Security Analysis of SDiTH

Kévin Carrier, Jean-Pierre Tillich, Valerian Hatey

Security Analysis of SDiTH

Kévin CARRIER, Jean-Pierre TILLICH and Valerian HATEY

CY Cergy Paris Université
ENSEA (École Nationale Supérieure de l'Électronique et de ses Applications)

20 novembre 2023

1. Security Analysis in NIST specification
2. Correction of the analysis
   - First problem
   - Second problem
   - Third problem
3. Improved attack with projective space

Syndrom Decoding problem

Let us recall the Syndrome Decoding problem

Problem (Syndrome Decoding SD($\mathbf{H}, \mathbf{s}, t$))

Given a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, *a syndrome* $\mathbf{s} \in \mathbb{F}_q^{n-k}$ *and a distance* $t \in [\![0, n]\!]$, *one wants to find a vector* $\mathbf{e} \in \mathbb{F}_q^n$ *such that* $\mathbf{He} = \mathbf{s}$ *and* $|\mathbf{e}| = t$.

SDITH security

Based on d-split syndrom decoding problem (a key recovery attack on SDiTH is as hard as solving the d-split Syndrome Decoding problem)

Problem (d-split Syndrome Decoding SD$(d, \mathbf{H}, \mathbf{s}, t)$)

Given a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$ and a distance $t \in [\![0, n]\!]$, one wants to find a vector $\mathbf{e} = (\mathbf{e}_1, \ldots, \mathbf{e}_d) \in (\mathbb{F}_q^{n/d})^d$ such that $\mathbf{He} = \mathbf{s}$ and $\forall i \in [\![1, d]\!]$, $|\mathbf{e}_i| = t/d$.

SDiTH framework

Solve the SD$(d, \mathbf{H}, \mathbf{s}, t)$ problem with :

- a random $\mathbf{H}$
- $\mathbf{s}$ from an injected solution ($\mathbf{s} = \mathbf{He}$ such that $|\mathbf{e}_i| = t/d$)

Reduce SD(d) to $SD(1)$?

In the specification, they use **[Feneuil,Joux,Rivain,2022]** results to obtain the following bound

Reduction from SD(d) to SD(1)

$$T_{\mathbf{SD}(d)} \geqslant \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}} T_{\mathbf{SD}(1)}$$

Idea :

- Consider an algorithm $\mathcal{A}$ which solves SD(d)
- Let $(\mathbf{H}, \mathbf{s}, t)$ be an instance of SD(1)
- Run $\mathcal{A}$ on $(\mathbf{H}_\sigma, \mathbf{s}_\sigma, t)$ with a random permutation at each step
- There is a probability $\frac{\binom{n/d}{t/d}^d}{\binom{n}{t}}$ of obtaining an instance of SD(d) at each step.

Problem

This bound is tight only when $t \to 0$ (we will see why)

How they analyze security

What algorithm used to solve SD(1) ?

1. ISDs (Information Set Decoding) as usual
2. In the case of SDiTH the field size is big $q = 256$ or $q = 251$
3. They claim that it is enough to stop at Stern's algorithm **[Stern,1989]**
4. Asymptotically with n and q **[May,Meurer,Thomae,2011]** and **[Becker,Joux,May,Meurer,2012]** are equivalent to **[Prange,1962]** (**[Canto Torres,2016]**)
5. But non asymptotically ?

Security assessment in the specification

- Based on Stern algorithm
- They perform non-asymptotic analysis to gain security bits
- They use Peters' tricks **[Peters,2010]** to optimize the non-asymptotic complexity of Stern

Stern algorithm

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$ and $t \in [\![0, n]\!]$
Output : $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{He} = \mathbf{s}$ and $|\mathbf{e}| = t$
Parameters : $p \in \left[\!\left[0, \frac{\min(t,k)}{2}\right]\!\right]$ and $\ell \in [\![0, n-k-t+2p]\!]$

Repeat as many time as necessary :

1. draw $I \subseteq [\![1, n]\!]$ of size k uniformly at random
2. $J \leftarrow [\![1, n]\!] \setminus I$
3. $\mathbf{P} \leftarrow \mathbf{H}_J^{-1}\mathbf{H}_I$
4. $\mathbf{y} \leftarrow \mathbf{H}_J^{-1}\mathbf{s}$
5. $\mathbf{P}' \leftarrow$ the ℓ first rows of $\mathbf{P}$ and $\mathbf{y}' \leftarrow$ the ℓ first positions of $\mathbf{s}$
6. Bet that :
   - $\mathbf{e}_I = (\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{F}_q^{k/2}$ such that $|\mathbf{x}_1| = |\mathbf{x}_2| = p$
   - $\mathbf{e}_J = (\mathbf{0}_l, \mathbf{x}_3) \in \mathbb{F}_q^l \times \mathbb{F}_q^{n-k-l}$ such that $|\mathbf{x}_3| = t - 2p$

How to find $\mathbf{e}$ with betting ?

With the bet, we have :

$$\begin{pmatrix} \mathbf{P}'\mathbf{x}_2 \\ \mathbf{x}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{y}' - \mathbf{P}'\mathbf{x}_1 \\ \mathbf{y}'' - \mathbf{P}''(\mathbf{x}_1 + \mathbf{x}_2) \end{pmatrix}$$

7. $\mathcal{L}_1 \leftarrow \left\{ \mathbf{y}' - \mathbf{P}'\mathbf{x}_1 \ : \ \mathbf{x}_1 \in \mathbb{F}_q^{\lfloor k/2 \rfloor} \times 0^{k - \lfloor k/2 \rfloor} \text{ and } |\mathbf{x}_1| = p \right\}$
   $\mathcal{L}_2 \leftarrow \left\{ \mathbf{P}'\mathbf{x}_2 \ : \ \mathbf{x}_2 \in 0^{\lfloor k/2 \rfloor} \times \mathbb{F}_q^{k - \lfloor k/2 \rfloor} \text{ and } |\mathbf{x}_2| = p \right\}$
8. For all $(\mathbf{y}' - \mathbf{P}'\mathbf{x}_1, \mathbf{P}'\mathbf{x}_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ such that $\mathbf{y}' - \mathbf{P}'\mathbf{x}_1 = \mathbf{P}'\mathbf{x}_2$
   - If $|\mathbf{P}(\mathbf{x}_1 + \mathbf{x}_2) - \mathbf{y}| = t - 2p$, return $\mathbf{e}$, such that
     - $\mathbf{e}_I \stackrel{\text{def}}{=} \mathbf{x}_1 + \mathbf{x}_2$
     - $\mathbf{e}_J \stackrel{\text{def}}{=} \mathbf{y} - \mathbf{P}(\mathbf{x}_1 + \mathbf{x}_2)$

Stern complexity

$$T_{\mathsf{SD}(1)} = \left(\frac{T_{\mathrm{Gauss}} + T_{\mathrm{lists}} + T_{\mathrm{checks}}}{\left(1 - (1 - p_{\mathbf{e}})^{N_{\mathrm{sol}}}\right)} \right) log_2(q)$$

1. T_{Gauss} : complexity of Gaussian elimination
2. T_{lists} : complexity of constructing lists
3. T_{checks} : complexity of collision checking
4. $p_{\mathbf{e}}$: probability of finding a particular solution
5. N_{sol} : expected number of solutions to the problem

Peters trick to construct $\mathcal{L}_1$ and $\mathcal{L}_2$

1. $\mathcal{L}_1 = \left\{ \mathbf{P}'\mathbf{x}_1 - \mathbf{y}' \ : \ \mathbf{x}_1 \in \mathbb{F}_q^{\lfloor k/2 \rfloor} \text{ and } |\mathbf{x}_1| = p \right\}$ and
   $\mathcal{L}_2 = \left\{ \mathbf{P}'\mathbf{x}_2 \ : \ \mathbf{x}_2 \in \mathbb{F}_q^{k/2-\lfloor k/2 \rfloor} \times \text{ and } |\mathbf{x}_2| = p \right\}$
2. Choose an enumeration order σ on $\left\{ \mathbf{x} \in \mathbb{F}_q^{\lfloor k/2 \rfloor} \ : \ |\mathbf{x}| = p \right\}$ such that $\mathbf{x}_{\sigma(i)}$ and $\mathbf{x}_{\sigma(i+1)}$ differ by 1 on at most two successive coordinates depending on the case :
   1. **On the same support** : $\mathbf{x}_{\sigma(i)} = (\dots, d, \dots, 0) \Rightarrow \mathbf{x}_{\sigma(i+1)} = (\dots, d+1, \dots, 0) \Rightarrow$ One coordinate
   2. **Change of support** : $\mathbf{x}_{\sigma(i)} = (\dots, q-1, 0, \dots) \Rightarrow \mathbf{x}_{\sigma(i+1)} = (\dots, 0, 1 \dots, 0) \Rightarrow$ Two successive coordinates
3. Precompute the sums of two successive columns of $\mathbf{P}'$
4. Construct an element of the list $\Rightarrow$ one addition of one column of size ℓ (except the first which requires $p-1$ or p addition of columns depending of the list)

$$T_{\text{lists}} = \ell k + \ell(2p-1) + \ell\left(|\mathcal{L}_1| + |\mathcal{L}_2|\right) \tag{1.1}$$

Second and third tricks

The second trick comes into collision checking

$$|\mathbf{P}(\mathbf{x}_1 - \mathbf{x}_2) - \mathbf{y}| = t - 2p \tag{1.2}$$

- Verify that the Hamming weight of a certain vector of length $n - k - \ell$ equals $t - 2p$
- Remarks that the vector is actually a random vector when it does not come from a wanted solution
- On the wrong candidates, check the Hamming weight over $\frac{q}{q-1}(t - 2p + 1)$ positions on average
- Checking on each coordinate for each position : $2p$ additions, $2p\frac{q-2}{q-1}$ multiplications

$$T_{\text{checks}} = \frac{q}{q-1}(t - 2p + 1)2p\left(1 + \frac{q-2}{q-1}\right)\frac{|\mathcal{L}_1|\,|\mathcal{L}_2|}{q^\ell}$$

Third trick

The third trick concerns Gaussian elimination but they neglect it because they claim that its cost is negligible

Their complexity result

Let's summarize their results :

Complexité

- $L_1 := |\mathcal{L}_1| = \binom{\lfloor \frac{k}{2} \rfloor}{p}(q-1)^p$
- $L_2 := |\mathcal{L}_2| = \binom{k-\lfloor \frac{k}{2} \rfloor}{p}(q-1)^p$
- $T_{\text{Gauss}} = \frac{1}{2}(n-k)^2(n+k)$
- $T_{\text{lists}} = \ell k + \ell(2p-1) + \ell\left(|\mathcal{L}_1| + |\mathcal{L}_2|\right)$
- $T_{\text{checks}} = \frac{q}{q-1}(t-2p+1)2p\left(1+\frac{q-2}{q-1}\right)\frac{|\mathcal{L}_1|\,|\mathcal{L}_2|}{q^\ell}$
- $p_{\mathbf{e}} = \dfrac{L_1 L_2 \binom{n-k-\ell}{t-2p}}{\binom{n}{t}(q-1)^{2p}}$
- $N_{\text{sol}} = 1$

Practical results

Table – SDiTH security in the specification

Parameter Sets	Category	Security Specification				SD Parameters				
		p	l	$T_{SD(1)} \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}}$	target	q	n	k	t	d
SDitH-L1-gf256	I	1	2	143.46	143	256	230	126	79	1
SDitH-L1-gf251	I	1	2	143.45	143	251	230	126	79	1
SDitH-L3-gf256	III	2	5	207.67	207	256	352	193	120	2
SDitH-L3-gf251	III	2	5	207.61	207	251	352	193	120	2
SDitH-L5-gf256	V	2	5	272.35	272	256	480	278	150	2
SDitH-L5-gf251	V	2	5	272.29	272	251	480	278	150	2

The choice of a non-asymptotic analysis

Advantage

Bit gain on security

Inconvenience

- Very accurate calculations $\Rightarrow$ easy to make mistakes
- We are exposed to more attacks which bring non-asymptotic gains

1 Security Analysis in NIST specification

2 Correction of the analysis
- First problem
- Second problem
- Third problem

3 Improved attack with projective space

They do not take into account the N_{sol} factor

Definition

- Let $N_{\mathrm{sol}}(d)$ denote the expected number of solutions of $\mathrm{SD}(d)$
- $d_{\mathsf{GV}}(n,k) \stackrel{\text{def}}{=} \inf\left(\left\{t \in \left[\!\left[0, n - \frac{n}{q}\right]\!\right] \ : \ \binom{n}{t}(q-1)^t \geqslant q^{n-k}\right\}\right)$

1. With their parameters, they take $t = d_{\mathsf{GV}}(n,k)$, they assume that the factor $N_{\mathrm{sol}}(1)$ is one (we only have the injected solution)
2. It is only true asymptotically with n
3. With their range of parameters $N_{\mathrm{sol}}(1) \in [\![412, 1056]\!] \Rightarrow$ between 8.7 and 10 bits lost

They consider that Gaussian elimination is negligible

In the case of **SDitH-L1-gf256** with $q, n, k, t, d = 256, 230, 126, 79, 1$

Their optimal cases :

- $T_{\text{lists}} = 15.974459404437539$
- $T_{\text{checks}} = 20.231483911937776$
- $T_{\text{Gauss}} : 20.883072906964095$
- $T = 143.4589084369141$

The optimal case if we neglect Gaussian elimination :

- $T_{\text{lists}} = 16.559421905158693$
- $T_{\text{checks}} = 12.231483911937778$
- $T_{\text{Gauss}} : 20.883072906964095$
- $T = 140.4941924784691$

It's the opposite

- The cost of Gaussian elimination is even exactly their dominant term
- We have to consider **[Bernstein,Lange,Peters,2008]** and **[Peters,2010]** to improve T_{Gauss}

The reduction bound is not tight

They do not take into account N_{sol}

- **Without** N_{sol} : $p_{\mathbf{e}}(1) \geqslant \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}} p_{\mathbf{e}}(d)$
- **With** N_{sol} : $p_{\mathbf{e}}(1)N_{\text{sol}}(1) \geqslant p_{\mathbf{e}}(d)\frac{\binom{n/d}{t/d}^d}{\binom{n}{t}}\left(1 + (N_{\text{sol}}(d) - 1)\frac{\binom{n}{t}}{\binom{n/d}{t/d}^d}\right)$

The precise reduction from SD(d) to SD(1) give

1. $N_{\text{sol}}(d) = 1 \Rightarrow p_{\mathbf{e}}(1)N_{\text{sol}}(1) \geqslant \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}} p_{\mathbf{e}}(d)N_{\text{sol}}(d)$
2. $N_{\text{sol}}(d) >> 1 \Rightarrow p_{\mathbf{e}}(1)N_{\text{sol}}(1) \geqslant p_{\mathbf{e}}(d)N_{\text{sol}}(d)$

$N_{\text{sol}}(d) \in [\![30, 748]\!]$ so we are in the second case giving

$$T_{\mathsf{SD}(d)} \geqslant T_{\mathsf{SD}(1)}$$

Consequences

Their bound is therefore optimistic, with the real bound they gain security bits

Stern algorithm adapted to SD(d)

it's interesting to directly solve the problem SD(d) here. We therefore adapt Stern to SD(d)

Principle of adaptation

1. Split the information set I as $I_1 \cup \cdots \cup I_d$ where I_i is of size $\frac{k}{d}$ and is randomly chosen in I.
2. Make a bet on each of them.

The fact that

$$T_I(d) \leqslant T_I(1)$$

makes us gain in complexity compared to the theoretical bound.

Result after correction

Table – Correction of the analysis

Parameter Sets	Security Specification			Correction			SD Parameters				
	p	l	$T_{SD(1)} \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}}$	p	l	$T_{\mathsf{SD}(d)}$	q	n	k	t	d
SDitH-L1-gf256	1	2	143.46	1	2	134.61	256	230	126	79	1
SDitH-L1-gf251	1	2	143.45	1	2	133.61	251	230	126	79	1
SDitH-L3-gf256	2	5	207.67	2	5	206.16	256	352	193	120	2
SDitH-L3-gf251	2	5	207.61	2	5	205.02	251	352	193	120	2
SDitH-L5-gf256	2	5	272.35	2	5	271.30	256	480	278	150	2
SDitH-L5-gf251	2	5	272.29	2	5	269.81	251	480	278	150	2

Remark

We have not yet corrected the complexity of Gaussian elimination, we have to consider **[Bernstein,Lange,Peters,2008]** and **[Peters,2010]**

1 Security Analysis in NIST specification

2 Correction of the analysis
- First problem
- Second problem
- Third problem

3 Improved attack with projective space

Observation

Suppose we are trying to solve the problem SD(1) in the case of a $\mathbf{0}$ syndrome.

We get a good solution if and only if we find $\mathbf{x}_1$ and $\mathbf{x}_2$ such that

$$\mathbf{P}'\mathbf{x}_1 = \mathbf{P}'\mathbf{x}_2 \text{ and } |\mathbf{P}(\mathbf{x}_1 - \mathbf{x}_2)| = t - 2p \tag{3.1}$$

We calculate $(q-1)$ times too much solution

$(\mathbf{x}_1, \mathbf{x}_2)$ checks (3.1) $\Leftrightarrow$ for all α non zero $(\alpha\mathbf{x}_1, \alpha\mathbf{x}_2)$ checks (3.1)

How to exploit that ?

Adaptation of Stern's algorithm in projective space

Impact on complexity

We divide the size of the lists by a $(q-1)$ factor $\Rightarrow$ we hope to gain a factor $(q-1)$ on the complexity

What if the syndrome is non-zero ?

Reduction of the Syndrome Decoding problem to the low weight code words search

The reduction :

We look for $\mathbf{e}$ such that $\mathbf{He} = s$ and $|\mathbf{e}| = t$.
Let $\mathcal{C}$ be the code associated with the parity check matrix $\mathbf{H}$.

1. Find $\mathbf{z}$ such that $\mathbf{Hz} = \mathbf{s} \quad (\mathbf{z} = (\mathbf{0}, \mathbf{s}))$
2. Let $\mathbf{H_z}$ a parity check matrix of the code $\langle \mathcal{C}, \mathbf{z} \rangle$.
3. Find $\mathbf{e}$ such that $\mathbf{H_z e} = \mathbf{0}$

Analysis

The algorithm finds $\mathbf{e}$ such that $\mathbf{He} = \alpha s$ (for a $\alpha \in \mathbb{F}_q$).
(So just divide by α and we solved the problem)

Proof on board

What about the complexity

- The reduction cost almost nothing (a factor $\frac{q}{q-1}$)
- We reduced the problem to dimension $k + 1 \Rightarrow$ possible loss

Notations

Projective space

For a space $\mathcal{E}$ over $\mathbb{F}_q$

- $\forall \mathbf{x}, \mathbf{y} \in \mathcal{E}, \quad \mathbf{x} \sim \mathbf{y} \iff \exists \alpha \in \mathbb{F}_q^*, \ \mathbf{x} = \alpha \mathbf{y}$
- $[\mathbf{x}] \stackrel{\text{def}}{=} \{\mathbf{y} \in \mathcal{E} \ : \ \mathbf{y} \sim \mathbf{x}\}$
- $\mathcal{E}/_{\sim} \stackrel{\text{def}}{=} \{[\mathbf{x}] \ : \ \mathbf{x} \in \mathcal{E}\}$

Choose the right representatives

To exploit the idea, we want to take the lists in projective space to divide their size as well as the number of collisions by $(q-1)$:

$$\mathcal{L}'_1 = \left\{ [\mathbf{P}'\mathbf{x}_1] \ : \ [\mathbf{x}_1] \in \mathbb{F}_q^{\lfloor \frac{k+1}{2} \rfloor} / \sim \text{ and } |\mathbf{x}_1| = p \right\}$$

$$\mathcal{L}'_2 = \left\{ [\mathbf{P}'\mathbf{x}_2] \ : \ [\mathbf{x}_2] \in \mathbb{F}_q^{k+1-\lfloor \frac{k+1}{2} \rfloor} / \sim \text{ and } |\mathbf{x}_2| = p \right\}$$

We must take precautions

- A collision of equivalence classes does not necessarily induce a collision $[\mathbf{P}'\mathbf{x}_1] = [\mathbf{P}'\mathbf{x}_2] \not\Rightarrow \mathbf{P}'\mathbf{x}_1 = \mathbf{P}'\mathbf{x}_2$
- Only a proportion of $\frac{1}{q-1}$ of the pairs $(\mathbf{x}'_1, \mathbf{x}'_2) \in ([\mathbf{x}_1], [\mathbf{x}_2])$ induce a collision
- We cannot do a naive search for such a couple otherwise we lose the gain of the factor $(q-1)$

The solution

We build good representatives who induce a collision

Particular representative

For all $[\mathbf{x}] \in \mathbb{F}_q^{\frac{k+1}{2}} / \sim$ we define $\overline{\mathbf{x}}$ as the unique representative of $[\mathbf{x}]$ such that :

- If $\mathbf{P}'\mathbf{x} = \mathbf{0}$: the first non-zero symbol of $\overline{\mathbf{x}}$ is 1
- If $\mathbf{P}'\mathbf{x} \neq \mathbf{0}$: the first non-zero symbol of $\mathbf{P}'\overline{\mathbf{x}}$ is 1

We now have the desired property

Property

$$\left[\mathbf{P}'\mathbf{x}_1\right] = \left[\mathbf{P}'\mathbf{x}_2\right] \Leftrightarrow \mathbf{P}'\overline{\mathbf{x_1}} = \mathbf{P}'\overline{\mathbf{x_2}}$$

The algorithm

The only things that change with the classic Stern algorithm are the following :

7. $\mathcal{L}'_1 \leftarrow \left\{ \mathbf{P}'\overline{\mathbf{x}_1} \ : \ [\mathbf{x}_1] \in \left(\mathbb{F}_q^{\lfloor \frac{k+1}{2} \rfloor} \times 0^{k+1-\lfloor \frac{k+1}{2} \rfloor} \right) / \sim \text{ and } |\mathbf{x}_1| = p \right\}$

   $\mathcal{L}'_2 \leftarrow \left\{ \mathbf{P}'\overline{\mathbf{x}_2} \ : \ [\mathbf{x}_2] \in \left(0^{\lfloor \frac{k+1}{2} \rfloor} \times \mathbb{F}_q^{k+1-\lfloor \frac{k+1}{2} \rfloor} \right) / \sim \text{ and } |\mathbf{x}_2| = p \right\}$

8. The collision checking is done with $(\overline{\mathbf{x}_1}, \overline{\mathbf{x}_2})$

The possible losses on the theoretical gain of the factor $(q-1)$:

problem one

We must adapt Peters' tips to our new algorithm $\Rightarrow$ potential losses

problem two

We go from dimension k to dimension $(k+1)$ $\Rightarrow$ potential losses

Adaptation of tricks

The tricks on list construction :

- Use as before an enumeration order on $\left\{\hat{\mathbf{x}} \in \mathbb{F}_q^{(k+1)/2} \ : \ \hat{\mathbf{x}} = (0, \dots, 0, 1, \dots) \text{ and } |\hat{\mathbf{x}}| = p\right\}$ to compute $\mathbf{P}'\hat{\mathbf{x}}$ in one addition of one columns by elements.
- Normalize $\mathbf{P}'\hat{\mathbf{x}}$ by its first non-zero coordinate to get $\mathbf{P}'\overline{\mathbf{x}}$ (We lose a factor of 2 here on the trick)

$$T'_{\text{lists}} \approx \frac{2}{q} T_{\text{lists}}$$

The second trick on collision checking :

We just need to normalize $\hat{\mathbf{x}}$ on collisions to get $\overline{\mathbf{x}}$ which takes $2p$ operations per collisions (negligible). We can then apply the trick as before.

$$T'_{\text{checks}} \approx \frac{1}{q} T_{\text{checks}}$$

The real gain

For the other complexity values, we have :

- $T'_{\text{Gauss}} \approx T_{\text{Gauss}}$
- $N'_{\text{sol}} \approx N_{\text{sol}}$
- $p'_{\mathbf{e}} \approx p_{\mathbf{e}}/4 \quad (\frac{\binom{n-k-1-\ell}{t-2p}}{\binom{n-k-\ell}{t-2p}} \approx \frac{1}{4}$ for the range of parameters)

Expected gain

- We lose 2 bits due to the transition from dimension k to dimension $k+1$.
- We lose 1 bits if T'_{lists} control T'_{checks}
- We should have an additional gain between 5 and 6 bits with the projections

Remark

We can adapt the Stern projectif algorithm to the problem SD(d) as previously

Result

Table – SDiTH security with our results

	Security Specification			Correction			d-split-Stern-proj			SD Parameters				
	p	l	$T_{SD(1)} \frac{\binom{n/d}{t/d}^d}{\binom{n}{t}}$	p	l	$T_{SD(d)}$	p	l	$T_{SD(d)}$	q	n	k	t	d
L1-gf256	1	2	143.46	1	2	134.61	2	4	130.07	256	230	126	79	1
L1-gf251	1	2	143.45	1	2	133.91	2	4	129.36	251	230	126	79	1
L3-gf256	2	5	207.67	2	5	206.16	2	4	200.73	256	352	193	120	2
L3-gf251	2	5	207.61	2	5	205.02	2	4	199.65	251	352	193	120	2
L5-gf256	2	5	272.35	2	5	271.30	2	4	266.77	256	480	278	150	2
L5-gf251	2	5	272.29	2	5	269.81	2	4	265.35	251	480	278	150	2

Table – Bit gain on security

Parameter Sets	Gains		
	Gain-Correction	Gain-proj	Total Gain
SDitH-L1-gf256	8.84	4.54	13.39
SDitH-L1-gf251	9.54	4.54	14.09
SDitH-L3-gf256	1.51	5.43	6.94
SDitH-L3-gf251	2.59	5.36	7.96
SDitH-L5-gf256	1.04	4.53	5.57
SDitH-L5-gf251	2.48	4.46	6.94

The loss of 0.5 bits on the expected results is linked to the fact that the Gaussian elimination is not negligible.

Work in coming

1. Improvement of T_{Gauss} with **[Bernstein,Lange,Peters,2008]** and **[Peters,2010]**
2. Apply the adaptation of projective spaces to **[Dumer,1991]** to see if we obtain a gain
3. Try to generalize the idea to the Grassmannian (generalization of projective spaces with several dimensions)