Recent progress in code-based cryptography

Laboratoire Hubert Curien, UMR CNRS 5516,
Bâtiment F 18 rue du professeur Benoît Lauras
42000 Saint-Etienne
France
pierre.louis.cayrel@univ-st-etienne.fr

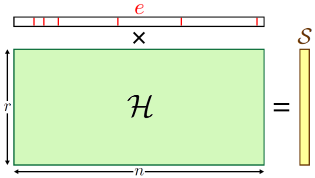
June, 21st 2012

Syndrome decoding problem

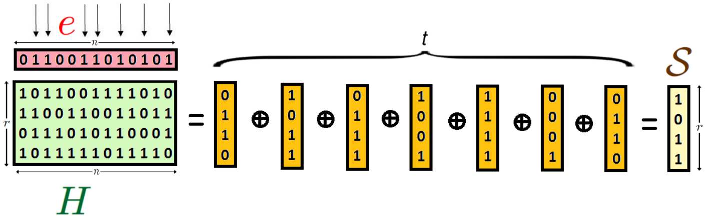
1. **Input.**

   $\mathcal{H}$: matrix of size $r \times n$
   $\mathcal{S}$: vector of $\mathbb{F}_2^r$
   t : integer

2. **Problem.** Does there exist a vector e of $\mathbb{F}_2^n$ of weight t such that :



- Problem NP-complete
  E.R. Berlekamp, R.J. McEliece and H.C. van Tilborg 1978


e
n
0 1 1 0 0 1 1 0 1 0 1 0 1
r
1 0 1 1 0 0 1 1 1 1 0 1 0
1 1 0 0 1 1 0 0 1 1 0 1 1
0 1 1 1 0 1 0 1 1 0 0 0 1
1 0 1 1 1 1 1 0 1 1 1 1 0
n
H
t
0 1 1 0
1 0 1 1
0 1 1 1
1 0 0 1
1 1 1 1
0 0 0 1
0 1 1 0
S
1 0 1 1
r

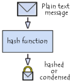
What can we do with this problem ?

- encryption

- signature

- identification

- hash function



- stream cipher

Menu

Error-correcting codes

- make possible the correction of errors when the communication is done on a noisy channel.
  - we add redundancy to the information transmitted.

$$\begin{array}{ccccc} & & \text{Noise} & & \\ & & \downarrow e & & \\ c = \boxed{m}\boxed{r} & \longrightarrow & \boxed{\text{Channel}} & \longrightarrow & c' = c + e \end{array}$$

  - by correcting the errors when the message is corrupted.
- stronger than a control of parity, they can detect and correct errors.

We use them :

- DVD,CD : reduce the effects of dust ...
- Phone : improve the quality of the communication.
- cryptography ?

Linear codes

- most used in error correction
- error correcting codes for which redundancy depends linearly on the information
- can be defined by a generator matrix :
  - c is a word of the code $\mathcal{C}$ if and only if :

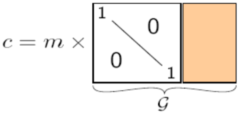


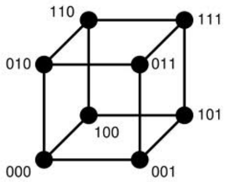
Figure: $\mathcal{G}$: generator matrix in systematic form

The generator matrix $\mathcal{G}$:

- is a $r \times n$ matrix;
- rows of $\mathcal{G}$ form a basis for the code $\mathcal{C}$.

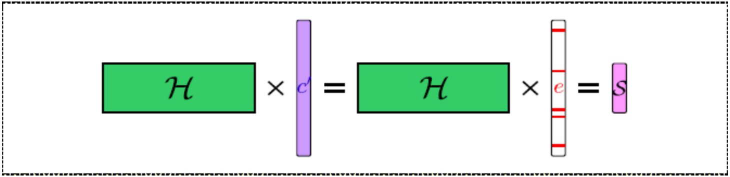
Minimum distance

- The Hamming weight of a word *c* is the number of non-zero coordinates.
- The minimum distance *d* of a code is the minimum of the Hamming weight between two words of the code.
- It is also the smallest weight of a non-zero vector.



The parity check matrix $\mathcal{H}$ is orthogonal to $\mathcal{G}$:

- it's a $r \times n$ matrix;
- it's the generator matrix of the dual;
- the code $\mathcal{C}$ is the kernel of $\mathcal{H}$.
  - $c \in \mathcal{C}$ if and only if $\mathcal{H} \cdot c = 0$.
- $\mathcal{S} = \mathcal{H} \cdot c' = \mathcal{H} \cdot c + \mathcal{H} \cdot e$ is the syndrome of the error.

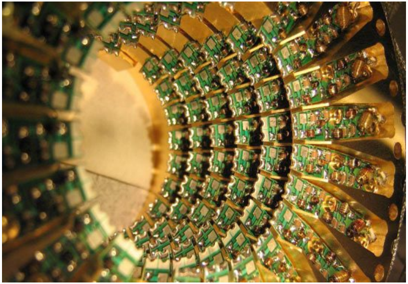


Code based cryptosystems

- introduced at the same time than RSA by McEliece

\+ advantages :

- faster than RSA ;
- not based on number theory problem (PQ secure) ;
- does not need cryptoprocessors ;
- based on hard problem (syndrome decoding problem ...)



– disadvantages :

- size of public keys (several thousand bits...)

Point of View

Top 25 Technology Predictions

By Dave Evans, Chief Futurist, Cisco IBSG Innovations Practice

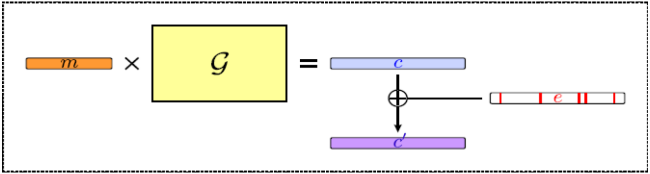
1. By 2029, 11 petabytes of storage will be available for $100—equivalent to 600+ years of continuous, 24-hour-per-day, DVD-quality video. (Source: Cisco IBSG, 2009)
2. In the next 10 years, we will see a 20-time increase in home networking speeds. (Source: Cisco IBSG, 2009)
3. By 2013, wireless network traffic will reach 400 petabytes a month. Today, the entire global network transfers 9 exabytes per month. (Source: FCC Head Julius Genachowski)
4. By the end of 2010, there will be a billion transistors per human—each costing one ten-millionth of a cent. (Sources: Intel Corporation; Cisco IBSG, 2006-2009; IBM)
5. The Internet will evolve to perform instantaneous communication, regardless of distance. (Source: Cisco IBSG, 2009)
6. The first commercial quantum computer will be available by mid-2020. (Source: Cisco IBSG, 2009)
7. By 2020, a $1,000 personal computer will have the raw processing power of a human brain. (Sources: Hans Moravec, Robotics Institute, Carnegie Mellon University, 1998; Cisco IBSG, 2006-2009)

How does the McEliece PKC work ?

- generate a code for which we have a decoding algorithm and $\mathcal{G}'$ the generator matrix.
  - this is the private key.

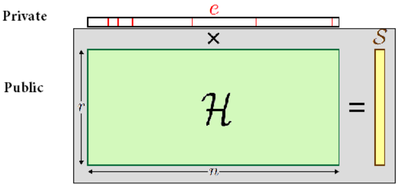
- transform $\mathcal{G}'$ to obtain $\mathcal{G}$ which seems random.
  - this is the public key.

- encrypt a message *m* by computing :
  - $c' = m \times \mathcal{G} \oplus e$ with *e* a random vector of weight *t*.



A dual construction using $\mathcal{H}$ instead of $\mathcal{G}$?

- Security equivalent to McEliece scheme.
- Private key :
  - $\mathcal{C}$ a $[n, r, d]$ code which corrects t errors,
  - $\mathcal{H}'$ a parity check matrix of $\mathcal{C}$,
  - a $r \times r$ invertible matrix Q,
  - a $n \times n$ permutation matrix P.
- Public key : $\mathcal{H} = Q\mathcal{H}'P$.
- Encryption :
  - $\phi_{n,t} : m \mapsto e$, with e of weight t.
  - $e \mapsto y = \mathcal{H}e$



- Decryption : decode $Q^{-1}y = (Q^{-1}Q)\mathcal{H}'Pe$ in Pe, then $P^{-1}Pe$ gives e.

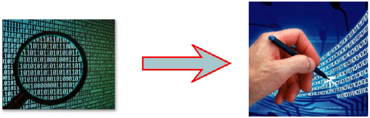
Hardware?

- Eisenbarth *et al.* "MicroEliece: McEliece for Embedded Devices", CHES'09.
- Shoufan *et al.* "A Novel Processor Architecture for McEliece Cryptosystem and FPGA Platforms", ASAP 2009
- Heyse. "Low-Reiter: Niederreiter Encryption Scheme for Embedded Microcontrollers", PQCrypto 2010
- Strenzke. "A Smart Card Implementation of the McEliece PKC", WISTP 2010
- Heyse. "CCA2 secure McEliece based on Quasi Dyadic Goppa Codes for Embedded Devices", PQCrypto 2011

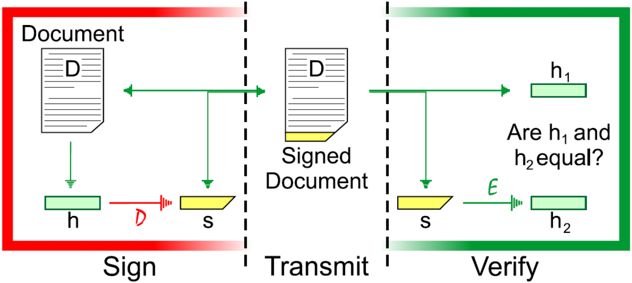
New results

- Decoding algorithms :
  - list-decoding algorithm (Bernstein 2011),
  - improved decoding algorithms (Barreto et al. 2011)
- Hardware :
  - side-channel :
    - DPA/SPA (Molter et al. 2011 and Avanzi et al. 2010),
    - timing attacks (Strenzke 2011)
  - efficient implementation :
    - QD-McEliece (Heyse 2011 and Cayrel et al. 2012),
    - Low-Reiter (Heyse 2010),
    - Micro-Eliece (Eisenbarth et al. 2010).
- Alternative to binary codes :
  - QC Alternant codes (Berger et al. 2009),
  - QD Goppa codes (Misoczki et al. 2010),
  - wild McEliece (Bernstein et al. 2010)
- Cryptanalysis :
  - decoding attacks (Finiasz et al. 2009, Bernstein et al. 2011, May et al. 2011, Becker et al. 2012),
  - structural attacks (Faugère et al. 2010 and Umana et al. 2009)

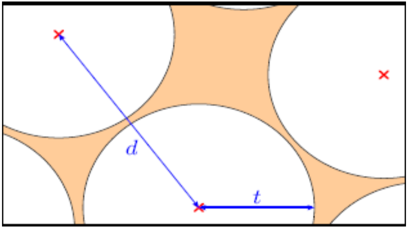
Several rank A+ conferences : Asiacrypt, Eurocrypt, Crypto, CHES



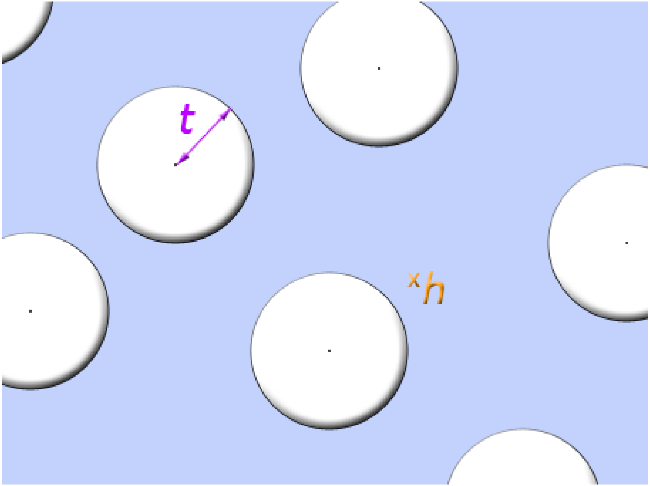
- PKC $\rightarrow$ signature.
  - RSA yes
  - McEliece and Niederreiter not directly

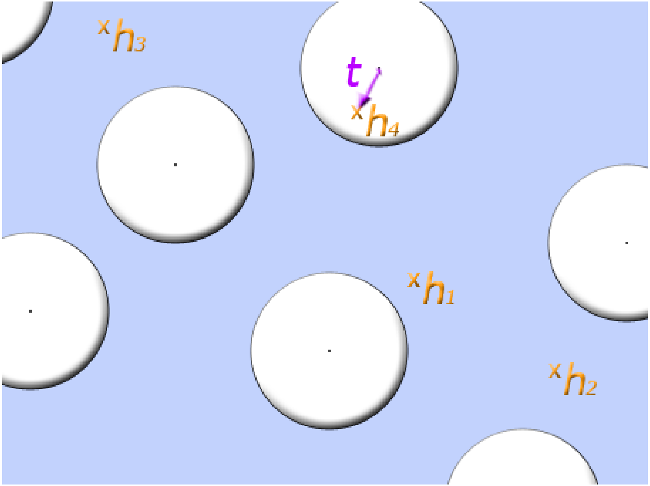


- **Problem:** McEliece and Niederreiter not invertible.
  - if we take $y \in \mathbb{F}_2^n$ random and a code $\mathcal{C}[n,k,d]$ for which we are able to decode $t = \frac{d}{2}$ errors, it is almost impossible to decode y in a word of $\mathcal{C}$.



- **Solution:**
  - the hash value has to be decodable !


t
x
h


h3
t
h4
h1
h2

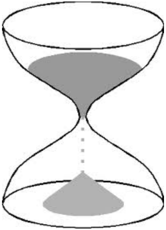
CFS signature scheme

- d the message to sign, we compute $M = h(d)$
- h a hash function with values in $\mathbb{F}_2^r$
  - we search $e \in \mathbb{F}_2^n$ of given weight t with $h(M) = \mathcal{H}e$
- let γ be a decoding algorithm ($i.e. \gamma(\mathcal{S}) = e$ *with* $\omega(e) = t$)

1. $i \leftarrow 0$
2. `while` $h(M|i)$ `is not a decodable syndrome do` $i \leftarrow i + 1$
3. `compute` $e = \gamma(h(M|i))$

Figure: CFS signature scheme

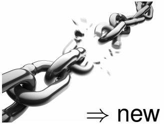
- signer sends $\{e, j\}$ such that $h(M|j) = \mathcal{H}e$



- we need a dense family of codes : Goppa codes
- binary Goppa codes $[2^m, 2^m - tm, 2t + 1]$
  - t small
  - the probability for a random element to be decodable (in a ball of radius t centered on the codewords) is $\approx \frac{1}{t!}$
- we take $n = 2^m, m = 16, t = 9.$
- we have 1 chance over $9! = 362880$ to have a decodable word.

signature cost	$t!t^2m^3$	12×10^{11} op. $\approx$ 1 min on FPGA
signature length	$(t-1) \times m + \log_2 t$	131 bits
verification cost	$t^2 m$	1 296 op.
PK size	$tm2^m$	1 MB

- cons :
  - decode several words ($t!$) before to find a good one
    - 70 times slower than RSA
  - t small leads to very big parameters
    - public key of 1 MB



$\Rightarrow$ new PK size : several MB, time to sign : several weeks ...

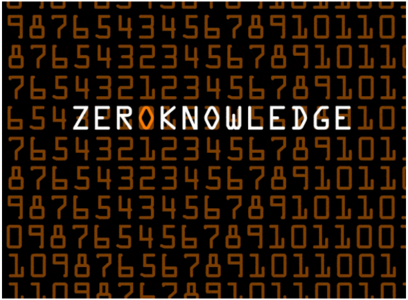
- solution : use structured codes (smaller public key size around 720 KB) and a GPU to have a signature in less than 2 minutes ...

1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

New results

- Signature algorithm :
  - Parallel CFS (Finiasz 2010);
  - One-Time Signature scheme (Barreto et al. 2010).
- Additional properties :
  - (Threshold)-ring signatures (Aguilar et al. 2009 and Dallot et al. 2009);
  - "blind" (Overbeck 2009).
- Alternative to binary Goppa codes :
  - QD Goppa codes (Barreto et al. 2010).
- Implementation :
  - software CFS (Landais et al. 2012).

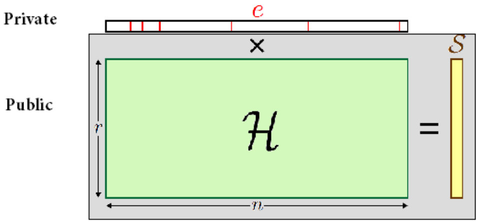
No efficient hardware implementations



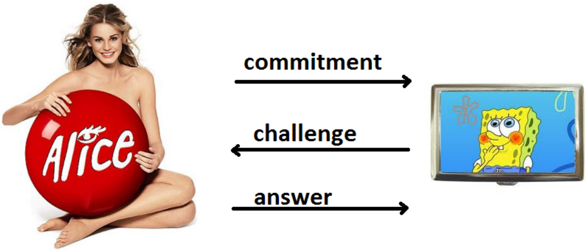
- zero-knowledge,
- the security is based on the syndrome decoding problem.

The protocol

- generate a random matrix $\mathcal{H}$ of size $r \times n$
- choose an integer t which is the weight
  - $(\mathcal{H}, t)$ are public
- each user chooses e of n bits and weight t.
  - this is the private key
- each user computes : $\mathcal{S} = \mathcal{H}e$.
  - just once for $\mathcal{H}$ fixed
  - $\mathcal{S}$ is public



- A wants to prove to B that she knows the secret e which satisfies $\mathcal{S} = \mathcal{H}e$ and $\omega(e) = t$ but she doesn't want to divulgate it.



- The protocol is on λ rounds and each of them is defined as follows.

A chooses y of n bits randomly and a permutation σ of $\{1, 2, \ldots, n\}$.
A sends to B : c_1, c_2, c_3 such that :

$$c_1 = h(\sigma|\mathcal{H}y);\ c_2 = h(\sigma(y));\ c_3 = h(\sigma(y \oplus e))$$

$\xrightarrow{\textbf{commitment}}$

A chooses y of n bits randomly and a permutation σ of $\{1, 2, \ldots, n\}$.
A sends to B : c_1, c_2, c_3 such that :

$$c_1 = h(\sigma|\mathcal{H}y);\ c_2 = h(\sigma(y));\ c_3 = h(\sigma(y \oplus e))$$

commitment $\longrightarrow$

challenge $\longleftarrow$

B sends to A a random $b \in \{0, 1, 2\}$.

A chooses y of n bits randomly and a permutation σ of $\{1, 2, \ldots, n\}$.
A sends to B : c_1, c_2, c_3 such that :

$$c_1 = h(\sigma|\mathcal{H}y);\ c_2 = h(\sigma(y));\ c_3 = h(\sigma(y \oplus e))$$

commitment →

← **challenge**

B sends to A a random $b \in \{0, 1, 2\}$.

Three possibilities:

1. if $b = 0$: A reveals y and σ
2. if $b = 1$: A reveals $(y \oplus e)$ and σ
3. if $b = 2$: A reveals $\sigma(y)$ and $\sigma(e)$

answer →

1. if $b = 0$: B checks that c_1, c_2 are correct
2. if $b = 1$: B checks that c_1, c_3 are correct
3. if $b = 2$: B checks that c_2, c_3 are correct and that $\omega(\sigma(e)) = t$

- for each round : probability to cheat is $\frac{2}{3}$.
  - for a security of $\frac{1}{2^{80}}$, we need 150 rounds.

Idea : Replace the random matrix $\mathcal{H}$ by the parity check matrix of a certain family of codes : *the double-circulant codes*.

- Let ℓ be an integer.
- a random double circulant matrix $\ell \times 2\ell$ $\mathcal{H}$ is defined as :

$$\mathcal{H} = (I|A) \ ,$$

where A is a *cyclic matrix*, of the form :

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_\ell \\ a_\ell & a_1 & a_2 & \cdots & a_{\ell-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_2 & a_3 & a_4 & \cdots & a_1 \end{pmatrix} ,$$

where $(a_1, a_2, a_3, \cdots, a_\ell)$ is a random vector of $\mathbb{F}_2^\ell$.

- Store $\mathcal{H}$ needs only ℓ bits.

- the minimum distance is the same as random matrices,
- the syndrom decoding is still hard,
- very interesting for implementation in low ressource devices.

- Let n equal 2ℓ
- **Private data :** the secret e of bit-length n.
- **Public data :** n bits ($\mathcal{S}$ of size ℓ and the first row of H, ℓ bits).

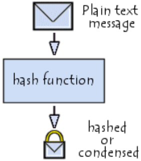
- at least $\ell = 347$ and $t = 74$ for a security of 2^{85}
- public and secret key sizes of $n = 694$ bits

New results

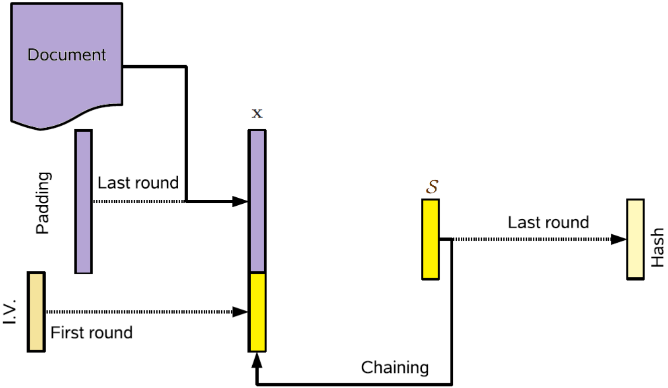
- Reducing soundness :
  - q-SD (Cayrel 2010);
  - Reduced communication (Aguilar et al. 2011)
- Efficient implementation :
  - Stern for low-resource devices (Cayrel et al. 2008)

No (other) efficient hardware implementations

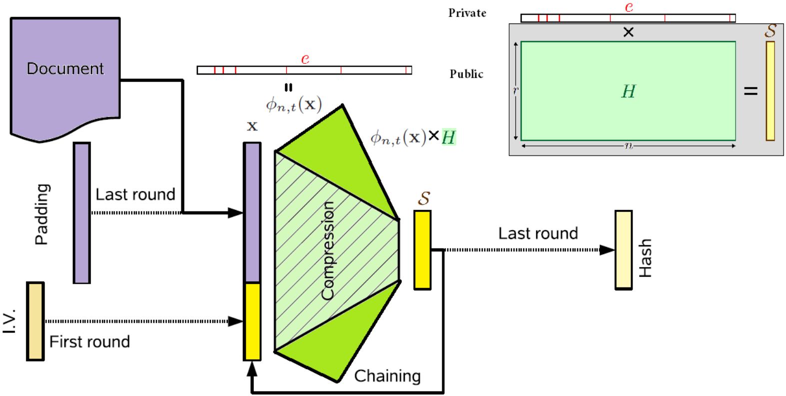
Hash-function and pseudo-random number generator



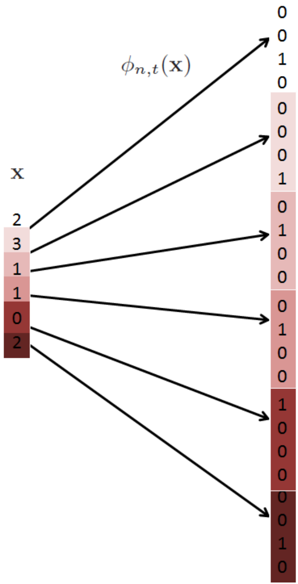
How to hash with codes ?



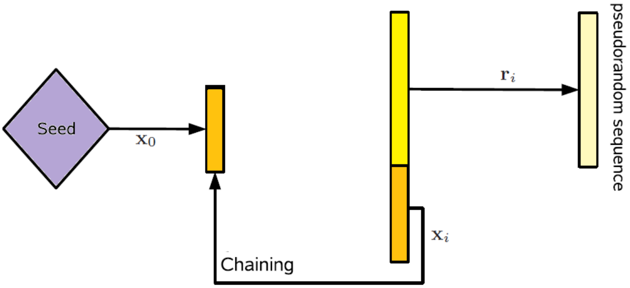
How to hash with codes ?



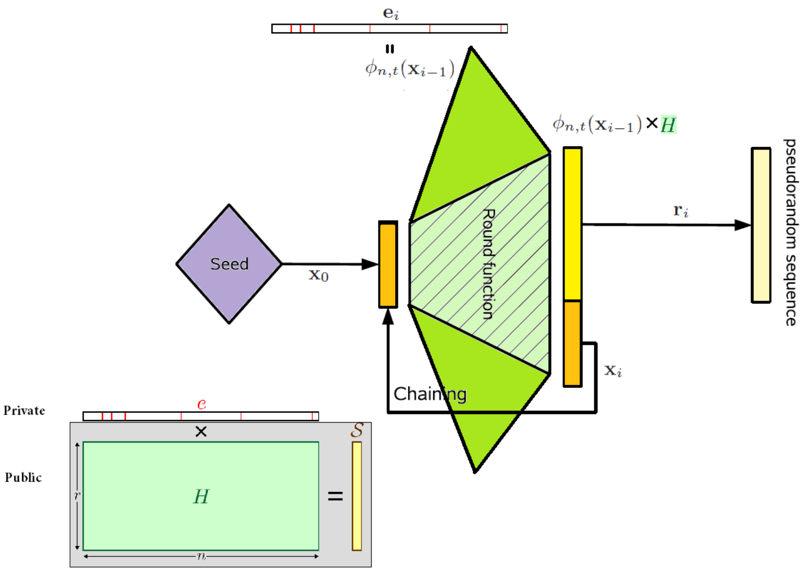
How $\phi_{n,t}$ could work?



How to generate pseudo-random sequences ?



How to generate pseudo-random sequences ?



New results

- Hash-function algorithms :
  - FSB (Augot et al. 2008);
  - SFSB (Meziani et al. 2011);
  - RFSB (Bernstein et al. 2011).

- Hash-function attacks :
  - QC codes (Fouque et al. 2008);
  - FSB day (Bernstein et al. 2009).

- Stream-cipher algorithms :
  - SYND (Finiasz et al. 2007);
  - 2SC (Meziani et al. 2011);
  - XSYND (Meziani et al. 2012).

No efficient hardware implementations

Encryption :

- Study of the QC/QD constructions ;
- Identity-based encryption.

Signature :

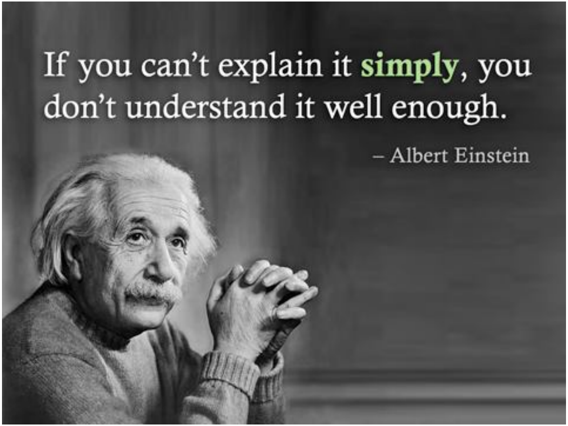
- FPGA implementation ;
- Smaller public keys.

Identification :

- 3-pass and soundness 1/2 ;
- Efficient implementation.

Secret-key :

- Fast schemes ;
- Study of side-channel attacks.


If you can't explain it simply, you don't understand it well enough.
– Albert Einstein