Algebraic attack on HFE revisited

Jintai Ding, Dieter S. Schmidt &
Fabian Werner

Multivariate Public Key Cryptosystems

- small field $k = GF(q)$
- extension Field $K = GF(q^n)$
- up to isomorphism:

$$K = \frac{k[t]}{\langle g(t) \rangle}$$

$g(t)$ *irreducible*, $\langle g(t) \rangle$ *is the ideal generated by* $g(t)$

- K can be seen as vector space over k
- → MVPKCS exploit this

up and down

- up for going from k to K ("upwards")
- down for going from K to k ("downwards")

$$up(x_0, \ldots, x_{n-1}) = x_0 + x_1 * t + \ldots + x_{n-1} * t^{n-1}$$

$$down(x_0 + x_1 * t + \ldots + x_{n-1} * t^{n-1}) = (x_0, x_1, \ldots, x_{n-1})$$

up and down

- up for going from k to K (“upwards”)
- down for going from K to k (“downwards”)

$$up(x_0, \ldots, x_{n-1}) = x_0 + x_1 * t + \ldots + x_{n-1} * t^{n-1}$$

$$down(x_0 + x_1 * t + \ldots + x_{n-1} * t^{n-1}) = (x_0, x_1, \ldots, x_{n-1})$$

example:

$$up(1,2,3) = 1 + 2 * t + 3 * t^2$$

$$down(a + b * t^2) = (a, 0, b)$$

MVPKCS: Tactic

$$\begin{tikzcd}
 & & \mathit{apply}\ \ P(X) \arrow[dr, "\mathit{down}"] & & \\
\overline{x}=(x_0,\ldots,x_{n-1}) \arrow[r] & S \arrow[ur, "\mathit{up}"] & & T \arrow[r] & \overline{c}=(c_0,\ldots,c_{n-1})
\end{tikzcd}$$

MVPKCS: Tactic

$$\begin{tikzcd} & & \mathit{apply}\ P(X) \arrow[dr, "\mathit{down}"] & & \\ \bar{x}=(x_0,\dots,x_{n-1}) \arrow[r] & S \arrow[ur, "\mathit{up}"] & & T \arrow[r] & \bar{c}=(c_0,\dots,c_{n-1}) \end{tikzcd}$$

$$ENC(x_0,\dots,x_{n-1}) = T \circ \mathit{down} \circ P \circ \mathit{up} \circ S\ (x_0,\dots,x_{n-1})$$

- hide operation on k^n in a univariate operation on K
- private parameters: P, S,T (only key owner knows them)
- trapdoor: $P(X)=C$ is “easy” whereas $ENC(\bar{x})=\bar{c}$ is “hard”

Example: MI / simple HFE

k = GF(3)
K = GF(3 ^ 3)

$\overline{x} = (1,0,1)$ $\qquad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{S} * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$

Example: MI / simple HFE

k = GF(3)
K = GF(3 ^ 3)

$$up(1,1,0)=1+t$$

$$\bar{x}=(1,0,1) \qquad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{S} * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Example: MI / simple HFE

k = GF(3)
K = GF(3 ^ 3)

$$P(X)=X^2$$
$$P(1+t)=1+2*t+t^2$$

$$up(1,1,0)=1+t$$

$$\bar{x}=(1,0,1) \qquad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{S} * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Example: MI / simple HFE

k = GF(3)
K = GF(3 ^ 3)

$$P(X)=X^2$$
$$P(1+t)=1+2*t+t^2$$

$$up(1,1,0)=1+t$$

$$down(1+2*t+t^2)=(1,2,1)$$

$$\bar{x}=(1,0,1) \quad \underbrace{\begin{pmatrix}1&1&0\\0&1&1\\0&1&0\end{pmatrix}}_{S}*\begin{pmatrix}1\\0\\1\end{pmatrix}=\begin{pmatrix}1\\1\\0\end{pmatrix}$$

$$\underbrace{\begin{pmatrix}0&1&1\\0&1&0\\1&2&1\end{pmatrix}}_{T}*\begin{pmatrix}1\\2\\1\end{pmatrix}=\begin{pmatrix}0\\2\\0\end{pmatrix}$$

$$\rightarrow \bar{c}=(0,2,0)$$

Public key

k = GF(3)
K = GF(3 ^ 3)

$$P(X)=X^2$$

$$P(up(S(\bar{x})))=\ldots$$

$$up(x_1+x_2, x_2+x_3, x_2)= (x_1+x_2)+(x_2+x_3)*t+(x_2)*t^2$$

$$\xrightarrow{\text{down}}$$

$$\bar{x}=(x_1, x_2, x_3) \quad \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{S} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1+x_2 \\ x_2+x_3 \\ x_2 \end{pmatrix} \qquad \underbrace{\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}}_{T} * down(P(up(S(\bar{x}))))=\ldots$$

Public key

k = GF(3)
K = GF(3 ^ 3)

$$P(X)=X^2$$
$$P(up(S(\bar{x})))=\ldots$$

down

$$up(x_1+x_2, x_2+x_3, x_2)=$$
$$(x_1+x_2)+(x_2+x_3)*t+(x_2)*t^2$$

$$\bar{x}=(x_1, x_2, x_3)\ \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{S} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1+x_2 \\ x_2+x_3 \\ x_2 \end{pmatrix} \qquad \underbrace{\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}}_{T} * down(P(up(S(\bar{x}))))=\ldots$$

$$\rightarrow \begin{pmatrix} x_1 x_2 - x_1 x_3 + x_3^2 \\ -x_1 x_2 + 2*x_1 x_3 - x_2^2 \\ x_1^2 - x_1 x_2 + x_1 x_3 + x_2^2 - x_2 x_3 + x_3^2 \end{pmatrix}$$

public key

Public key

k = GF(3)
K = GF(3 ^ 3)

$P(X)=X^2$

$P(up(S(\bar{x})))=\ldots$

down

$$up(x_1+x_2, x_2+x_3, x_2)=(x_1+x_2)+(x_2+x_3)*t+(x_2)*t^2$$

$$\bar{x}=(x_1,x_2,x_3)\quad \underbrace{\begin{pmatrix}1&1&0\\0&1&1\\0&1&0\end{pmatrix}}_{S}*\begin{pmatrix}x_1\\x_2\\x_3\end{pmatrix}=\begin{pmatrix}x_1+x_2\\x_2+x_3\\x_2\end{pmatrix}\qquad \underbrace{\begin{pmatrix}0&1&1\\0&1&0\\1&2&1\end{pmatrix}}_{T}*down(P(up(S(\bar{x}))))=\ldots$$

$$\rightarrow\begin{pmatrix}1*0-1*1+1^2\\-1*0+2*(1*1)-0^2\\1^2-1*0+1*1+0^2-0*1+1^2\end{pmatrix}=\begin{pmatrix}0\\2\\0\end{pmatrix}$$

keys $\leftrightarrow$ P(X)

- public key of HFE and most other MVPKCS:

$$\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} = ENC(x_0, \dots, x_{n-1}) = T \circ down \circ P \circ up \circ S\,(x_0, \dots, x_{n-1})$$

  - always of degree 2

keys ↔ P(X)

- public key of HFE and most other MVPKCS:

$$\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} = ENC(x_0, \ldots, x_{n-1}) = T \circ down \circ P \circ up \circ S\,(x_0, \ldots, x_{n-1})$$

  - always of degree 2, because we choose

$$MI: P(X) = X^{q^{\Theta}+1} \qquad HFE: P(X) = \sum_{l,k=0}^{r} a_{lk}\, X^{q^l+q^k} + \sum_{l=0}^{r} b_l\, X^{q^l}$$

  - why? X^{q^l} is linear → $X^{q^i+q^k}$ is quadratic
  - HFE: **H**idden **F**ield **E**quation → this is P(X)

Decryption

- Decryption: Encryption backwards

$$
\begin{tikzcd}
 & & \textit{solve}\ \ P(X) = up(T^{-1}\bar{c}) \arrow[dl, "down"'] & & \\
\bar{x} = (x_0, \ldots, x_{n-1}) & S^{-1} \arrow[l] & & T^{-1} \arrow[ul, "up"'] & \bar{c} = (c_0, \ldots, c_{n-1}) \arrow[l]
\end{tikzcd}
$$

Decryption

- Decryption: Encryption backwards

$$solve\ \ P(X) = up(T^{-1}\bar{c})$$

down *up*

$$\bar{x} = (x_0, \ldots, x_{n-1}) \leftarrow S^{-1} \qquad\qquad T^{-1} \leftarrow \bar{c} = (c_0, \ldots, c_{n-1})$$

Berlekamp algorithm
complexity: $O(nd^2 + \log(d) + d^3)$ $d = degree(P(X))$

- efficiency depends on degree of P(X)

Decryption

- Decryption: Encryption backwards

$$\begin{tikzcd}
 & & \text{solve } \; P(X)=up(T^{-1}\,\bar{c}) \arrow[dl, "down"'] & & \\
\bar{x}=(x_0,\dots,x_{n-1}) & S^{-1} \arrow[l] & & T^{-1} \arrow[ul, "up"'] & \bar{c}=(c_0,\dots,c_{n-1}) \arrow[l] \arrow[llll, bend left, "??"]
\end{tikzcd}$$

$$\begin{pmatrix} f_1(x_1,\dots,x_3) \\ f_2(x_1,\dots,x_3) \\ f_3(x_1,\dots,x_3) \end{pmatrix} = \begin{pmatrix} x_1 x_2 - x_1 x_3 + x_3^2 \\ -x_1 x_2 + 2 * x_1 x_3 - x_2^2 \\ x_1^2 - x_1 x_2 + x_1 x_3 + x_2^2 - x_2 x_3 + x_3^2 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

Breaking HFE

- recover plaintext from public key, ciphertext
- → solve

$$\begin{pmatrix} f_1(x_1,\ldots,x_n) \\ \vdots \\ f_n(x_1,\ldots,x_n) \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$

- 1965: Buchberger discovered constructive method for computing “Groebner Bases” for ideals
  - GB for $\langle f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n\rangle$ will reveal plaintext

How does it work?

- GB $g_1,\dots,g_r$ in "lex" order yields triangular form:

$$\begin{array}{c} \langle f_1(x_1,\dots,x_n)-c_1 \\ \vdots \\ f_n(x_1,\dots,x_n)-c_n\rangle \end{array} = \begin{array}{r} \langle g_1(x_1,\dots,x_{n-1},x_n) \\ \vdots \\ g_{r-1}(x_{n-1},x_n) \\ g_r(x_n)\rangle \end{array}$$

- solve $g_r(x_n)=c_n$ (easy, since $x_n \in k$ and k is small)
- plug x_n into $g_{r-1}(x_{n-1},x_n)$, solve $g_{r-1}(x_{n-1},x_n)-c_{n-1}$
- ...

How efficient?

- in 2002, Faugère came up with an algorithm (F_4) for computing GBs in a more efficient way
  - in 2002, he was able to break the HFE challenge set by its inventor (k=GF(2), n=80)
  - claim of Faugere/Joux: complexity is $O(n^{\log d})$
  - → HFE insecure?

- important: base field was assumed to be GF(2)!

Our claim

- HFE can defeat F_4 by just setting k=GF(q) where q is 11, 13, ...
- characteristic is very important, because of the

field equations

field equations, multivariate version

- we use them to separate
  - solutions to a polynomial system in some desired field
  - solutions to the same system in the algebraic closure
- in short: for finite field $F = GF(q)$ and polynomials $f_1, \ldots, f_n \in F[x_1, \ldots, x_n]$, we have

$$Variety(f_1, \ldots, f_n) \cap F^n = Variety(f_1, \ldots, f_n, x_1^q - x_1, \ldots, x_n^q - x_n)$$

  - by adding the field equation we get only those solutions in the actual field, not in its algebraic closure

additional solutions

- example: $p(x)=x^2+1$ has no solutions in $\mathbb{R}$ but there are solutions in $\mathbb{C}$
- over a finite field: over $k=GF(3)$, p is still irreducible
  - → we can consider $K=GF(3)[x]/\langle p(x)\rangle$
  - coset $x+\langle p(x)\rangle$ is a root
  - computing a GB without field equations will "somehow" keep the solution $x+\langle p(x)\rangle$

HFE and field equations?

- base field $GF(2)$ → field equations of degree 2 → easy to use in GB computation
- base field $GF(11)$ → field equations of degree 11
  - basic algebra: have to store polynomials having $\binom{n+11}{11}$ different possible terms
  - exceeds 4 GB(!) of memory for n > 32

Breaking HFE

- so we cannot compute a GB for

  $\langle f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n,x_1^q-x_1,\ldots,x_n^q-x_n\rangle$

  because: GB algorithm makes use of $x_i^{11}-x_i$ causing a push to degree 11 → kills him because of size of the polynomials

- what we can do is compute GB for

  $\langle f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n\rangle$

- but then we get addional solutions in the alg. closure

how many extra solutions?

- every root of $P(X)-C$ is also a root of

$f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n$

- since we don't have field equations: P spilts completely → $|V(P(X)-C)|=deg(P(X))$
- P is randomly chosen
  - → solutions are randomly scattered in K (or in extensions)
  - → high chance that they will be different in the last coordinate
  - → final GB must reach degree $deg(P(X))=11^2+11=132$

even better!

- Indeed: not really satisfactory
  - GB uses monomial order that makes use of the degree (eliminates high degree polys first)
    - compute GB in degree respective order
    - use "FGLM" to convert to "lex" order

- Bezout theory sais:

$$|V(f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n)|=2^n$$

exponential growth

- having

  $|V(f_1(x_1,\ldots,x_n)-c_1,\ldots,f_n(x_1,\ldots,x_n)-c_n)|=2^n$

we know that FGLM must fail to compute final basis because its complexity relies on the number of solutions

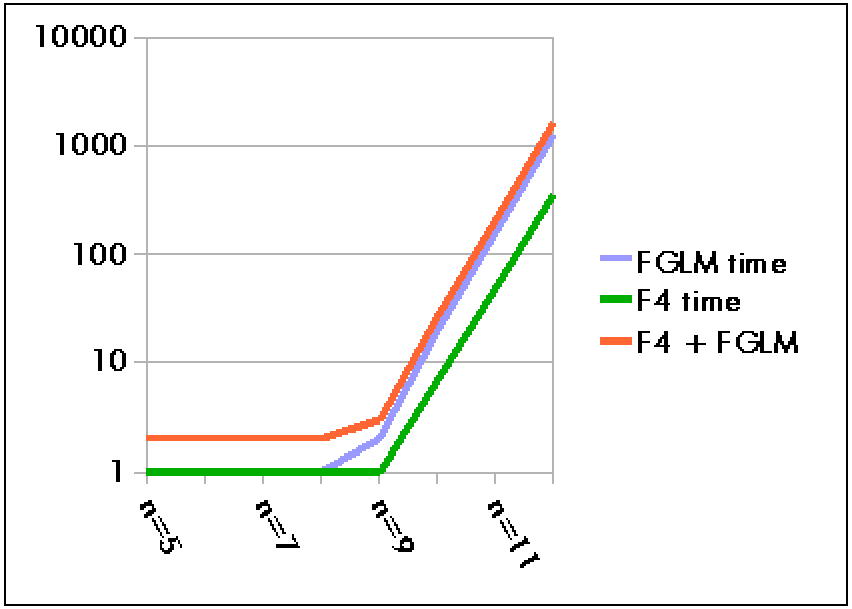
[more precise: relies on the dimension of a quotient = amount of standard monomials but...

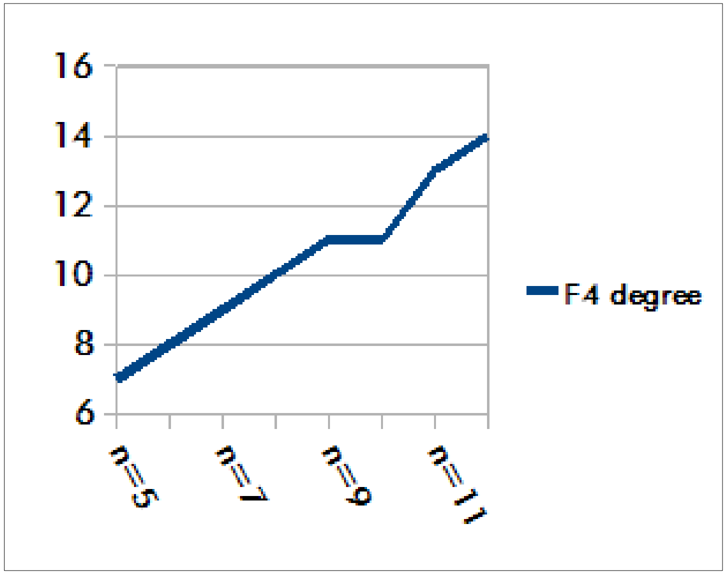
$deg(g_r(x_n))=2^n$

→ there are 2^n standard monomials]

seriously...

– implemented everything in MAGMA and used F_4 to solve HFE systems with $deg(P(X)) = 13^2 + 13$ over $GF(13)$





One last twist

- one last modification
- other reviewer's comment: “if HFE is secure then you do not need this”
  - HFE secure against GB attacks!
  - there is a second attack by Kipnis and Shamir
  - idea: lift up everything to K[X]
  - use bijections between functions on vectorspace and polynomials over the big field

KS: bijections

- bijections between linear function on k and K polynomials:

$$T \leftrightarrow \; p_T(X) = \sum_{i=0}^{n-1} a_i * X^{q^i} \quad \text{(represented by a sequence in } K^n\text{)}$$

- also: quadratic functions and polynomials

$$Q \leftrightarrow \; p_Q(X) = \sum_{j,k=0}^{n-1} a_{j\,k} * X^{q^j + q^k}$$

(represented by a symmetric matrix in $K^{n \times n}$)

KS: attack

- matrix equation: $G(X) = T(P(S(X)))$
- can be rewritten as $T^{-1}(G(X)) = P(S(X))$
- T is a sequence, it can be shown that

$$T^{-1}(G(X)) = t_1 * G^{*1} + \ldots + t_n * G^{*n} =: G'$$

- and that for the correct choices for the t_i, G' has rank r (security bound on degree of P)
- → formulate "G' has rank r" as system of equations

Projection defeats KS

- however, this can be defeated easily by doing a "projection"
- → form the public key as usual, then substitute x_n by a random linear combination of $x_1, \ldots, x_{n-1}$
- KS equation becomes

$$G(X) = \pi(T(P(S(X))))$$

- desired relation can not be established:

$$T^{-1}(G(X)) \neq P(S(X))$$

Thank you for listening!!

- Questions? Remarks?

Thank you for listening!!

- Questions? Remarks?

→ "suggestions for questions"™ :

1) Why not simply increasing the size of the base field? Why increasing the characteristic?
2) Can't we "hardwire" the field equations into the GB algorithm like "whenever you encounter x_i^q , substitute it by x_i" (because then the GB-algorithm would not explicitly make use of them and consequently would not push itself up to degree q)
3) Doesn't the projection destroy the argumentation on the size of the variety? What if desired relation can be derived somehow?