Preliminary Cryptanalysis of the Biscuit Signature Scheme

Charles Bouillaguet, **Julia Sauvage**

Sorbonne Université, CNRS, LIP6

April 11, 2024

Biscuit

Biscuit signature scheme [Bettale et al., 23]

- Submission to the NIST competition for additional post-quantum signatures
- `MPC-in-the-Head`-based Signature
- **Structured** algebraic equations

Biscuit

Biscuit signature scheme [Bettale et al., 23]

- Submission to the NIST competition for additional post-quantum signatures
- `MPC-in-the-Head`-based Signature
- **Structured** algebraic equations

Biscuit polynomial system

Public Key :

- m quadratic polynomials p_i in n variables $(m \approx n)$ over $\mathbb{F}_q$
- $p_i(\mathbf{x}) = u_i(\mathbf{x}) + v_i(\mathbf{x}) \times w_i(\mathbf{x})$
- u_i, v_i and w_i **affine forms**
  $(u_i(\mathbf{x}) = a_0 x_0 + \cdots + a_{n-1} x_{n-1}$ with $a_i \in \mathbb{F}_q)$

Secret Key :

- $\mathbf{s}$ with $p_i(\mathbf{s}) = 0$ for $i \in \{1, \ldots, m\}$

Security of Biscuit Signature Scheme

Attacks

- Key-Recovery: Solving the system (Public Key)
- Forgery: Solving a subsystem + Kales-Zaverucha attack

Security of Biscuit Signature Scheme

Attacks

- Key-Recovery: Solving the system (Public Key)
- Forgery: Solving a subsystem + Kales-Zaverucha attack

Biscuit NIST Specification

- Combinatory algo : $q^{\frac{3}{4}n}$
- Asymptotic complexity Hybrid Method : $2^{2.01n}$

Security of Biscuit Signature Scheme

Attacks

- Key-Recovery: Solving the system (Public Key)
- Forgery: Solving a subsystem + Kales-Zaverucha attack

Biscuit NIST Specification

- Combinatory algo : $q^{\frac{3}{4}n}$
- Asymptotic complexity Hybrid Method : $2^{2.01n}$

New algorithms

- Direct : $n^3 q^{\frac{n}{2}}$
- New hybrid approach: $2^{1.59n}$

Hybrid Method and New Idea

Hybrid method [Bettale et al., 2012]

1. Choose an optimal k.
2. Guess the value of k variables.
3. Groebner basis algorithm on m polynomials and $n - k$ variables.

- Asymptotic complexity known at m/n and q fixed.

Hybrid Method and New Idea

Hybrid method [Bettale et al., 2012]

1. Choose an optimal k.
2. Guess the value of k variables.
3. Groebner basis algorithm on m polynomials and $n - k$ variables.

- Asymptotic complexity known at m/n and q fixed.

New idea for Biscuit-like systems

$$p_i(\mathbf{x}) = u_i(\mathbf{x}) + v_i(\mathbf{x}) \times w_i(\mathbf{x})$$

We guess $v_i(\mathbf{x}) = a \in \mathbb{F}_q$. We have now:

$$\begin{aligned} p_i(\mathbf{x}) &= u_i(\mathbf{x}) + a \times w_i(\mathbf{x}) \\ v_i(\mathbf{x}) &= a \end{aligned}$$

$\hookrightarrow$ $m - 1$ polynomials in $n - 2$ variables.

Attacks

Direct attack algorithm

1. Guess $n/2$ values
2. Get the n linear equations
3. Complexity : $n^3 q^{\frac{n}{2}}$

- Better than the combinatory algorithm ($q^{3/4n}$)

Attacks

Direct attack algorithm

1. Guess $n/2$ values
2. Get the n linear equations
3. Complexity : $n^3 q^{\frac{n}{2}}$

- Better than the combinatory algorithm ($q^{3/4n}$)

Modified Hybrid method

1. Choose an optimal k.
2. Guess k values.
3. Groebner basis algorithm on $m - k$ polynomials and $n - 2k$ variables.

- Asymptotic complexity known at m/n and q fixed.

Security Estimations and Asymptotic Complexity

Asymptotic Complexity in $2^{\alpha n}$

	Classical		New	
q	k/n	α	k/n	α
16	0.182	2.01	0.269	1.59
256	0.049	2.39	0.086	2.24

Security Estimations and Asymptotic Complexity

Asymptotic Complexity in $2^{\alpha n}$

	Classical		New	
q	k/n	α	k/n	α
16	0.182	2.01	0.269	1.59
256	0.049	2.39	0.086	2.24

Estimating time cost

- `MQ-estimator`

  $\hookrightarrow$ Use asymptotic complexity, constants $= 1$
- Exhaustive search on k

Results on Key-Recovery Cost

Key recovery cost for Biscuit (`MQ-estimator v1.1.0`, jan 2023)

Version		Parameters				Classical		New	
	Level	q	n	m	**sec.**	T	k	**T**	k
`v1`	**I**	16	64	67	**160**	151	11	**124**	17
	II		87	90	**210**	201	13	**163**	26
	III		118	121	**276**	266	21	**215**	31
`v2`	**I**	256	50	52	**143**	140	0	**133**	3
	II		89	92	**207**	232	3	**222**	5
	III		127	130	**272**	326	4	**312**	9

Forgery Attack

Forgery

- Kales-Zaverucha forgery attack [Kales et al., 20].

Property for Biscuit Signature Scheme [Bettale et al., 23]

- $\mathbf{s}'$ partial solution for $m - u$ polynomials
- Verifier accepts $\mathbf{s}'$ with proba q^{-u}
- Time cost of the Kales-Zaverucha attack depends on this probability

- We solve a sub-system before the Kales-Zaverucha attack
- **Problem: Choosing the optimal u**

Forgery Attack

Interesting case

If the subsystem is underdetermined $(m - u < n)$:

- $t = n - (m - u)$
- We can freely add t linear dependencies $\hookrightarrow$ We still have a solution (with great probability)

Algorithm in this case

- With $i \in \{1, \ldots, t\}$, we set $v_i(\mathbf{x}) = 0$:
- $p_i = u_i(\mathbf{x}) + v_i(\mathbf{x}) \times w_i(\mathbf{x})$ becomes :

$$u_i(\mathbf{x}) = 0$$
$$v_i(\mathbf{x}) = 0$$

$\hookrightarrow$ We have now $n - 2t$ polynomials in $n - 2t$ variables to solve.

Cost of Forgery

Version			Parameters						KZ attack	
			N	τ	q	n	m	**sec.**	**T**	u
v1	I	short	256	18	16	64	67	**143**	**116**	4
		fast	16	34					**120**	4
	II	short	256	30		87	90	**208**	**162**	3
		fast	16	54					**163**	1
	III	short	256	40		118	121	**274**	**215**	3
		fast	16	73					**215**	0
v2	I	short	256	18	256	50	52	**143**	**131**	4
		fast	32	28					**133**	0
	II	short	256	25		89	92	**207**	**199**	10
		fast	32	40				**210**	**205**	9
	III	short	256	33		127	130	**272**	**265**	16
		fast	32	53				**275**	**271**	14

Thank you !

Generalization ?

LWE with binary error

$A \times s + e = b$ with

$$\begin{pmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ a_{1,0} & \cdots & a_{1,n-1} \\ \vdots & \ddots & \vdots \\ a_{m-1,0} & \cdots & a_{m-1,n-1} \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{m-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{pmatrix}$$

- $s \in \mathbb{F}_q^n$ the secret.
- $e \in \{0, 1\}^m$ an unknown error vector.
- $A \in \mathbb{F}_q^{m \times n}$ and $b \in \mathbb{F}_q^m$ public.

Generalization ?

LWE with binary error

$A \times s + e = b$ with

$$\begin{pmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ a_{1,0} & \cdots & a_{1,n-1} \\ \vdots & \ddots & \vdots \\ a_{m-1,0} & \cdots & a_{m-1,n-1} \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{m-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{pmatrix}$$

- $s \in \mathbb{F}_q^n$ the secret.
- $e \in \{0,1\}^m$ an unknown error vector.
- $A \in \mathbb{F}_q^{m \times n}$ and $b \in \mathbb{F}_q^m$ public.

Linear equations

$\alpha_i(s) = e_i$ with $0 \leq i \leq m-1$

And :

$\alpha_i(x) = a_{i,0}x_0 + \cdots + a_{i,n-1}x_{n-1} - b_i$

Generalization ?

Arora Ge

- Arora Ge: $(\alpha_i(s))(\alpha_i(s) - 1) = 0$
  $\hookrightarrow$ Quadratic polynomial in n variables over $\mathbb{F}_q$.
- Solve with the Hybrid method

Generalization ?

Arora Ge

- Arora Ge: $(\alpha_i(s))(\alpha_i(s) - 1) = 0$
  $\hookrightarrow$ Quadratic polynomial in n variables over $\mathbb{F}_q$.
- Solve with the Hybrid method

Our idea

- Guess an optimal k e_i
  $\hookrightarrow$ **Cost : $\mathbf{2^k}$** (independent of the field)
- solve $\mathbf{m} - \mathbf{k}$ polynomials of $\mathbf{n} - \mathbf{k}$ variables over $\mathbb{F}_q$.

Generalization ?

Arora Ge

- Arora Ge: $(\alpha_i(s))(\alpha_i(s)-1)=0$
  $\hookrightarrow$ Quadratic polynomial in n variables over $\mathbb{F}_q$.
- Solve with the Hybrid method

Our idea

- Guess an optimal k e_i
  $\hookrightarrow$ **Cost : $\mathbf{2^k}$** (independent of the field)
- solve $\mathbf{m}-\mathbf{k}$ polynomials of $\mathbf{n}-\mathbf{k}$ variables over $\mathbb{F}_q$.

Interest

- Little improvement of the classical Arora-Ge algorithm
- Exhaustive comparison with lattice-based algorithms needed