A Public Key encryption scheme based on the Polynomial Reconstruction problem

Daniel Augot Matthieu Finiasz

Eurocrypt 2003 – Warsaw

INRIA

Reedsolomon Codes

Definition

⇒ Reed-Solomon code of length n and dimension k

- ◇ Choose a set of n distinct points $\{x_1, \dots, x_n\}$ in a field (here $\mathbb{F}_{2^m}$). This is the support of the code.
- ◇ A message m is a polynomial of degree less than k over $\mathbb{F}_{2^m}$ (with $k < n$).
- ◇ The codeword c_m associated to the message m is its evaluation on the support: the n-tuple $(m(x_1), \dots, m(x_n))$.

As $k < n$ the transmitted codeword contains some redundancy: k values are enough to recover the polynomial m using interpolation.

⇒ if some errors are added to c_m, m can still be recovered using a decoding algorithm:

- ◇ Euclid's algorithm $\rightarrow$ correct up to $\frac{n-k}{2}$ errors
- ◇ Guruswami-Sudan algorithm $\rightarrow$ correct up to $n - \sqrt{nk}$ errors

Polynomial Reconstruction

Given n pairs $(x_i, y_i)_{i=1..n}$, find a polynomial $\mathcal{P}$ of degree less than k such that $\mathcal{P}(x_i) = y_i$ for at least t values of i.

⇒ if all x_i are distinct, this corresponds to decoding $n - t$ errors in a Reed-Solomon code of dimension k and length n

Possible attacks:

- ◇ exhaustive search on correct positions
- ◇ exhaustive search on wrong positions / decoding attack (Sudan algorithm)

⇒ as stated by Naor and Pinkas, if $\binom{n}{k}$ and $\binom{n}{t}$ are exponential in n and if $t < \sqrt{kn}$ the problem is hard

⚠ you also need $t > k + 1$ for the problem to be hard (interpolation)

The Cryptosystem
Preliminaries

The secret key of the system is composed of:

- ◇ a codeword c, evaluation of a polynomial of degree exactly $k - 1$
- ◇ an error pattern E of Hamming weight W

The public key is simply the sum $(c + E)$.

⇒ If W is well chosen, recovering the secret key from the public key is exactly an instance of the PR problem.

Messages to be encrypted are polynomials of degree $k - 2$ in $\mathbb{F}_{2^m}$.

The Cryptosystem

Encoding

ciphertext

element of $\mathbb{F}_{2^{80}}$

error of weight w

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

codeword of degree k−2
corresponding to m

public key

$\boxed{\text{codeword of degree k−1} \; + \; \text{error of weight W}}$

The Cryptosystem

Decoding

$\Rightarrow$ First shorten the code on the positions for which E is non-zero. We get:

$$\bar{y} = \bar{c}_m + \alpha\,\bar{c} + \cancel{\alpha\bar{E}} + \bar{e}$$

$\bar{c}_m + \alpha\,\bar{c}$ belongs to the shortened code and $\bar{e}$ is an error pattern of weight smaller or equal to w

$\Rightarrow$ if w is well chosen, one can decode $\bar{y}$ in the shortened code

$\Rightarrow$ the polynomial of degree $k-1$ corresponding to $c_m + \alpha\,c$ can be recovered

- c_m was chosen of degree $k-2$
- c is known (it's part of the secret key)
- α can be found by looking at the term of degree $k-1$
- c_m can then be recovered and so m too

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

Attacks

Note that once you know any of α, e or m you can get the two others, however you get no information at all about the secret key.

$\Rightarrow$ we distinguish two independent categories of attacks

* Secret Key recovery
  - search on good positions
  - search on error positions

* Message recovery $\sim$ decoding in a Reed-Solomon code plus one word $(c+E)$
  - exhaustive search on α
  - search on error positions (try to find e)
  - search on good positions (try to find m)

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

Secret Key recovery

$\Rightarrow$ Recovering the secret key is as difficult as solving an instance of the Polynomial Reconstruction problem

However some attacks exist:

$\Rightarrow$ Error Set Decoding: takes full advantage of the code structure. Shorten the code on β random positions (hoping they correspond to non-null positions of E) and try to decode in the shortened code.

$\Rightarrow$ You can't choose a W too close to the Sudan bound

$\Rightarrow$ Information Set Decoding: consider the code as a random code and try to find k positions containing no errors.

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

Message Recovery

$\Rightarrow$ Decoding in RS+1: that is decoding in the code of dimension $k+1$

$\Rightarrow$ exhaustive search on α

$\Rightarrow$ algebraic method ?

$\Rightarrow$ Error Set Decoding: consists in shortening the code on some positions (hoping they were erroneous) and try to decode, but there is no decoding algorithm

$\Rightarrow$ this is of no use

$\Rightarrow$ Information Set Decoding: exactly as for Key Recovery except the dimension of the code is one more, and the error is of smaller weight

$\Rightarrow$ efficient when W is large as $w = n - W - \sqrt{(n-W)k}$

Note that instead of ISD attacks, the Canteaut-Chabaud algorithm can be used as it is far more efficient than exhaustive search.

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

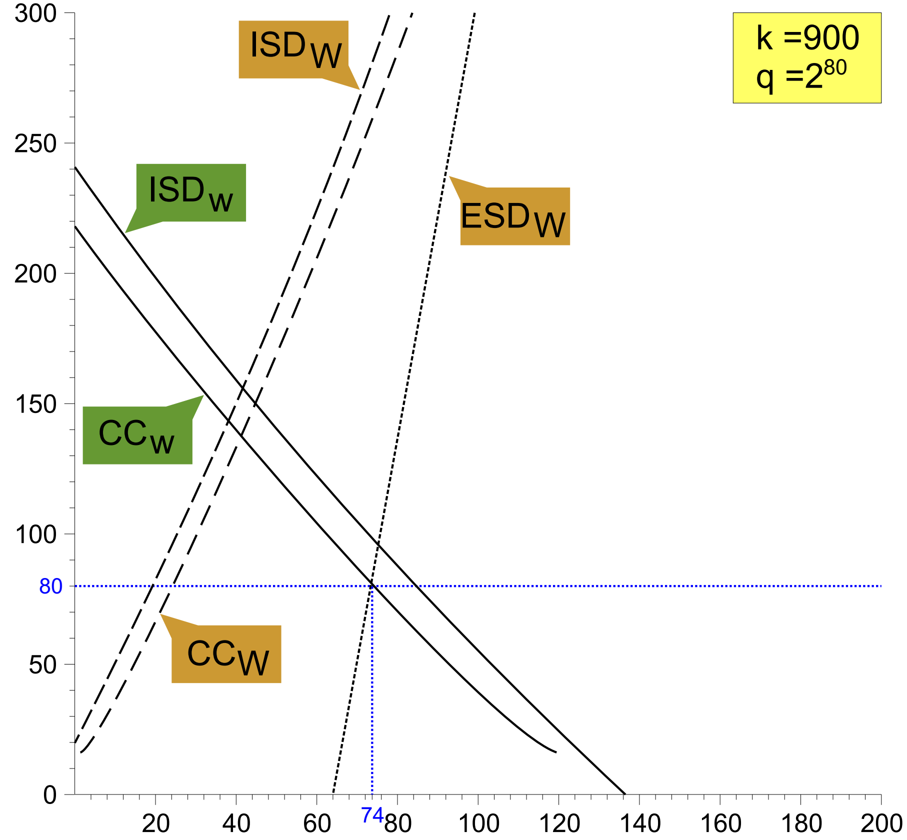
Secure Parameters

As usual, we intend to reach a security of 2^{80} binary operations.

$\Rightarrow$ n can't be very small: that is at least 1024

$\Rightarrow$ We choose $k = 900$

$\Rightarrow$ optimal for the transmission rate $\frac{k}{n}$



security against the different attacks as a function of W

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

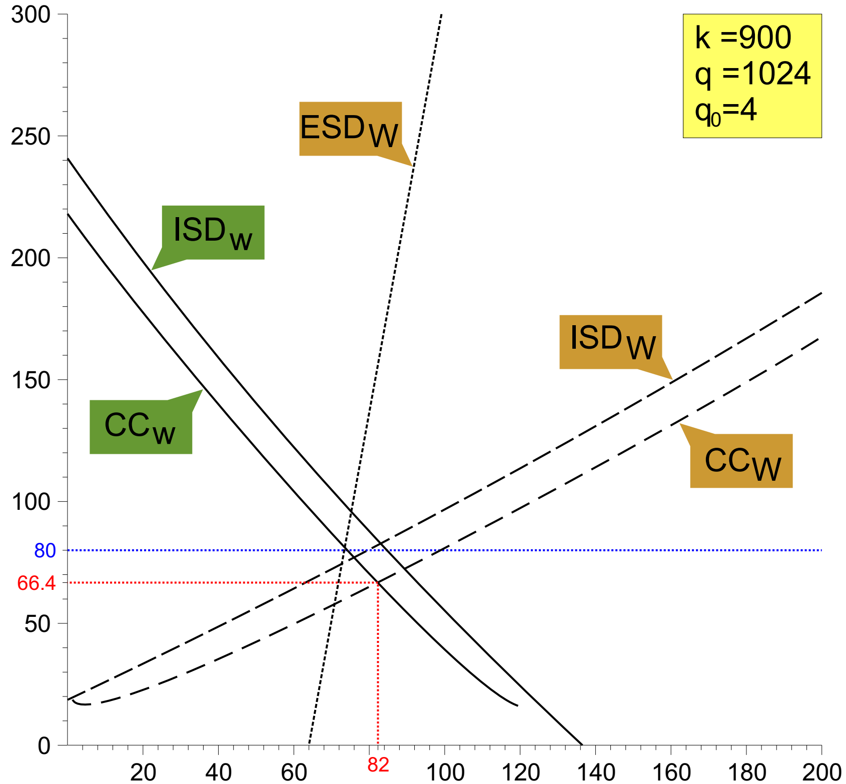
Shortening the public key

Parameters are: $n = 1024$ and $\mathbb{F}_q = \mathbb{F}_{2^{80}}$
$\Rightarrow$ the public key is $80 \times 1024 = 81920$ bits long

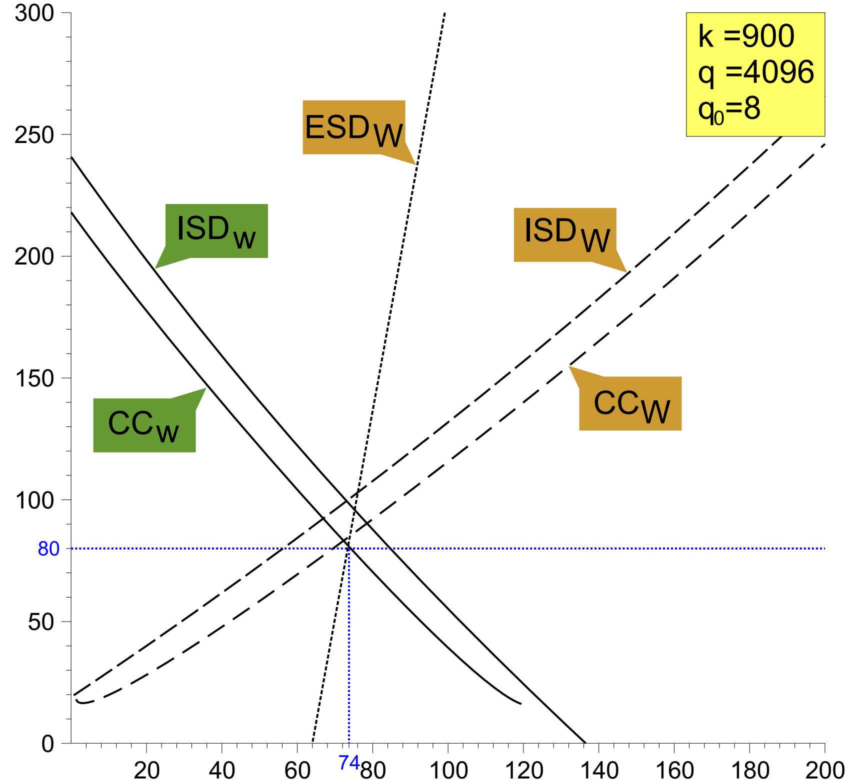
We can shorten this key by considering a subfield-subcode
$\Rightarrow$ the support is of length 1024 so we can use the subcode over $\mathbb{F}_{2^{10}}$ without any loss of dimension.
 $\Rightarrow$ the public key is $c + E$ with c a code word of the $[1024, 900]_{2^{10}}$ RS and E an error of weight W with coordinates in $\mathbb{F}_{2^{10}}$. Encryption is still done in $\mathbb{F}_{2^{80}}$
$\Rightarrow$ Now the key is 10240 bits long

We can still shorten the key with subfield-subcodes
$\Rightarrow$ this time we accept a dimension loss and consider the subcode $[1024, k']_{2^2}$
 $\Rightarrow$ we have $n - k' = 5 \times (n - k)$, that is $k' = 404$
$\Rightarrow$ the key would be 2048 bits long, but the system can no longer be secure

$$y = c_m + \boxed{\alpha}\boxed{(c + E)} + \boxed{e}$$



with the dimension loss ISD_W and CC_W become too easy and the system is insecure



by placing ourselves in $\mathbb{F}_{2^{84}}$ we can optimize the dimension loss. The key is 3072 bits long

$$y = c_m + \boxed{\alpha}\boxed{(c + E)} + \boxed{e}$$

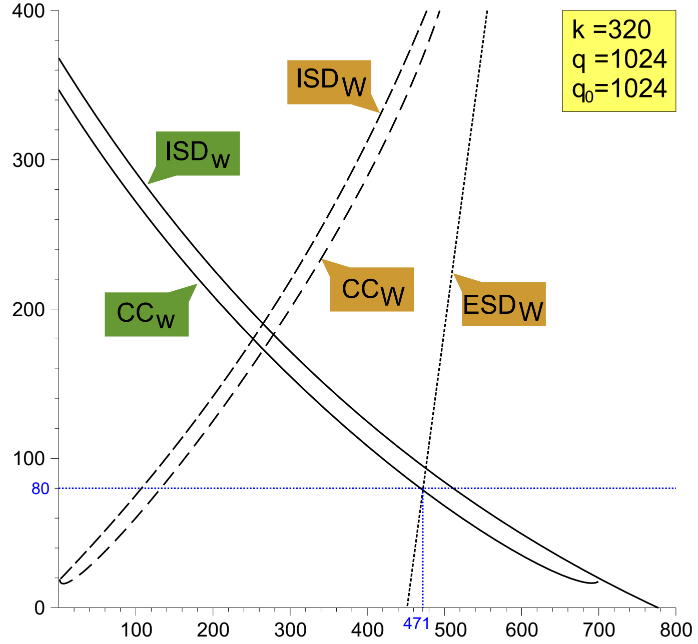
Efficiency

The optimal version of the scheme has the following properties:

- public key size: 3072 bits
- transmission rate: $\frac{k-1}{n} = 0.88$ for $k = 900$
- encryption complexity: $O(n \log q)$ per bit
- decryption complexity: $O(\frac{(n-W)^2}{k} \log q)$ per bit of plaintext
- block size: 75600 bits of plaintext

$\Rightarrow$ decryption can go faster for a large W

$\Rightarrow$ we can use $k = 320$ and $W = 470$

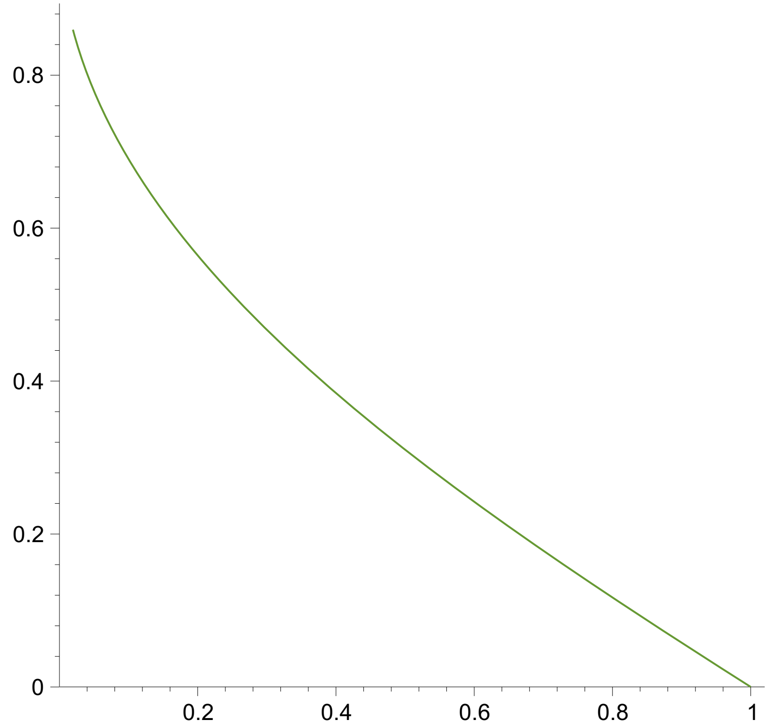


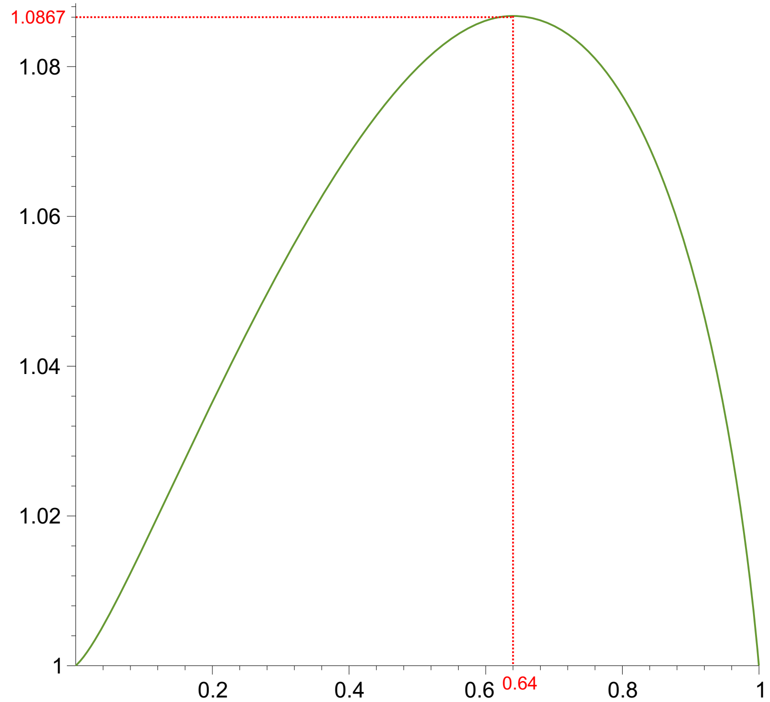
$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

Asymptotic Behavior

We want to see if the security is scalable

$\Rightarrow$ all the parameters of the system are linear in n





Optimal value of $\frac{W}{n}$ *as a function of* $\frac{k}{n}$ S *as a function of* $\frac{k}{n}$, $Security = S^n$

With $n = 1024$ one could reach a security as high as 2^{122}

$$y = c_m + \boxed{\alpha}\boxed{(c + E)} + \boxed{e}$$

. . .

We can evaluate precisely the security of this system against all kinds of attack, except the Decoding in RS+1 attack

$\Rightarrow$ Attack by J.-S. Coron: takes advantage of the code structure and recovers the message in a few minutes

How can the system be fixed?

- $\diamond$ change the system parameters
- $\diamond$ change the kind of code used
- $\diamond$ change the way the public key is added to c_m

$$y = c_m + \boxed{\alpha}\boxed{(c+E)} + \boxed{e}$$

Conclusion

We obtain a new public key cryptosystem

- $\star$ very easy to generate keys in large number
- $\star$ fast encryption/decryption
- $\star$ true exponential security against most attacks
- $\star$ possibility to have transmission rates close to 1
- $\star$ resistant to quantum computing

But it first needs a little fix. . .