,j}, \beta_i, \gamma \in \mathbb{F}_q$$

Given $\mathbf{y} \in \mathbb{F}_q^m$ we can efficiently find $\mathbf{x} \in {\mathbb{F}_q}^n$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$.

1. Pick values for the vinegar variables randomly
2. Solve linear system of m equations and m variables to find the values of the the oil variables.

We hide the structure of $\mathcal{F}$ by composing it with a random invertible linear map $\mathcal{T}$ to get the public key $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. The public key $(\mathcal{F}, \mathcal{T})$ can be used to find preimages of $\mathcal{P}$.

Signature scheme:

Key generation : Pick $\mathcal{F}$,$\mathcal{T}$ randomly, compute $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$
Signing : Hash and sign: $\mathbf{s} = \mathcal{P}^{-1}(\mathcal{H}(d))$
Verification : check if $\mathcal{P}(\mathbf{s}) = \mathcal{H}(d)$

The public key consists of m quadratic polynomials in n variables, so roughly $m\frac{n^2}{2}\log_2(q)$ bits

Example

For 128 bits of security we have $m \approx 50$, $n \approx 150$, and $q = 2^8$, So

$$|pk| \approx 50 \times \frac{150^2}{2} \times 8 \text{ bits } \approx 560 \text{ KB}.$$

The public key consists of m quadratic polynomials in n variables, so roughly $m\frac{n^2}{2}\log_2(q)$ bits

Example

For 128 bits of security we have $m \approx 50$, $n \approx 150$, and $q = 2^8$, So

$$|pk| \approx 50 \times \frac{150^2}{2} \times 8 \text{ bits } \approx 560 \text{ KB}.$$

Optimization by Petzoldt reduces $|pk|$ to $\frac{m^3}{2}\log_2(q)$

Example

$$|pk| \approx \frac{50^3}{2} \times 8 \text{ bits } \approx 62 \text{ KB}$$

The hardness of solving polynomial systems depends on the size of the field.

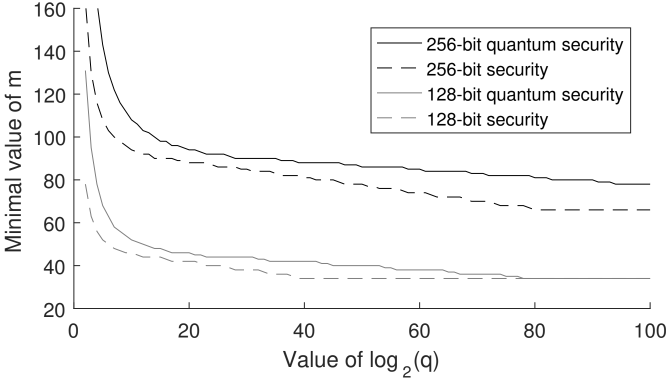


Figure: The number of variables needed such that solving a polynomial system is hard for different finite fields

The idea is to use two fields:

- A small field $\mathbb{F}_2$ for the public and secret keys i.e. $\mathcal{P}, \mathcal{F}$ and $\mathcal{T}$
- A large extension for output of $\mathcal{H}$ and the signatures. e.g. $\mathbb{F}_{2^{32}}$

The maps $\mathcal{P}, \mathcal{F}$ and $\mathcal{T}$ are defined over $\mathbb{F}_2$, but lifted to a large extension field.

Key generation is identical to UOV over $\mathbb{F}_2$, signature generation and verification is identical to UOV over the large field.

Forging a signature for a document d requires finding a solution to a multivariate system over $\mathbb{F}_{31}$.

$$\begin{aligned}
18x_1^2 + 7x_1x_2 + 5x_3 + 22x_1x_4 + 29x_4x_5 + 3x_5 &\equiv 20 \pmod{31} \\
6x_2x_3 + 12x_3^2 + 25x_2x_6 + 7x_3x_4 + 11x_3x_5 + 30x_6^2 &\equiv 11 \pmod{31} \\
\underbrace{15x_1x_2 + 9x_2x_3 + 12x_3x_4 + 25x_2 + 28x_5x_6}_{\mathcal{P}(\mathbf{x})} &\equiv \underbrace{8}_{\mathcal{H}(d)} \pmod{31}
\end{aligned}$$

Forging a signature for a document d requires finding a solution to a multivariate system over $\mathbb{F}_{2^{32}}$.

$$\underbrace{\begin{array}{r} x_1^2 + x_1x_2 + x_3 + x_1x_4 + x_4x_5 + x_5 \\ x_2x_3 + x_3^2 + x_2x_6 + x_3x_4 + x_3x_5 + x_6^2 \\ x_1x_2 + x_2x_3 + x_3x_4 + x_2 + x_5x_6 \end{array}}_{\mathcal{P}(\mathbf{x})} \begin{array}{c} = \\ = \\ = \end{array} \underbrace{\begin{array}{l} 1 + \alpha^2 + \cdots + \alpha^{30} \\ 1 + \alpha + \cdots + \alpha^{29} \\ \alpha + \alpha^5 + \cdots + \alpha^{31} \end{array}}_{\mathcal{H}(d)}$$

Direct attack

A direct attack tries to solve the system $\mathcal{P}(\mathbf{s}) = \mathcal{H}(d)$ to forge a signature $\mathbf{s}$.

- Theoretically: Degree of regularity of the system is the same as in the case of UOV over the large field.
- Experimentally: The Algebraic solver F_4 is not significantly better at attacking the new scheme than in the case of original UOV over the large field.

Key recovery attack

Tries to recover the secret key $(\mathcal{F}, \mathcal{T})$ from the public key $\mathcal{P}$. This attack is fully equivalent to key recovery attack against UOV over $\mathbb{F}_2$, so attacks are well understood.

- We use a secret key in 'normal form', i.e. $\mathcal{T}$ of the form

$$\begin{pmatrix} \mathbf{1}_{v\times v} & \mathbf{T} \\ \mathbf{0}_{m\times v} & \mathbf{1}_{m\times m} \end{pmatrix}$$

- We store the randomness used to generate the public key, and recompute the secret key each signing session needed.
- Message recovery mode ($\pm 15\%$ of |sig|).
- Trade off between |sig| and |pk|.

Table: Parameter sets achieving security level 2 of NIST

(q,m,n)	|sig|	|pk|	|sk|	KeyGen	Sign	Verify
$(2^8,63,256)$	0.3 KB	15.5 KB	32B	21 Mc	5.6 Mc	4.9 Mc
$(2^{48},49,242)$	1.7 KB	7.3 KB	32B	15 Mc	34 Mc	24 Mc

KU LEUVEN

sl	(q, m, n)	|sig|	|pk|	|sk|	KeyGen	Sign	Verify
2	$(2^8, 63, 256)$	0.3 KB	16 KB	32B	21	6	5
4	$(2^8, 90, 351)$	0.4 KB	45 KB	32B	81	22	17
5	$(2^8, 117, 404)$	0.5 KB	97 KB	32B	146	36	30

Disadvantages:

- Public key size
  (But 10x smaller than other MQ schemes)
- no security reduction

Advantages:

- Signature size
- Secret key size (minimal)
- Based on UOV
  (since 1999)

KU LEUVEN