Cryptanalysis of a Generalized Subset-Sum Pseudorandom Generator

Charles Bouillaguet, Florette Martinez, Damien Vergnaud

Journées C2, 18 Octobre 2023

Introduction

Randomness in Cryptography...

RSA

- p, q random, $N = p \times q$.
- $p, q \rightarrow N$ easy
- $N \rightsquigarrow p, q$ hard

Randomness in Cryptography...

RSA

- p, q random, $N = p \times q$.
- $p, q \rightarrow N$ easy
- $N \rightsquigarrow p, q$ hard

What if?

$$N_1 = p \times q_1 \text{ and } N_2 = p \times q_2$$

Randomness in Cryptography...

RSA

- p, q random, $N = p \times q$.
- $p, q \rightarrow N$ easy
- $N \rightsquigarrow p, q$ hard

What if?

$$N_1 = p \times q_1 \text{ and } N_2 = p \times q_2$$

It happens (see Lenstra *et al* and Heninger *et al* in 2012)

... Is important.

Random Number Generators

True Random Number Generators

- Collects noises.
- Pool of entropy
- Good-quality randomness
- Expensive!

Pseudo-Random Number Generators (PRNG)

- Takes a seed
- Expands the seed in a flow of pseudo random number
- Deterministic algorithm
- Faster

The Knapsack Generator

Subset Sum Problem

We know a n-uple of weights $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{n-1}) \in \{0, \ldots, M\}^n$ and an integer q and we search for a binary vector $\mathbf{v} = (v_0, \ldots, v_{n-1})$ such that

$$\sum_{i=0}^{n-1} v_i \omega_i = q \bmod 2^n.$$

If $M \simeq 2^n$ it is the hardest.

Knapsack Generator by Rueppel and Massey in 1985

Secret Key: $(u_0, \ldots, u_{n-1}) \in \{0,1\}^n$ and $\omega \in \{0, \ldots, 2^n\}^n$

Knapsack Generator by Rueppel and Massey in 1985

Secret Key: $(u_0, \ldots, u_{n-1}) \in \{0,1\}^n$ and $\omega \in \{0, \ldots, 2^n\}^n$

First step: We use a LFSR to obtain a stream of weak pseudo-random numbers

$$(u_0, \ldots, u_{n-1}) \rightarrow \boxed{LFSR} \rightarrow (\mathbf{v}_0, \mathbf{v}_1, \ldots)$$

where $\mathbf{v}_i$ are binary n-uplets

Knapsack Generator by Rueppel and Massey in 1985

Secret Key: $(u_0, \ldots, u_{n-1}) \in \{0,1\}^n$ and $\omega \in \{0, \ldots, 2^n\}^n$

$$(u_0, \ldots, u_{n-1}) \rightarrow \boxed{LFSR} \rightarrow (\mathbf{v}_0, \mathbf{v}_1, \ldots)$$

Second step: We hide the weakness using the Subset Sum Problem

$$\mathbf{v}_i \rightarrow \boxed{< \ , \omega >} \rightarrow q_i$$

If $\mathbf{v}_i = (v_0, \ldots, v_{n-1})$ then $< \mathbf{v}_i, \omega >= \sum_{j=0}^{n-1} v_j \omega_j \bmod 2^n$.

Knapsack Generator by Rueppel and Massey in 1985

Secret Key: $(u_0, \ldots, u_{n-1}) \in \{0,1\}^n$ and $\omega \in \{0, \ldots, 2^n\}^n$

$$(u_0, \ldots, u_{n-1}) \rightarrow \boxed{LFSR} \rightarrow (\mathbf{v}_0, \mathbf{v}_1, \ldots)$$

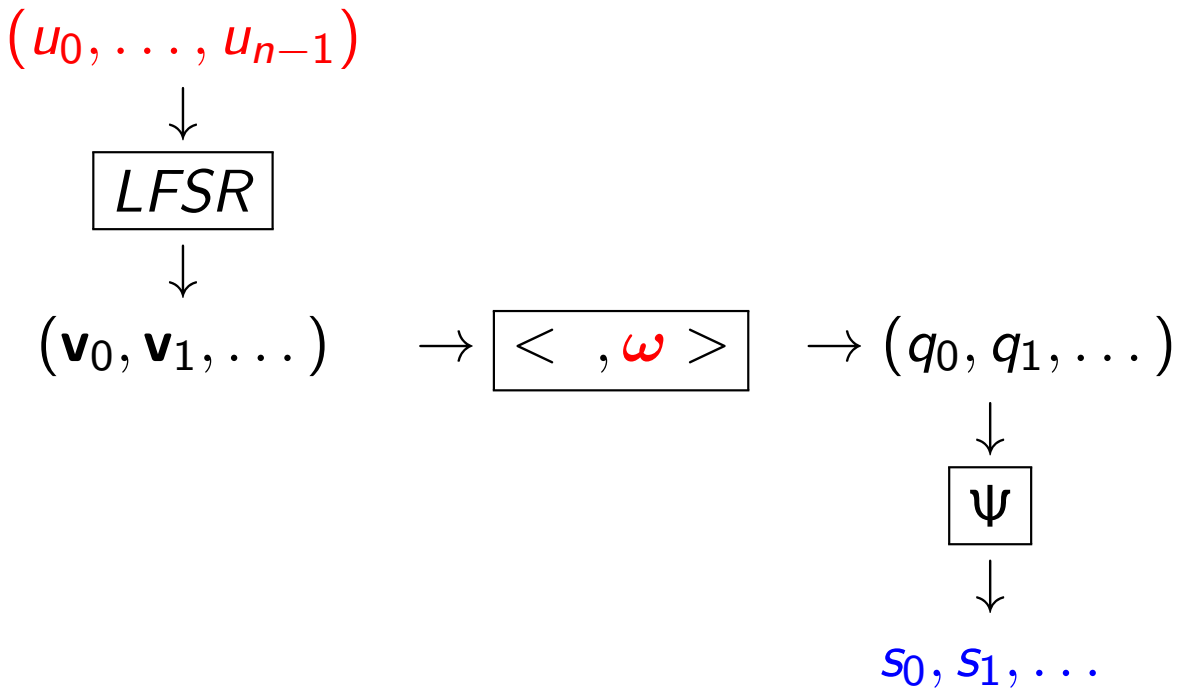
$$\mathbf{v}_i \rightarrow \boxed{< \ , \omega >} \rightarrow q_i$$

Third step: We hide more by truncating

$$q_i \rightarrow \boxed{\Psi} \rightarrow s_i$$

where Ψ truncates the ℓ last bits.

Knapsack Generator by Rueppel and Massey in 1985



Knellwolf and Meier attack in 2011

Weakness : the secret key in unbalanced $(u_0, \ldots, u_{n-1})$ (n bits) + $(\omega_0, \ldots, \omega_{n-1})$ (n^2 bits)

Knellwolf and Meier attack in 2011

Weakness : the secret key in unbalanced $(u_0, \ldots, u_{n-1})$ (n bits) + $(\omega_0, \ldots, \omega_{n-1})$ (n^2 bits)
Attack : we guess $(u_0, \ldots, u_{n-1})$

Knellwolf and Meier attack in 2011

Weakness : the secret key in unbalanced $(u_0, \ldots, u_{n-1})$ (n bits) + $(\omega_0, \ldots, \omega_{n-1})$ (n^2 bits)
Attack : we guess $(u_0, \ldots, u_{n-1})$ and obtain $\boldsymbol{\omega}$ as solution of

$$\begin{pmatrix} \mathbf{v}_0 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix} \times \boldsymbol{\omega} \simeq 2^{\ell} \times \begin{pmatrix} s_0 \\ \ldots \\ s_{n-1} \end{pmatrix} \bmod 2^n$$

Knellwolf and Meier attack in 2011

Weakness : the secret key in unbalanced $(u_0, \ldots, u_{n-1})$ (n bits) + $(\omega_0, \ldots, \omega_{n-1})$ (n^2 bits)
Attack : we guess $(u_0, \ldots, u_{n-1})$ and obtain $\boldsymbol{\omega}$ as solution of

$$\begin{pmatrix} \mathbf{v}_0 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix} \times \boldsymbol{\omega} \simeq 2^{\ell} \times \begin{pmatrix} s_0 \\ \ldots \\ s_{n-1} \end{pmatrix} \bmod 2^n$$

$\rightarrow$ We can solve that using lattice techniques

Solving approximate linear system

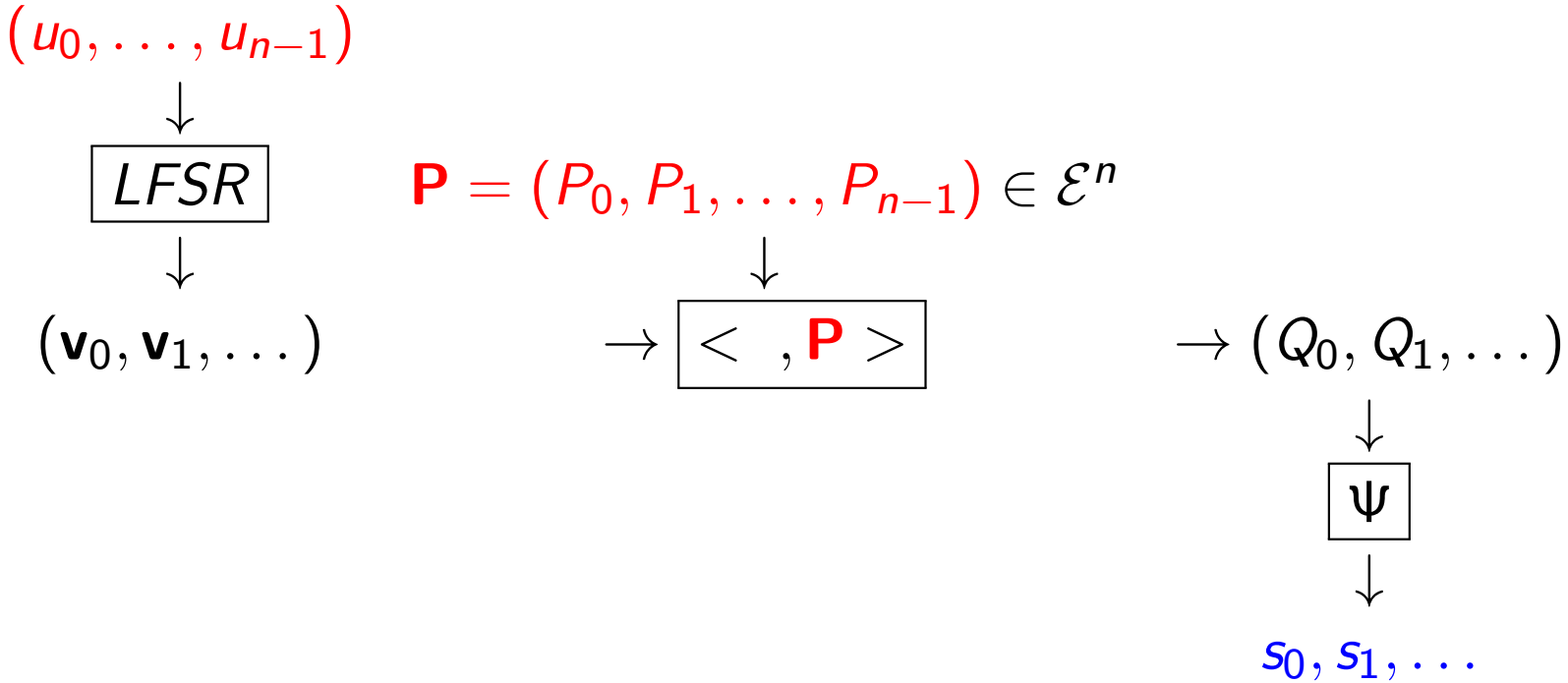
- In $(\mathbb{Z}/n\mathbb{Z}, +)$, if P'_1 and P'_2 have the same r leading bits as P_1 and P_2 then $P'_1 + P'_2$ and $P_1 + P_2$ have the same $r - 1$ leading bits.

Solving approximate linear system

- In $(\mathbb{Z}/n\mathbb{Z}, +)$, if P'_1 and P'_2 have the same r leading bits as P_1 and P_2 then $P'_1 + P'_2$ and $P_1 + P_2$ have the same $r - 1$ leading bits.
- In an elliptic curve, there is no nice behaviour between the addition and the encoding of a point.

Elliptic Knapsack Generator

Elliptic Knapsack Generator by Von zur Gathen and Shparlinski



where $< \mathbf{v}_i, \mathbf{P} >= \sum_{j=0}^{n-1} v_j P_j$ and Ψ truncates the ℓ last bits of the abscissa of Q.

KM attack: we guess $(u_0, \ldots, u_{n-1})$

KM attack: we guess $(u_0, \ldots, u_{n-1})$ and we have

$$\begin{pmatrix} \mathbf{v}_0 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix} \times \mathbf{P}$$

is close to a vector of points which abscisses are

$$2^\ell \times \begin{pmatrix} s_0 \\ \ldots \\ s_{n-1} \end{pmatrix}.$$

KM attack: we guess $(u_0, \ldots, u_{n-1})$ and we have

$$\begin{pmatrix} \mathbf{v}_0 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix} \times \mathbf{P}$$

is close to a vector of points which abscisses are

$$2^\ell \times \begin{pmatrix} s_0 \\ \ldots \\ s_{n-1} \end{pmatrix}.$$

We cannot do anything directly.

Naive attack: we guess $(u_0, \ldots, u_{n-1})$ and everything to undo Ψ.

Naive attack: we guess $(u_0, \ldots, u_{n-1})$ and everything to undo Ψ. Then we have

$$\mathbf{P} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix}^{(-1)} \times \begin{pmatrix} Q_0 \\ Q_1 \\ \ldots \\ Q_{n-1} \end{pmatrix} \bmod |\mathcal{E}|.$$

Time complexity in $\mathcal{O}(2^{n+\ell \times n})$.

Naive attack: we guess $(u_0, \ldots, u_{n-1})$ and everything to undo Ψ.
Then we have

$$\mathbf{P} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \ldots \\ \mathbf{v}_{n-1} \end{pmatrix}^{(-1)} \times \begin{pmatrix} Q_0 \\ Q_1 \\ \ldots \\ Q_{n-1} \end{pmatrix} \bmod |\mathcal{E}|.$$

Time complexity in $\mathcal{O}(2^{n+\ell \times n})$.
Better than the exhaustive search but not practical.

The attack

Initial guess

Guess $u_0, \ldots, u_{n-1}$ and derive all of the $\mathbf{v}_i$'s

The attack

Initial guess

Guess $u_0, \ldots, u_{n-1}$ and derive all of the $\mathbf{v}_i$'s

Find good triplets

Search for $n/2$ triplets (i, j, k) s.t. $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$

Context

- A, B, C lists of size N made of uniformly random n-uplets
- $\mathbf{v}_i \in A$, $\mathbf{v}_j \in B$, $\mathbf{v}_k \in C$

$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k)$ when $n = 1$

$\mathbf{v}_i$	0	0	0	0	1	1	1	1
$\mathbf{v}_j$	0	0	1	1	0	0	1	1
$\mathbf{v}_k$	0	1	0	1	0	1	0	1
$\mathbf{v}_i + \mathbf{v}_j$	0	0	1	1	1	1	2	2

$$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k) = 3/8$$

$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k)$ when $n = 1$

$\mathbf{v}_i$	0	0	0	0	1	1	1	1
$\mathbf{v}_j$	0	0	1	1	0	0	1	1
$\mathbf{v}_k$	0	1	0	1	0	1	0	1
$\mathbf{v}_i + \mathbf{v}_j$	0	0	1	1	1	1	2	2

$$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k) = 3/8$$

When $n \geq 1$:

$$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k) = (3/8)^n$$

We define $Y =$ "the number of good triplets in A, B, C"

$$\mathbb{E}(Y) = N^3 \left(\frac{3}{8}\right)^n$$

$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k)$ when $n = 1$

$\mathbf{v}_i$	0	0	0	0	1	1	1	1
$\mathbf{v}_j$	0	0	1	1	0	0	1	1
$\mathbf{v}_k$	0	1	0	1	0	1	0	1
$\mathbf{v}_i + \mathbf{v}_j$	0	0	1	1	1	1	2	2

$$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k) = 3/8$$

When $n \geq 1$:

$$\mathbb{P}(\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k) = (3/8)^n$$

We define $Y =$ "the number of good triplets in A, B, C"

$$\mathbb{E}(Y) = N^3 \left(\frac{3}{8}\right)^n$$

For now $N \simeq \left(\frac{8}{3}\right)^{n/3}$

A Sub-Quadratic Algorithm

The good triplets have a bias as $w(\mathbf{v}_k) = w(\mathbf{v}_i) + w(\mathbf{v}_j)$.

A Sub-Quadratic Algorithm

The good triplets have a bias as $w(\mathbf{v}_k) = w(\mathbf{v}_i) + w(\mathbf{v}_j)$.
Then we often have $w(\mathbf{v}_i) \simeq w(\mathbf{v}_j) \simeq n/3$ and $w(\mathbf{v}_k) \simeq 2n/3$

A Sub-Quadratic Algorithm

The good triplets have a bias as $w(\mathbf{v}_k) = w(\mathbf{v}_i) + w(\mathbf{v}_j)$.
Then we often have $w(\mathbf{v}_i) \simeq w(\mathbf{v}_j) \simeq n/3$ and $w(\mathbf{v}_k) \simeq 2n/3$

1: **function** FindTriplet(A, B, C, ϵ)
2: $A' \leftarrow \{\mathbf{v}_i \in A \mid w(\mathbf{v}_i) \leq n/3 + \epsilon\}$
3: $B' \leftarrow \{\mathbf{v}_j \in B \mid w(\mathbf{v}_j) \leq n/3 + \epsilon\}$
4: $C' \leftarrow \{\mathbf{v}_k \in C \mid w(\mathbf{v}_k) \geq 2n/3 - \epsilon\}$
5: **for all** $\mathbf{v}_i, \mathbf{v}_j \in A' \times B'$ **do**
6: **if** $\mathbf{v}_i + \mathbf{v}_j \in C$ **then**
7: **return** $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$
8: **return** $\perp$

For $\epsilon = 1/6$, the algorithm succeed with overwhelming probability in time $\mathcal{O}(N^{1.654}) \simeq \mathcal{O}(2^{0.78n})$.

The attack

Initial guess

Guess $u_0, \ldots, u_{n-1}$ and derive all of the $\mathbf{v}_i$'s

Find good triplets

Search for $n/2$ triplets (i, j, k) s.t. $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$

The attack

Initial guess

Guess $u_0, \ldots, u_{n-1}$ and derive all of the $\mathbf{v}_i$'s

Find good triplets

Search for $n/2$ triplets (i, j, k) s.t. $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$

Retrieve the Q_i

If one knows (i, j, k) s.t. $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$ then $Q_i + Q_j = Q_k$ is satisfied.

Search for (R_i, R_j, R_k) such that:

- $R_i + R_j = R_k$
- $\Psi(R_b) = s_b$ for $b \in \{i, j, k\}$

Let (R_i, R_j) be such that $\Psi(R_b) = s_b$ for $b \in \{i, j\}$. There are 2^{2l} such couples.

Let (R_i, R_j) be such that $\Psi(R_b) = s_b$ for $b \in \{i, j\}$. There are 2^{2l} such couples.
If the encoding of R_i, R_j is only close to the encoding of Q_i, Q_j then the encoding $R_k = R_i + R_j$ has **no reason** to be close to the encoding of $Q_k = Q_i + Q_j$.

Let (R_i, R_j) be such that $\Psi(R_b) = s_b$ for $b \in \{i, j\}$. There are 2^{2l} such couples.
If the encoding of R_i, R_j is only close to the encoding of Q_i, Q_j then the encoding $R_k = R_i + R_j$ has **no reason** to be close to the encoding of $Q_k = Q_i + Q_j$.
Hence

$$\mathbb{P}(\Psi(R_k) = s_k) = \frac{2^\ell}{|\mathcal{E}|}$$

We have $2^\ell \times 2^\ell$ (R_i, R_j) so as long as $2^{3\ell} < |\mathcal{E}|$, we only have (Q_i, Q_j, Q_k) and $(-Q_i, -Q_j, -Q_k)$ remaining.

- As we want to solve a linear system and $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$, we discard v_k and Q_k for each triplets we computed.

- As we want to solve a linear system and $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$, we discard v_k and Q_k for each triplets we computed.
- We known $xQ_{i_1}, xQ_{j_1}, \ldots, xQ_{i_{n/2}}, xQ_{j_{n/2}}$ but not the sign of the ordinates.

- As we want to solve a linear system and $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$, we discard v_k and Q_k for each triplets we computed.
- We known $xQ_{i_1}, xQ_{j_1}, \ldots, xQ_{i_{n/2}}, xQ_{j_{n/2}}$ but not the sign of the ordinates.
- Fixing the sign of Q_{i_t} fixes the sign of Q_{j_t}.

- As we want to solve a linear system and $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$, we discard v_k and Q_k for each triplets we computed.
- We known $xQ_{i_1}, xQ_{j_1}, \ldots, xQ_{i_{n/2}}, xQ_{j_{n/2}}$ but not the sign of the ordinates.
- Fixing the sign of Q_{i_t} fixes the sign of Q_{j_t}.
- We need and exhaustive search on the signs of the $n/2$ couples.

- As we want to solve a linear system and $\mathbf{v}_i + \mathbf{v}_j = \mathbf{v}_k$, we discard v_k and Q_k for each triplets we computed.
- We known $xQ_{i_1}, xQ_{j_1}, \ldots, xQ_{i_{n/2}}, xQ_{j_{n/2}}$ but not the sign of the ordinates.
- Fixing the sign of Q_{i_t} fixes the sign of Q_{j_t}.
- We need and exhaustive search on the signs of the $n/2$ couples.

For each choice of signs we compute:

$$\mathbf{P} = \begin{pmatrix} v_{i_1} \\ v_{j_1} \\ \ldots \\ v_{i_{n/2}} \\ v_{j_{n/2}} \end{pmatrix}^{(-1)} \times \begin{pmatrix} Q_{i_1} \\ Q_{j_1} \\ \ldots \\ Q_{i_{n/2}} \\ Q_{j_{n/2}} \end{pmatrix} \bmod |\mathcal{E}|$$

For all $(u_0, \ldots, u_{n-1})$ in $\{0,1\}^n$:

- derive all the $\mathbf{v}_i$ and find $n/2$ good triplets in $\mathcal{O}(2^{0.78n})$
- for each good triplet derive (Q_i, Q_j) and $(-Q_i, -Q_j)$ in $\mathcal{O}(2^{2\ell})$
- derive the P_i's for the $2^{n/2-1}$ possible signs combinations
- check consistency

For all $(u_0, \ldots, u_{n-1})$ in $\{0,1\}^n$:

- derive all the $\mathbf{v}_i$ and find $n/2$ good triplets in $\mathcal{O}(2^{0.78n})$
- for each good triplet derive (Q_i, Q_j) and $(-Q_i, -Q_j)$ in $\mathcal{O}(2^{2\ell})$
- derive the P_i's for the $2^{n/2-1}$ possible signs combinations
- check consistency

The complexity is

$$\mathcal{O}(2^n \times (2^{0.78n} + (n/2 \times 2^{2\ell}) + 2^{n/2-1}))$$

that is to say $\mathcal{O}(2^{1.78n})$ binary operations (with $\ell = \log_2(n)$).

Experimental results

When $n = 16$ and the initial sequence $(u_0, \ldots, u_{n-1})$ is known.

- When $|\mathcal{E}| = 65111$.

ℓ	1	2	3	4	5	6
m	1000	1000	1000	1000	1000	1885
time	6.9s	5.3s	5.6s	5.02s	5.7s	26.7s

- When $|\mathcal{E}| = 1099510687747$.

ℓ	1	2	3	4	5	6	7	8	9
m	1885	1885	1885	1885	1885	1885	1885	1885	1750
time	2.1s	2.1s	2.08s	2.5s	2.6s	2.1s	3.5s	8.3s	26.7s

Thank you for your attention