Can we use convolutional codes in the McEliece Cryptosystem?

P. Almeida, M. Beltrá, D. Napp, C. Sebastião

8th June 2023

- In block codes a long block of fixed length is transmitted:

$$\mathbf{u}G = \mathbf{v}$$

- In convolutional codes a continuous sequence of shorter vectors is transmitted:

$$\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_s) \Longrightarrow \mathbf{u}_s D^s + \cdots + \mathbf{u}_2 D^2 + \mathbf{u}_1 D + \mathbf{u}_0 =: \mathbf{u}(D)$$

the *information vector*.

- In block codes a long block of fixed length is transmitted:

$$\mathbf{u}G = \mathbf{v}$$

- In convolutional codes a continuous sequence of shorter vectors is transmitted:

$$\mathbf{u} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_s) \Longrightarrow \mathbf{u}_s D^s + \cdots + \mathbf{u}_2 D^2 + \mathbf{u}_1 D + \mathbf{u}_0 =: \mathbf{u}(D)$$

the *information vector*.

Encoding with a convolutional encoder

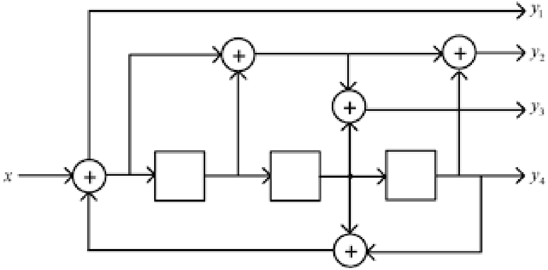
$$\cdots + \mathbf{u}_2 D^2 + \mathbf{u}_1 D + \mathbf{u}_0 \xrightarrow{G(D)=G_0+G_1D+G_2D^2+\ldots+G_mD^m}$$

$$\cdots + \underbrace{(\mathbf{u}_2 G_0 + \mathbf{u}_1 G_1 + \mathbf{u}_0 G_2)}_{\mathbf{v}_2} D^2 + \underbrace{(\mathbf{u}_1 G_0 + \mathbf{u}_0 G_1)}_{\mathbf{v}_1} D + \underbrace{\mathbf{u}_0 G_0}_{\mathbf{v}_0}$$

Definition

A *convolutional code* $\mathcal{C}$ of rate k/n is an $\mathbb{F}[D]$-submodule of $\mathbb{F}[D]^n$ of rank k given by a polynomial *encoder matrix* $G(D) \in \mathbb{F}^{k\times n}[D]$,

$$\mathcal{C} = \mathrm{Im}_{\,\mathbb{F}[D]} G(D) = \left\{ \mathbf{u}(D)G(D) : \ \mathbf{u}(D) \in \mathbb{F}^k[D] \right\}$$



The polynomial:

$$\begin{aligned}\mathbf{u}(D)G(D) &= (\mathbf{u}_0 + \mathbf{u}_1 D + \cdots + \mathbf{u}_s D^s)(G_0 + G_1 D + \cdots + G_m D^m) \\ &= \mathbf{u}_0 G_0 + (\mathbf{u}_1 G_0 + \mathbf{u}_0 G_1)D + (\mathbf{u}_2 G_0 + \mathbf{u}_1 G_1 + \mathbf{u}_0 G_2)D^2 + \cdots\end{aligned}$$

Can be represented by constant matrices:

$$\begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_s \end{bmatrix} \begin{bmatrix} G_0 & G_1 & \cdots & G_{\mu+\nu} & & & & & & \\ & G_0 & G_1 & \cdots & G_{\mu+\nu} & & & & & \\ & & \ddots & \ddots & & \ddots & & & & \\ & & & G_0 & G_1 & \cdots & G_{\mu+\nu} & & & \\ & & & & \ddots & \ddots & & \ddots & & \\ & & & & & G_0 & G_1 & \cdots & G_{\mu+\nu} \end{bmatrix}$$

Original McEliece PKC:

Original McEliece PKC:

Secret key: G, S and P where

- $G \in \mathbb{F}^{k\times n}$ be an encoder of an (n,k) block code $\mathcal{C}$ capable of correcting t errors,
- $S \in \mathbb{F}^{k\times k}$ an invertible matrix
- $P \in \mathbb{F}^{n\times n}$ a permutation matrix.

Original McEliece PKC:

Secret key: G, S and P where

- $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code $\mathcal{C}$ capable of correcting t errors,
- $S \in \mathbb{F}^{k \times k}$ an invertible matrix
- $P \in \mathbb{F}^{n \times n}$ a permutation matrix.

Public key: $G' = SGP$ and t.

Original McEliece PKC:

Secret key: G, S and P where

- $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code $\mathcal{C}$ capable of correcting t errors,
- $S \in \mathbb{F}^{k \times k}$ an invertible matrix
- $P \in \mathbb{F}^{n \times n}$ a permutation matrix.

Public key: $G' = SGP$ and t.

The codes used: **Goppa codes** or **QC MDPC.**

Original McEliece PKC:

Secret key: G, S and P where

- $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code $\mathcal{C}$ capable of correcting t errors,
- $S \in \mathbb{F}^{k \times k}$ an invertible matrix
- $P \in \mathbb{F}^{n \times n}$ a permutation matrix.

Public key: $G' = SGP$ and t.

The codes used: **Goppa codes** or **QC MDPC.**

A major drawback

Requires very large keys

Original McEliece PKC:

Secret key: G, S and P where

- $G \in \mathbb{F}^{k\times n}$ be an encoder of an (n,k) block code $\mathcal{C}$ capable of correcting t errors,
- $S \in \mathbb{F}^{k\times k}$ an invertible matrix
- $P \in \mathbb{F}^{n\times n}$ a permutation matrix.

Public key: $G' = SGP$ and t.

The codes used: **Goppa codes** or **QC MDPC.**

A major drawback

Requires very large keys

How to reduce them?

Change the G. It would be ideal to use GRS

A new Variant of the McEliece cryptosystem

Classic McEliece cryptosystem:

Encoder G of a linear block code allows to correct t errors:

$$G' = S\ G\ P$$

S an invertible matrix and P a permutation. Alice sends

$$\mathbf{y} = \mathbf{u}G' + \mathbf{e}$$

Bob computes

$$\mathbf{y}P^{-1} = \mathbf{u}SG + \mathbf{e}P^{-1}$$

and decodes

$$(\mathbf{u}S)G \Longrightarrow \mathbf{u}S \Longrightarrow \mathbf{u}$$

Proposal:

We construct our public convolutional encoder $G'(D)$ as

$$G'(D) = S(D)\ G\ P(D^{-1}, D).$$

Proposal:

We construct our public convolutional encoder $G'(D)$ as

$$G'(D) = S(D)\; G\; P(D^{-1}, D).$$

Alice sends

$$\mathbf{y}(D) = \mathbf{u}(D)G'(D) + \mathbf{e}(D) \Longrightarrow$$

Bob computes

$$\mathbf{y}(D)T(D^{-1}, D) = (\mathbf{u}(D)S(D))G + \mathbf{e}(D)P^{-1}(D^{-1}, D)$$

and finally decodes

$$(\mathbf{u}(D)S(D))G \Longrightarrow \mathbf{u}(D)S(D) \Longrightarrow \mathbf{u}(D)$$

- Let $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code admitting an efficient decoding algorithm which can correct up to t errors.

- Let $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code admitting an efficient decoding algorithm which can correct up to t errors.
- An invertible polynomial matrix

$$S(D) = S_1 D + S_2 D^2 + \cdots + S_{m-1} D^{m-1} \in \mathbb{F}^{k \times k}[D],$$

whose inverse is in $\mathbb{F}^{k \times k}(D)$

- Let $G \in \mathbb{F}^{k \times n}$ be an encoder of an (n, k) block code admitting an efficient decoding algorithm which can correct up to t errors.
- An invertible polynomial matrix

$$S(D) = S_1 D + S_2 D^2 + \cdots + S_{m-1} D^{m-1} \in \mathbb{F}^{k \times k}[D],$$

whose inverse is in $\mathbb{F}^{k \times k}(D)$

- An invertible rational polynomial matrix

$$P(D^{-1}, D) = P_{-1} D^{-1} + P_0 + P_1 D,$$

whose inverse is of the form

$$T(D^{-1}, D) = P^{-1}(D^{-1}, D) = T_{-1} D^{-1} + T_0 + T_1 D, \quad (1)$$

and such that each row of each coefficient matrix T_i, $i \in \{-1, 0, 1\}$, has no more than ρ nonzero elements.

Summary:

Secret key: $S(D)$, G, and $P(D^{-1}, D)$.

Public key: $G'(D) = S(D)GP(D^{-1}, D)$ and t/ρ.

Summary:

Secret key: $S(D)$, G, and $P(D^{-1}, D)$.

Public key: $G'(D) = S(D)GP(D^{-1}, D)$ and t/ρ.

Encryption: Alice selects an error vector $\mathbf{e}(D)$ satisfying

$$\mathrm{wt}((\mathbf{e}_i, \mathbf{e}_{i+1}, \mathbf{e}_{i+2})) \leq \frac{t}{\rho},$$

for all $0 \leq i \leq s + m - 2$, and encrypts $\mathbf{u}(D)$ as

$$\mathbf{y}(D) = \mathbf{u}(D)G'(D) + \mathbf{e}(D).$$

Decryption: Bob multiplies $\mathbf{y}(D)$ from the right by $T(D^{-1}, D) = P^{-1}(D^{-1}, D)$ to obtain

$$\mathbf{y}(D)T(D^{-1}, D) = \mathbf{u}(D)S(D)G + \mathbf{e}(D)T(D^{-1}, D),$$

he decodes each coefficient using G and finally he recovers the message $\mathbf{u}(D)$ from $\mathbf{u}(D)S(D)$.

We need to protect G with $S(D)GP(D^{-1},D)$

We need to protect G with $S(D)GP(D^{-1}, D)$

We impose the following conditions on $P(D^{-1}, D)$ and $T(D^{-1}, D)$:

- each nonzero column of P_i has at least two nonzero elements;
- each nonzero row of T_i has exactly two nonzero elements.

Using Generalized Reed-Solomon codes

We need to protect G with $S(D)GP(D^{-1}, D)$

We impose the following conditions on $P(D^{-1}, D)$ and $T(D^{-1}, D)$:

- each nonzero column of P_i has at least two nonzero elements;
- each nonzero row of T_i has exactly two nonzero elements.

Does there exist a large class of such matrices?

Using Generalized Reed-Solomon codes

We need to protect G with $S(D)GP(D^{-1}, D)$

We impose the following conditions on $P(D^{-1}, D)$ and $T(D^{-1}, D)$:

- each nonzero column of P_i has at least two nonzero elements;
- each nonzero row of T_i has exactly two nonzero elements.

Does there exist a large class of such matrices?

How to build them?

Lemma

Let T be a block matrix of the form

$$T = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where A_{11} and A_{22} are non singular. Then,

a) $|T| = |A_{11}|\,|A_{22} - A_{21}A_{11}^{-1}A_{12}|$.

b) If T is non singular, the inverse of T is

$$\begin{bmatrix} (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & -A_{11}^{-1}A_{12}(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \\ -A_{22}^{-1}A_{21}(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \end{bmatrix}.$$

building $T(D^{-1}, D)$

We propose a class of matrices $T(D^{-1}, D)$ of the following form:

$$T(D^{-1}, D) = \Pi \left[\begin{array}{c|c} A(D^{-1}, D) & \beta A(D^{-1}, D) \\ \hline A(D^{-1}, D) & A(D^{-1}, D) \end{array} \right],$$

with$n$ even, $\beta \notin \{0, 1\}$, $\Pi \in \mathbb{F}^{n \times n}$ be a permutation matrix and the matrices $A = A(D^{-1}, D)$ are randomly generated satisfying the following conditions:

1. A is an upper triangular matrix;
2. The entries of the principal diagonal of A are of the form D^j, with $j \in \{-1, 0, 1\}$, in such a way that there are δ_j entries with power D^j, satisfying

$$\delta_{-1} = \delta_1;$$

3. Each row of A has at most one entry of the form γD^j for each exponent $j \in \{-1, 0, 1\}$, with $\gamma \in \mathbb{F} \setminus \{0\}$;
4. All nonzero entries of a column of A have the same exponent of D.

Costruction of $S(D)$

- As for the construction of $S(D) = S_1 D + S_2 D^2 + \cdots + S_{m-1} D^{m-1}$ we only require, besides of being invertible, to have the first coefficients with rank less than k.

Costruction of $S(D)$

- As for the construction of $S(D) = S_1 D + S_2 D^2 + \cdots + S_{m-1} D^{m-1}$ we only require, besides of being invertible, to have the first coefficients with rank less than k.
- These weak restrictions on $S(D)$ will allow to generate large parts of the S_i completely at random.

Strong Keys

Strong Keys are interesting to hinder ISD attacks. Consider:

$$\begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_s \end{bmatrix} \begin{bmatrix} G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & & & & & \\ & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & & & & \\ & & \ddots & \ddots & & \ddots & & & & \\ & & & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & & \\ & & & & \ddots & \ddots & & \ddots & & \\ & & & & & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{s+m} \end{bmatrix}$$

$\Longrightarrow$

$$\mathbf{u}_0 \widetilde{G} = \mathbf{y}_I$$

Strong Keys

Strong Keys are interesting to hinder ISD attacks. Consider:

$$\begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_s \end{bmatrix} \begin{bmatrix} G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & & & & \\ & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & & & \\ & & \ddots & \ddots & & \ddots & & & \\ & & & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} & & \\ & & & & \ddots & \ddots & & \ddots & \\ & & & & & G'_0 & G'_1 & \cdots & G'_{\mu+\nu} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{s+m} \end{bmatrix}$$

$\Longrightarrow$

$$\mathbf{u}_0 \widetilde{G} = \mathbf{y}_I$$

We require:

- $\mathcal{C} = \operatorname{Im} \widetilde{G}$ to have distance $=1$
- *the reciprocal code* $\widetilde{\mathcal{C}}^{\mathbf{r}} = \operatorname{Im} \widetilde{G^r}$ to have distance 1.

Many strong keys

n	k	m	(d_{-1}, d_0, d_1)	$(r_1, r_2, \ldots, r_{m-1})$	percentage strong keys
72	48	6	$(24, 24, 24)$	$(16, 32, 48, 32, 16)$	34.4%
72	48	10	$(24, 24, 24)$	$(16, 16, 24, 32, 48, 32, 24, 16, 16)$	23.4%
108	72	6	$(36, 36, 36)$	$(24, 48, 72, 48, 24)$	64.4%
108	72	10	$(36, 36, 36)$	$(24, 24, 36, 48, 72, 48, 36, 24, 24)$	44.2%
108	84	6	$(36, 36, 36)$	$(28, 56, 84, 56, 28)$	71.6%
108	84	10	$(36, 36, 36)$	$(28, 28, 42, 56, 84, 56, 42, 28, 28)$	55.2%
120	84	6	$(40, 40, 40)$	$(28, 56, 84, 56, 28)$	77.0%
120	84	10	$(40, 40, 40)$	$(28, 28, 42, 56, 84, 56, 42, 28, 28)$	60.4%
144	96	6	$(48, 48, 48)$	$(32, 64, 96, 64, 32)$	83.4%
144	96	10	$(48, 48, 48)$	$(32, 32, 48, 64, 96, 64, 48, 32, 32)$	62.2 %
144	108	6	$(48, 48, 48)$	$(36, 72, 108, 72, 36)$	89.0%
144	108	10	$(48, 48, 48)$	$(36, 36, 54, 72, 108, 72, 54, 36, 36)$	74.0%
180	120	6	$(60, 60, 60)$	$(40, 80, 120, 80, 40)$	89.6%
180	120	10	$(60, 60, 60)$	$(40, 40, 60, 80, 120, 80, 60, 40, 40)$	76.8%
180	132	6	$(60, 60, 60)$	$(44, 88, 132, 88, 44)$	90.8%
180	132	10	$(60, 60, 60)$	$(44, 44, 66, 88, 132, 88, 66, 44, 44)$	83.6%

Table: *Percentage of strong keys.*

ATTACKS AGAINST THE PROPOSED CRYPTOSYSTEM

There are two main attacks to the McEliece PKC

- Plaintext recovery
  - ISD attacks on the full rank sliding matrix
  - Sequential plaintext recovery attacks
- Structural attacks

Let

$$\mathbf{y}_{\text{total}} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{s+m} \end{bmatrix},$$
$$\mathbf{u}_{\text{total}} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_s \end{bmatrix},$$
$$\mathbf{e}_{\text{total}} = \begin{bmatrix} \mathbf{e}_0 & \mathbf{e}_1 & \cdots & \mathbf{e}_{s+m} \end{bmatrix},$$

$$G_{\text{total}} = \begin{bmatrix} G'_0 & G'_1 & G'_2 & \cdots & G'_m & & & & \\ & G'_0 & G'_1 & G'_2 & \cdots & G'_m & & & \\ & & \ddots & \ddots & \ddots & & \ddots & & \\ & & & G'_0 & G'_1 & G'_2 & & \cdots & G'_m \end{bmatrix}.$$

An attacker could consider

$$\mathbf{y}_{\text{total}} = \mathbf{u}_{\text{total}} G_{\text{total}} + \mathbf{e}_{\text{total}}$$

Let

$$\mathbf{y}_{\text{total}} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{s+m} \end{bmatrix},$$
$$\mathbf{u}_{\text{total}} = \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_s \end{bmatrix},$$
$$\mathbf{e}_{\text{total}} = \begin{bmatrix} \mathbf{e}_0 & \mathbf{e}_1 & \cdots & \mathbf{e}_{s+m} \end{bmatrix},$$

$$G_{\text{total}} = \begin{bmatrix} G'_0 & G'_1 & G'_2 & \cdots & G'_m & & & & \\ & G'_0 & G'_1 & G'_2 & \cdots & G'_m & & & \\ & & \ddots & \ddots & \ddots & & \ddots & & \\ & & & G'_0 & G'_1 & G'_2 & \cdots & G'_m \end{bmatrix}.$$

An attacker could consider

$$\mathbf{y}_{\text{total}} = \mathbf{u}_{\text{total}} G_{\text{total}} + \mathbf{e}_{\text{total}}$$

Far too large matrices even with optimization of ISD algorithms

Sequential plaintext recovery attacks

If an attacker is able to obtain $\mathbf{u}_0, \mathbf{e}_0$, then $\Longrightarrow$
$D^{-1}(\mathbf{y}(D) - \mathbf{u}_0 G'(D) - \mathbf{e}_0)$ and attack $\mathbf{u}_1, \mathbf{e}_1$ and so on.

Sequential plaintext recovery attacks

If an attacker is able to obtain $\mathbf{u}_0, \mathbf{e}_0$, then $\Longrightarrow$ $D^{-1}(\mathbf{y}(D) - \mathbf{u}_0 G'(D) - \mathbf{e}_0)$ and attack $\mathbf{u}_1, \mathbf{e}_1$ and so on.

However, the equations that involve only $\mathbf{u}_0$ are represented by

$$\mathbf{u}_0 \widetilde{G} = \mathbf{y}_I + \mathbf{e}_I$$

and the code generated by the rows of $\widetilde{G}$ is $\widetilde{\mathcal{C}}$. If $G'(D)$ is a strong key then $\widetilde{\mathcal{C}}$ has distance equal to 1 and then recovering $\mathbf{u}_0$ is difficult in the presence of errors.

If one consider the code generated by $\mathcal{G} = UG\Delta\Gamma$, with $U \in \mathbb{F}^{k\times k}$ non singular, $\Delta \in \mathbb{F}^{n\times n}$ non singular diagonal and $\Gamma \in \mathbb{F}^{n\times n}$ a permutation matrix, then, any triplet

$$\{\mathcal{S}(D) = S(D)U^{-1},\, \mathcal{G} = UG\Delta\Gamma,\, \mathcal{P}(D^{-1}, D) = (\Delta\Pi)^{-1}P(D^{-1}, D)\}$$

can be used to decode the ciphertext.

Structural attacks

If one consider the code generated by $\mathcal{G} = UG\Delta\Gamma$, with $U \in \mathbb{F}^{k\times k}$ non singular, $\Delta \in \mathbb{F}^{n\times n}$ non singular diagonal and $\Gamma \in \mathbb{F}^{n\times n}$ a permutation matrix, then, any triplet

$$\{\mathcal{S}(D) = S(D)U^{-1},\ \mathcal{G} = UG\Delta\Gamma,\ \mathcal{P}(D^{-1}, D) = (\Delta\Pi)^{-1}P(D^{-1}, D)\}$$

can be used to decode the ciphertext.
Again, far too many possibilities

Public key sizes and ciphertext sizes

	n	k	m	s	WF Full Rank	Public Key	Ciphertext size
New	72	48	6	31	$2^{128.88}$	169344	19152
	72	48	10	32	$2^{130.16}$	266112	21672
	108	72	6	21	$2^{131.77}$	381024	21168
	108	72	6	47	$2^{257.22}$	381024	40824
	108	72	10	20	$2^{131.64}$	598752	23436
	120	84	6	19	$2^{130.65}$	493920	21840
	120	84	10	17	$2^{129.85}$	776160	23520
	120	84	10	45	$2^{259.47}$	776160	47040
	144	96	6	15	$2^{130.17}$	774144	25344
	144	108	10	40	$2^{259.51}$	1368576	58752
	144	108	10	83	$2^{512.95}$	1368576	108288
	180	120	6	28	$2^{256.46}$	1209600	50400
	180	132	6	63	$2^{513.10}$	1330560	100800
	180	132	10	31	$2^{260.38}$	2090880	60480
Classic McEliece	2960	2288			2^{128}	1537536	672
	6960	5413			2^{256}	8373911	1547
	8192	6528			2^{256}	10862592	1664
GRS with 2-weight mask	784	496			$2^{128.1}$	1428480	2880
	1820	1384			$2^{256.0}$	6637664	4360
Expanded RS	1258	1031			$2^{256.0}$	4624198	2724
Wild McEliece with extension degree 2	852	618			$2^{128.0}$	$\approx$ 712000	1170
	858	672			$2^{128.0}$	624960	930
	892	712			$2^{128.0}$	634930	900

Table: *Parameters, work forces and public key sizes (in bits) of PKC*

Conclusions

- The keys obtained are significantly smaller

Conclusions

- The keys obtained are significantly smaller
- The proposed scheme seems secure but many many possible variants using convolutional codes are possible, i.e, it allows a lot of flexibility (we are waiting for the attacks)
  - Use of convolutional codes with low degree instead of block code
  - Avoid starting and finishing from the zero state
  - Using particular matrices P for allowing more errors at the beginning, etc

- The keys obtained are significantly smaller
- The proposed scheme seems secure but many many possible variants using convolutional codes are possible, i.e, it allows a lot of flexibility (we are waiting for the attacks)
  - Use of convolutional codes with low degree instead of block code
  - Avoid starting and finishing from the zero state
  - Using particular matrices P for allowing more errors at the beginning, etc
- One main drawback is that the length of the messages are longer than the ones used in most common public-key encryption schemes (this seems difficult to avoid when using convolutional codes).

Thanks for your attention and the organization!