An efficient structural attack on NIST submission DAGS

Élise Barelli\textsuperscript{1} and Alain Couvreur\textsuperscript{2,3}

\textsuperscript{1}Université de Versailles Saint Quentin
\textsuperscript{2}INRIA
\textsuperscript{3}LIX, École polytechnique

Asiacrypt 2018
**Context**

- **DAGS** is a proposal to NIST call for post quantum cryptography.
- McEliece-like public key encryption scheme (+ conversion to a KEM).
- Based on quasi–dyadic alternant codes.
- Original parameters:

<table>
<thead>
<tr>
<th>Security</th>
<th>$n$</th>
<th>$\dim \mathcal{C}_{pub}$</th>
<th>Ground field</th>
<th>$\mathcal{G}$</th>
<th>Key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>832</td>
<td>416</td>
<td>$\mathbb{F}_{32}$</td>
<td>$(\mathbb{Z}/2\mathbb{Z})^4$</td>
<td>6.8 kB</td>
</tr>
<tr>
<td>192</td>
<td>1216</td>
<td>512</td>
<td>$\mathbb{F}_{64}$</td>
<td>$(\mathbb{Z}/2\mathbb{Z})^5$</td>
<td>8.5 kB</td>
</tr>
<tr>
<td>256</td>
<td>2112</td>
<td>704</td>
<td>$\mathbb{F}_{64}$</td>
<td>$(\mathbb{Z}/2\mathbb{Z})^6$</td>
<td>11.6 kB</td>
</tr>
</tbody>
</table>

**Note.** Parameters have been updated (see further).
1 Prerequisites

2 Description of the attack

3 Complexity and implementation
(Generalised) Reed–Solomon codes

Definition 1 (Reed–Solomon codes)

Let $n, k$ be positive integers $k \leq n$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with distinct entries

$$\text{RS}_k(\mathbf{x}) \overset{\text{def}}{=} \{(f(x_1), \ldots, f(x_n)) \mid \deg(f) < k\}.$$
(Generalised) Reed–Solomon codes

Definition 1 (Reed–Solomon codes)
Let $n, k$ be positive integers $k \leq n$. Let $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with distinct entries

$$\text{RS}_k(x) \overset{\text{def}}{=} \{(f(x_1), \ldots, f(x_n)) \mid \deg(f) < k\}.$$ 

Definition 2 (Generalised Reed–Solomon codes)
Let $n, k$ be positive integers $k \leq n$. Let $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with distinct entries and $y = (y_1, \ldots, y_n) \in (\mathbb{F}_q^\times)^n$.

$$\text{GRS}_k(x, y) \overset{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid \deg(f) < k\}.$$
(Generalised) Reed–Solomon codes

**Definition 1 (Reed–Solomon codes)**

Let $n, k$ be positive integers $k \leq n$. Let $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with distinct entries

$$RS_k(x) \overset{\text{def}}{=} \{(f(x_1), \ldots, f(x_n)) \mid \deg(f) < k\}.$$ 

**Definition 2 (Generalised Reed–Solomon codes)**

Let $n, k$ be positive integers $k \leq n$. Let $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with distinct entries and $y = (y_1, \ldots, y_n) \in (\mathbb{F}_q \times)^n$.

$$GRS_k(x, y) \overset{\text{def}}{=} \{(y_1 f(x_1), \ldots, y_n f(x_n)) \mid \deg(f) < k\}.$$ 

**Claim.** For such codes one can correct up to $\frac{n-k}{2}$ errors in polynomial time.
### Alternant codes

**Definition 3 (Alternant codes)**

Let $n, k$ be positive integers $k \leq n$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ be a vector with distinct entries and $\mathbf{y} = (y_1, \ldots, y_n) \in (\mathbb{F}_{q^m}^\times)^n$. An alternant code is a code of the form

$$\text{GRS}_r(\mathbf{x}, \mathbf{y}) \cap \mathbb{F}_q^n.$$
**Alternant codes**

**Definition 3 (Alternant codes)**

Let $n, k$ be positive integers $k \leq n$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ be a vector with distinct entries and $\mathbf{y} = (y_1, \ldots, y_n) \in (\mathbb{F}_{q^m})^n$. An alternant code is a code of the form

$$\text{GRS}_r(\mathbf{x}, \mathbf{y}) \cap \mathbb{F}_{q}^n.$$
Alternant codes

Definition 3 (Alternant codes)

Let $n, k$ be positive integers $k \leq n$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n_{q^m}$ be a vector with distinct entries and $\mathbf{y} = (y_1, \ldots, y_n) \in (\mathbb{F}^\times_{q^m})^n$. An alternant code is a code of the form

$$\text{GRS}_r(\mathbf{x}, \mathbf{y}) \cap \mathbb{F}^n_q.$$ 

Fact 1. Alternant codes inherit from generalised Reed–Solomon decoding algorithms.
## Alternant codes

### Definition 3 (Alternant codes)

Let $n, k$ be positive integers $k \leq n$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ be a vector with distinct entries and $\mathbf{y} = (y_1, \ldots, y_n) \in (\mathbb{F}_{q^m}^\times)^n$. An alternant code is a code of the form

$$\text{GRS}_r(\mathbf{x}, \mathbf{y}) \cap \mathbb{F}_q^n.$$ 

**Fact 1.** Alternant codes inherit from generalised Reed–Solomon decoding algorithms.

**Fact 2.** Their parameters are not as good as GRS codes, but they are much less structured which is interesting for cryptography.
History – McEliece (1978)

- 1978: McEliece’s original proposal based on binary Goppa codes (special case of alternant codes). Public key: 32kB for ≈ 80 bits of security\(^1\).

\(^1\)With respect to Prange algorithm
1978 : McEliece’s original proposal based on binary Goppa codes (special case of alternant codes). Public key : 32kB for $\approx 80$ bits of security\textsuperscript{1}.


During these 40 years many attempts to get shorter keys.

\textsuperscript{1}With respect to Prange algorithm
History – McEliece (1978)

- 1978: McEliece’s original proposal based on binary Goppa codes (special case of alternant codes). Public key: 32kB for \( \approx 80 \) bits of security\(^1\).

During these 40 years many attempts to get shorter keys. How?

\(^1\)With respect to Prange algorithm
Idea 1: Reducing the extension degree

The larger the $m$ the worse the parameters. But:

Fact. The larger the $m$ the worse the parameters. But:
Idea 1: Reducing the extension degree

\[ F_{q^m} \supset GRS_k(x, y) \cap F_q^n \]

**Fact.** The larger the \( m \) the worse the parameters. But:

- Case \( m = 1 \) is broken (Sidelnikov, Shestakov 1992);
- Some specific cases of \( m = 2 \) and 3 called *wild Goppa codes* are broken too:
  - C., Otmani, Tillich, 2014;
  - Faugère, Perret, de Portzamparc, 2014
Idea 2: Using codes with a non trivial automorphism group

- **Advantage.** Permits to reduce the public key size with almost no incidence on the security
Idea 2: Using codes with a non trivial automorphism group

- **Advantage.** Permits to reduce the public key size with almost no incidence on the security \textit{w.r.t. message security attacks}. 

Some tempting choices of using large groups lead to key recovery attacks: Otmani, Tillich, Dallot (2008); Faugère, Otmani, Perret, Tillich (2010); Faugère, Otmani, Perret, Tillich, de Portzamparc (2016).
Idea 2: Using codes with a non trivial automorphism group

- **Advantage.** Permits to reduce the public key size with almost no incidence on the security w.r.t. message security attacks.
- **But, may affect the security w.r.t. key recovery attacks.**
Idea 2: Using codes with a non trivial automorphism group

- **Advantage.** Permits to reduce the public key size with almost no incidence on the security **w.r.t. message security attacks.**
- **But, may affect the security w.r.t. key recovery attacks.**

Some tempting choices of using large groups lead to key recovery attacks:

- Otmani, Tillich, Dallot (2008);
- Faugère, Otmani, Perret, Tillich (2010);
- Faugère, Otmani, Perret, Tillich, de Portzamparc (2016).
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. $\text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. $\text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes, i.e. GRS$_k(x, y) \cap \mathbb{F}_q^n$ with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. \( \text{GRS}_k(x, y) \cap \mathbb{F}_q^n \) with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. $\text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. GRSp(x, y) ∩ F^n_q with an automorphism group acting as:
DAGS scheme’s public keys are Quasi–dyadic alternant codes. i.e. \( \text{GRS}_k(x, y) \cap \mathbb{F}_q^n \) with an automorphism group acting as:
In short: automorphism group $G$ is $\cong (\mathbb{Z}/2\mathbb{Z})^\gamma$ for some $\gamma > 0$.

- **Public key.** An $\mathbb{F}_q[G]$-basis of $\text{GRS}_k(x, y) \cap \mathbb{F}_q^n$;
- **Secret Key.** The pair $(x, y)$.

**Important.** The extension degree $m$ is 2.

$$\begin{align*}
\mathbb{F}_q^2 & \quad \text{GRS}_k(x, y) \subseteq \mathbb{F}_q^n \\
\mathbb{F}_q & \quad \text{GRS}_k(x, y) \cap \mathbb{F}_q^n
\end{align*}$$
Section 2

Description of the attack
Tool 1: the conductor

In $\mathbb{F}_q^n$ we denote by $\star$ the component wise product:

$$u \star v \overset{\text{def}}{=} (u_1 v_1, \ldots, u_n v_n).$$

Then, the star product of two codes $A, B \subseteq \mathbb{F}_q^n$:

$$A \star B \overset{\text{def}}{=} \text{Span}\{a \star b \mid a \in A, \ b \in B\}$$

**Definition 4**

Let $\mathcal{U}, \mathcal{V} \subseteq \mathbb{F}_q^n$ be two codes:

$$\text{Cond}(\mathcal{U}, \mathcal{V}) = \{x \in \mathbb{F}_q^n \mid x \star \mathcal{U} \subseteq \mathcal{V}\}$$

**Remark**

*Equivalently, the conductor is the largest code $\mathcal{X}$ satisfying $\mathcal{X} \star \mathcal{U} \subseteq \mathcal{V}$.***
Why are conductors good for?

Illustrative example.
- Suppose the public key is $\text{GRS}_k(x, y)$
- Suppose we obtained $\text{GRS}_{k-1}(x, y)$ (for instance by brute force search)

Lemma 5

$$\text{Cond}(\text{GRS}_{k-1}(x, y), \text{GRS}_k(x, y)) = \text{RS}_2(x) = \text{Span}\{1, x\}.$$ 

Idea of the proof.

The largest space of polynomials $S$ such that

$$S \cdot \mathbb{F}_q[X]_{<k-1} \subseteq \mathbb{F}_q[X]_{<k}$$

is $\mathbb{F}_q[X]_{<2} = \text{Span}\{1, X\}$. 
Why are conductors good for?

Illustrative example.

- Suppose the public key is $\text{GRS}_k(x, y)$
- Suppose we obtained $\text{GRS}_{k-1}(x, y)$ (for instance by brute force search)

**Lemma 6**

\[
\text{Cond}(\text{GRS}_{k-1}(x, y), \text{GRS}_k(x, y)) = \text{RS}_2(x) = \text{Span}\{1, x\}.
\]

Fundamental fact: the result does not depend on $y$!
With alternant codes, things become harder...

Lemma 7

\[
\text{Cond}(\text{GRS}_{k-1}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_2(x) \cap \mathbb{F}_q^n.
\]
With alternant codes, things become harder...

Lemma 7

$$\text{Cond}(\text{GRS}_{k-1}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_2(x) \cap \mathbb{F}_q^n.$$ 

- Good news: typically equality holds.
With alternant codes, things become harder...

Lemma 7

\[
\text{Cond}(\text{GRS}_{k-1}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_2(x) \cap \mathbb{F}_q^n.
\]

- Good news: typically equality holds.
- Bad news: typically \( \text{RS}_2(x) \cap \mathbb{F}_q^n = \text{Span}\{(1, \ldots, 1)\} \).
With alternant codes, things become harder...

One has to increase the gap between the degrees.

**Lemma 8**

*For any* $0 \leq a < k$,

\[
\text{Cond}(\text{GRS}_{k-a}(x, y), \text{GRS}_k(x, y)) = \text{RS}_{a+1}(x).
\]
With alternant codes, things become harder...

One has to increase the gap between the degrees.

**Lemma 8**

*For any* \(0 \leq a < k\),

\[
\text{Cond}(\text{GRS}_{k-a}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_{a+1}(x) \cap \mathbb{F}_q^n.
\]
With alternant codes, things become harder...

One has to increase the gap between the degrees.

**Lemma 8**

*For any* $0 \leq a < k$,

\[
\text{Cond}(\text{GRS}_{k-a}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_{a+1}(x) \cap \mathbb{F}_q^n.
\]

**Idea.** Choose $a$ so that $\text{RS}_{a+1}(x) \cap \mathbb{F}_q^n \neq \text{Span}\{(1, \ldots, 1)\}$. 
With alternant codes, things become harder...

One has to increase the gap between the degrees.

**Lemma 8**

For any $0 \leq a < k$,

$$\text{Cond}(\text{GRS}_{k-a}(x, y) \cap \mathbb{F}_q^n, \text{GRS}_k(x, y) \cap \mathbb{F}_q^n) \supseteq \text{RS}_{a+1}(x) \cap \mathbb{F}_q^n.$$ 

**Idea.** Choose $a$ so that $\text{RS}_{a+1}(x) \cap \mathbb{F}_q^n \neq \text{Span}\{(1, \ldots, 1)\}$.

For instance $\text{RS}_{q+1}(x) \cap \mathbb{F}_q^n$ contains $x^q + x$ (image of $x$ by $\text{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_q}$).
(Very Naive attack)

Recall that $\mathcal{C}_{\text{pub}} = \text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ and $m = 2$. We look for $\text{GRS}_{k-q}(x, y) \cap \mathbb{F}_q^n$

- For any $\mathcal{D} \subseteq \mathcal{C}_{\text{pub}} \cap \mathbb{F}_q^n$ of codimension $2q$, compute $\text{Cond}(\mathcal{D}, \mathcal{C}_{\text{pub}})$.
- If the conductor $\neq \text{Span}\{(1, \ldots, 1)\}$, you probably found $\text{RS}_{q+1}(x) \cap \mathbb{F}_q^n$. Deducing $x$ from this code is rather easy.
Description of the attack

(Very Naive attack)

Recall that $\mathcal{C}_{\text{pub}} = \text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ and $m = 2$.

We look for $\text{GRS}_{k-q}(x, y) \cap \mathbb{F}_q^n$

- For any $\mathcal{D} \subseteq \mathcal{C}_{\text{pub}} \cap \mathbb{F}_q^n$ of codimension $2q$, compute $\text{Cond}(\mathcal{D}, \mathcal{C}_{\text{pub}})$.
- If the conductor $\neq \text{Span}\{(1, \ldots, 1)\}$, you probably found $\text{RS}_{q+1}(x) \cap \mathbb{F}_q^n$. Deducing $x$ from this code is rather easy.

$\implies$ Cost $\tilde{O}(q^{2q \cdot (\dim \mathcal{C}_{\text{pub}} - 2q)})$. e.g. For DAGS$_1$ : $> 2^{112640}$ operations.
(Very Naive attack)

Recall that $\mathcal{C}_{\text{pub}} = \text{GRS}_k(x, y) \cap \mathbb{F}_q^n$ and $m = 2$. We look for $\text{GRS}_{k-q}(x, y) \cap \mathbb{F}_q^n$

- For any $\mathcal{D} \subseteq \mathcal{C}_{\text{pub}} \cap \mathbb{F}_q^n$ of codimension $2q$, compute $\text{Cond}(\mathcal{D}, \mathcal{C}_{\text{pub}})$.
- If the conductor $\neq \text{Span}\{(1, \ldots, 1)\}$, you probably found $\text{RS}_{q+1}(x) \cap \mathbb{F}_q^n$. Deducing $x$ from this code is rather easy.

$\rightarrow$ Cost $\tilde{O}(q^{2q \cdot (\dim \mathcal{C}_{\text{pub}} - 2q)})$. e.g. For DAGS\_1: $> 2^{112640}$ operations.
$\rightarrow$ Up to now we never used the automorphism group.
Description of the attack

Tool 2: The invariant code

Consider the code

\[ \mathcal{C}_{pub}^G \overset{\text{def}}{=} \{ \mathbf{c} \in \mathcal{C}_{pub} \mid \forall \sigma \in \mathcal{G}, \, \sigma(\mathbf{c}) = \mathbf{c} \}. \]

Theorem 9 (Proved under some heuristic)

\[ \text{Cond}((\text{GRS}_{k-q}(\mathbf{x}, \mathbf{y}) \cap \mathbb{F}_q^n)^G, \mathcal{C}_{pub}) = \text{RS}_{q+2}(\mathbf{x}) \cap \mathbb{F}_q^n. \]
Tool 2 : The invariant code

Consider the code

\[ C_{pub}^G \overset{\text{def}}{=} \{ c \in C_{pub} \mid \forall \sigma \in G, \sigma(c) = c \}. \]

Theorem 9 (Proved under some heuristic)

\[ \text{Cond}( (\text{GRS}_{k-q}(x,y) \cap \mathbb{F}_q^n)^G, C_{pub} ) = \text{RS}_{q+2}(x) \cap \mathbb{F}_q^n. \]
**Description of the attack**

**Tool 2 : The invariant code**

Consider the code

\[ C_{pub}^G \triangleq \{ c \in C_{pub} \mid \forall \sigma \in G, \sigma(c) = c \}. \]

**Theorem 9 (Proved under some heuristic)**

\[ \text{Cond}\left( (\text{GRS}_{k-q}(x, y) \cap \mathbb{F}_q^n)^G, C_{pub} \right) = \text{RS}_{q+2}(x) \cap \mathbb{F}_q^n. \]

→ Enumerate \( D \subseteq C_{pub}^G \) of codimension \( \frac{2q}{|G|} \).

→ Cost \( \tilde{O}(q^{2q} \frac{\dim C_{pub} - 2q}{|G|}) \).
Description of the attack

Tool 2 : The invariant code

Consider the code

\[ C_{\text{pub}}^G \overset{\text{def}}{=} \{ c \in C_{\text{pub}} \mid \forall \sigma \in G, \; \sigma(c) = c \}. \]

Theorem 9 (Proved under some heuristic)

\[
\text{Cond}\left((\text{GRS}_{k-q}(x, y) \cap \mathbb{F}_q^n)^G, C_{\text{pub}}\right) = \text{RS}_{q+2}(x) \cap \mathbb{F}_q^n.
\]

\[ \rightarrow \text{Enumerate } D \subseteq C_{\text{pub}}^G \text{ of codimension } \frac{2q}{|G|}. \]

\[ \rightarrow \text{Cost } \tilde{O}(q^{\frac{4q}{|G|}} \cdot \frac{\dim C_{\text{pub}} - 2q}{|G|}). \]

\[ \rightarrow \text{Next, using some classical coding theoretic operations (shortening) we can reduce the cost to } \tilde{O}(q^{\frac{4q}{|G|}}). \]
Section 3

Complexity and implementation
In practice

The average work factor will be:

|                | Claimed security | $q$  | $|G|$ | Work factor |
|----------------|------------------|------|------|-------------|
| DAGS_1         | 128 bits         | $2^5$| $2^4$| $2^{70}$    |
| DAGS_3         | 192 bits         | $2^6$| $2^5$| $2^{80}$    |
| DAGS_5         | 256 bits         | $2^6$| $2^6$| $2^{58}$    |
Second approach using polynomial system solving

Brute force search can be replaced by the resolution of a system of polynomial equations of degree 2.

Note. Magma implementation on personal computer.
Second approach using polynomial system solving

Brute force search can be replaced by the resolution of a system of polynomial equations of degree 2.

|      | Claimed security | $q$  | $|G|$ | 1st approach | 2nd approach |
|------|------------------|------|------|--------------|--------------|
| DAGS_1 | 128 bits | $2^5$ | $2^4$ | $2^{70}$ | $\approx 20\text{mn}$ |
| DAGS_3 | 192 bits | $2^6$ | $2^5$ | $2^{80}$ | - |
| DAGS_5 | 256 bits | $2^6$ | $2^6$ | $2^{58}$ | $< 1\text{mn}$ |

Note. Magma implementation on personal computer.
Second approach using polynomial system solving

Brute force search can be replaced by the resolution of a system of polynomial equations of degree 2.

|                | Claimed security | $q$  | $|G|$ | 1st approach Work factor | 2nd approach Running times |
|----------------|------------------|------|------|--------------------------|---------------------------|
| DAGS_1         | 128 bits         | $2^5$| $2^4$| $2^{70}$                 | $\approx$ 20mn           |
| DAGS_3         | 192 bits         | $2^6$| $2^5$| $2^{80}$                 | -                         |
| DAGS_5         | 256 bits         | $2^6$| $2^6$| $2^{58}$                 | $< 1$mn                   |

Note. Magma implementation on personal computer.

Note 1. DAGS authors changed their proposal to be out of reach of the first version of the attack (see DAGS’ website).
Second approach using polynomial system solving

Brute force search can be replaced by the resolution of a system of polynomial equations of degree 2.

|                | Claimed security | $q$ | $|G|$ | 1st approach Work factor | 2nd approach Running times |
|----------------|------------------|-----|------|--------------------------|-----------------------------|
| DAGS_1         | 128 bits         | $2^5$ | $2^4$ | $2^{70}$                  | $\approx 20$mn              |
| DAGS_3         | 192 bits         | $2^6$ | $2^5$ | $2^{80}$                  | -                           |
| DAGS_5         | 256 bits         | $2^6$ | $2^6$ | $2^{58}$                  | $< 1$mn                     |

Note. Magma implementation on personal computer.

Note 1. DAGS authors changed their proposal to be out of reach of the first version of the attack (see DAGS’ website).

Note 2. Bardet, Bertin and Otmani, are currently working on improving the 2nd version. They are able to break original DAGS_3 in $< 20$mn.
Questions?