White-Box Implementation Techniques for the HFE family

Pierre Galissant, Louis Goubin

Université de Versailles-St-Quentin-en-Yvelines

UFR des Sciences
CAMPUS DE VERSAILLES

White-Box Model

Multivariate Cryptography and HFE

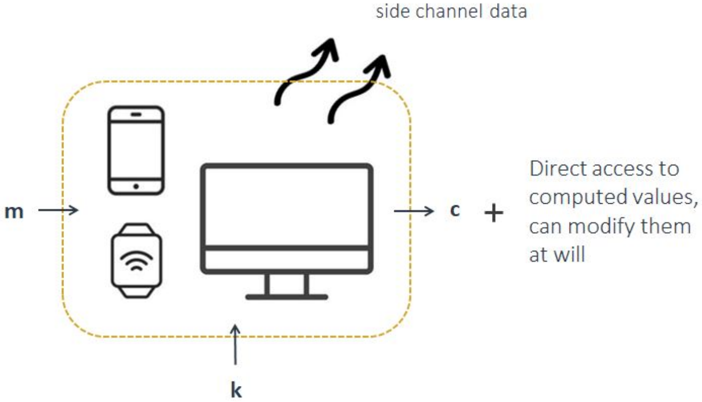
Implementation of HFE

- White-Box Security
- Fixing Parameters

Perspectives

The White-Box Model

↪ Introduced by Chow et al. in 2002



The White-Box Model (2)

Some Context

- ↪ Mainly focused on the symmetric setting
- ↪ WhiBox contest 17 and 19 : All AES implementations were broken
- ↪ Mainly based on masking or tables based solutions

For the Public-Key Setting

- ↪ WhiBox 21 and 24 contest on ECDSA : all implementations quickly broken
- ↪ Almost nothing on the public-key setting

Framework

White-Box Compiler

A white-box compiler $\mathcal{C}$ is probabilistic algorithm that on the input of a keyed-algorithm A and a key $k \in K$, outputs an implementation of A_k noted $\mathcal{C}_A(k)$ that aims to achieve security properties in the white-box model.

Security Notions

- $\hookrightarrow$ Unbreakability : "It should be hard to extract the key from the targeted code"
- $\hookrightarrow$ Incompressibility : "It should be hard to produce a smaller code that is functionally equivalent to the target"

Incompressibility

1. Draw at random a key k in private keyspace K

Incompressibility

1. Draw at random a key k in private keyspace K
2. The adversary $\mathcal{A}$ gets the program $\mathcal{C}_A(k)$ from the compiler

Incompressibility

1. Draw at random a key k in private keyspace K
2. The adversary $\mathcal{A}$ gets the program $\mathcal{C}_A(k)$ from the compiler
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ knowing $\mathcal{C}_A(k)$

Incompressibility

1. Draw at random a key k in private keyspace K
2. The adversary $\mathcal{A}$ gets the program $\mathcal{C}_A(k)$ from the compiler
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ knowing $\mathcal{C}_A(k)$
4. The adversary $\mathcal{A}$ succeeds if $\mathcal{P}$ and $\mathcal{C}_A(k)$ are equivalent and $size(\mathcal{P}) < \sigma$.

Incompressibility

1. Draw at random a key k in private keyspace K
2. The adversary $\mathcal{A}$ gets the program $\mathcal{C}_A(k)$ from the compiler
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ knowing $\mathcal{C}_A(k)$
4. The adversary $\mathcal{A}$ succeeds if $\mathcal{P}$ and $\mathcal{C}_A(k)$ are equivalent and $size(\mathcal{P}) < \sigma$.

Definition

We define the probability of the adversary $\mathcal{A}$ to succeed in the σ-incompressibility game by:

$$Succ_{\mathcal{A},\mathcal{C}_A} := \mathbb{P}[k \leftarrow K; \mathcal{P} = \mathcal{A}(\mathcal{C}_A(k))\,; \mathcal{P} \approx \mathcal{C}_A(k);\ (size(\mathcal{P}) < \sigma)]$$

We say that $\mathcal{C}_A$ is (σ,τ,ϵ)-incompressible if for any adversary $\mathcal{A}$, Time($\mathcal{A}$)+Time($\mathcal{P}$) $<$ τ implies $Succ_{\mathcal{A},\mathcal{C}_A} \leq \epsilon$.

Multivariate Cryptography

General Idea

$\hookrightarrow$ Public Key : $\begin{cases} P_1(x_1, \ldots, x_n) \\ \vdots \\ P_m(x_1, \ldots, x_n) \end{cases}$ over a field $\mathbb{F}$, mostly of degree 2

Multivariate Cryptography

General Idea

↪ Public Key : $\begin{cases} P_1(x_1, \ldots, x_n) \\ \vdots \\ P_m(x_1, \ldots, x_n) \end{cases}$ over a field $\mathbb{F}$, mostly of degree 2

↪ Given the public key and $(y_1, ..., y_n) \in \mathbb{F}^n$, it is hard to compute $(x_1, ..., x_n) \in \mathbb{F}^n$,

s.t. $\begin{cases} P_1(x_1, \ldots, x_n) = y_1 \\ \vdots \\ P_m(x_1, \ldots, x_n) = y_m \end{cases}$ ...

Multivariate Cryptography

General Idea

↪ Public Key : $\begin{cases} P_1(x_1, \ldots, x_n) \\ \vdots \\ P_m(x_1, \ldots, x_n) \end{cases}$ over a field $\mathbb{F}$, mostly of degree 2

↪ Given the public key and $(y_1, ..., y_n) \in \mathbb{F}^n$, it is hard to compute $(x_1, ..., x_n) \in \mathbb{F}^n$,

s.t. $\begin{cases} P_1(x_1, \ldots, x_n) = y_1 \\ \vdots \\ P_m(x_1, \ldots, x_n) = y_m \end{cases}$ ...

↪ ... but given a secret-key, the inversion is easy.

Multivariate Cryptography (2)

$\hookrightarrow$ For quadratic equations : the MQ problem is NP-hard problem

$\hookrightarrow$ But with a trapdoor, so more cryptanalysis needed

Multivariate Cryptography (2)

↪ For quadratic equations : the MQ problem is NP-hard problem

↪ But with a trapdoor, so more cryptanalysis needed

↪ Among the most famous multivariate public-key schemes:

- Big Field
  - C^* (Matsumoto and Imai, 1983)
  - HFE (Hidden Field Equations, by Patarin, 1996)
  - GeMSS (Casanova et al. 2018)
- Oil and Vinegar
  - UOV (Unbalanced Oil and Vinegar, by Goubin, Kipnis, Patarin, 1999)
  - Rainbow (Ding, Schmidt, 2005)
- New candidates to NIST PQC like VOX,PROV,..

HFE - (Hidden Field Equation)

Signature Algorithm described by Patarin (1996)

- ↪ F of degree 2 over $\mathbb{F}_2^n$
- ↪ S, T affine bijections over $\mathbb{F}_2^n$.
- ↪ The public-key is $P = S \circ \pi \circ F \circ \pi^{-1} \circ T$.

$$
\begin{tikzcd}
 & \mathbb{F}_{2^n} \arrow[r, "F"] \arrow[d, "\pi"'] & \mathbb{F}_{2^n} \arrow[d, "\pi"] & \\
\mathbb{F}_2^n \arrow[r, "T"] \arrow[rrr, bend right, "P"'] & \mathbb{F}_2^n \arrow[r] & \mathbb{F}_2^n \arrow[r, "S"] & \mathbb{F}_2^n
\end{tikzcd}
$$

HFE - (Hidden Field Equation)

Signature Algorithm described by Patarin (1996)

- $\hookrightarrow$ F of degree 2 over $\mathbb{F}_2^n$ Inverting F is easy its degree D is not too big.
- $\hookrightarrow$ S, T affine bijections over $\mathbb{F}_2^n$. Inverting S, T and π is easy
- $\hookrightarrow$ The public-key is $P = S \circ \pi \circ F \circ \pi^{-1} \circ T$. Inverting P is hard

$$
\begin{tikzcd}
& \mathbb{F}_{2^n} \arrow[r, "F"] \arrow[d, "\pi"'] & \mathbb{F}_{2^n} \arrow[d, "\pi"] & \\
\mathbb{F}_2^n \arrow[r, "T"] \arrow[rrr, bend right, "P"'] & \mathbb{F}_2^n \arrow[r] & \mathbb{F}_2^n \arrow[r, "S"] & \mathbb{F}_2^n
\end{tikzcd}
$$

The IP Problem and White-Box

Isomorphism of polynomials (IP Problem)

$\hookrightarrow$ Given $P = S \circ F \circ T$ and F systems of polynomials of degree 2 over a field $\mathbb{F}_2$ and secret S, T affine bijections over $\mathbb{F}_2^n$

The IP Problem and White-Box

Isomorphism of polynomials (IP Problem)

- ↪ Given $P = S \circ F \circ T$ and F systems of polynomials of degree 2 over a field $\mathbb{F}_2$ and secret S, T affine bijections over $\mathbb{F}_2^n$
- ↪ Find S and T

The IP Problem and White-Box

Isomorphism of polynomials (IP Problem)

- ↪ Given $P = S \circ F \circ T$ and F systems of polynomials of degree 2 over a field $\mathbb{F}_2$ and secret S, T affine bijections over $\mathbb{F}_2^n$
- ↪ Find S and T

Links with the White-Box Model

- ↪ Composition has "incompressibility" properties if F has succinct representation
- ↪ Similarly to C^* , $F = x^3$: $Size(S, T) = 2n^2$ and $Size(P) = n \times \sigma(n, 2)$

HFE - Perturbations

A Way to Enhance Security

- The Projection Perturbation: Fix p coordinates of the public key
- The Minus Perturbation: Remove a coordinates from the public key
- The Hat Plus perturbation: Add a quadratic form Q whose image is a vector space of dimension t:

$$Q(X) = \sum_{0 \leq i \leq t} \beta_i \times p_i(x_1, ..., x_n)$$

HFE - Black-Box Security

Black-Box Security

- $\hookrightarrow$ Best Invertion through Gröbner Basis computation : $\mathcal{O}\left(\binom{n+d_{reg}}{d_{reg}}^{\omega}\right)$
- $\hookrightarrow$ Best Key-Recovery : $\mathcal{O}\left(\left(n^2\binom{2d+2}{d} + n\binom{2d+2}{d}^2\right)^{\omega}\right)$, with $d = \lceil \log_2 D) \rceil$
- $\hookrightarrow$ Structural attacks : F is a monomial or F is only over $\mathbb{F}_2$

Implementation of HFE

Our Contributions

1. We propose the first white-box implementation technique for the HFE family

Implementation of HFE

Our Contributions

1. We propose the first white-box implementation technique for the HFE family
2. We reduce the unbreakability of our implementation to the study of a particular instance of the IP problem

Implementation of HFE

Our Contributions

1. We propose the first white-box implementation technique for the HFE family
2. We reduce the unbreakability of our implementation to the study of a particular instance of the IP problem
3. We revisit the notion of incompressibility in the public-key setting and state a precise conjecture regarding the incompressiblity of our construction

Implementation of HFE

Our Contributions

1. We propose the first white-box implementation technique for the HFE family
2. We reduce the unbreakability of our implementation to the study of a particular instance of the IP problem
3. We revisit the notion of incompressibility in the public-key setting and state a precise conjecture regarding the incompressiblity of our construction
4. We propose a challenge implementation to motivate the study of our implementation technique

Starting Point: Affine Multiple Attack

A Structural Remark

- ↪ Generalization of Patarin's attack on C^* over $\mathbb{F}_2^n$:

$$F(x) = x^3 = y \implies xy^2 = x^4 y$$

- ↪ Frobenius is linear, we get n equations for $P = S \circ F \circ T$:

$$\sum a_{i,j} x_i y_j + \sum b_i x_i + \sum c_j y_j + d = 0$$

Starting Point: Affine Multiple Attack

A Structural Remark

↪ Generalization of Patarin's attack on C^* over $\mathbb{F}_2^n$:

$$F(x) = x^3 = y \implies xy^2 = x^4 y$$

↪ Frobenius is linear, we get n equations for $P = S \circ F \circ T$:

$$\sum a_{i,j} x_i y_j + \sum b_i x_i + \sum c_j y_j + d = 0$$

Affine Multiple Attack to invert P

1. Get pairs $(x, P(x))$ to solve a linear system in $a_{i,j}, b_i, c_j$ and d
2. Once these coefficients are known, plugging y allow the recovery of x by Gaussian reduction.

Affine Multiple Attack (2)

Definition

Let $F \in \mathbb{F}_{2^n}[x]$. The polynomial $A(x, y) \in \mathbb{F}_{2^n}[x, y]$ is said to be an affine multiple of F if $A(x, y) = 0 \mod (F(x) - y)$ and A is $\mathbb{F}_2$-linear in x.

$\hookrightarrow$ $d_{\mathit{aff}} :=$ maximum Hamming weight of the monomials in y in the polynomial $A(x, y)$.

Affine Multiple Relations

Existence

The vector space $\mathbb{F}_{2^n}(y)[x]/_{(P(x)-y)}$ of dimension $D = deg(F)$ over $\mathbb{F}_{2^n}(y)$. The $D+1$ $\mathbb{F}_{2^n}$-linear polynomials $(1, x^{2^0}, x^{2^1}, ..., x^{2^{D-1}})$ are linearly dependent

We now need an algorithm to compute the dependency relation :

$$a + \sum_{i=0}^{D-1} a_i x^{2^i} = 0 \bmod (P(x) - y), \quad a, a_o, ..., a_{D-1} \in \mathbb{F}_{2^n}(y)$$

Affine Multiple Relations (2)

In practice

$\hookrightarrow$ Compute the $b_{i,j}$ such that $x^{2^i} = \sum_{j=0}^{D-1} b_{i,j} x^j \bmod (P(x) - y)$. This translates to the relations:

$$a + \sum_{i=0}^{D-1} a_i \sum_{j=0}^{D-1} b_{i,j} x^j = 0$$

$\hookrightarrow$ Solve the linear system over $\mathbb{F}_{2^n}(y)$:

$$\begin{pmatrix} 1 & b_{0,0} & \cdots & b_{D-1,0} \\ 0 & b_{0,1} & \cdots & b_{D-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b_{0,D-1} & \cdots & b_{D-1,D-1} \end{pmatrix} \times \begin{pmatrix} a \\ a_0 \\ \vdots \\ a_{D-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(and normalize to get the a_k over $\mathbb{F}_{2^n}[y]$)

Affine Multiple Relations (3)

Prohibitive Costs for Standard HFE

Essentially, we perform modular reductions and solve a linear system over $\mathbb{F}_{2^n}(y)$. The complexity is then :

$$\mathcal{O}(M(n, 2^D)D^\omega)$$

We need to focus on polynomials of small degree D.

Affine Multiple Relations (3)

Prohibitive Costs for Standard HFE

Essentially, we perform modular reductions and solve a linear system over $\mathbb{F}_{2^n}(y)$. The complexity is then :

$$\mathcal{O}(M(n, 2^D)D^\omega)$$

We need to focus on polynomials of small degree D.

Black-Box Security

$\hookrightarrow$ Black-box security depends on F, and perturbations.

The Construction

WBHFE algorithm

- ↪ Compute $A(x, y)$
- ↪ Compute the coordinates $A_i(x_1, ..., x_n, y_1, ..., y_n)$ (for $i \in [\![1, n]\!]$) of A through the isomorphism π
- ↪ Compute the composition :

$$\tilde{\mathbf{A}}_{\mathbf{i}}(x_1, ..., x_n, y_1, ..., y_n) = A_i(S(x_1, ..., x_n), T^{-1}(y_1, ..., y_n))$$

- ↪ To get a preimage of y, $P(x) = y$, plug $y = (y_1, ..., y_n)$ into the $\tilde{\mathbf{A}}_{\mathbf{i}}$ to get a linear system in $x = (x_1, ..., x_n)$

Adding Perturbations

Projection

$\hookrightarrow$ We simply project the coordinates on the affine multiple

Hat Plus and Minus

$\hookrightarrow$ We change the modulus ! For any integer $m_s > 0$ and split *Im*$(Q) = \cup_{i=1}^{m_s} V_i$ where $\#V_i = \delta_i \geq 2$. We set $V_i = \{v_{i,1}, ..., v_{i,\delta_i}\}$:

$$H_i(x) = \prod_{j=1}^{\delta_i} (y - F(x) - v_{i,j})$$

Size of the construction

Size of the WB Code

As we store the n polynomials $\tilde{A}_i$, which are of degree d_{aff} in y_i and linear in x_i, the size is upper bounded by :

$$n^2\sigma(n, d_{\mathit{aff}})$$

Constraint

The size (and the running time) is exponential in d_{aff} : we need to minimize it !

White-Box Security

A conjecture for White-Box Security

$\hookrightarrow$ Based on incompressibility of IP instances.

Usual White-Box Attacks

$\hookrightarrow$ Can we assess the efficiency of DCA, DFA, etc .. ?

Incompressibility of IP Instances

1. Draw at random two secrets S, T in $\mathsf{Aff}_n(\mathbb{F}_2)$

Incompressibility of IP Instances

1. Draw at random two secrets S, T in $\mathsf{Aff}_n(\mathbb{F}_2)$
2. The adversary $\mathcal{A}$ is given an IP instance $(\tilde{P}_i)_{i \in [\![1,m]\!]}$ composed of $(P_i)_{i \in [\![1,m]\!]}$, S and T

Incompressibility of IP Instances

1. Draw at random two secrets S, T in $\mathsf{Aff}_n(\mathbb{F}_2)$
2. The adversary $\mathcal{A}$ is given an IP instance $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ composed of $(P_i)_{i\in[\![1,m]\!]}$, S and T
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ that allows to evaluate $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ for every element $(\mathbb{F}_2)^n$

Incompressibility of IP Instances

1. Draw at random two secrets S, T in $\mathsf{Aff}_n(\mathbb{F}_2)$
2. The adversary $\mathcal{A}$ is given an IP instance $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ composed of $(P_i)_{i\in[\![1,m]\!]}$, S and T
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ that allows to evaluate $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ for every element $(\mathbb{F}_2)^n$
4. The adversary $\mathcal{A}$ wins if $size(\mathcal{P}) < \sigma$

Incompressibility of IP Instances

1. Draw at random two secrets S, T in $\mathsf{Aff}_n(\mathbb{F}_2)$
2. The adversary $\mathcal{A}$ is given an IP instance $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ composed of $(P_i)_{i\in[\![1,m]\!]}$, S and T
3. The adversary $\mathcal{A}$ returns a program $\mathcal{P}$ that allows to evaluate $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ for every element $(\mathbb{F}_2)^n$
4. The adversary $\mathcal{A}$ wins if $size(\mathcal{P}) < \sigma$

Definition

Let $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ be an IP instance with polynomials in n variables over $\mathbb{F}_2$, with known polynomials $(P_i)_{i\in[\![1,m]\!]}$ and secrets S, T and let $\mathcal{A}$ an adversary. We say that $(\tilde{P}_i)_{i\in[\![1,m]\!]}$ is (σ,τ,ϵ)-incompressible if there is no adversary $\mathcal{A}$ that wins the σ-incompressibility game with probability ϵ and Time($\mathcal{A}$)+Time($\mathcal{P}$) $< \tau$.

Incompressibility of IP Instances (2)

What do we know about this problem ?

$\hookrightarrow$ The instance is structured, in particular, it corresponds to a HFE public key

Incompressibility of IP Instances (2)

What do we know about this problem ?

- ↪ The instance is structured, in particular, it corresponds to a HFE public key
- ↪ Could be useful for multivariate keys, but not studied that much
- ↪ Best known attack is complete key recovery (i.e. unbreakability)
- ↪ We use a variation where the central polynomial is of degree 3 or 4

Discussion on general attacks : DCA, DFA, ...

Multivariate Cryptography is nice for White-Box !

$\hookrightarrow$ It diffuses entirely by composition : HFE public-keys are already IP instances

$\hookrightarrow$ For combinatorial reasons, usual white-box attacks are not known to solve the IP problem

White-Box Model

Multivariate Cryptography and HFE

Implementation of HFE

- White-Box Security
- Fixing Parameters

Perspectives

Summary on Constraints

Black-Box Security from the State of the Art

- ↪ Best Invertion through Gröbner Basis computation : $\mathcal{O}\left(\binom{n+d_{reg}}{d_{reg}}^{\omega}\right)$
- ↪ Best Key-Recovery : $\mathcal{O}\left(\left(n^2\binom{2d+2}{d} + n\binom{2d+2}{d}^2\right)^{\omega}\right)$
- ↪ Structural attacks : F is a monomial or F is only over $\mathbb{F}_2$

White-Box Size and Security

- ↪ Minimize d_{aff} on affine multiples compatible with perturbations
- ↪ Check the parameters against the best attacks in the white-box model

Fixing Parameters for $\lambda = 80$

Nude HFE

- $\hookrightarrow$ $F = x^6 + Ax^5 + Bx^3,\ A, B \in \mathbb{F}_2^n$
- $\hookrightarrow$ $log(n) = 11.8$, $log(size) = 45.7$ ($\approx$ 10TB)

Variations of pC^{*-}

- $\hookrightarrow$ $F = x^3 + Ax^2,\ A \in \mathbb{F}_2^n$
- $\hookrightarrow$ $n = 101, p = 12, a = 21$, $\log(size) = 30.5$ ($\approx$ 187MB)

Variations of $C^{*\hat{+}-}$ (Challenge: github.com/p-galissant/WBHFE)

- $\hookrightarrow$ $F = x^3 + Ax^2,\ A \in \mathbb{F}_2^n$
- $\hookrightarrow$ $n = 85, t = 9, a = 5$, $log(size) = 29.5$ ($\approx$ 93MB)

Perspectives

Further understanding of affine multiples

- $\hookrightarrow$ Can we avoid exhaustive search to find interesting affine multiples ?
- $\hookrightarrow$ Can we extend this technique to other trapdoors ?

Polynomial Composition and White-Box

- $\hookrightarrow$ Can we improve our understanding of the compression of IP instances ?
- $\hookrightarrow$ Can we use composition problems for other white-box implementations ?

Thank You