Security Analysis of Multivariate Polynomials for Hashing

Luk Bettale[1], Jean-Charles Faugère, Ludovic Perret

SALSA
LIP6 UPMC INRIA Paris-Rocquencourt, France

INSCRYPT 2008

[1]author partially supported by DGA/MRIS (french secretary of defense)

Outline

Description of multivariate hash functions

A family of functions based on the difficulty of solving polynomial system.

- Use Merkle-Damgård,
- Compression function $\Leftrightarrow$ Algebraic system.

Let $\mathbb{K} = \mathbb{F}_q$, $f : \mathbb{K}^{m+n} \longrightarrow \mathbb{K}^m$

$$\begin{cases} f_1(y_1, \ldots, y_m, x_1, \ldots, x_n) \\ \vdots \\ f_m(y_1, \ldots, y_m, x_1, \ldots, x_n) \end{cases}$$

Multivariate hash functions – Parameters

Jintai Ding and Bo-Yin Yang.
Multivariate polynomials for hashing.
INSCRYPT 2007.

Three constructions proposed by Ding and Yang :

- "Dense" *cubic* random polynomials,
- "Sparse" *cubic* random polynomials,
- Composition of 2 quadratic systems (degree 4 polynomials).
  - also proposed by Billet, Robshaw and Peyrin (ACISP 2007)

Parameter sets ($m = n$)

160-bits hash

$\#\mathbb{K} = 2^8, n = 20$
$\#\mathbb{K} = 2^4, n = 40$

256-bits hash

$\#\mathbb{K} = 2^{16}, n = 16$
$\#\mathbb{K} = 2^8, n = 32$
$\#\mathbb{K} = 2^4, n = 64$

Multivariate hash functions – Parameters

Jintai Ding and Bo-Yin Yang.
Multivariate polynomials for hashing.
INSCRYPT 2007.

Three constructions proposed by Ding and Yang :

- "Dense" *cubic* random polynomials,
- "Sparse" *cubic* random polynomials,
- Composition of 2 quadratic systems (degree 4 polynomials).
  - also proposed by Billet, Robshaw and Peyrin (ACISP 2007)

Parameter sets ($m = n$)

160-bits hash

$\#\mathbb{K} = 2^8, n = 20 \rightarrow$ broken
$\#\mathbb{K} = 2^4, n = 40$

256-bits hash

$\#\mathbb{K} = 2^{16}, n = 16 \rightarrow$ broken
$\#\mathbb{K} = 2^8, n = 32$
$\#\mathbb{K} = 2^4, n = 64$

Multivariate hash functions – Parameters

Jintai Ding and Bo-Yin Yang.
Multivariate polynomials for hashing.
INSCRYPT 2007.

Three constructions proposed by Ding and Yang :

- "Dense" *cubic* random polynomials,
- "Sparse" *cubic* random polynomials,
- Composition of 2 quadratic systems (degree 4 polynomials).
  - also proposed by Billet, Robshaw and Peyrin (ACISP 2007)

Parameter sets ($m = n$)

160-bits hash

$\#\mathbb{K} = 2^8, n = 20 \rightarrow$ broken
$\#\mathbb{K} = 2^4, n = 40$

256-bits hash

$\#\mathbb{K} = 2^{16}, n = 16 \rightarrow$ broken
$\#\mathbb{K} = 2^8, n = 32 \rightarrow$ broken
$\#\mathbb{K} = 2^4, n = 64$

General attacks on multivariate hash functions

Preimage attack

Given a digest $(h_1, \ldots, h_m)$ and a fixed $IV = (v_1, \ldots, v_m)$:

Find $(x_1, \ldots, x_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_1 = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_m = 0 \end{cases}$$

Collision attack

Given a fixed $IV = (v_1, \ldots, v_m)$:

Find any non-zero difference $(\delta_1, \ldots, \delta_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \end{cases}$$

General attacks on multivariate hash functions

Preimage attack (degree 3)

Given a digest $(h_1, \ldots, h_m)$ and a fixed $IV = (v_1, \ldots, v_m)$:

Find $(x_1, \ldots, x_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_1 = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_m = 0 \end{cases}$$

Collision attack

Given a fixed $IV = (v_1, \ldots, v_m)$:

Find any non-zero difference $(\delta_1, \ldots, \delta_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \end{cases}$$

Preimage attack (degree 3)

Given a digest $(h_1, \ldots, h_m)$ and a fixed $IV = (v_1, \ldots, v_m)$:

Find $(x_1, \ldots, x_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_1 = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - h_m = 0 \end{cases}$$

Collision attack (degree 2)

Given a fixed $IV = (v_1, \ldots, v_m)$:

Find any non-zero difference $(\delta_1, \ldots, \delta_n)$ such that:

$$\begin{cases} f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \\ \vdots \\ f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) = 0 \end{cases}$$

Roadmap of the algebraic attack

1. randomly choose a non-zero difference $(\delta_1, \ldots, \delta_n)$.
2. fix the values of $(y_1, \ldots, y_m)$ to the IV and build the system $f' = (f'_1, \ldots, f'_m)$ with:

$$\begin{aligned} f'_1 &= f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \\ &\vdots \\ f'_m &= f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \end{aligned}$$

3. compute the solutions of $f' = 0$.
4. if we find a solution, then we have a collision, else come back to step 1.

Roadmap of the algebraic attack

1. randomly choose a non-zero difference $(\delta_1, \ldots, \delta_n)$.
2. fix the values of $(y_1, \ldots, y_m)$ to the IV and build the system $f' = (f'_1, \ldots, f'_m)$ with:

$$\begin{aligned} f'_1 &= f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \\ &\vdots \\ f'_m &= f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \end{aligned}$$

3. solve the system $f' = 0$.
4. if we find a solution, then we have a collision, else come back to step 1.

Roadmap of the algebraic attack

1. randomly choose a non-zero difference $(\delta_1, \ldots, \delta_n)$.
2. fix the values of $(y_1, \ldots, y_m)$ to the IV and build the system $f' = (f'_1, \ldots, f'_m)$ with:

$$\begin{aligned} f'_1 &= f_1(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_1(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \\ &\vdots \\ f'_m &= f_m(v_1, \ldots, v_m, x_1, \ldots, x_n) - f_m(v_1, \ldots, v_m, x_1 + \delta_1, \ldots, x_n + \delta_n) \end{aligned}$$

3. solve the system $f' = 0$.
4. if we find a solution, then we have a collision, else come back to step 1.

Tools for solving polynomial systems ?

About Gröbner bases

Definition

A set $G \subset \mathbb{K}[x_1, \ldots, x_n]$ is a Gröbner basis w.r.t. a monomial ordering $\prec$ of a polynomial ideal $\mathcal{I}$ if :
$\forall f \in \mathcal{I}, \exists g \in G$ such that $LM_{\prec}(g)$ divide $LM_{\prec}(f)$

About Gröbner bases

Definition

A set $G \subset \mathbb{K}[x_1, \ldots, x_n]$ is a Gröbner basis w.r.t. a monomial ordering $\prec$ of a polynomial ideal $\mathcal{I}$ if :
$\forall f \in \mathcal{I}, \exists g \in G$ such that $LM_{\prec}(g)$ divide $LM_{\prec}(f)$

Property

*A Gröbner basis for the lexicographic order of a **zero-dimensional** system has the following shape :*

$$\begin{cases} g_1(x_1), \\ g_2(x_1, x_2), \\ \vdots \\ g_{k_2}(x_1, x_2), \\ g_{k_3}(x_1, x_2, x_3), \\ \vdots \\ g_{k_n}(x_1, \ldots\ldots, x_n), \end{cases}$$

About Gröbner bases

Definition

A set $G \subset \mathbb{K}[x_1, \ldots, x_n]$ is a Gröbner basis w.r.t. a monomial ordering $\prec$ of a polynomial ideal $\mathcal{I}$ if :
$\forall f \in \mathcal{I}, \exists g \in G$ such that $LM_{\prec}(g)$ divide $LM_{\prec}(f)$

Property

*A Gröbner basis for the lexicographic order of a **zero-dimensional** system has the following shape :*

$$\begin{cases} g_1(x_1), \\ g_2(x_1, x_2), \\ \vdots \\ g_{k_2}(x_1, x_2), \\ g_{k_3}(x_1, x_2, x_3), \\ \vdots \\ g_{k_n}(x_1, \ldots\ldots, x_n), \end{cases}$$

Zero-dim solving strategy:

1. *Compute a DRL Gröbner basis.*
2. *Compute a lex Gröbner basis with a change ordering algorithm.*

Gröbner bases algorithms

Algorithms

- Buchberger : the historical algorithm
- F4 : linear algebra on matrices
- F5 : no useless computations for semi-regular systems

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases (F4).

Journal of Pure and Applied Algebra 139, June 1999.

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).

ISSAC 2002, July 2002.

Algorithms

- Buchberger : the historical algorithm
- F4 : linear algebra on matrices
- F5 : no useless computations for semi-regular systems

Complexity of F5 : $\mathcal{O}\left(\left(m \cdot \mathrm{C}_{n+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right)$, with $2 \leq \omega \leq 3$

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases (F4).

Journal of Pure and Applied Algebra 139, June 1999.

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).

ISSAC 2002, July 2002.

Algorithms

- Buchberger : the historical algorithm
- F4 : linear algebra on matrices
- F5 : no useless computations for semi-regular systems

Complexity of F5 : $\mathcal{O}\left(\left(m \cdot \mathrm{C}_{n+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right)$, with $2 \leq \omega \leq 3$

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases (F4).

Journal of Pure and Applied Algebra 139, June 1999.

Jean-Charles Faugère.

A new efficient algorithm for computing Gröbner bases without reduction to zero (F5).

ISSAC 2002, July 2002.

Algorithms

- F5 : no useless computations for semi-regular systems

Complexity of F5 : $\mathcal{O}\left(\left(m \cdot \mathrm{C}_{n+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right)$, with $2 \leq \omega \leq 3$

Semi-regular systems

- A system of unrelated polynomials
- The degree of regularity (d_{reg}) can be known a priori.
- The more equations we have, the more d_{reg} decrease.

Magali Bardet, Jean-Charles Faugère, and Bruno Salvy.
On the complexity of Gröbner basis computation of semi-regular over-defined algebraic equations.
Proc. ICPSS.

Algorithms

- F5 : no useless computations for semi-regular systems

Complexity of F5 : $\mathcal{O}\left(\left(m \cdot \mathrm{C}_{n+d_{\mathrm{reg}}-1}^{d_{\mathrm{reg}}}\right)^{\omega}\right)$, with $2 \leq \omega \leq 3$

Semi-regular systems

- A system of unrelated polynomials $\approx$ a random system
- The degree of regularity (d_{reg}) can be known a priori.
- The more equations we have, the more d_{reg} decrease.

Magali Bardet, Jean-Charles Faugère, and Bruno Salvy.
On the complexity of Gröbner basis computation of semi-regular over-defined algebraic equations.
Proc. ICPSS.

Solving the system – Naive approach

Back to our system

$f'_i \in \mathbb{K}[x_1, \ldots, x_n]$ for $1 \leq i \leq n$

$$\begin{cases} f'_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f'_n(x_1, \ldots, x_n) = 0 \end{cases}$$

Specificity $(n = m)$

- Square system and no field equations $\Rightarrow 2^n$ solutions
- Random system (experimentally semi-regular) $\Rightarrow d_{reg} = n + 1$
- n big (≥ 16)

Solving the system – Naive approach

Back to our system

$f'_i \in \mathbb{K}[x_1, \ldots, x_n]$ for $1 \leq i \leq n$

$$\begin{cases} f'_1(x_1, \ldots, x_n) = 0 \\ \vdots \\ f'_n(x_1, \ldots, x_n) = 0 \end{cases}$$

Specificity $(n = m)$

- Square system and no field equations $\Rightarrow 2^n$ solutions
- Random system (experimentally semi-regular) $\Rightarrow d_{reg} = n + 1$
- n big (≥ 16)

Cannot be solved directly

Solving the system – Hybrid approach

Solution

We specialize k variables of the system (random guess)

$\Rightarrow$ the system becomes over-defined

- The degree of regularity decreases.
- The number of solutions is 0 or 1.
- We have to compute $\#\mathbb{K}^k$ Gröbner bases.

Luk Bettale, Jean-Charles Faugère, and Ludovic Perret.

Cryptanalysis of the TRMS signature scheme of PKC'05.

AFRICACRYPT 2008.

Jean-Charles Faugère, and Ludovic Perret.

On the security of UOV.

SCC 2008.

Solving the system – Hybrid approach

Solution

We specialize k variables of the system (random guess)

$\Rightarrow$ the system becomes over-defined

- The degree of regularity decreases.
- The number of solutions is 0 or 1.
- We have to compute $\#\mathbb{K}^k$ Gröbner bases.

Luk Bettale, Jean-Charles Faugère, and Ludovic Perret.

Cryptanalysis of the TRMS signature scheme of PKC'05.

AFRICACRYPT 2008.

Jean-Charles Faugère, and Ludovic Perret.

On the security of UOV.

SCC 2008.

A tradeoff between exhaustive search and Gröbner bases computation.

Hybrid approach in practice – Dense construction

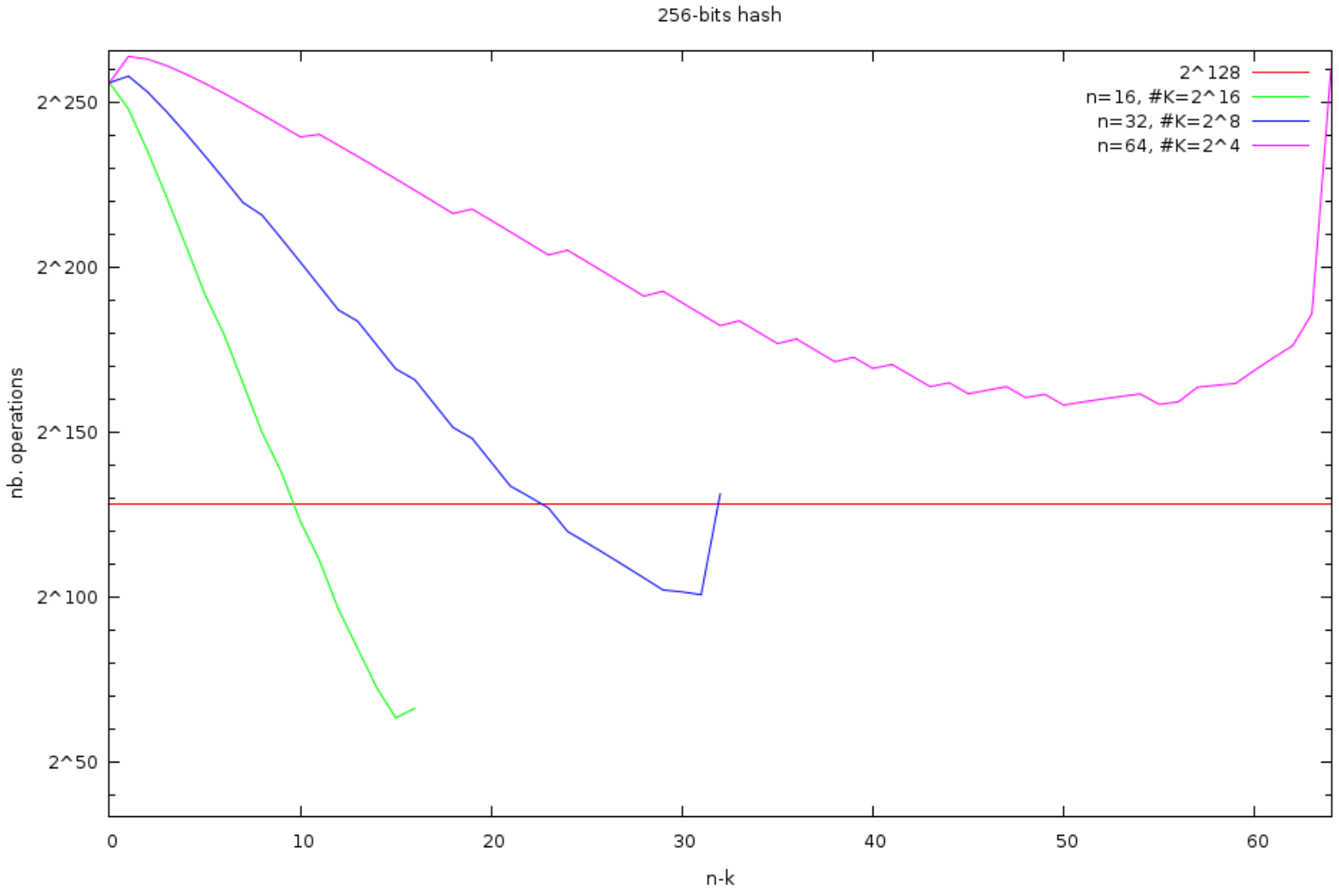
Computations done with the FGb library (bi-pro Xeon 2.4 GHz with 64GB of RAM).

Collision attack

$\#\mathbb{K}$	n	$n-k$	k	T_{F_5}	$\mathrm{Nop}_{\mathrm{F}_5}$	N	$\mathrm{N}_{\mathrm{gen}}$
2^8	20	18	2	51 h.	2^{41}	2^{57}	2^{80}
		17	3	2h45min.	2^{37}	2^{61}	
		16	4	643.1 s.	2^{34}	2^{66}	
2^{16}	16	15	1	≈ 1 h.	$2^{36.9}$	$2^{52.9}$	2^{128}
		14	2	126 s.	$2^{32.3}$	$2^{64.3}$	

Figure: Time and complexity of solving

Theoretical complexity of several parameters



Roadmap – sparse construction

1. randomly choose a non-zero difference $\boldsymbol{\delta}$ of **low** fixed Hamming weight $w(\boldsymbol{\delta})$
2. fix the values of y to the IV and build the system $f' = f(\mathbf{v}, \mathbf{x} + \boldsymbol{\delta}) - f(\mathbf{v}, \mathbf{x}) = 0$.
3. compute the solutions of f'
4. if we find a solution, then we have a collision, else come back to step 1

Results – Sparse construction

Computations done with the MAGMA[2] computational algebra system (bi-pro Xeon 2.4 GHz with 64GB of RAM).

Collision attack

$\#\mathbb{K}$	n	$n-k$	ϵ	$w(\delta)$	time min/max		prob
2^8	20	20	0.2%	4	0.5 s.	48 h.	1/4
2^{16}	16	16	0.2%	5	0.1 s.	311.9 s.	1/3
2^8	32	32	0.1%	2	0.4 s.	690.3 s.	1/15

Figure: Time of solving and probability to find a collision

- Less variables to be fixed (or none).
- Sparse systems seem much easier to solve.

[2] http://magma.maths.usyd.edu.au/

Conclusion

- Sparse systems should be avoided.
- Dense systems:
  - Some parameters proposed at INSCRYPT 2007 are broken
  - Parameters sets with $n < 40$ are not recommended

	$\#\mathbb{K}$	n	comp. time (cycles/byte)	security (nb. ops)	generic (nb. ops)
✗	2^8	20	67300	2^{67}	2^{80}
	2^4	40	4233000	2^{105}	
✗	2^{16}	16	48200	2^{64}	2^{128}
✗	2^8	32	3040000	2^{101}	
	2^4	64	9533000	2^{158}	

Figure: Tradeoff speed/security for dense construction