Syndrome Decoding in the Non-Standard Cases

Matthieu Finiasz

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

LASEC

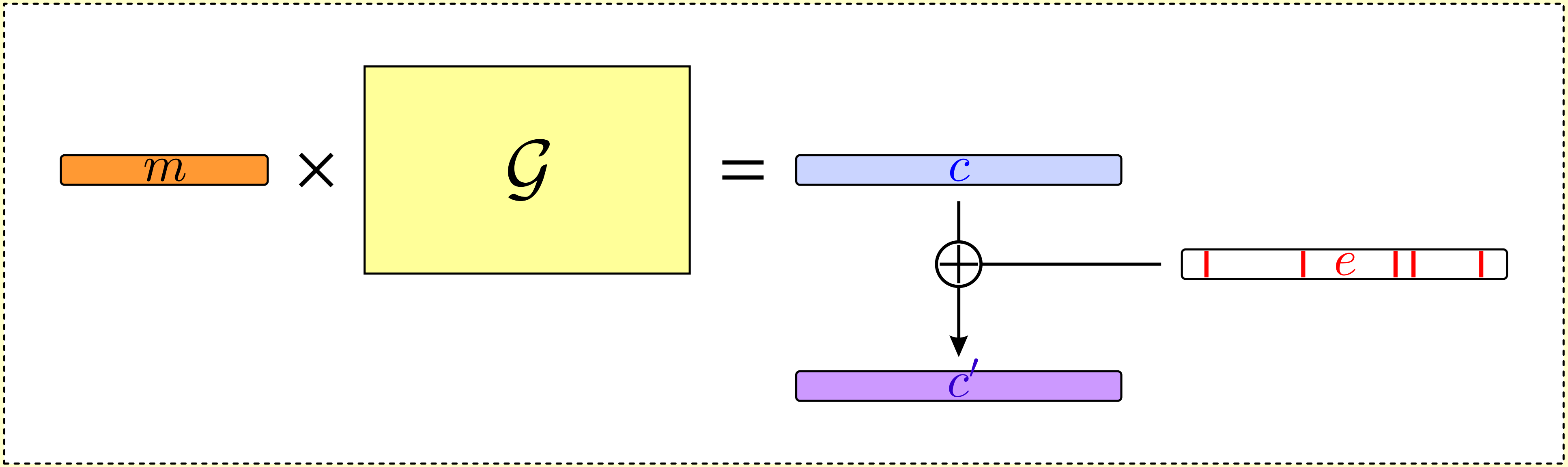
Outline

Part I

The Problem of Syndrome Decoding

What Does Decoding Mean?

- A code $\mathcal{C}$ can be defined by a $k \times n$ generator matrix $\mathcal{G}$
  - a message m is encoded into a codeword c, adding some noise e gives a word $c' = c \oplus e$.



- Decoding consists in finding the closest codeword to c'.

Parity Check Matrix and Syndromes

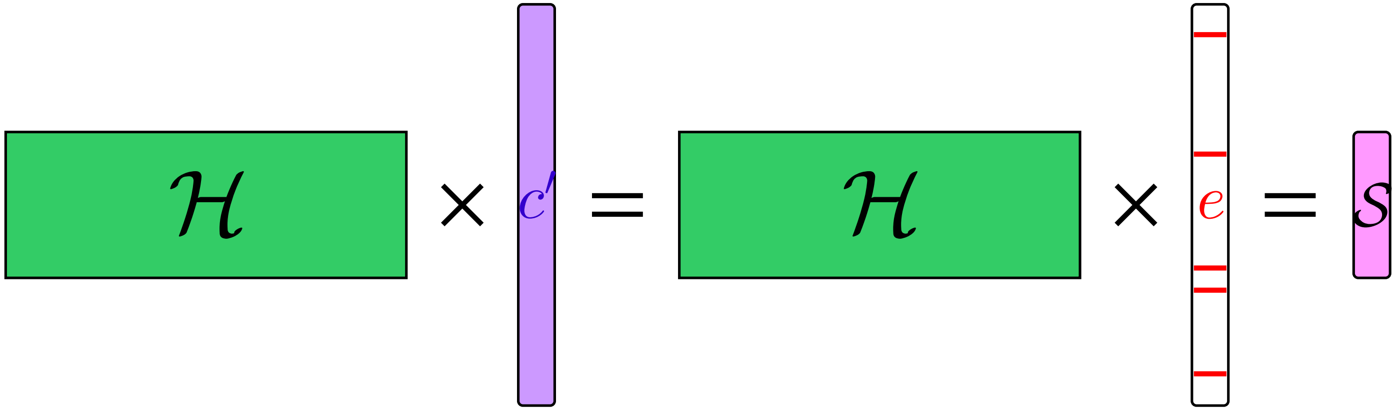
- A *parity check matrix* $\mathcal{H}$ of the code $\mathcal{C}$ is such that:

$$c \in \mathcal{C} \quad \text{iff} \quad \mathcal{H} \cdot c = 0.$$

  - Using $\mathcal{H}$ one can make decoding independent of c:

$$\mathcal{H} \cdot c' = \mathcal{H} \cdot (c \oplus e) = \cancel{\mathcal{H} \cdot c} \oplus \mathcal{H} \cdot e = \mathcal{S}.$$

    → $\mathcal{S}$ is the *syndrome* of c' (or of e).



- Find the word of syndrome $\mathcal{S}$ of lowest weight.

The Problem of Syndrome Decoding

Syndrome Decoding: (SD)

Input: an $n-k \times n$ binary matrix $\mathcal{H}$, an $n-k$ bit vector $\mathcal{S}$ and a weight w.
Output: an n bit vector e of Hamming weight $\leq w$ such that $\mathcal{H} \cdot e = \mathcal{S}$.

- It is a sort of “bounded” decoding: maximum-likelihood decoding is not in NP.
- NP-complete [Berlekamp - McEliece - van Tilborg 1978] $\longrightarrow$ some instances are hard.

Known Techniques for Solving SD

- Birthday techniques:
  - standard with 1 list
  - memory saving with 4 lists [Joux 2002]
  - generalized birthday with 2^a lists [Wagner 2002]
- Decoding techniques:
  - information set decoding [Canteaut - Chabaud 1998]
  - iterative decoding [Fossorier - Kobara - Imai 2003]
- Lattice-based techniques?

Part II

The Cryptosystems of McEliece and Niederreiter

- The public key is a scrambled Goppa code generator matrix $\mathcal{G}' = \mathcal{Q} \times \mathcal{G} \times \mathcal{P}$. $(\mathcal{G}, \mathcal{P}, \mathcal{Q})$ is the private key.

Encryption: $E_{\mathcal{G}'}(m)$

Pick e of weight $\leq t$.
Compute $c' = E_{\mathcal{G}'}(m) = m \times \mathcal{G}' \oplus e$.

Decryption: $D_{(\mathcal{G},\mathcal{P},\mathcal{Q})}(c')$

Compute $c' \times \mathcal{P}^{-1} = m \times \mathcal{Q} \times \mathcal{G} \oplus e'$.
Decode to remove e' and recover $m \times \mathcal{Q}$, and multiply by $\mathcal{Q}^{-1}$ to get m.

- Similar to McEliece, but the message is coded in the error e instead of the codeword.
  - The public key is $\mathcal{H}' = \mathcal{P} \times \mathcal{H} \times \mathcal{Q}$ where $\mathcal{H}$ is a parity check matrix.
  - The message is coded into a word e of given weight.
  - The ciphertext is the syndrome $\mathcal{S} = \mathcal{H}' \times e$.

- Both systems have equivalent security
  $\longrightarrow$ decryption requires to solve an instance of SD.

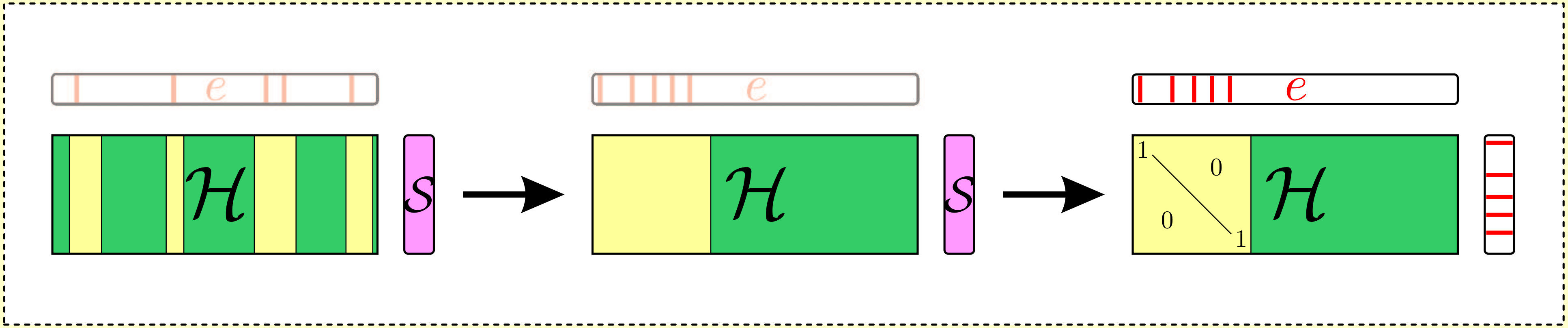
Usual Parameters

- The original McEliece parameters are $n = 1024$, $k = 524$ and $t = 50$ $\longrightarrow$ not secure enough.
- "Better" parameters are $n = 2048$, $k = 1718$, $t = 33$.
- The corresponding instances of SD are very specific:
  - there is always a single solution,
  - parameters correspond to Goppa codes: $\frac{n-k}{w} = \log n$,
    $\longrightarrow$ w is a little below the Gilbert-Varshamov bound.

Most research was focused on this type of parameters, they are believed to be among the hard instances of SD.

Information Set Decoding (ISD)

- Find k positions containing no non-zero positions of e.
  - ▷ This is called an information set.
  - → A Gaussian elimination on the $n-k$ other gives e.



- Probability of success $= \frac{\binom{n-w}{k}}{\binom{n}{k}} = \frac{\binom{n-k}{w}}{\binom{n}{w}} \simeq \left(\frac{n-k}{n}\right)^w$.
  - → Complexity $= \mathcal{O}\left(\mathcal{P}oly\,(n)\left(\frac{n}{n-k}\right)^w\right)$.

- There is a single solution
  - generalized birthday does not apply
  - simply list words of weight $\frac{w}{2}$ and look for the collision
  - complexity is of order $\mathcal{O}\left(n^{\frac{w}{2}}\right)$.

- If $n - k > \sqrt{n}$, birthdays are less efficient than ISD
  → useful only for codes correcting very few errors.

- "Standard case" refers to the kind of instances of SD derived from McEliece or Niederreiter cryptosystems:
  - a single solution exists
  - close to the Gilbert-Varshamov bound.
- These are the cases that have been the most studied
  - the best algorithm is quite complex
  - less research was done for other parameters
    $\longrightarrow$ generic algorithms are used.

Part III

McEliece-Based Signatures

The Problem of Code-Based Signatures

[Courtois - Finiasz - Sendrier 2001]

- One needs to decrypt a "random" ciphertext
  - ▷ some (most) syndromes/words can't be decoded.
  - ▷ some (most) messages can't be signed!
- A simple solution exists:
  - ▷ get the highest possible probability of success
    - → increase the density of decodable syndromes.
  - ▷ hash a lot of "equivalent" documents
    - → append a counter, for example.

⚠ The counter is part of the signature.

The Signature Algorithm

Signature Algorithm: $Sign(\mathsf{D})$

1. Initialize the counter $i = 0$
2. Hash D and i into a syndrome: $\mathcal{S}_i = Hash(\mathsf{D}||i)$
3. Try to decode $\mathcal{S}_i$ into a word e_i

   → if it fails $i{+}{+}$ and go back to 2
4. Return $Sign(\mathsf{D}) = (i, e_i)$.

- The average number of attempts is:

$$\mathcal{N}_{attempts} = \frac{\mathcal{N}_{\mathcal{S}}}{\mathcal{N}_e} = \frac{2^{n-k}}{\binom{n}{t}} \simeq t!$$

Reaching Non-Standard Parameters

- For efficiency, we need codes correcting very few errors
  - fewer errors also gives shorter signatures!
  - we proposed $n = 2^{16}$, $n - k = 144$ and $t = 9$.

- Near the limit where birthday techniques become more efficient than ISD ($n - k$ is very small):

$$\left(\frac{n}{n-k}\right)^t \approx 2^{79.5} \quad \text{and} \quad n^{\lceil \frac{w}{2} \rceil} = 2^{80}$$

- Can another algorithm be more efficient yet?

A Problem a Little Different from SD

- Forging a signature does not simply consist in solving one instance of SD:
  - there are many instances sharing the same matrix
  - among these some give a solution
  - a large majority has no solution.
- An attacker needs to solve “one of many” instances
  - is this easier (attacks can be parallelized)?
  - is this harder (most instances are unusable)?
  - how can we improve birthday techniques?

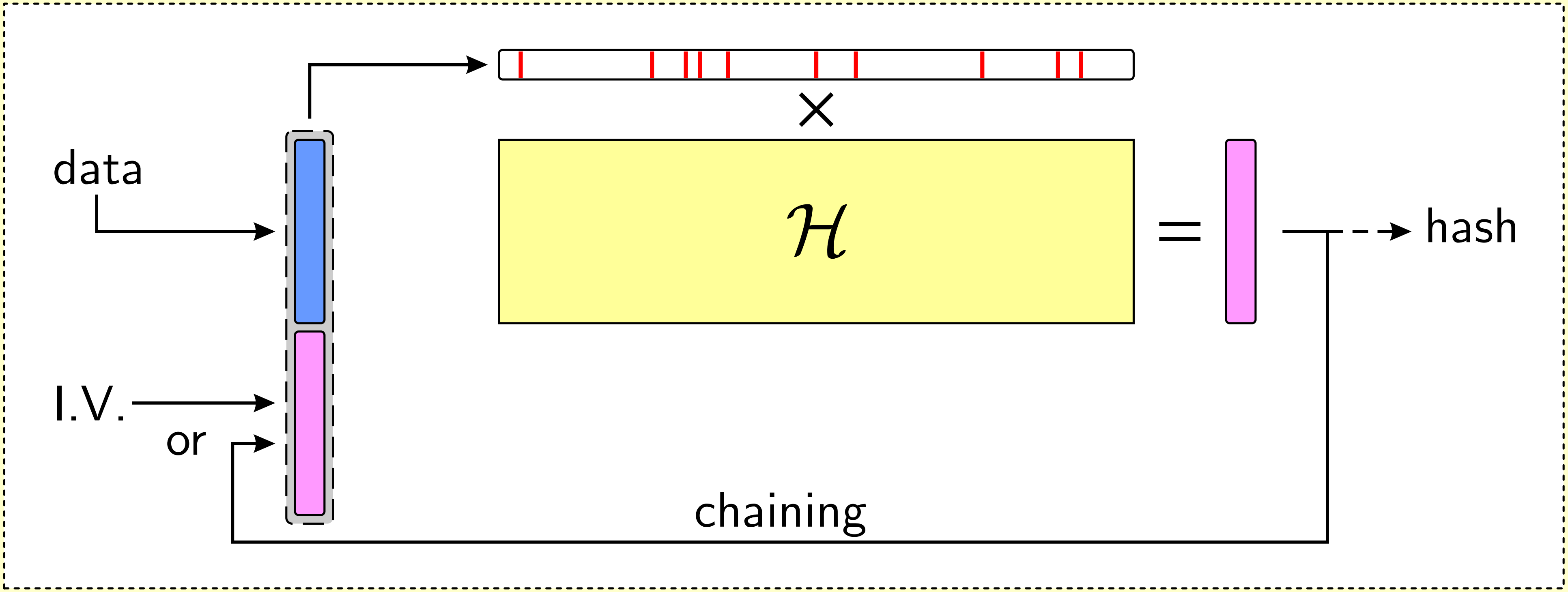
Part IV

Provably Secure Syndrome-Based Hash Functions

Main Idea

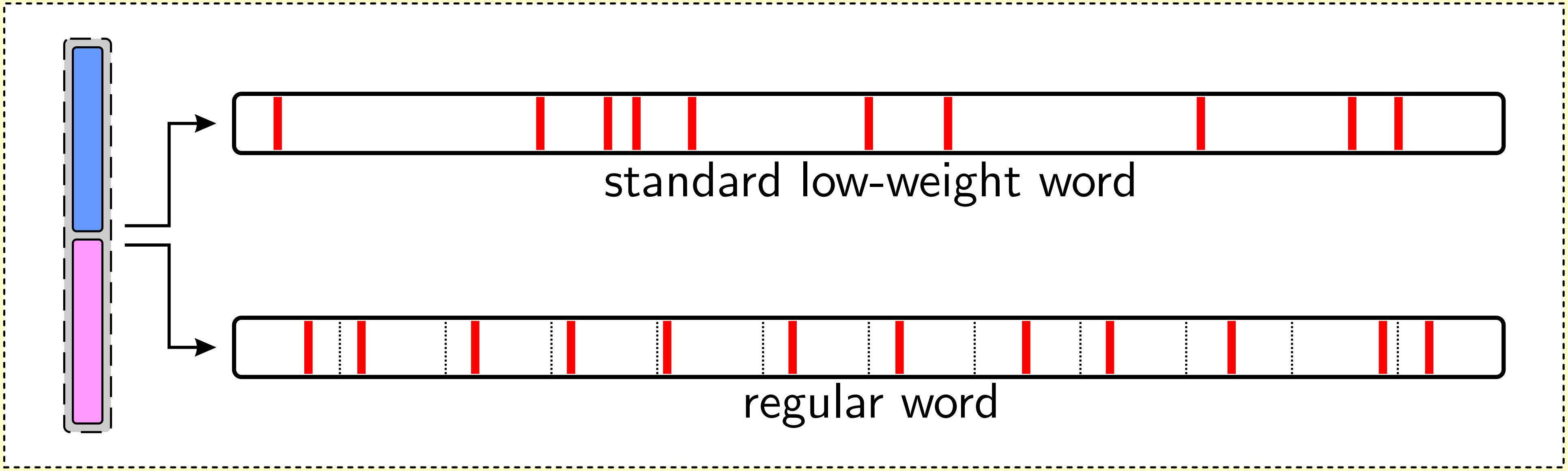
[Augot - Finiasz - Sendrier 2005]

- Design a compression function for which inversion and collision search requires to solve an instance of SD
  - take a large random binary matrix, convert the input into a low weight word and output its syndrome.



Constraints on the Parameters

- It has to compress
  - we have to choose a w such that $\binom{n}{w} > 2^{n-k}$,
  - there are many solutions to SD for inversion/collision.
- It has to be fast
  - one to one conversion to constant weight word is slow
    → use regular words.



Security

- SD with regular word is still NP-complete
  - collision search or inversion requires to solve an instance of some new problems.

- In practice
  - the best attacks use Wagner's generalized birthday
  - secure parameters are for example:

$$n = 21760, \quad n - k = 400 \quad \text{and} \quad w = 85.$$

- Parameters n and $n - k$ are similar to signature parameters, but w is huge → far from Goppa codes.

Compared to Standard SD

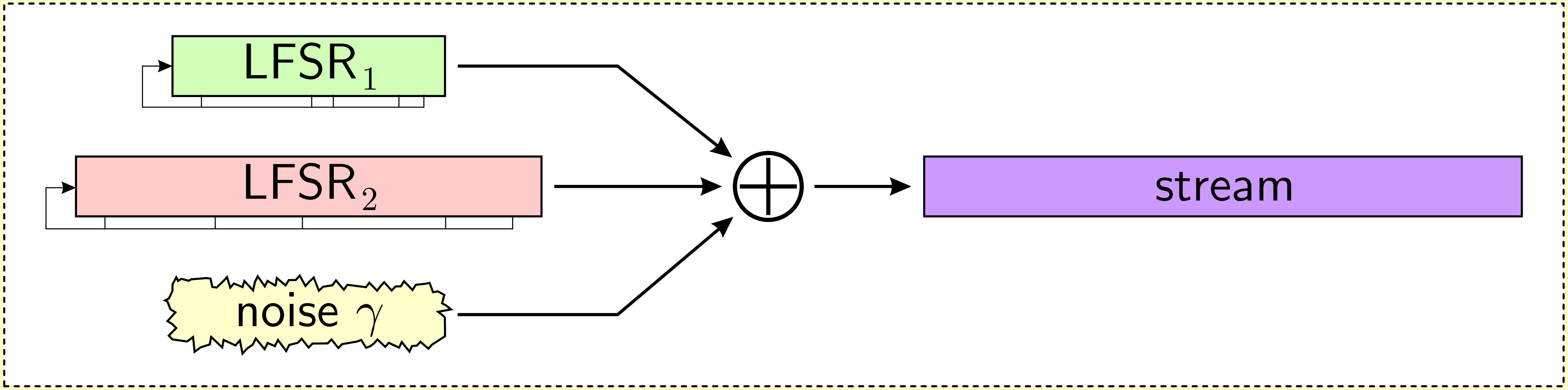
- Quite a few differences compared to attacks on McEliece:
  - there are many solutions
  - a truly random binary matrix is used
    - is this harder in average than a scrambled Goppa?
  - though still NP-complete the problems are not SD
    - instances can be split in subparts
    - ISD attacks can surely be improved
    - it has been studied only very little

Part V

The Multiple of Low Weight Problem

A Key Problem of Correlation Attacks

- Correlation attacks approximate a stream-cipher by two LFSRs and some noise



- In order to recover the initialization of LFSR_1:
  - find a multiple K of weight w of LFSR_2
  - multiply the stream by K → suppress LFSR_2
  - results in a decoding problem with noise γ^w.

Multiple of Low Weight Problem: (MLW)

Input: a polynomial P, a degree d and a weight w.
Output: a polynomial K of degree $\leq d$, weight $\leq w$ and such that $P|K$.

- This is a re-writing of the SD problem, with a truncated cyclic code:
  - compute the $d+1 \times d_P$ binary matrix with columns:
    $$\mathcal{H}_i = x^i \mod P(x), \quad i \in [0, d].$$
  - look for a word of weight $\leq w$ and syndrome 0.

Classical Cryptanalytic Setting

- When attacking a stream cipher, the smaller w and d, the less stream bits will be required to decode
  - some kind of trade-off between weight and degree,
  - strong threshold: a small change on w and on d will change from no solution to many:
$$\mathcal{N}_{sol} \simeq \frac{\binom{d}{w}}{2^{d_P}},$$
  - finding several solutions is useful,
  - LFSR_2 will be about 100 bits long
    $\longrightarrow$ $d_P = n - k$ is small: ISD is inefficient.
- Use birthday techniques (either classical or generalized).

TCHo: the Trapdoor Stream Cipher

[Finiasz - Vaudenay 2006]

- Use a multiple of low weight as a trapdoor:
  - ▷ factor a polynomial K of degree d and weight w,
  - ▷ choose a factor P and use it for LFSR$_2$,
  - ▷ use a small LFSR$_1$ to encode the message,
  - ▷ add some noise γ and output a stream of length ℓ.

- For key recovery ⟶ find a single "unexpected" solution.
- For decryption ⟶ find many "expected" solutions.

⚠ d_P is much larger than before. Typical parameters are: $\ell = 50000$, $d_P = 6000$, $d_K = 15000$ and $w = 100$.

MLW Compared to Classical SD

- The main difference is the use of a truncated cyclic code instead of a "random" matrix
  - ▷ this has little influence on the security: $w \rightarrow w-1$.
- Key recovery for TCHo is very similar to classical SD.
- In the other cases, there is no limit for w
  - ▷ some solutions are easy to find (P itself!)
    $\rightarrow$ they are usually useless.
  - ▷ two types of hard-to-find solutions:
    - ▷ w with few solutions $\rightarrow$ ISD/birthday
    - ▷ w with loads of solutions $\rightarrow$ Wagner.
- The best strategy will depend on γ and the stream size.

Conclusion

- “Standard SD instances” have been extensively studied
  - I believe new techniques are possible, but any progress would be a breakthrough.
    - → I would compare this to the factoring problem.
- “Non-standard SD instances” have been less studied
  - new specific techniques are bound to appear,
    - → take advantage of specific parameters.
    - → take advantage of a specific setting.
  - parameters that are proposed are probably too tight
    - → expect attacks with little practical impact.
  - will these new attacks be generalized?