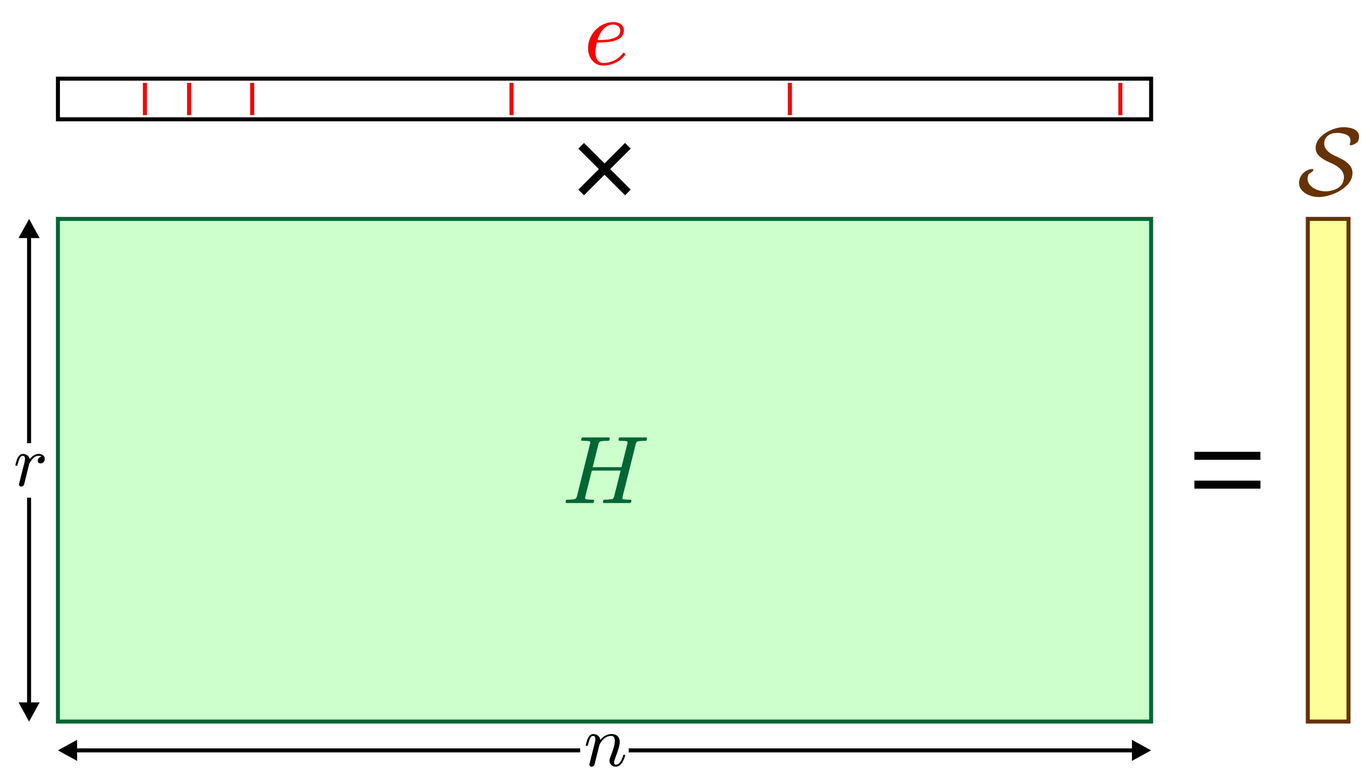
Security Bounds for the Design of Code-Based Cryptosystems

M. Finiasz and N. Sendrier

ENSTA ParisTech

INRIA

The Syndrome Decoding Problem



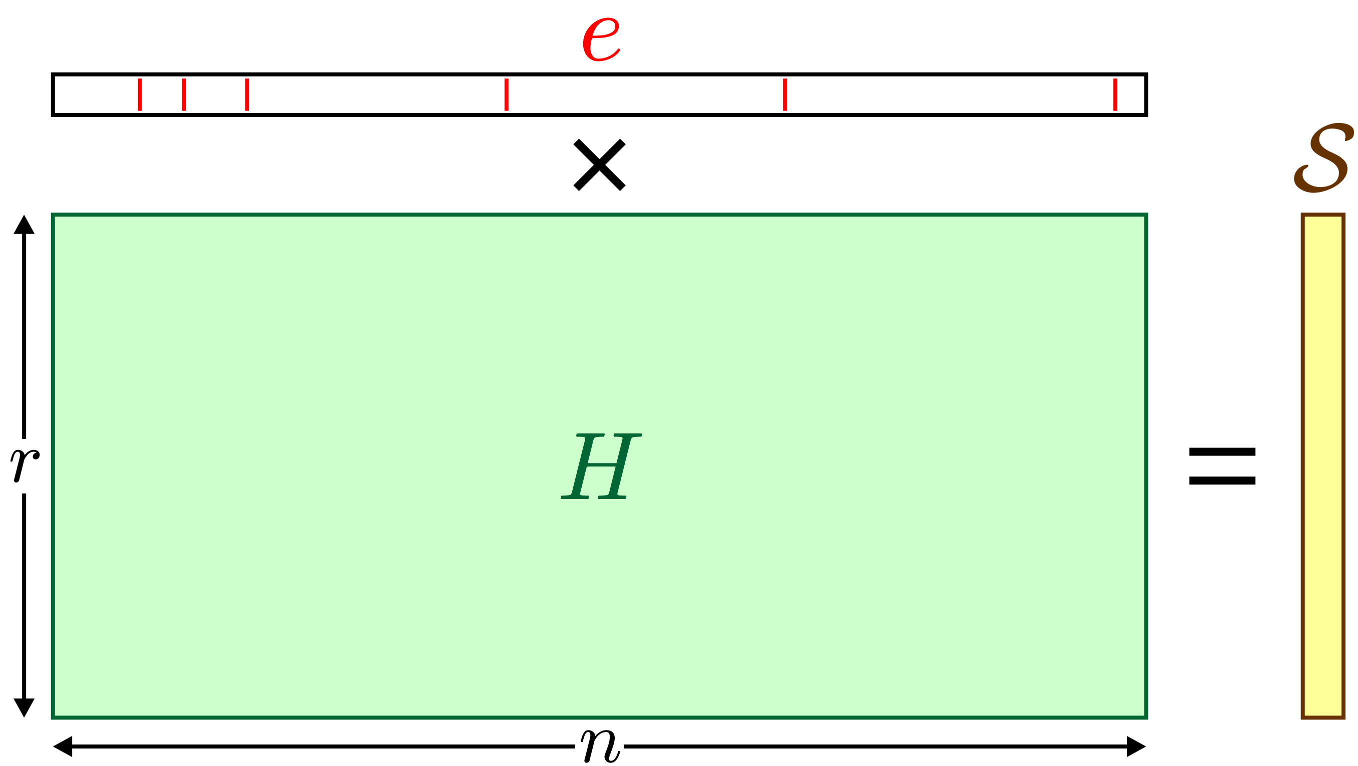
Syndrome Decoding (SD)

Does $e \in \{0,1\}^n$ of weight $\leq w$ such that $e \times H = \mathcal{S}$ exist?

▷ NP-complete problem.

[Berlekamp, McEliece, van Tilborg - 1978]

The Syndrome Decoding Problem



Computational Syndrome Decoding (CSD)

Find $e \in \{0,1\}^n$ of weight $\leq w$ such that $e \times H = \mathcal{S}$.

▷ The security of most code-based cryptosystems relies on the difficulty of solving this problem.

Our Point of View

- Depending on parameters (n, r, w), what is the difficulty of solving CSD?
  - ▷ we are looking for a lower bound:
    $\longrightarrow$ any attack on the system costs at least this.

- There are three families of attacks to look at:
  - ▷ we describe an idealized version of each attack,
    $\longrightarrow$ trying to take into account improvements to come.
  - ▷ we propose a lower bound for each of them (or an approximation of a lower bound).

Birthday Algorithm

- Build a list/hash table of XORs of $\frac{w}{2}$ columns of H:
  - look for 2 equals elements in this set
    - $\longrightarrow$ each such pair gives a solution to the CSD instance.

- The size L of the list to build is:
  - if $\binom{n}{w} > 2^r$ then $L = 2^{\frac{r}{2}}$,
  - else, if the problem has a single solution, $L = \binom{n}{\frac{w}{2}}$.

- In both cases, the complexity is $O(L \log L)$ with regards to time or memory.

- The basic technique has 2 drawbacks:
  - one manipulates r-bit long XORs,
  - in the second case, the solution is found $\frac{1}{2}\binom{w}{\frac{w}{2}}$ times.

- We thus improve/idealize the algorithm accordingly:
  - introduce a "window" of size ℓ
    - $\longrightarrow$ does not improve the asymptotic complexity,
  - store a list of smaller size.

- W_1 et W_2 are subsets of the words of weight $\frac{w}{2}$.

input: $H_0 \in \{0,1\}^{r \times n}$, $s \in \{0,1\}^r$

```
repeat                                                     (MAIN LOOP)
    P ← random n × n permutation matrix
    H ← H_0 P
    for all e ∈ W_1
        i ← h_ℓ(e H^T)                                     (BA 1)
        write(e, i)             // store e at index i of a structure
    for all e_2 ∈ W_2
        i ← h_ℓ(s + e_2 H^T)                               (BA 2)
        S ← read(i)       // extract the elements stored at index i
        for all e_1 ∈ S
            if e_1 H^T = s + e_2 H^T                       (BA 3)
                return (e_1 + e_2) P^T                     (SUCCESS)
```

- We make two assumptions:
  - for all pairs of words (e_1, e_2), the sum $e_1 + e_2$ is uniformly distributed,
  - if K_0 is the cost of a complete test, the total cost is:
$$\ell \cdot \sharp(\text{BA } 1) + \ell \cdot \sharp(\text{BA } 2) + K_0 \cdot \sharp(\text{BA } 3).$$
- Then, the cost of solving an instance of CSD is lower bounded by:

$$\boxed{\mathrm{WF}_{\mathrm{BA}}(n, r, w) = 2L \log(K_0 L) \text{ with } L = \min\left(\sqrt{\binom{n}{w}}, 2^{r/2}\right).}$$

$\rightarrow$ L is the size of W_1 and, in average, of W_2.

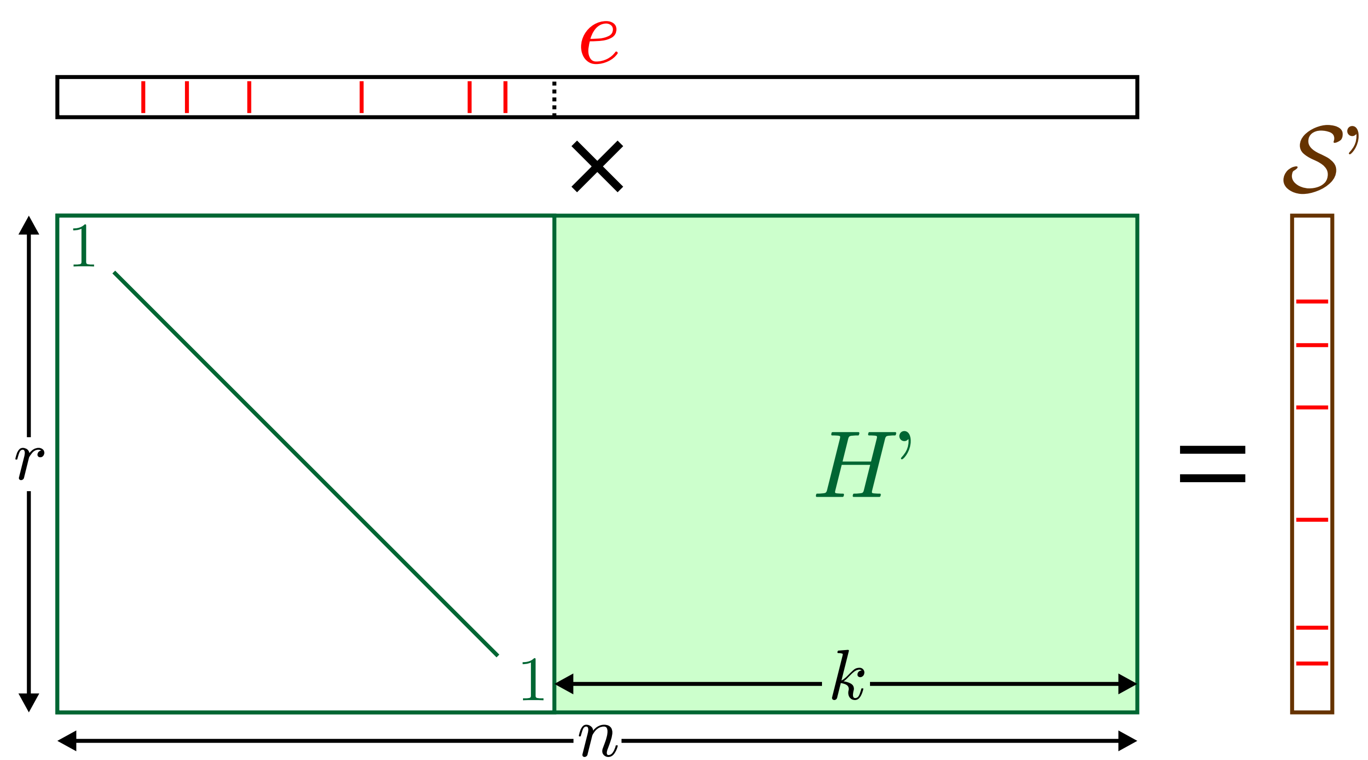
- We make two assumptions:
  - for all pairs of words (e_1, e_2), the sum $e_1 + e_2$ is ~~uniformly distributed~~,
  - if K_0 is the cost of a complete test, the total cost is:
$$\ell \cdot \sharp(\text{BA } 1) + \ell \cdot \sharp(\text{BA } 2) + K_0 \cdot \sharp(\text{BA } 3).$$
- Then, the cost of solving an instance of CSD is lower bounded by:

$$\boxed{\mathrm{WF}_{\mathrm{BA}}(n,r,w) = \sqrt{2}L\log(K_0L) \text{ with } L = \min\left(\sqrt{\binom{n}{w}}, 2^{r/2}\right).}$$

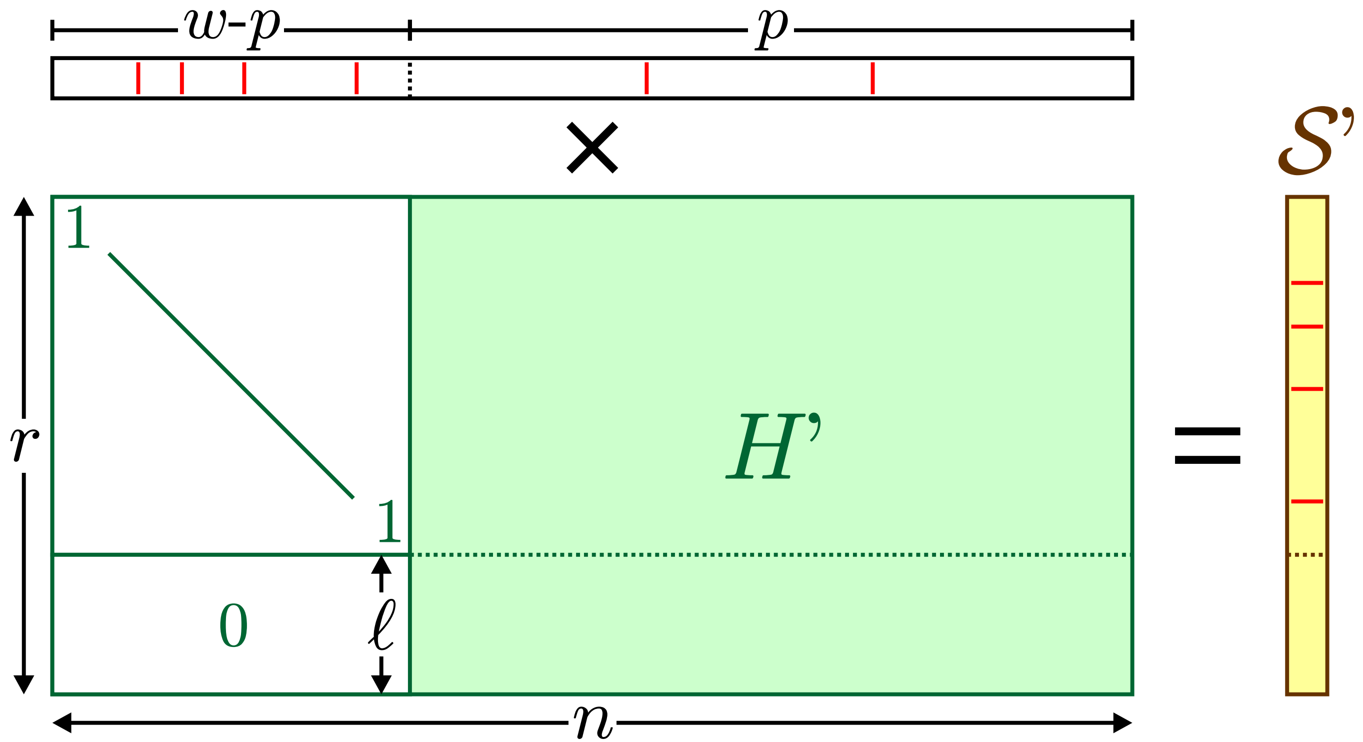
$\longrightarrow$ the attacker might choose better sets W_1 and W_2.

Information Set Decoding (ISD)

- The idea is to look for an information set:
  → a set of k positions containing no errors.
- For CSD, this is equivalent to finding a set of r columns of H containing the w positions of a solution.



- Each Gaussian elimination tests $\binom{r}{w}$ solution candidates,
  - ▷ we want to increase this number.
- We introduce two parameters ℓ and p. [Stern 1989]
  - ▷ equality on a window of size ℓ → birthday algorithm.



- W_1 and W_2 are words of weight $\frac{p}{2}$ and length $k + \ell$.

input: $H_0 \in \{0,1\}^{r \times n}$, $s_0 \in \{0,1\}^r$

```
repeat                                              (MAIN LOOP)
   P ← random n × n permutation matrix
   (H', U) ← PGElim(H_0 P)          // partial Gaussian elimination
   s ← s_0 U^T
   for all e ∈ W_1
      i ← h_ℓ(e H'^T)                                   (ISD 1)
      write(e, i)                // store e at index i of a structure
   for all e_2 ∈ W_2
      i ← h_ℓ(s + e_2 H'^T)                             (ISD 2)
      S ← read(i)       // extract the elements stored at index i
      for all e_1 ∈ S
         if wt(s + (e_1 + e_2) H'^T) = w − p             (ISD 3)
            return (P, e_1 + e_2)                       (SUCCESS)
```

- Again, we make two assumptions:
  - for all pairs of words (e_1, e_2), the sum $e_1 + e_2$ is uniformly distributed,
  - if K_{w-p} is the cost of an $\textsc{isd}\ 3$ test, the total cost is:
    $$\ell \cdot \sharp(\textsc{isd}\ 1) + \ell \cdot \sharp(\textsc{isd}\ 2) + K_{w-p} \cdot \sharp(\textsc{isd}\ 3).$$
- For a CSD instance with a single solution:

$$\boxed{\mathrm{WF}_{\mathrm{ISD}}(n, r, w) \approx \min_p \frac{2\ell\binom{n}{w}}{\lambda\binom{r-\ell}{w-p}\sqrt{\binom{k+\ell}{p}}} \text{ with } \ell = \log\left(K_{w-p}\sqrt{\binom{k}{p}}\right).}$$

- With $\lambda = 1 - e^{-1}$, success probability of the "birthday".

- When $\binom{n}{w} > 2^r$, we distinguish between 2 cases:
  - either ISD 3 has less than a solution: $\binom{r}{w-p}\binom{k}{p} \ll 2^r$
    $\longrightarrow$ a similar formula applies,

$$\boxed{\mathrm{WF}_{\mathrm{ISD}}(n,r,w) \approx \min_p \frac{2\ell\, 2^r}{\lambda\binom{r-\ell}{w-p}\sqrt{\binom{k+\ell}{p}}} \text{ with } \ell = \log\left(K_{w-p}\sqrt{\binom{k}{p}}\right).}$$

  - or ISD 3 has several solutions: $\binom{r}{w-p}\binom{k}{p} > 2^r$
    $\longrightarrow$ a single iteration is enough, using smaller lists,

$$\boxed{\mathrm{WF}_{\mathrm{ISD}}(n,r,w) \approx \min_p \frac{2\ell\, 2^{r/2}}{\sqrt{\binom{r-\ell}{w-p}}} \text{ with } \ell = \log\left(K_{w-p}\frac{2^{r/2}}{\sqrt{\binom{r}{w-p}}}\right).}$$

- Not always very tight, especially for intermediate cases...

Generalized Birthday Algorithm (GBA)

- We first look at a modified problem with $f : \mathbb{N} \rightarrow \{0,1\}^r$
  - → Find $x_0, ..., x_{2^a-1} \in \mathbb{N}$ such that $\bigoplus_i f(x_i) = 0$.
  - ▷ We no longer have a length constraint n and w is a power of 2.
  - ▷ There is an infinite number of solutions.
- With the standard birthday algorithm:
  - ▷ pick a list W_1 of XORs of 2^{a-1} vectors $f(x_i)$,
  - ▷ same for W_2 and then look for collisions,
    - → the list size has to be $2^{r/2}$.
  - ▷ we do not benefit from the infinite number of solutions...

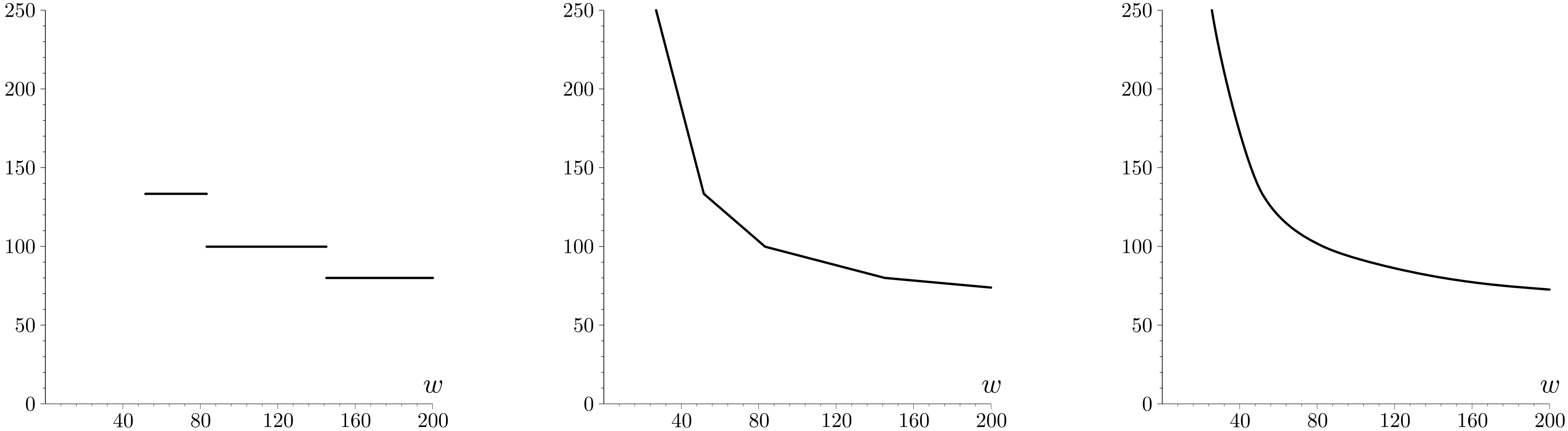
- Lists W_1 and W_2 are built so as to help collisions: elements are not chosen at random.
  - Start with 2^a lists $L_0, \ldots L_{2^a-1}$ each containing $2^{\frac{r}{a+1}}$ vectors $f(x_i)$,
  - pairwise merge lists L_{2j} and L_{2j+1} to obtain 2^{a-1} lists L'_j of XORs of 2 $f(x_i)$. Keep only elements starting with $\frac{r}{a+1}$ zeros.
    $\longrightarrow$ the L'_j still contain $2^{\frac{r}{a+1}}$ elements in average.
  - similarly merge again until 2 lists of XORs of 2^{a-1} vectors starting with $\frac{(a-1)r}{a+1}$ zeros remain.
- We end up with a single solution in average, and all manipulated lists are of size $2^{\frac{r}{a+1}}$.

- If w is not a power of 2:
  - ▷ choose different size lists $\longrightarrow$ difficult to analyse,
  - ▷ we only consider lists of XORs of $\frac{w}{2^a}$ elements.

- When the length constraint n is added:
  - ▷ the starting lists may be too small,
    $\longrightarrow$ use a smaller a and higher weight starting elements.
  - ▷ all lists contain the same elements,
    $\longrightarrow$ less distinct elements in the merged lists.

- We build the lists L'_j so that they only contain unique elements, bringing us back to the general case.

- We select 2^{a-1} distinct a-bit vectors s_j such that:
$$\bigoplus s_j = 0$$
  - ▷ in the L'_j lists we keep the XORs of weight $\frac{w}{2^{a-1}}$ having s_j as their first a bits,
  $\longrightarrow$ the $\binom{n}{w/2^{a-1}}$ possible vectors are distributed among the 2^{a-1} lists.
  - ▷ we then use GBA normally on vectors of length $r - a$.
- We obtain the following constraint on a:
$$\frac{1}{2^a}\binom{n}{\frac{2w}{2^a}} \geq 2^{\frac{r-a}{a}}.$$
  - ▷ The complexity of the attack is then $\frac{r-a}{a} 2^{\frac{r-a}{a}}$.



(a) (b) (c)

- Integer values for a give a complexity curve like (a),
  - zeroing a few bits in the lists L_j we obtain (b).

- Almost the same as a using non-integer values (c)
  - $\longrightarrow$ this is what should be used in our bound.

Bound on GBA applied to CSD

- Our complexity considers an idealized algorithm:
  - XORs of non-integer numbers of vectors,
  - non-integer number of lists,

    $\longrightarrow$ impossible to achieve better with GBA.

- For any parameter set (n, r, w) of CSD we have:

$$\boxed{\mathrm{WF}_{\mathrm{GBA}}(n, r, w) \geq \frac{r-a}{a} 2^{\frac{r-a}{a}} \text{ with } a \text{ such that } \frac{1}{2^a}\binom{n}{\frac{2w}{2^a}} = 2^{\frac{r-a}{a}}.}$$

Application to some Existing Cryptosystems

- We have to solve instances of CSD with a single "unexpected" solution,
  - below the Gilbert-Varshamov bound.
  - GBA can not be applied ($a < 1$ in the formula).
- Our bound on ISD gives a good approximation:

(m, w)	optimal p	optimal ℓ	binary work factor
$(10, 50)$	4	22	$2^{59.9}$
$(11, 32)$	6	33	$2^{86.8}$
$(12, 41)$	10	54	$2^{128.5}$

- In the $(10, 50)$ case, Canteaut-Chabaud costs $2^{64.2}$ and Bernstein-Lange-Peters $2^{60.5}$.

- Parameters similar to those of encryption:
  - ▷ only one instance out of $w!$ has a solution,
  - ▷ unlimited number of target syndromes,
    $\longrightarrow$ for GBA, we can use a syndrome list in addition.
    [Bleichenbacher]

- We use an unbalanced GBA: 3 small lists of XORs of columns of H, one large list of syndromes.
  - ▷ XORs of $\lceil \frac{w}{3} \rceil$, $w - \lceil \frac{w}{3} \rceil - \lfloor \frac{w}{3} \rfloor$ and $\lfloor \frac{w}{3} \rfloor$ columns,
  - ▷ we can't us any idealization (the gap is too large),
    $\longrightarrow$ still we can give practical complexities.

- The time and memory complexities are respectively $O(\mathcal{T}\log\mathcal{T})$ and $O(\mathcal{M}\log\mathcal{M})$.

  If $\frac{2^r}{\binom{n}{w-\lfloor w/3\rfloor}} \geq \sqrt{\frac{2^r}{\binom{n}{\lfloor w/3\rfloor}}}$:

$$\mathcal{T} = \frac{2^r}{\binom{n}{w-\lfloor w/3\rfloor}} \text{ and } \mathcal{M} = \frac{\binom{n}{w-\lfloor w/3\rfloor}}{\binom{n}{\lfloor w/3\rfloor}},$$

  otherwise:

$$\mathcal{T} = \mathcal{M} = \sqrt{\frac{2^r}{\binom{n}{\lfloor w/3\rfloor}}}.$$

- The time and memory complexities are respectively $O(\mathcal{T} \log \mathcal{T})$ and $O(\mathcal{M} \log \mathcal{M})$.

	$w = 8$	$w = 9$	$w = 10$	$w = 11$	$w = 12$
$m = 15$	$2^{51.0}/2^{51.0}$	$2^{60.2}/2^{43.3}$	$2^{63.1}/2^{55.9}$	$2^{67.2}/2^{67.2}$	$2^{81.5}/2^{54.9}$
$m = 16$	$2^{54.1}/2^{54.1}$	$2^{63.3}/2^{46.5}$	$2^{66.2}/2^{60.0}$	$2^{71.3}/2^{71.3}$	$2^{85.6}/2^{59.0}$
$m = 17$	$2^{57.2}/2^{57.2}$	$2^{66.4}/2^{49.6}$	$2^{69.3}/2^{64.2}$	$2^{75.4}/2^{75.4}$	$2^{89.7}/2^{63.1}$
$m = 18$	$2^{60.3}/2^{60.3}$	$2^{69.5}/2^{52.7}$	$2^{72.4}/2^{68.2}$	$2^{79.5}/2^{79.5}$	$2^{93.7}/2^{67.2}$
$m = 19$	$2^{63.3}/2^{63.3}$	$2^{72.5}/2^{55.7}$	$2^{75.4}/2^{72.3}$	$2^{83.6}/2^{83.6}$	$2^{97.8}/2^{71.3}$
$m = 20$	$2^{66.4}/2^{66.4}$	$2^{75.6}/2^{58.8}$	$2^{78.5}/2^{76.4}$	$2^{87.6}/2^{87.6}$	$2^{101.9}/2^{75.4}$
$m = 21$	$2^{69.5}/2^{69.5}$	$2^{78.7}/2^{61.9}$	$2^{81.5}/2^{80.5}$	$2^{91.7}/2^{91.7}$	$2^{105.9}/2^{79.5}$
$m = 22$	$2^{72.6}/2^{72.6}$	$2^{81.7}/2^{65.0}$	$2^{84.6}/2^{84.6}$	$2^{95.8}/2^{95.8}$	$2^{110.0}/2^{83.6}$

- We attack a compression function:
  - necessarily many solutions for inversion or collision search.
- Standard case for the application of GBA:
  - we directly use our formula with $2w$ for collisions, and w for inversion.
- More problematic case for ISD:
  - we are between the zones of application of our two formulas...

- Bounds on the complexity of GBA against FSB:

	n	r	w	inversion	collision
FSB_{160}	5×2^{18}	640	80	$2^{156.6}$	$2^{118.7}$
FSB_{224}	7×2^{18}	896	112	$2^{216.0}$	$2^{163.4}$
FSB_{256}	2^{21}	$1\,024$	128	$2^{245.6}$	$2^{185.7}$
FSB_{384}	23×2^{16}	$1\,472$	184	$2^{360.2}$	$2^{268.8}$
FSB_{512}	31×2^{16}	$1\,984$	248	$2^{482.1}$	$2^{359.3}$

- These are only bounds using an idealized algorithm. This does not give any attack.

Conclusion

- We described idealized version of known attacks against CSD:
  - ▷ these idealized versions have a complexity easier to analyse, allowing us to derive “simple” bounds
  - ▷ achieving better complexities than these bounds necessarily requires to change the algorithms.
    $\longrightarrow$ generalized birthday inside ISD?

- It is also interesting to note that existing algorithms have practical complexities very close to our bounds:
  - ▷ these algorithms are already almost optimal.