Code-Based Zero-Knowledge Proofs, Signatures and More

Kirill Morozov
UNT

6th Code-Based Cryptography Workshop (CBC 2018)

Davie, Florida, USA

April 6, 2018

“9am is too late at night even for me.”
– Response of an owl person to an invitation
(Traveling mathematical anecdote)

Plan of this talk

- Background
- McEliece and Niederreiter PKE and their variants
- Code-based commitments (Jain et al.)
- Zero-knowledge identification schemes and their apps
- Code-based signatures:

  Confirmer sig. and security of CFS scheme
- Other recent results
- Concluding remarks

Linear codes

- A binary linear (n,k) code C is a linear k-dimensional subspace of F_2^n
- Basis of the subspace can be written as $G \in \{0,1\}^{k \times n}$ called the generator matrix (rank(G) = k)
- $C = \{mG : m \in F_2^k\}$

Example: Take m=(1 1 0 0),

$$\mathbf{G} := \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}}_{\text{Systematic form}}$$

The codeword x = (1 0 0 0 1 1 0)
+ (0 1 0 0 1 0 1)
= (1 1 0 0 0 1 1)

Rem.: Obtaining m from x is **easy!**

Minimum distance d

Example:
$x = (0\ 1\ 0\ 0\ 1\ 1)$
$w_H(x) = 3$

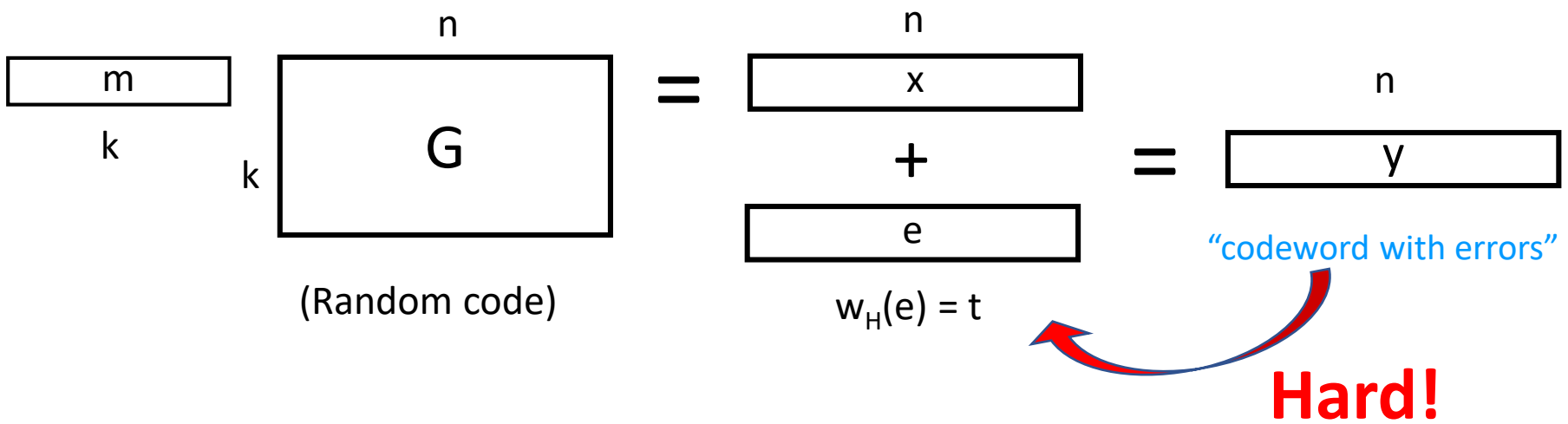
- Hamming weight
  $w_H(x) :=$ # non-zero entries of $x \in F_2^n$
- Hamming distance
  $d_H(x,y) := w_H(y-x) = w_H(y \oplus x) = w_H(y+x) =$ # positions where y and x differ

$y = (1\ 1\ 0\ 0\ 1\ 0)$
$d_H(x,y) = 2$

- Minimum distance of a code C
  $$d := \min_{x_1,x_2 \in C\,:\,x_1 \neq x_2} d_H(x_1,x_2)$$

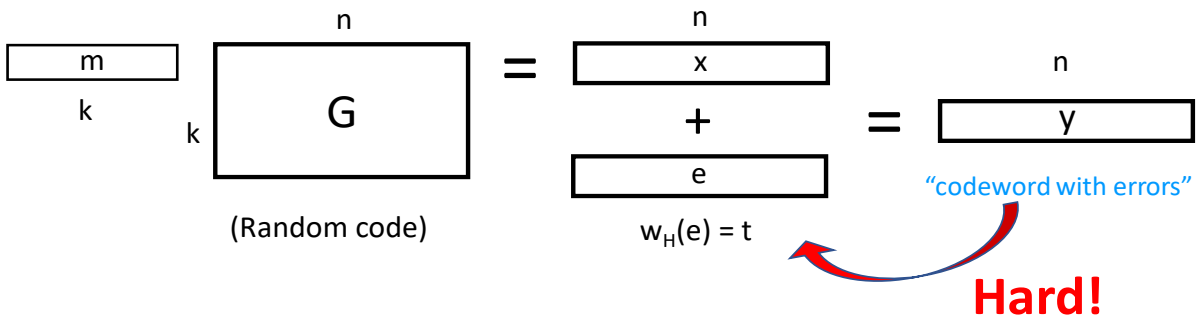
- Remark: Min. distance = minimum weight of a codeword in C
  (since x_1+x_2 is also a codeword)

General Decoding Problem



- Input: (G,y,t), where $G \leftarrow_R \{ F_2^{k \times n} \mid \mathrm{rank}(G)=k \}$, $e \leftarrow_R \{ F_2^n \mid w_H(e)=t \}$, $m \in F_2^k$, $y=mG+e$
- Output: (m,e)
- General Decoding is NP-complete
  [Berlekamp, McEliece, van Tilburg '78: IEEE Trans. Inf. Theory]

How to choose “t”?



- Input (G,y,t), output (m,e)
- Make the solution “e” unique!
- => (Hamming) weight of “e” must be $< w_H(x)$, $\forall x \in C$
- Finding minimum weight of G

  = minimum distance of G – NP-hard!
- [Coffey, Goodman ‘90]: Minimum weight of random G $\geq$ GV(n,k) with high probability (in “n”)

(Asymptotic) Gilbert-Varshamov bound for linear codes

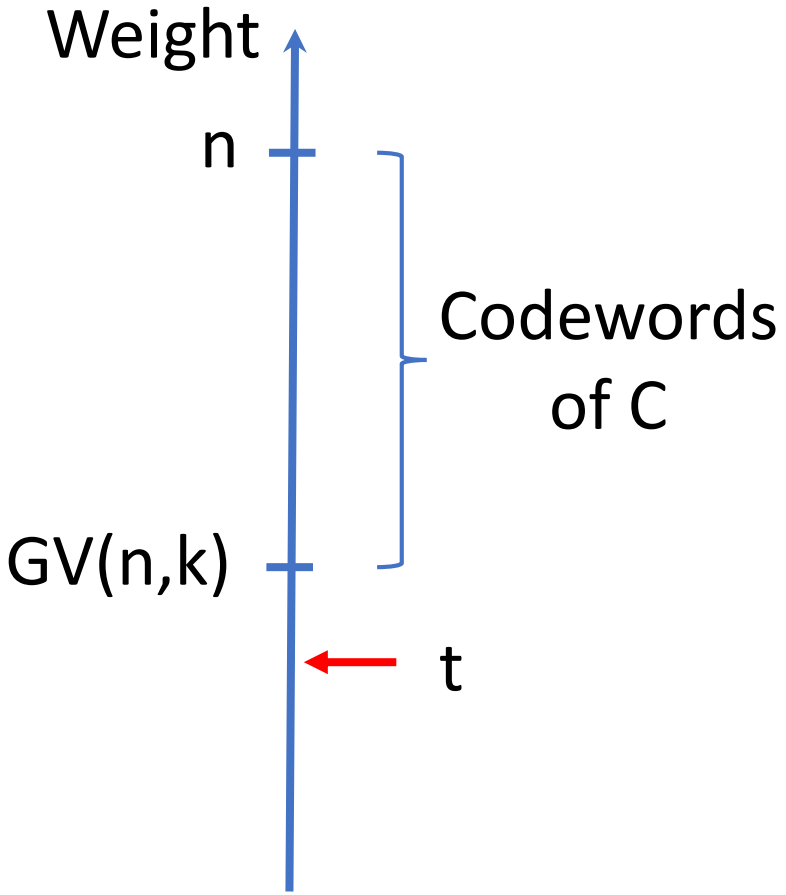
- $q \geq 2, \quad \forall\ 0 \leq d/n < 1-1/q, \quad 0 < \varepsilon \leq 1-H_q(d/n)$
- $\exists$ a code with rate $k/n \geq 1 - H_q(d/n) - \varepsilon$

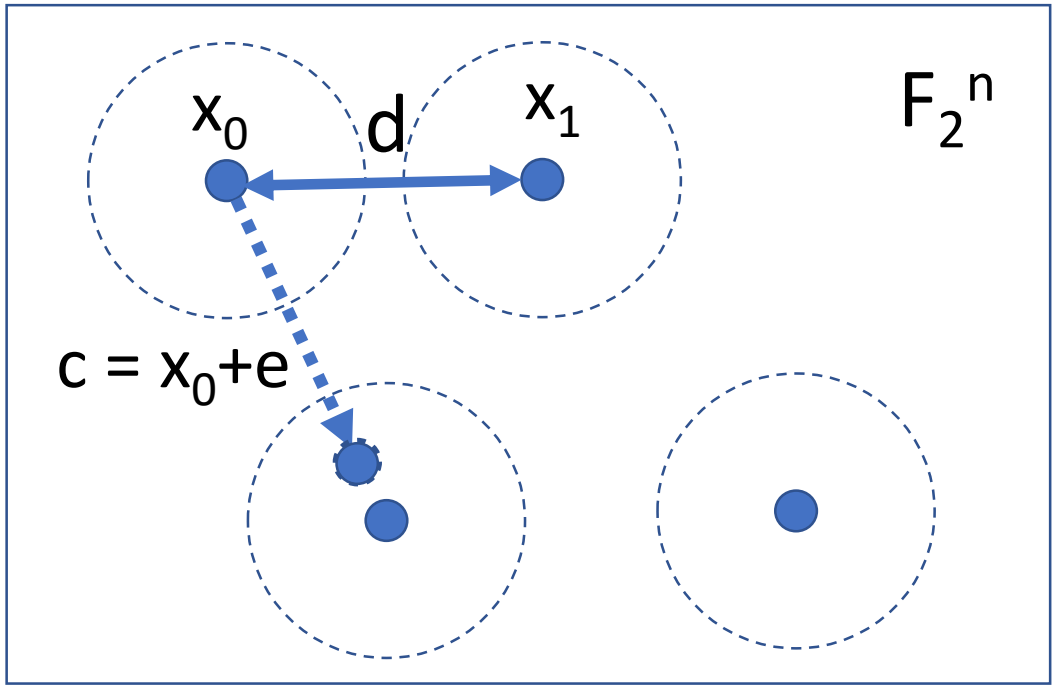
  and relative distance d/n ,

  where $H_q(x) = x \cdot \log_q(q-1) - x \cdot \log_q x - (1-x) \cdot \log_q (1-x)$

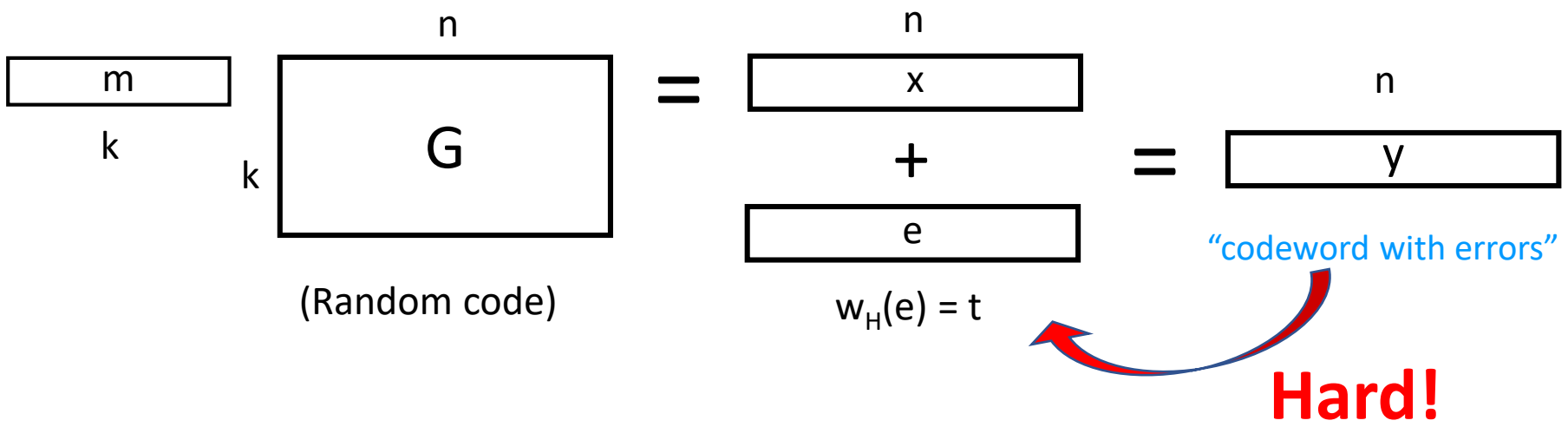
  (q-ary entropy function)
- => determine "d" for (n,k) by solving the equation

Choice of weight for “e”



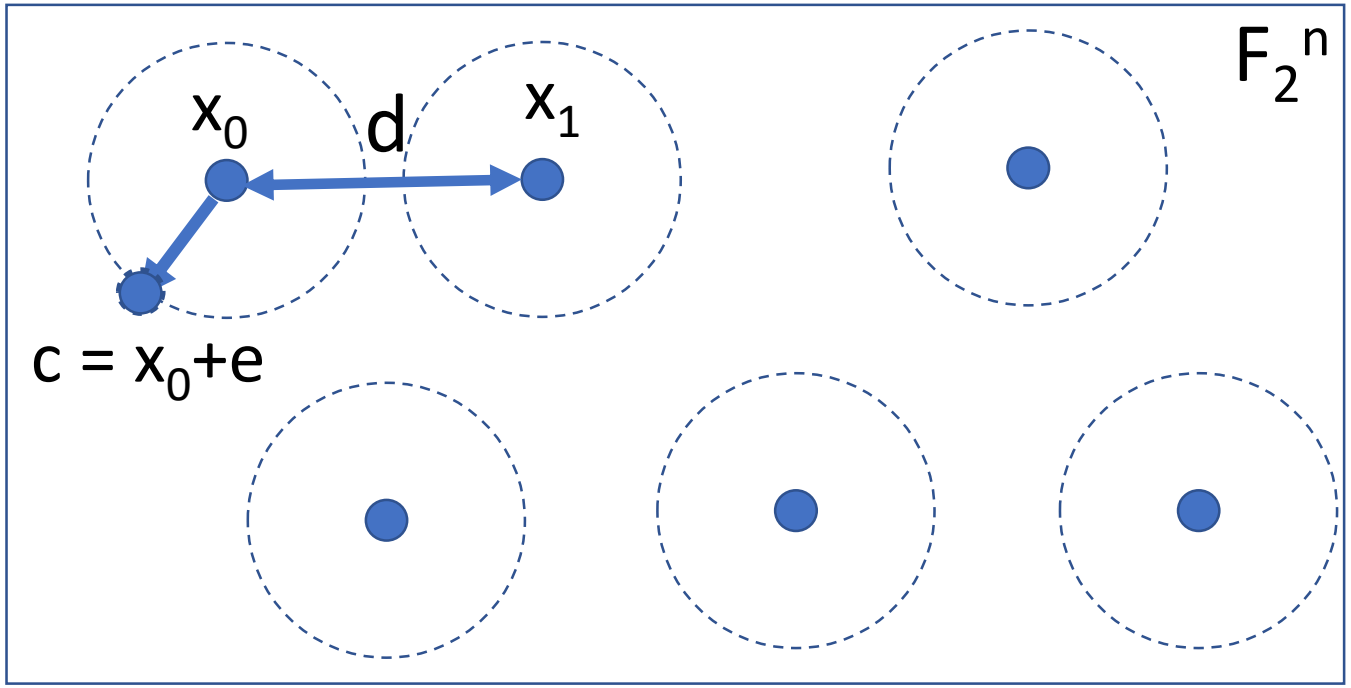


How about adding a trapdoor?



- Input (G,y,t), output (m,e)
- Make "m" a message, and "e" a "correctable" error

Nearest Codeword Decoding



$x_0, x_1 \in C$

- Correct arbitrary pattern of "t" errors <= min. dist. $d(C) \geq 2t+1$
- Example: $c = x_0 + e =$

$$
\begin{array}{r}
(1\ 1\ 0\ 0\ 0\ 1\ 1) \\
+\ (0\ 1\ 0\ 0\ 0\ 0\ 0) \\
= (1\ 0\ 0\ 0\ 0\ 1\ 1)
\end{array}
$$

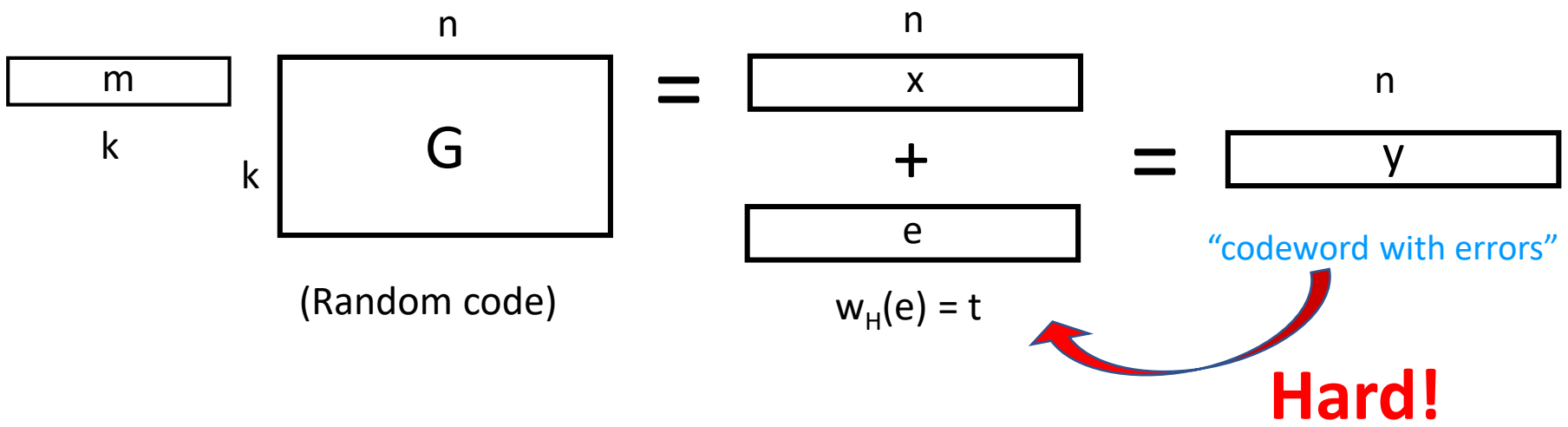
Need a “good” efficiently decodable code

“Good”:

- “d” as large as possible (because decod.alg. for random code are exp(t))
  – lots of candidates (LDPC, polar, …)
- Hard to deduct an efficient decoding alg. from a “scrambled” version of a code
  – many candidates from above drop out; still remain (among others):

    Goppa codes [McEliece ‘78], [Bernstein, Lange, Peters ‘10],
    and their generalizations: AG-codes [Janwa, Moreno ‘96],
    Generalized Srivastava codes [Cayrel, Hoffman, Persichetti ‘12] ;
    as well as LDPC codes [Baldi, Chiaraluce ‘07]
    and MDPC codes [Misoczki, Tillich, Sendrier, Barreto], …

How about adding a trapdoor? Easy!



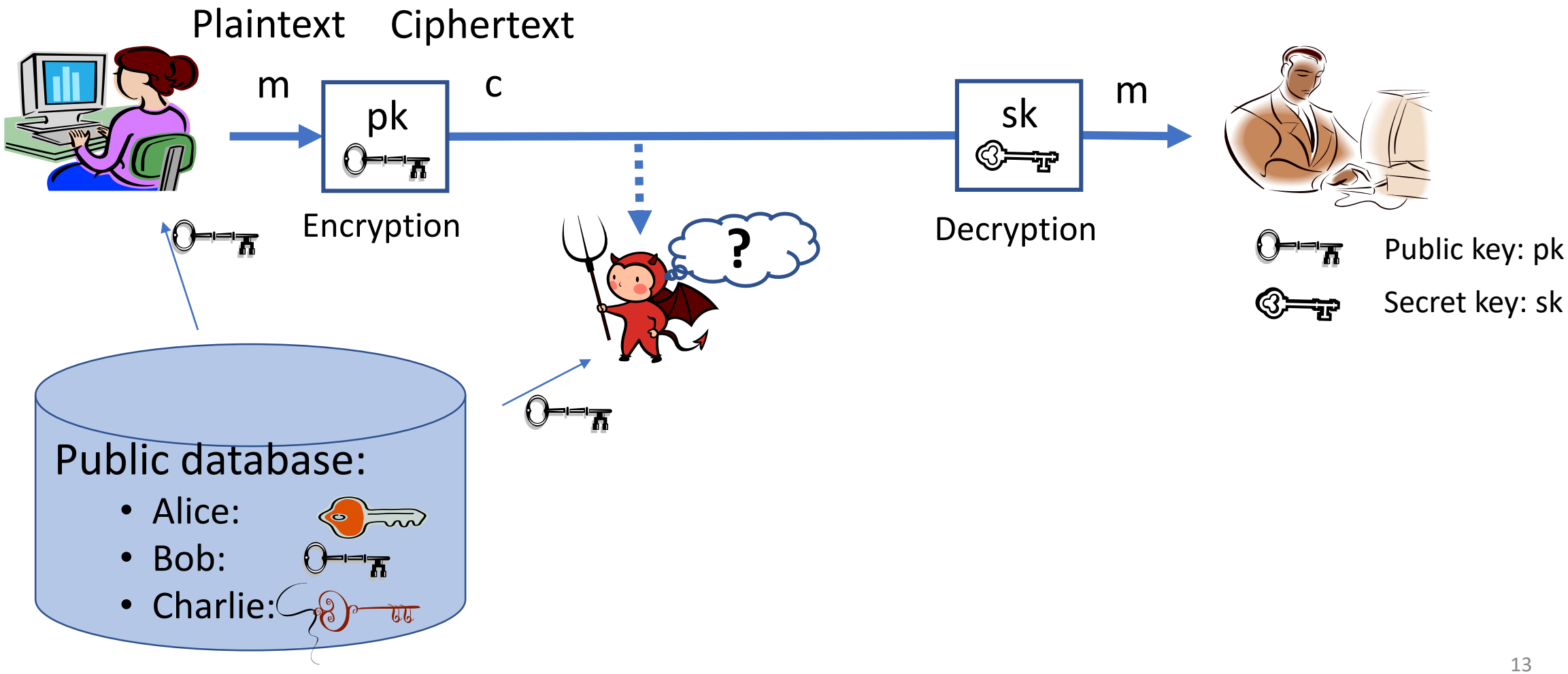
- Input (G,y,t), output (m,e)
- Make "m" a message, and "e" a "correctable" error

- Choose "G" as a "good" code
- "pk" = "scrambled" version of G
- "sk" = efficient decoding alg.

[McEliece '78]

Public-Key Encryption

Triple of alg.:
Key generation
Encryption
Decryption



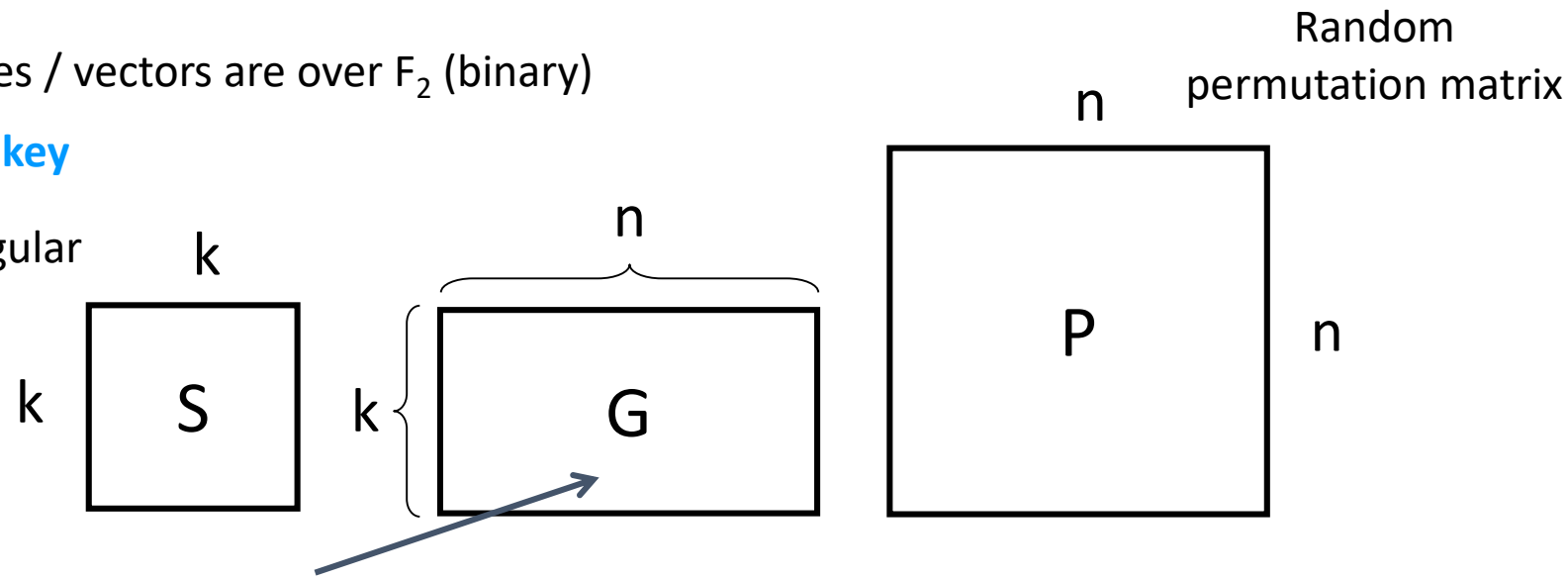
McEliece PKE: Key Generation

[McEliece, DSN Progress Report'78]

Matrices / vectors are over F_2 (binary)

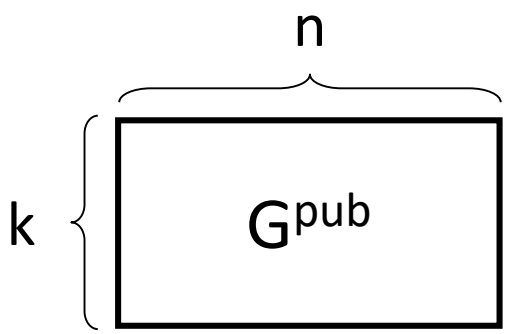
Secret key

Non-singular matrix



Generator matrix of an irreducible Goppa code [Goppa '70]
– a linear code over F_2 correcting up to t errors, $k \geq n - t \cdot \log_2 n$

Public key

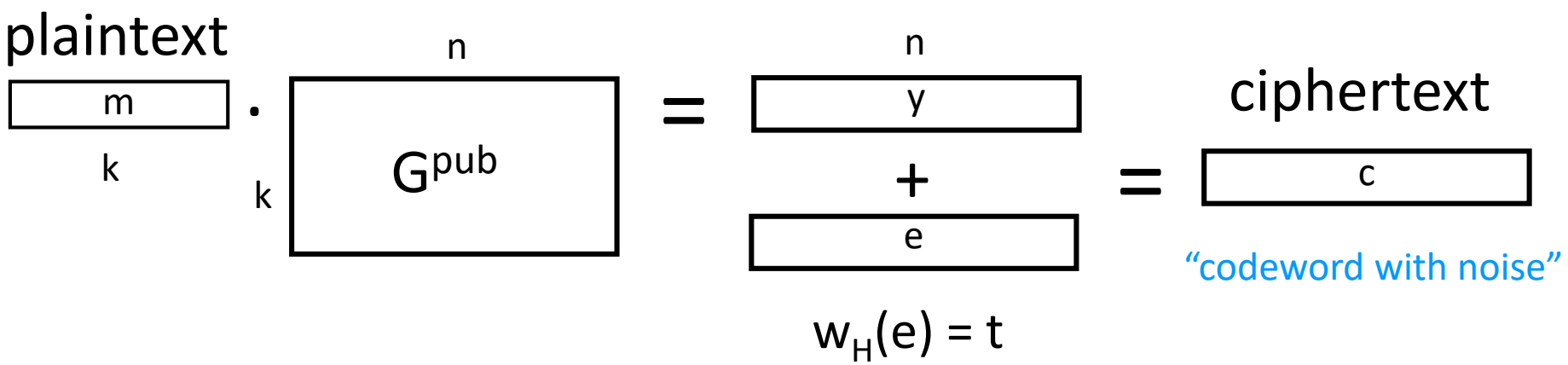


= SGP

Goppa Distinguishing Problem: Distinguish G^{pub} from random code

McEliece PKE: En(de)cryption

Encryption of $m \in \{0,1\}^k$: $c = mG^{pub} + e$



Decryption of the ciphertext $c = mG^{pub}+e = m(SGP)+e$:

1. $cP^{-1} = mSG + eP^{-1}$ – the error vector is still of weight t
2. Correct t errors using *decoding algorithm* of G: $cP^{-1} \Rightarrow mSG$
3. Solve a linear system to compute $mSG \Rightarrow mS$
4. $m = mSS^{-1}$ – since S is non-singular

Parity-Check Matrix

(n-k)-bit vector
called a "syndrome"

- By definition: $H \in F_2^{n-k \times n}$: $C = \{ x \in F_2^n : Hx^T = 0 \}$
- H is a basis of ker(G) s.t. $HG^T = 0$
- For $G = [I_k | A_{k \times n-k}]$, we have $H = [-(A_{k \times n-k})^T | I_{n-k}]$ (systematic form)

$$\mathbf{G} := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{H} := \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

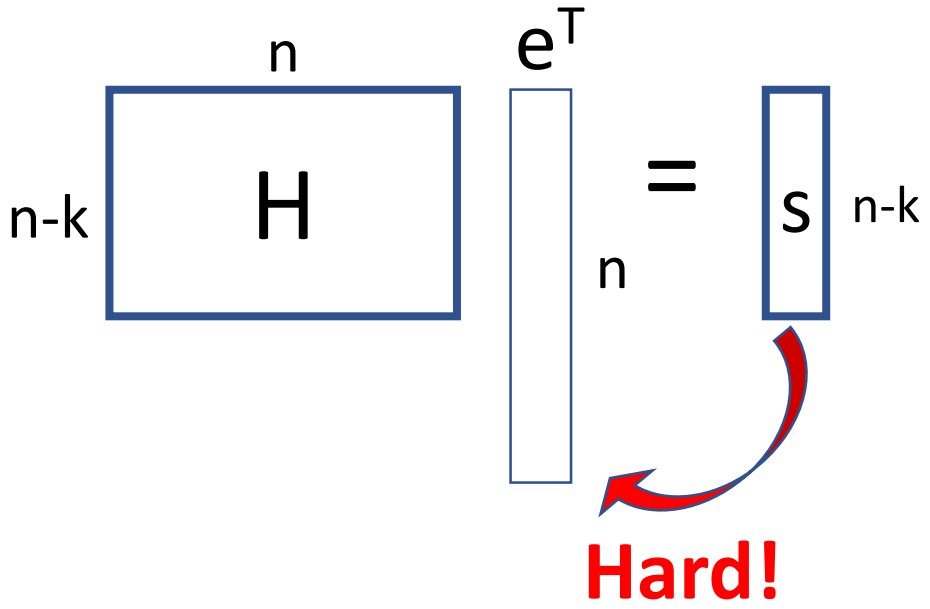
Syndrome

- Remember the general decoding: $c = mG + e$
- Compute the syndrome of c:
- $s := Hc^T = H(mG + e)^T = HG^Tm^T + He^T = He^T$

- => a dual version of the gen. dec. problem

Syndrome Decoding Problem

- Input: (H,s,t) s.t.
  - $H \in F_2^{(n-k) \times n}$, a parity-check matrix of a random (n,k) code
  - $s \in F_2^{n-k}$, a syndrome
  - t - an integer
- Find: $e \in F_2^n$, $w_H(e) \leq t$ s.t. $s = He^T$
- NP-complete
  [Berlekamp, McEliece, van Tilborg: IEEE Trans. Inf. Theory '78]



A “dual” version of the McEliece encryption

- Syndrome decoding: Input: (H,s,t), output: e
- [Niederreiter ‘86]:
- $e \rightarrow m$ (via encoding of messages as t-weight vectors)
- $s \rightarrow c$
- $H \rightarrow$ parity-check mat. of “good” eff. dec. code
- pk (similar to McEliece PKE): “scrambled” version of H
- sk: eff. dec. alg.

- Remark: [Niederreiter ‘86] proposed GRS as “good” codes – shown to be a bad choice by [Sidel’nikov, Shestakov ‘92], but Goppa codes are ok

Niederreiter PKE: Key Generation

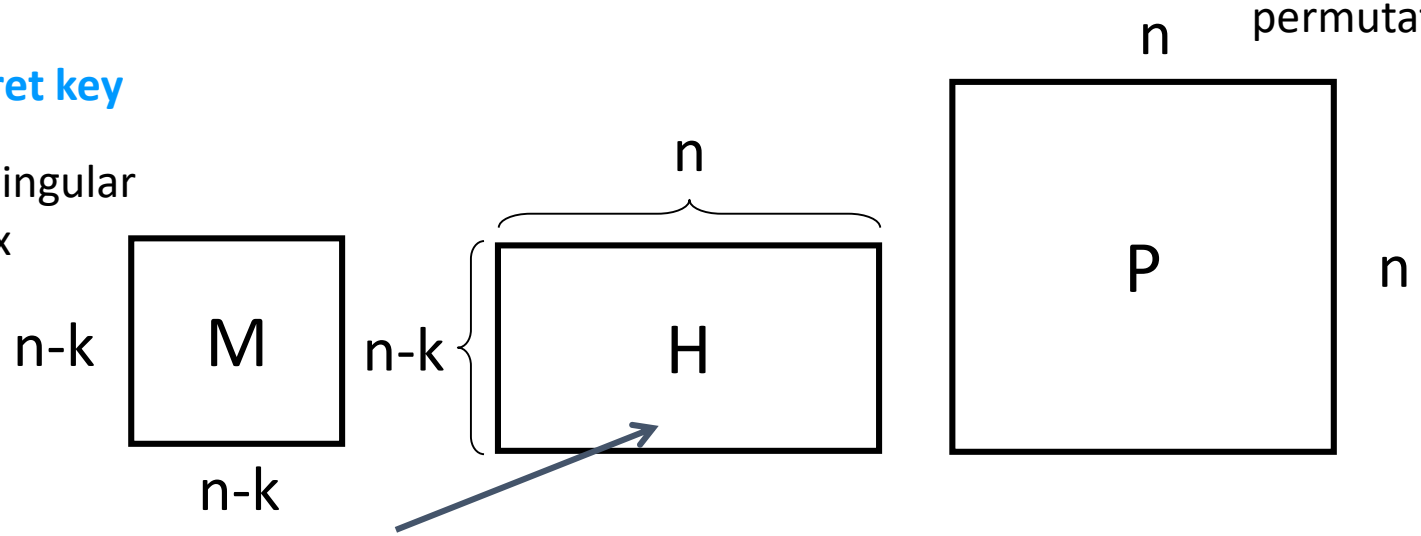
[Niederreiter, Problems of Control and Information Theory '86]

All the matrices and vectors are over F_2 (binary)

Secret key

Non-singular matrix

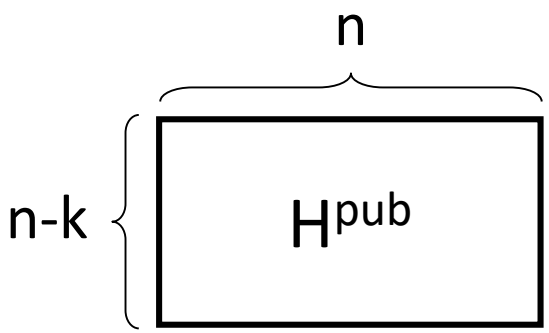
Random permutation matrix



Parity-check matrix of an irreducible Goppa code [Goppa '70]

– a linear code over F_2 correcting up to t errors, $k \geq n - t \cdot \log_2 n$

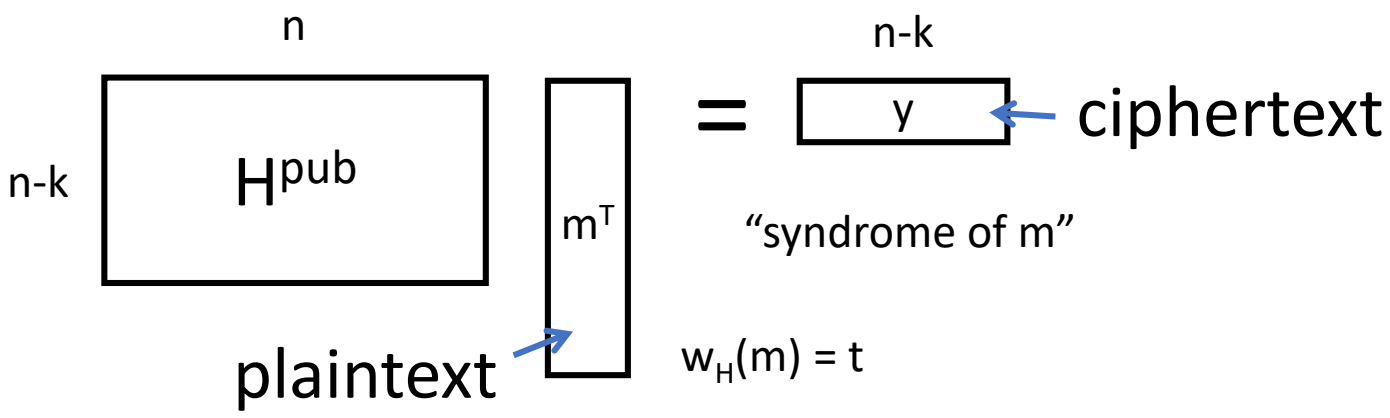
Public key



= MHP

Niederreiter: En(de)cryption

Encryption of the plaintext $m \in \{0,1\}^n$, $w_H(m)=t$: $c = H^{pub}\, m^T$



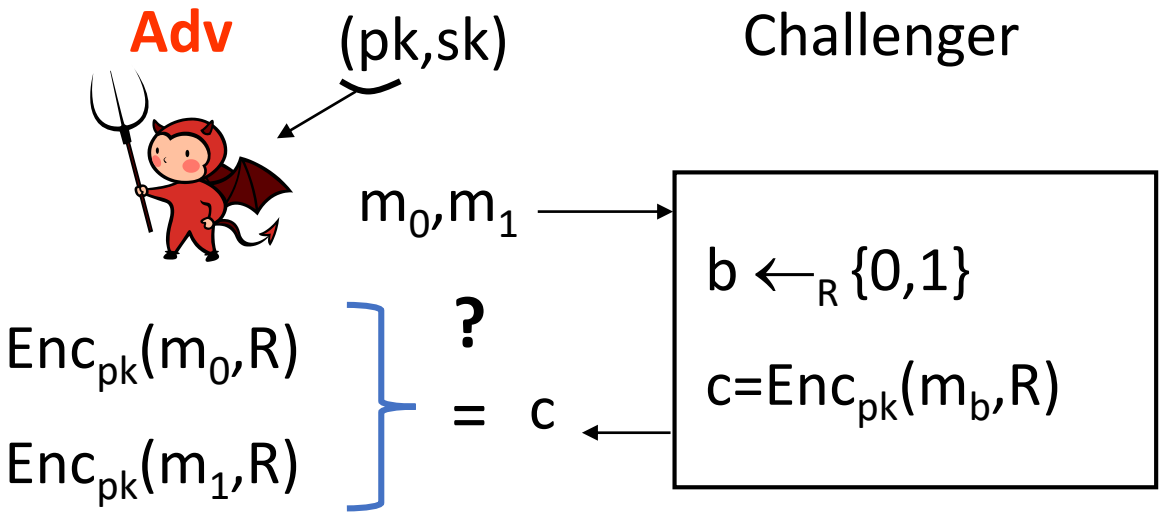
Decryption of the ciphertext $c = H^{pub}\, m^T = (MHP)m^T$:

1. $M^{-1}c = (HP)m^T = H(Pm^T)$ – since M is non-singular
2. Compute Pm^T using *decoding algorithm* of H
3. $m^T = P^{-1}Pm^T$

Semantic (IND-CPA) Security

- Introduced by Goldwasser and Micali [GM84]
- The ciphertext must leak "no useful information"
- **Game:**

Winning means distinguishing the ciphertexts with non-negligible probability

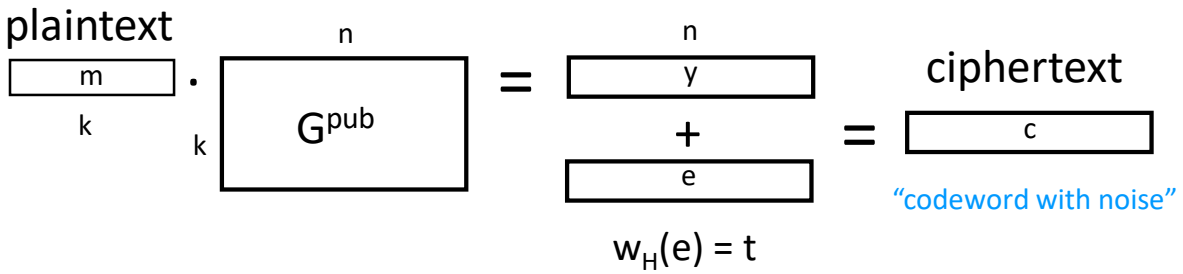


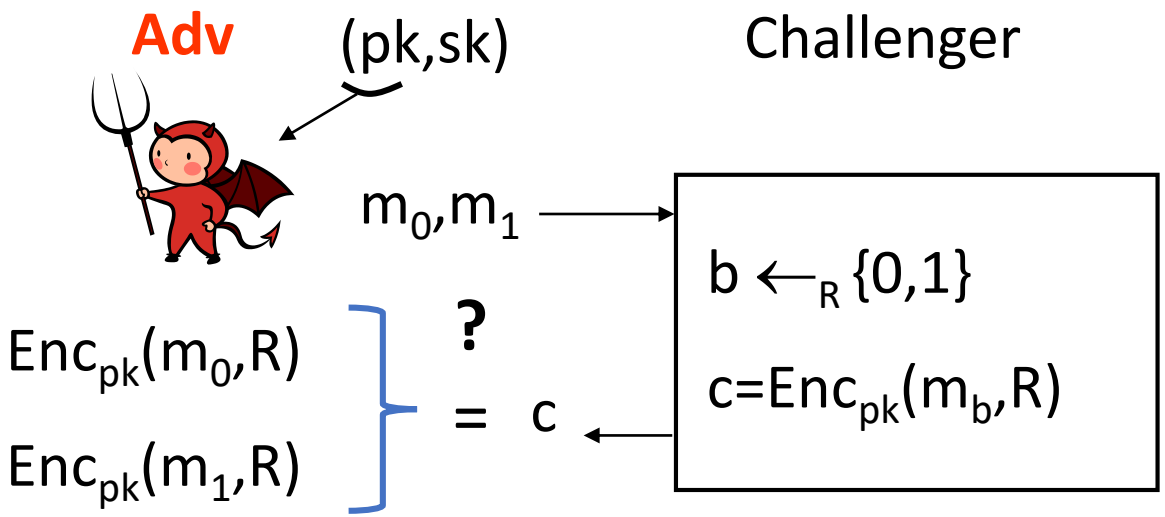
If winning probability is negligible, then the PKE is semantically secure or indistinguishable under chosen plaintext attack(IND-CPA)

What messages would you choose?

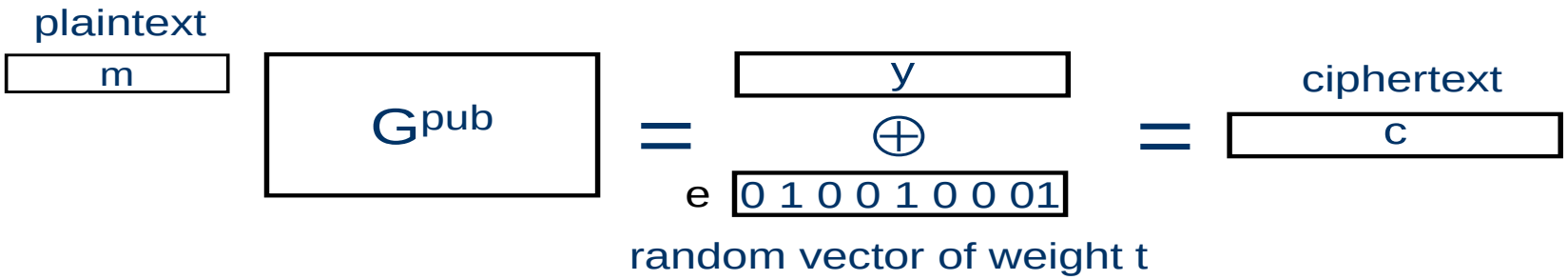
- $Enc_{ME}(m_0)$ must be easy to distinguish from $Enc_{ME}(m_1)$
- How about $m_0 = 0^k$?
- => $w_H(Enc_{ME}(m_0)) = t$
- While for $m_0 \neq 0^k$,

  $w_H(Enc_{ME}(m_0)) \geq t+1$





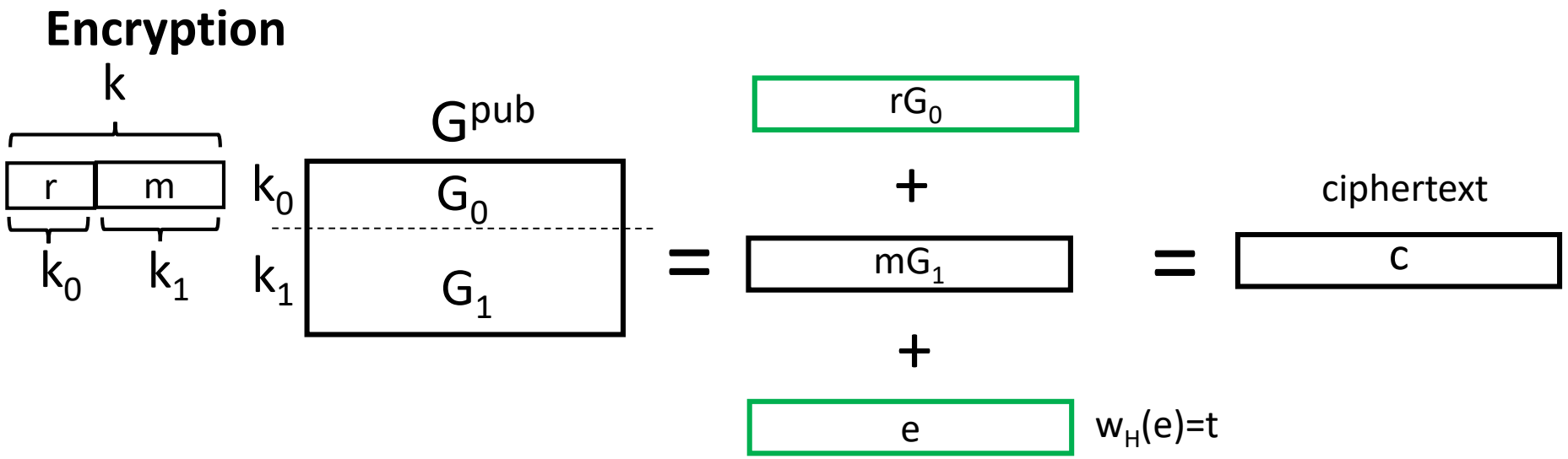
Textbook McEliece is not IND-CPA



- Given c, pick some plaintext m’
- Compute e’ = c+m’G^{pub} = (mG^{pub}+e) + m’G^{pub} = (m+m’)G^{pub}+e
- If e’ has weight t, conclude m=m’
  - If m≠m’, since $w_H((m+m')G^{pub}) \geq 2t+1$,
    we have $w_H((m+m')G^{pub}+e) \geq t+1$
- Plaintext-checkable => not IND-CPA

Randomized McEliece Encryption

Without random oracles



- [Nojima, Kobara, M, Imai'08]: If G^{pub} is indistinguishable from random => rG_0+e is pseudorandom (using [Katz, Shin '06])
- => Randomized McEliece is IND-CPA in the standard model assuming hardness of Gen.Dec. and Goppa Distinguishing
- Features in the code-based KEM proposals submitted to NIST competition

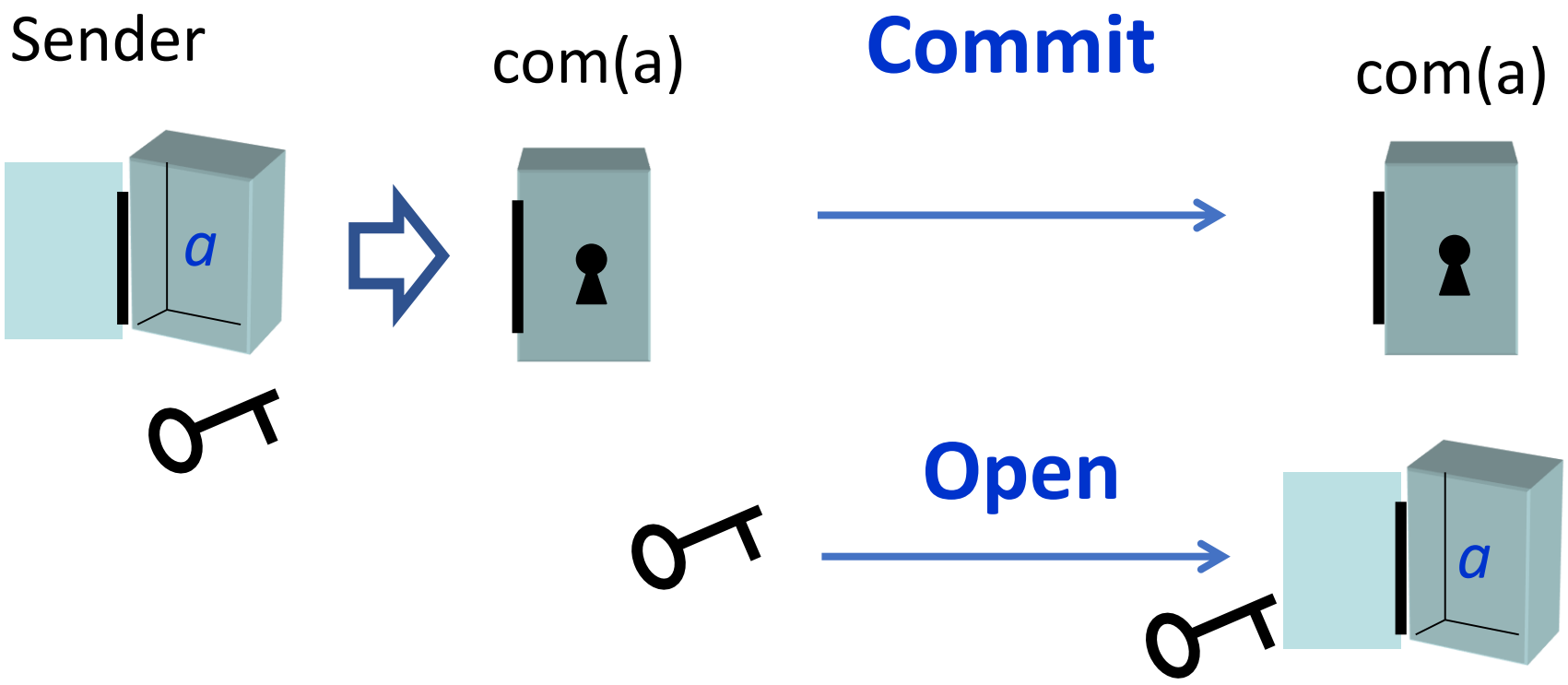
Randomized Niederreiter Encryption

- $c = H(e_r | e_m)^T$, $w_H(e_r) + w_H(e_m) = t$
- [Nojima, Kobara, M, Imai'08]: Rand. Niederreiter is IND-CPA assuming hardness of SD and Goppa Dist.
  (using [Fischer, Stern '96] pseudorandom generator

Remark on message length for both variants

- Rand.M. and Rand.N. require a relatively short message length because it “eats into” the security param. (“k” for R.M. and “t” for R.N.)
- Possibly okay for KEM
- Use conversions [Fujisaki, Okamoto ‘99], [Kobara, Imai ‘01] for IND-CCA security in the random oracle model

Commitment Scheme



Hiding: Receiver cannot learn the value

{Accept,Reject}

Binding: Sender cannot open any other value

Correctness: Honest sender is always accepts

Security of commitment schemes

To the best of my knowledge

- In the setting of lossless communications, two cases:

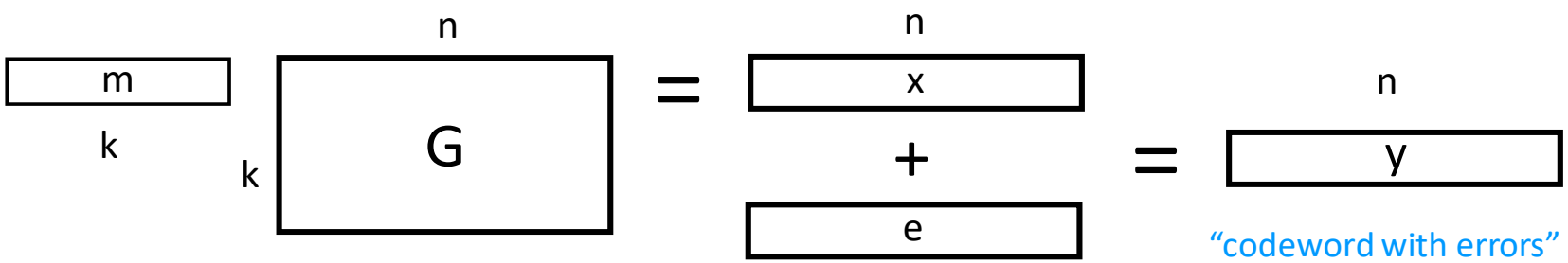
1. Computational binding, unconditional hiding
   (practical commitments in this scenario is an open problem)
2. Unconditional binding, computational hiding

   [Jain, Pietrzak, Krenn, Tentes '12]

Learning parity with noise (LPN) problem

- Secret $m \in \{0,1\}^k$
- Sample: $(g,\langle g,m\rangle)+e$,

  where $g \in \{0,1\}^k$, $\langle\cdot,\cdot\rangle$ – scalar product, $e \in Ber(\tau)$, $0< \tau< 0.5$
- Input: "n" samples ; Output: "m"

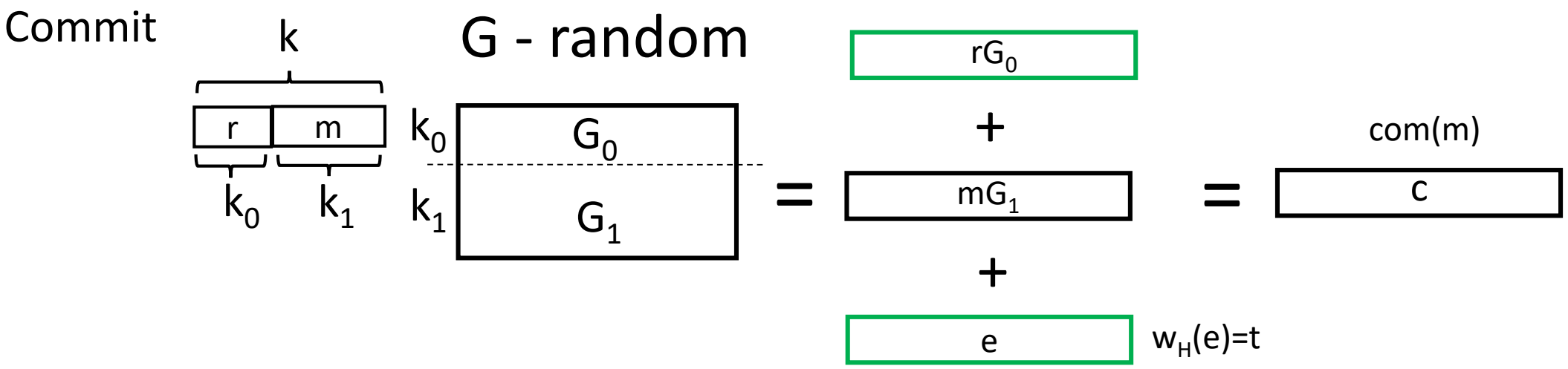


- "g" $\rightarrow$ column of G , "e" with Bernoulli distribution

LPN is equivalent to Gen. Dec.

- [Nojima et al. ‘08]: The case of fixed-weight random “e” is poly. reducible to the case of Bernoulli “e”
- [Jain et al. ‘12] mostly consider the “Exact LPN” (xLPN) problem, where “e” is fixed-weight – the <u>same</u> as Gen. Dec.

Jain et al's commitment (Asiacrypt '12)



- Commit: com(m) = (r|m)G+e = c , t < GV(n,k)/2
- Open: (r,m,e) , check if $w_H(e)=t$ and c = (r|m)G+e , if yes ACC o/w REJ
- Hiding (computational): As in the randomized McEliece PKE

Jain et al's commitment: Statistical binding

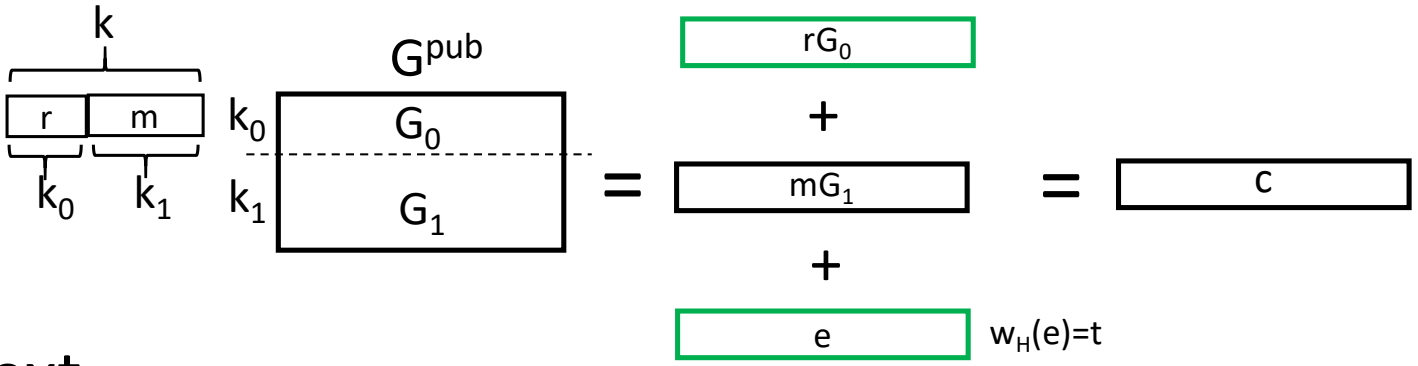
- Suppose $(r_1|m_1)G+e_1 = c = (r_2|m_2)G+e_2$, i.e. $\exists$ 2 ways to open "c"
- => $(r_1+r_2|m_1+m_2)G = e_1+e_2$ – impossible
  since $w_H(e_1+e_2) < GV(n,k)$ => cannot be a codeword

- Statistical = unconditional with negligible error

- Quiz: How to make it <u>perfect</u>? (= unconditional with <u>zero</u> error)

Perfectly binding code-based commitment

- [M, Roy, Sakurai '17]



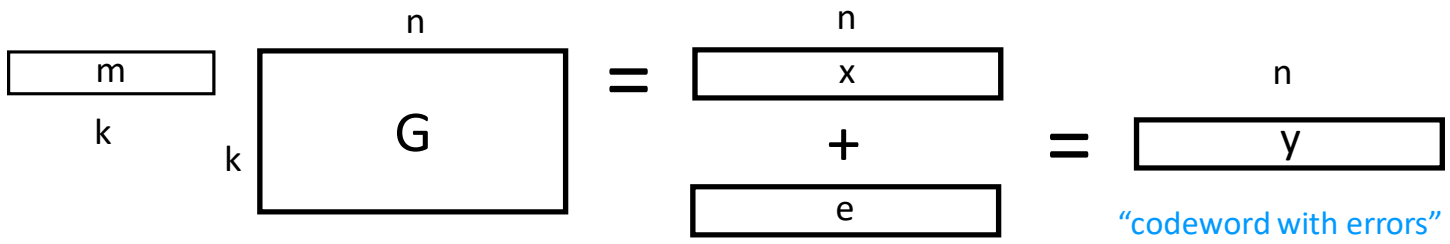
- Commitment
  = Rand. McEliece ciphertext
- Opening: Same as Jain et al.
- Binding is perfect as min. dist. is guaranteed for Goppa codes

- Note: Trapdoor is never used

Applications of commitments

- A lot… ☺
- E.g., a key primitive for constructing zero-knowledge (ZK) proofs
- = proof that yield nothing except validity of the statement
- Normally, the statement is in NP
  (solving the problem is hard, but checking the solution is easy)
- The trick is for Prover to let Verifier check correctness
  of the statement without revealing the solution

ZK proofs



- Example of the statement:

  "For a given (G,y,t) , I know (m,e) : y = mG+e"
- Public data: (G,y,t)
- Witness: (m,e)

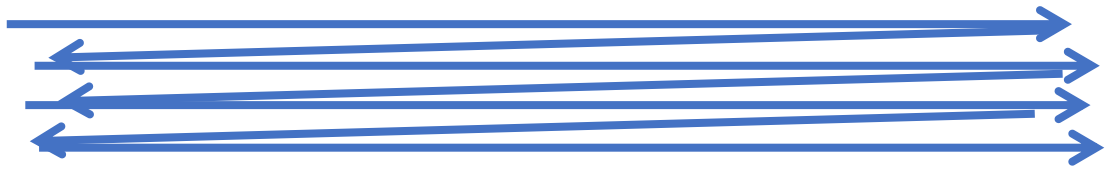
ZK Proofs – Definitions (informal)

Prover P

Proof of NP-statement

Verifier V

Witness



Public data

- Correctness: Honest V always accepts honest P
- Soundness: Cheating P (not knowing the witness) is rejected with overwhelming probability
- Zero-knowledge: Cheating V learns nothing, except for the validity of the statement

Remark: A proof may be non-interactive (out of scope)

Applications of ZK proofs

- A lot ☺
- Natural application: ZK identification scheme, where
- pk = public data , sk = witness
- Recent: Privacy and anonymity in cryptocurrencies (Z-Cash, …)

Very high-level idea – Example

Check digit

- Let a prover P have a credit card no. “1234 5678 9012 3456” (secret)
- P wants to prove to V that she has a number of proper format
- P generates a masking that preserves the check digit
- P : c_0 = com(masking) , c_1 = com(masked no.)
- V : flips a random $b \leftarrow_R \{0,1\}$, asks P to open c_b ,

  if b=0 verify correctness of mask.

  if b=1 verify the check digit

  if “fail” reject , o/w accept

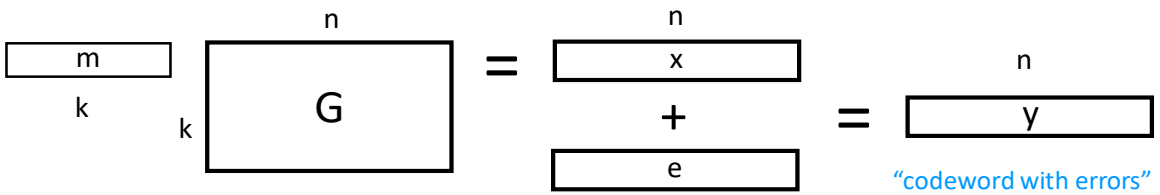
Example (cont)

- P's prob-ty to cheat is 1/2
  - becomes $(1/2)^r$ after "r" rounds
    (independent instances, randomly chosen masking)
- Soundness <= "masking AND masked no. => secret no."
  - But P never reveals both!
- Zero-knowledge <= It is possible to generate the protocol transcript without interacting with P – simulation :
  1) Masking is random
  2) Masked no. is "random looking" => easy to fake something with the last digit which verifies

High-level idea

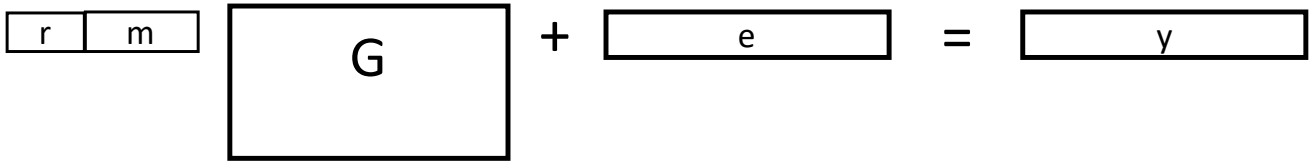
- Prover P randomly “scrambles” the problem
- P performs: com(masking) , com(masked witness)
- … (as before)

ZK proof for Gen. Dec.



- Code-based ZK identification scheme [Stern '93]
  – seminal result (after many failed attempts)
  – based on syndrome decoding problem (to be discussed later)
- [Veron '97]: Dual version of Stern's scheme based on Gen. Dec.
  – shown not ZK , fixed and extended by [Jain et al. '12]

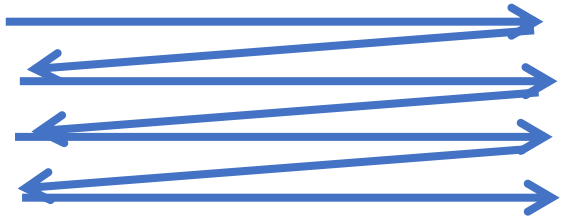
Jain et al.'s ZK Identification Scheme



[Jain, Krenn, Pietrzak, Tentes: Asiacrypt '12]

Prover

Witness:
(r,m,e)



Verifier

Public data:
(G,y)

- Two variants w.r.t. distribution of "e":
  1. Fixed-weight "e": Gen. Dec., soundness error 2/3,
  2. $t_1 \leq w_H(e) \leq t_2$ (variable weight from interval): Suited for LPN, soundness error 4/5
- Designed as proof of validity for their commitment

Jain et al.'s scheme for Gen. Dec.

- [Jain, Krenn, Pietrzak, Tentes: Asiacrypt'12]

Common public data: $G \in \{0,1\}^{k \times n}$

Prover

Secret key (witness): $((r|m),e)$
s.t. $r,m \in \{0,1\}^{k/2}$, $e \in \{0,1\}^n$, $w_H(e)=t$

Verifier

Public key: (y,t)
s.t. $c=(r|m)G+e$, $t=w_H(e)$

1. $v \leftarrow_R \{0,1\}^k$, $f \leftarrow_R \{0,1\}^n$, $\pi \leftarrow_R S_n$

$C_0 = Com(\pi'=\pi, t_0=vG+f)$
$C_1 = Com(t_1=\pi(f))$
$C_2 = Com(t_2=\pi(f+e))$

$\longrightarrow$

2. $b \leftarrow_R \{0,1,2\}$

$\longleftarrow$

Jain et al.'s Protocol (2|2)

$C_0 = Com(\pi'=\pi, t_0=vG+f)$
$C_1 = Com(t_1=\pi(f))$
$C_2 = Com(t_2=\pi(f+e))$

Common public data: $G \in \{0,1\}^{k \times n}$

Prover | Verifier

3.

If b=0: π', t_0, t_1 and open C_0, C_1 → Accept iff $\boxed{t_0+\pi'^{-1}(t_1) \in Code(G)}$

If b=1: π', t_0, t_2 and open C_0, C_2 → Accept iff $t_0+\pi'^{-1}(t_2)+c \in Code(G)$

If b=2: t_1, t_2 and open C_1, C_2 → Accept iff $w_H(t_1+t_2) = t$

- Soundness error is 2/3
- It reduces to $(2/3)^r$ by iterating this protocol independently r times

Why Soundness Error is 2/3?

- Because the cheating prover can always produce the commitments satisfying 2 challenges

Example:

- $C_0 = Com(\pi'=\pi, t_0=vG+f+c)$
- $C_1 = Com(t_1=\pi(f+c))$
- $C_2 = Com(t_2=\pi(f))$

Verification:

- b=0 : Accept iff $t_0+\pi'^{-1}(t_1) \in Code(G)$ => OK
- b=1 : Accept iff $t_0+\pi'^{-1}(t_2)+c \in Code(G)$ => OK
- b=2 : Accept iff $w_H(t_1+t_2) = t$ => **NG**

Proof

- Correctness:
- b=0: $t_0+\pi'^{-1}(t_1) = (vG+f)+\pi^{-1}(\pi(f)) = vG \in \mathrm{Code}(G)$
- b=1: $t_0+\pi'^{-1}(t_2)+c = (vG+f)+\pi^{-1}(\pi(f+e)) + (r|m)G+e = (v+(r|m))G \in \mathrm{Code}(G)$
- b=2: $w_H(t_1+t_2) = w_H(\pi(f)+\pi(f+e)) = w_H(\pi(e)) = t$

Soundness

- Proven by constructing a PPT machine called "witness extractor"
- Extracts a witness (with non-negligible probability) if any PPT prover is accepted by the honest verifier (with non-negl. prob.)
- Shows that the prover must know the witness, if it is accepted

Soundness (informal)

Suppose that all 3 challenges can be properly answered

- Then the binding property => openings to the same commitments must be consistent

- By verification for b=0,1:
- => $\pi'^{-1}(t_1+t_2)+c \in \mathrm{Code}(G)$
- => $c = (r'|m')G+\pi'^{-1}(t_1+t_2)$, where $(r'|m')$ is computed by elementary linear algebra
- … for b=2: $w_H(t_1+t_2)=t$ => $((r'|m') , \pi'^{-1}(t_1+t_2))$ is a valid witness

Rem.: Verification for b=0: Accept iff $t_0+\pi'^{-1}(t_1)\in\mathrm{Code}(G)$
for b=1: Accept iff $t_0+\pi'^{-1}(t_2)+c\in\mathrm{Code}(G)$

Proof of plaintext knowledge (PPK)

Prover P

$c = Enc_{pk}(m,R)$

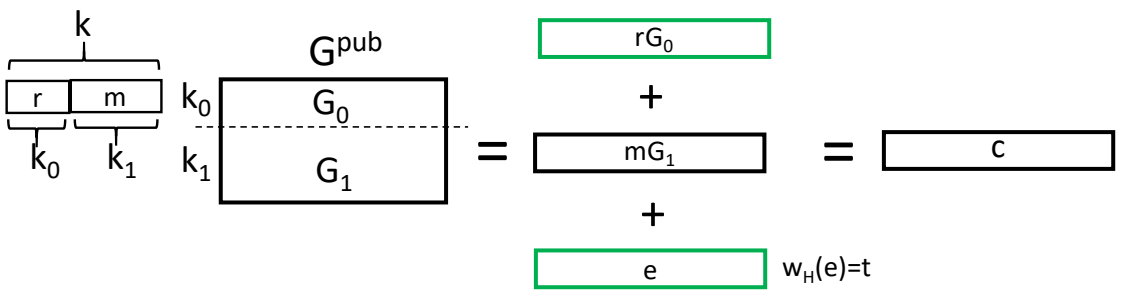
Proof: "I know the plaintext inside c, encrypted on pk"

Verifier V

(Does not know sk!)

- Proposed in [Aumann, Rabin '99 (manuscript)]: generic consr. (any PKE)
- [Kobara, M, Overbeck]: Stern's ZK id. scheme as proof of correctness for code-based oblivious transfer

PPK for code-based PKE



- [M, Takagi '12]: Veron's ZK id. => PPK for McEliece PKE
- [Jain et al. '12]: Veron's schme is not ZK
- [Hu, M, Takagi '13]: Jain et al.'s id. => PPK for McEliece

  Stern's id. => PPK for Niederreiter
- Rand. McEliece has the same "shape" as Jain et al.'s commitment

Issues to take care of

- Q: $G \rightarrow G^{pub}$ is ok? A: Yes.

  Actually, indist. of G^{pub} is not needed for the ZK proof, only for security of the Rand. McEliece
- Q: (pk,sk) $\leftarrow$KeyGen in id. scheme

  but (c ; (r,m,e)) can be generated maliciously

  (Although not G^{pub} assuming PKI)
- A: No problem, as [Bellare, Goldreich '92] show that soundness

  (they call it "validity") will hold for maliciously generated instances

Jain et al.'s main result

- Efficient zero-knowledge proofs for NP-statements using code-based commitments
- I.e., commit to inputs (binary strings) $m_0, m_1, \ldots, m_N$, prove in ZK that $m_0 = C(m_1, \ldots, m_N)$
- For small circuits, it works better than the "MPC-in-the-head paradigm" [Ishai, Kushilevitz, Ostrovsky, Sahai '07]
- Implication for code-based encryption (together with [Hu et al. '13]): **general ZK proofs on the encrypted data!**

Verifiable encryption refs:
[Asokan, Shoup, Waidner '98]
[Camenisch, Damgård '00]
[Camenisch, Shoup '03]

Simple example: Proof of equality

- Verifiable encryption with respect to equality relation

Prover P

$m, \quad c = (r|m)G^{pub}+e$

Proof: "This m is inside c, encrypted on pk"

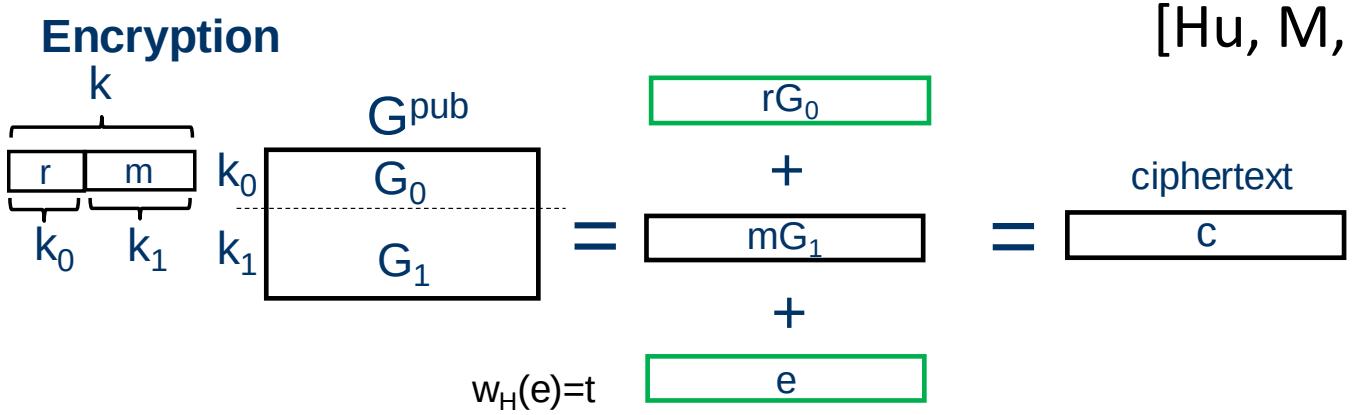
Verifier V

(Does not know sk!)

- Public data: c, G^{pub}, m
- Witness: (r,e) : $w_H(e)=t$ and $c = (r|m)G^{pub}+e$

Verifiable McEliece PKE (1|3)

[Hu, M, Takagi: AsiaCCS'2013]



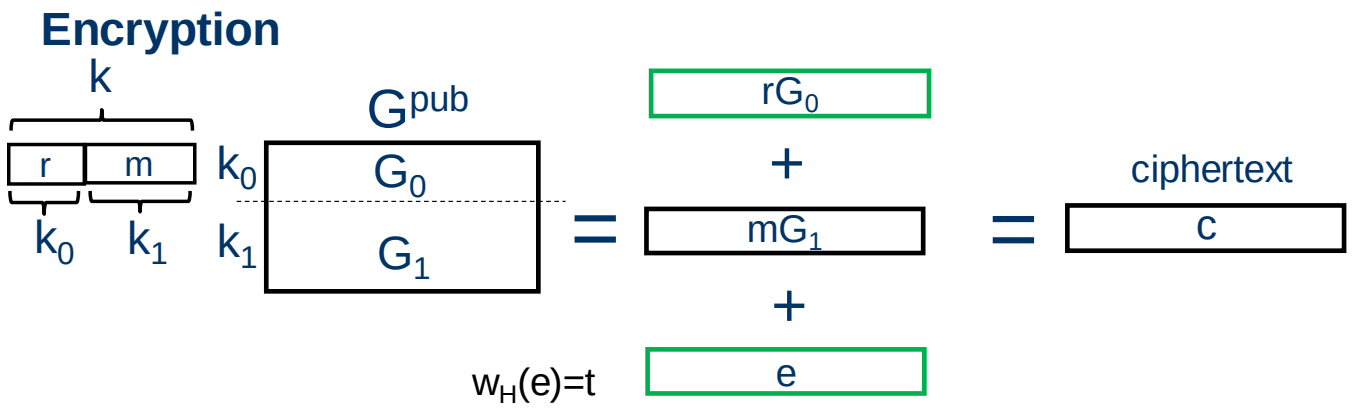
Prover **1.** $c = (r|m)G^{pub}+e,\ m'$ → Verifier

ZK proof for (c,G^{pub}) and $((r|m),e)$ as witness

⇄

- Informally: Proof that "c" is correctly formed (also PPK)
- Note: $\mathrm{rank}(G^{pub})=k$ is a necessary condition → It won't work for Rand. Niederreiter!

Verifiable McEliece PKE (2|3)

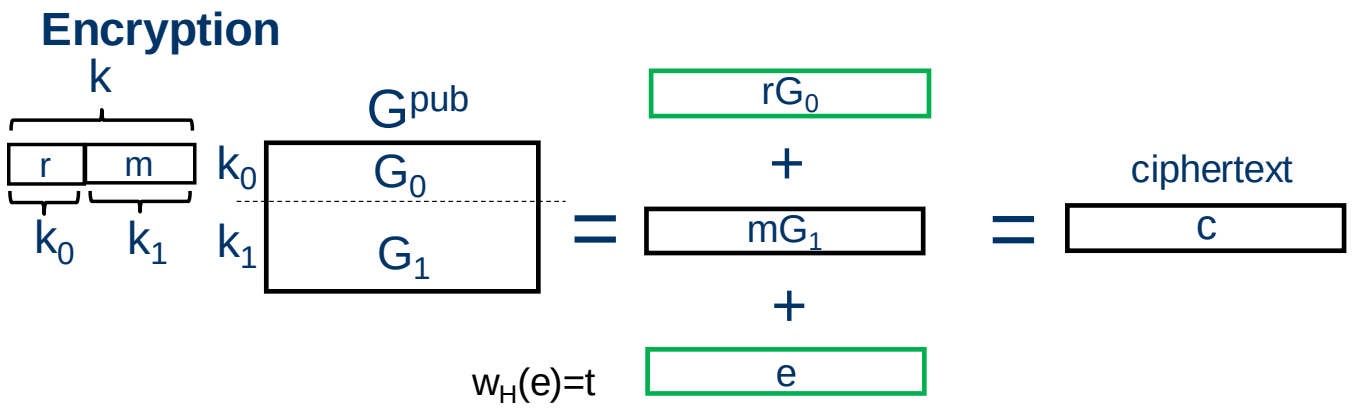


Prover **2.** $c' \leftarrow c \oplus m'G_1 = rG_0 \oplus e$ Verifier

ZK proof for (c', G_0) and (r,e) as witness

- Informally: Proof that m' indeed cancels

Verifiable McEliece PKE (3|3)



Intuition:

- If $m' \neq m$, we have $c' = rG_0 + e + (m'+m)G_1$
- $w_H((m' \oplus m)G_1) \geq d \geq 2t+1$,
  d is the minimum distance of the Goppa code
- => $e + (m' \oplus m)G_1 \geq t+1$ (error vector of weight $\geq t+1$)
- => rejected (by the soundness property)

Proof of inequality

Prover P

m’, c = (r|m)G^{pub}+e

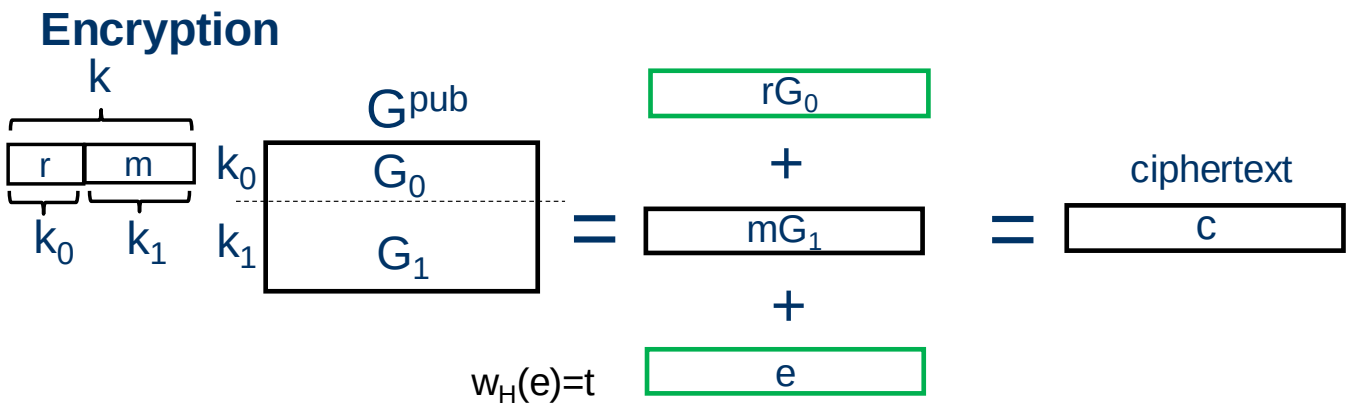
Proof: “This m’ is NOT inside c (encrypted on pk)”

Verifier V

(Does not know sk!)

- Public data: c, G^{pub}, m’
- Witness: (r,e) : $w_H(e)=t$ and c = (r|m)G^{pub}+e

Proof of Inequality (cont.)



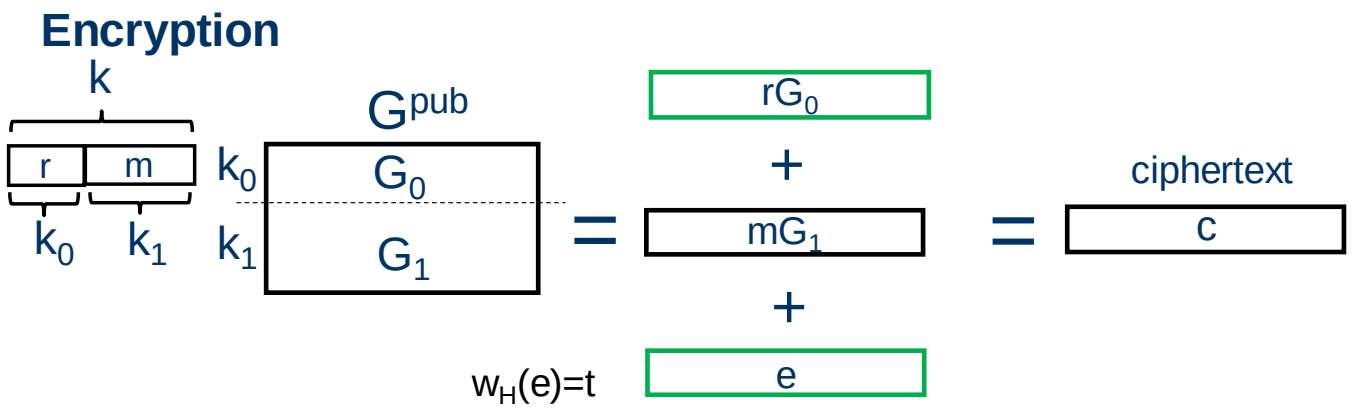
Prover **1.** $\xrightarrow{\quad m',\ c = (r|m)G^{pub}+e \quad}$ Verifier

Proof of valid opening for (c,G^{pub}) and $((r|m),e)$ with $w_H(e)=t$

$\rightleftarrows$

- Informally: Proof that "c" is correctly formed

Proof of Inequality (cont.)

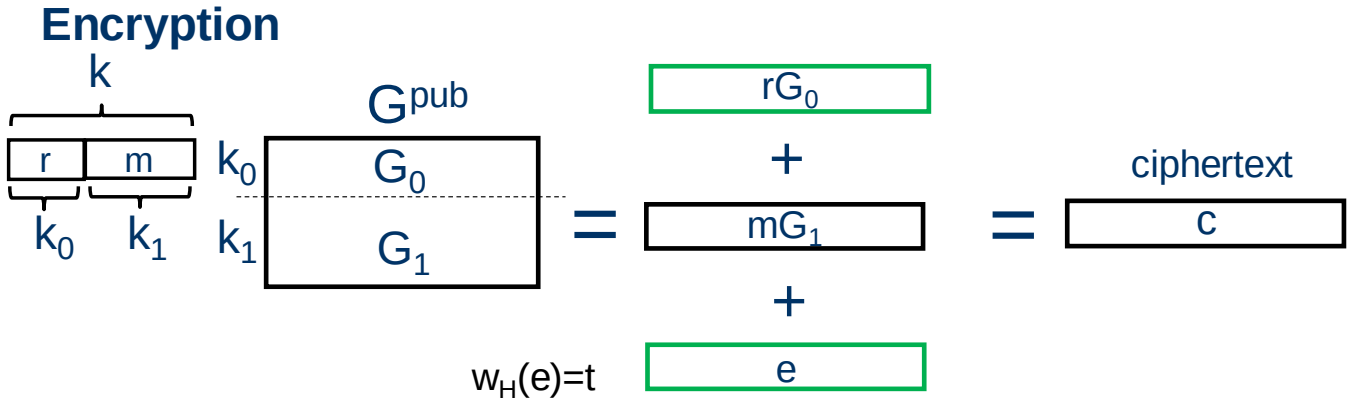


Prover **2.** $c' \leftarrow c + m'G_1$ Verifier

Proof of valid opening for (c',G_0) and (r,e) with $t+1 \leq w_H(e) \leq n$

- Informally: Proof that m' does NOT cancel

Proof of Inequality (cont.)



- If $m' \neq m$, we have $c' = rG_0 + e + (m'+m)G_1$
- $w_H((m'+m)G_1+e) \geq 2t+1-t \geq t+1$, (OK for proof of valid opening)
- If $m' \neq m \Rightarrow w_H((m'+m)G_1+e) = t$
  $\Rightarrow$ rejected (by the soundness property)

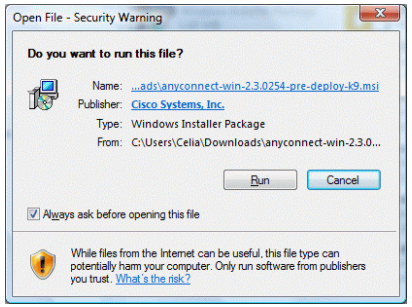
- Remark: We assume $\mathrm{rank}(G^{pub})=k$

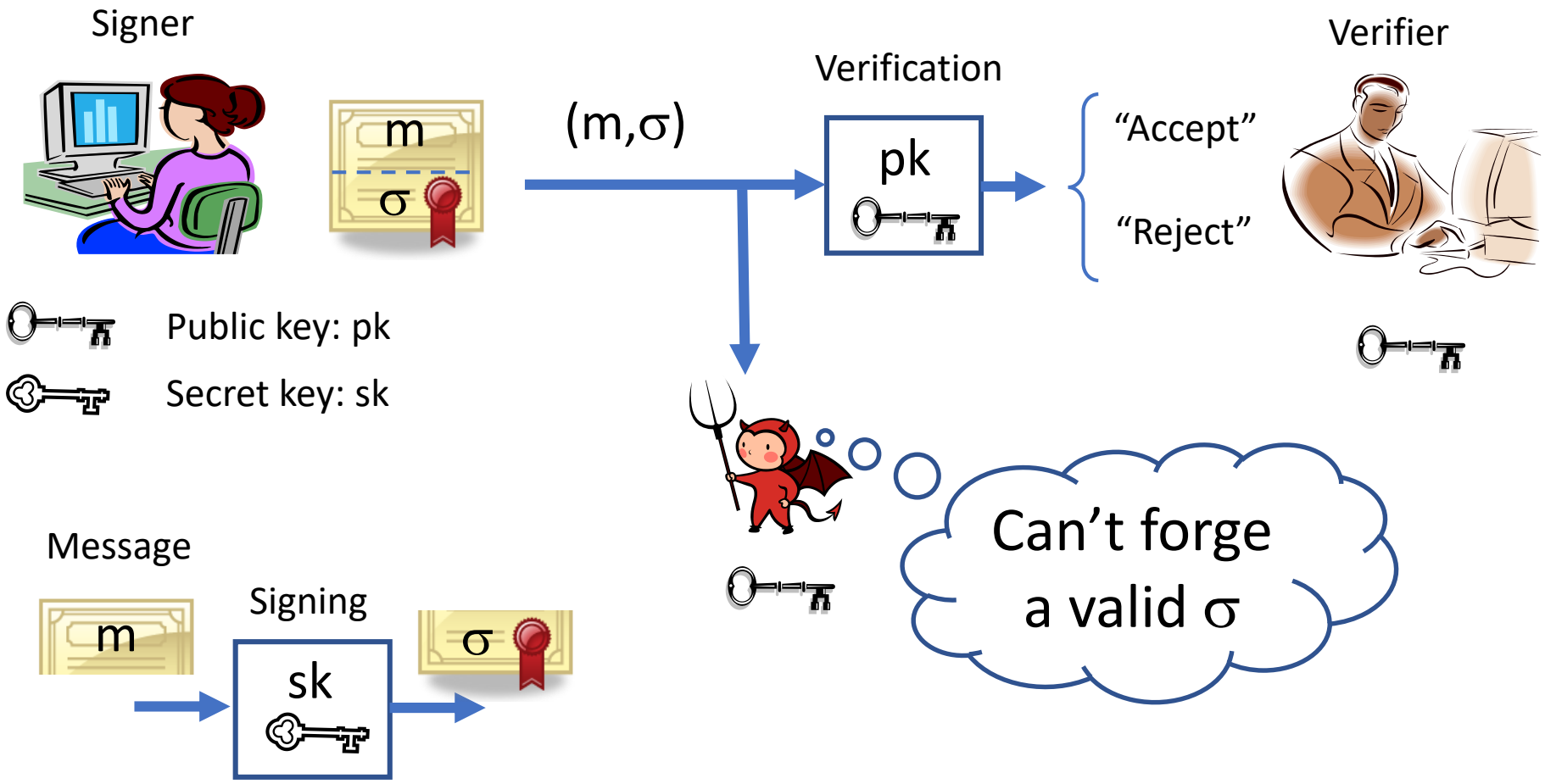
Application: Designated confirmer signature

- Let’s recap digital signatures…

Digital Signature

Triple of algs.: KeyGen
Signing
Verification





Undeniable signatures

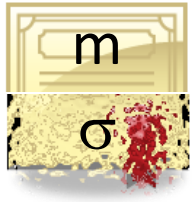
Chaum:
[Crypto '89]
[Eurocrypt '94]

- Controlled verification. Modern application: Privacy/anonymity?
- Designated confirmer sig.: Confirm/Deny delegated to a "confirmer"

Signer

Public key: pk_S

Secret key: sk_S



DCSign

(m,σ)

"Accept"

"Reject"

Verifier

pk_S, pk_C

Confirmer

Public key: pk_C

Secret key: sk_C

Confirm (a valid sig)

Deny (an invalid sig)

DCS: "Signature of Encryption"

- Generic framework by [ElAimani: Provsec'10]
- Key generation: (pk_S, sk_S) – key pair of SEUF-CMA signature
  (pk_C, sk_C) – key pair of IND-CPA public key encryption

Signer

(pk_S, sk_S)

Confirmer

(pk_C, sk_C)

m, $DCSig(m) = (c, \sigma)$,

where $c = Enc(m, pk_C)$

$\sigma = Sig(c, sk_S)$

Confirmation

Proof that c contains m

Denial

Proof that c does not contain m

Verifier

1. Verify the signature σ

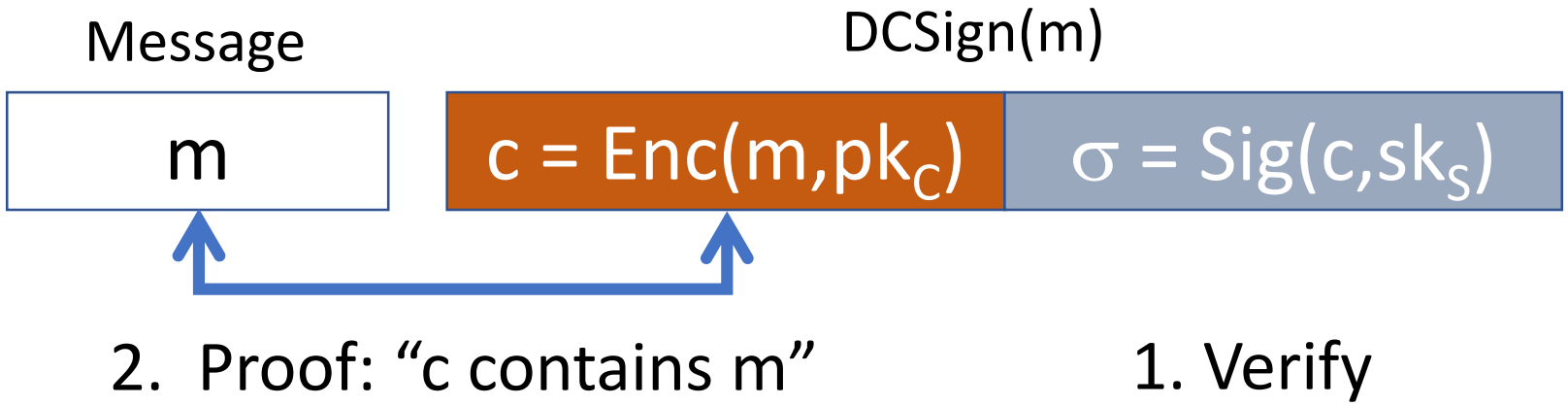
2. Check the proof

Efficient code-based DCS

- [Hu, M, Takagi, 15 (manuscript, Hu's PhD thesis)]:

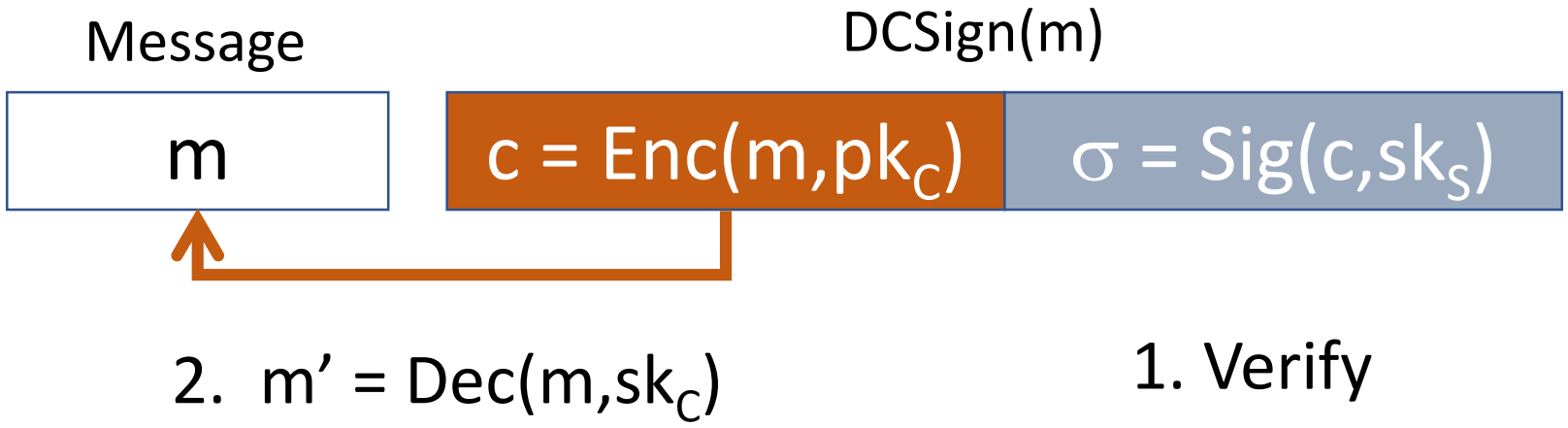
  Plug in all the previously considered results
  in the El Aimani framework

DCSign: Intuition



- "Proof" is a ZK proof with m and randomness as witness
- **Unforgeability:** Unforgeability of Sig
- **Invisibility:** Enc. prevents from connecting DCSign(m) with m
- **Soundness:** Unforgeability of Sig + Soundness of ZKP

Confirmation/Denial: Intuition

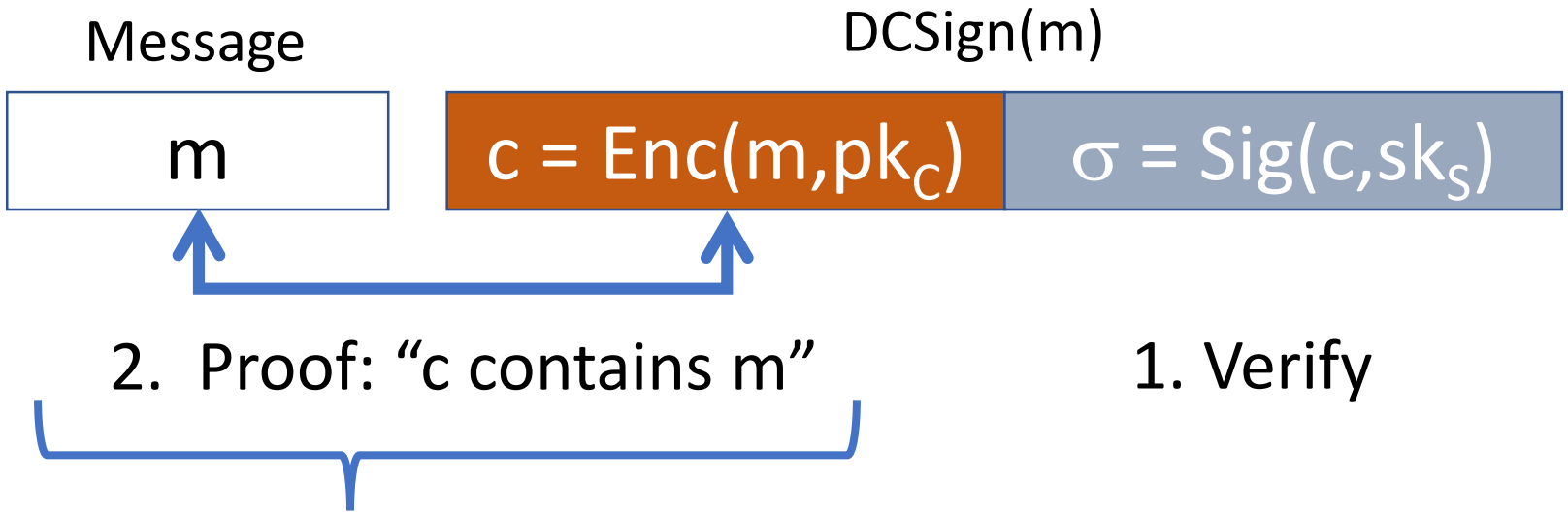


3. If m'=m, then Confirmation: Proof "c contains m"

   else Denial: Proof "c does not contain m"

- **Non-transferability** <= Proofs are ZK
- ***Challenge:*** To construct Confirmation/Denial

Convertibility: Intuition



- Make this proof non-interactive using the Fiat-Shamir paradigm and append it to DCSign(m)
- Then verification can be performed by anyone (holding pk_S) => conversion to an ordinary sig.

SEUF-CMA signature

- Existentially unforgeable under adaptive chosen message attack (EUF-CMA)

  = "Cannot forge a signature even with access to signature oracle"

- Strongly existentially …. (SEUF-CMA)

  = "On top of the above, cannot even forge (msg , sig')

  upon seeing (msg , sig)"

  = Cannot construct an "alternative" <u>valid</u> signature

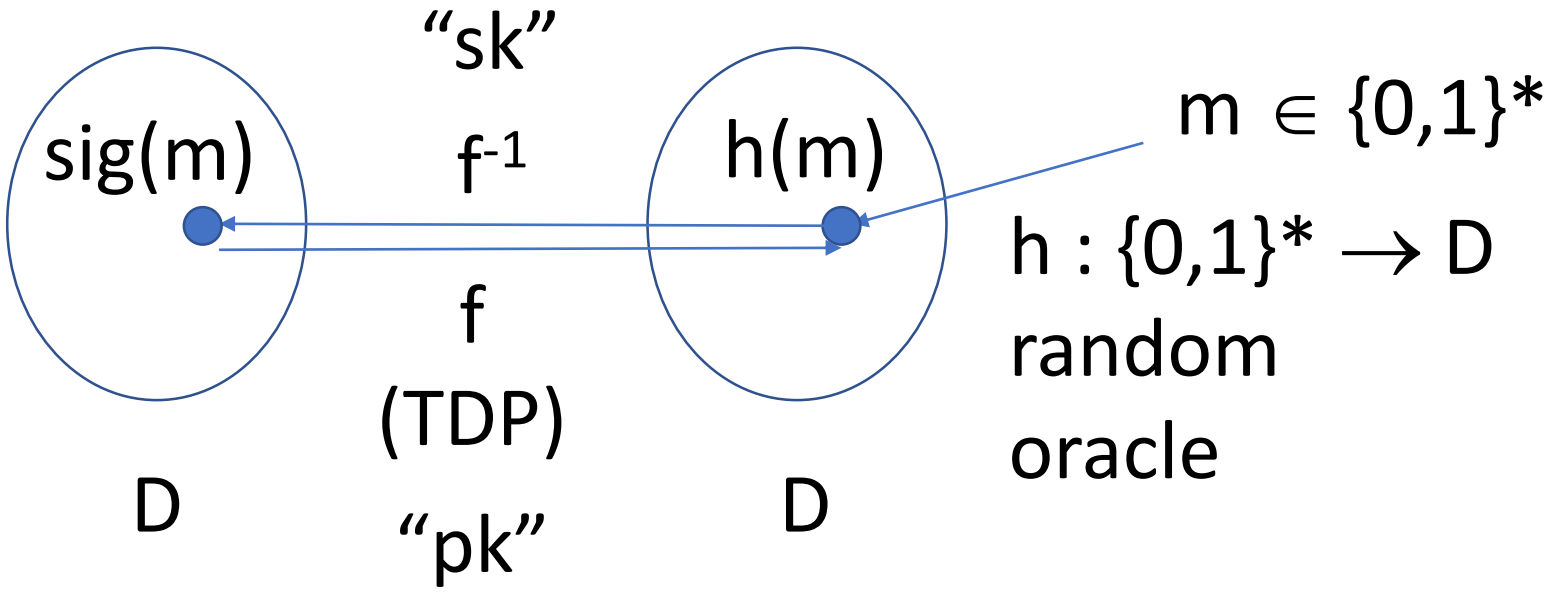
SEUF code-based signature?

To the best of my understanding, as ∃ different replies that lead to successful verification

- Stern-via-Fiat-Shamir signature is EUF-CMA but not SEUF-CMA, and neither is Jain et al.-via-FS
- [M, Roy, Steinwandt, Xu: Open Mathematics ‘18]:
- Courtois-Finiasz-Sendrier (CFS) [Asiacrypt ‘01] signature variant (simple modification) is SEUF-CMA and also secure against key substitution attacks

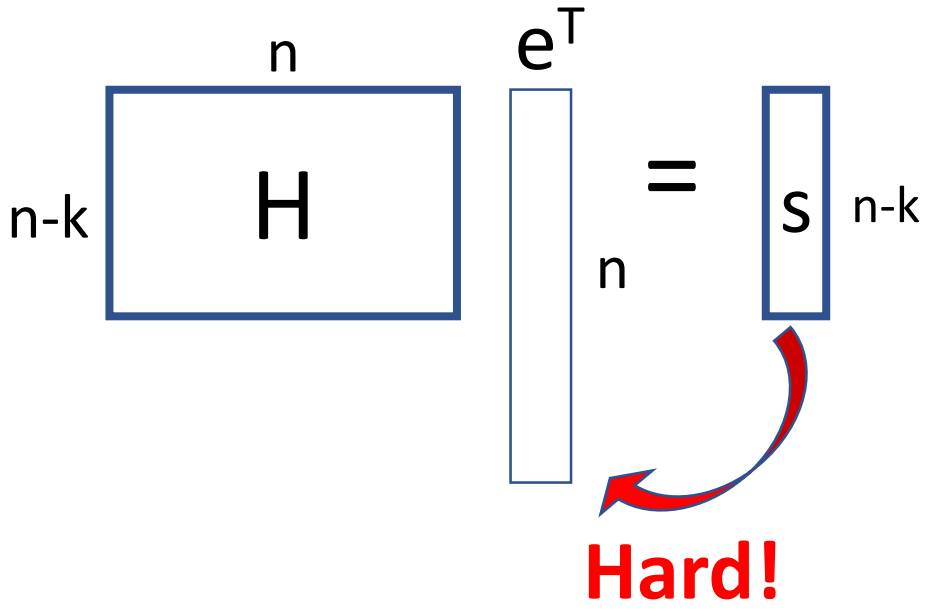
Full Domain Hash (FDH) Signatures

- [Rivest, Shamir, Adleman: Comm. ACM 78] – idea
- [Bellare and Rogaway: Eurocrypt '96] – formalization

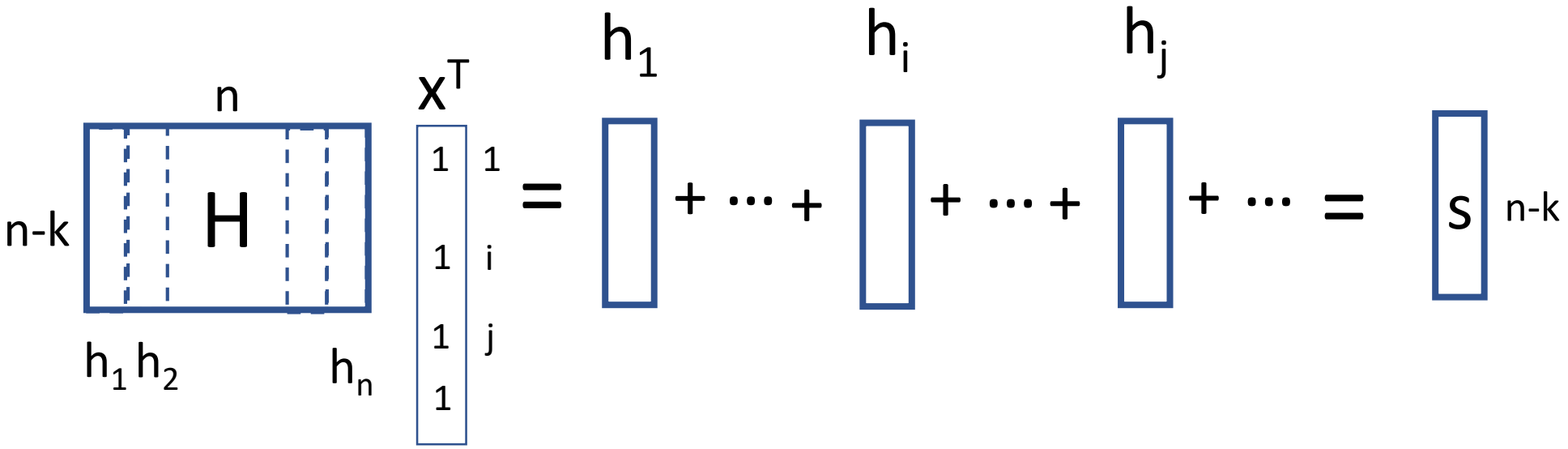


Syndrome Decoding Problem (reminder)

- Input: (H,s,t) s.t.
  - $H \in F_2^{(n-k) \times n}$, a parity-check matrix of a random [n,k] code
  - $s \in F_2^{n-k}$, a syndrome
  - t - an integer
- Find: $e \in F_2^n$, $w_H(e) \leq t$ s.t. $s = He^T$
- NP-complete
  [Berlekamp, McEliece, van Tilborg: IEEE Trans. Inf. Theory '78]

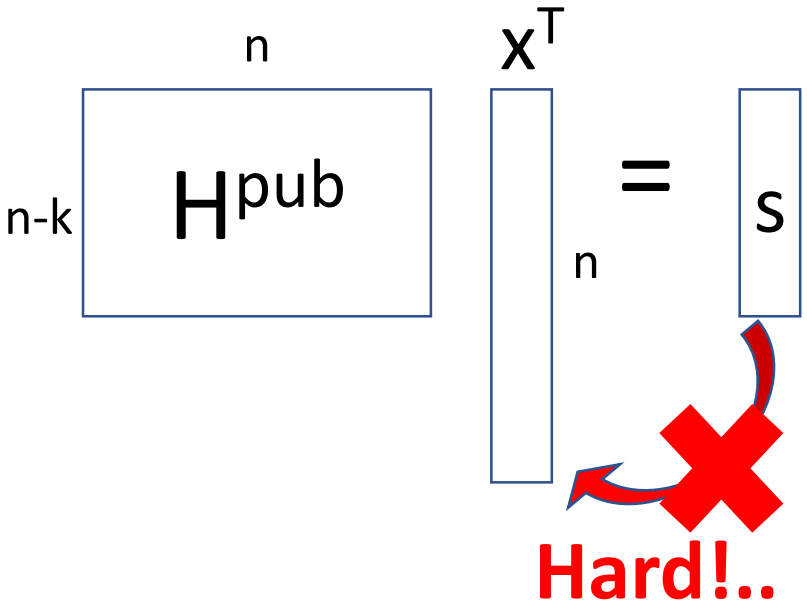


BTW: A variant of the subset sum problem



Embedding a trapdoor a la Niederreiter

- Input: (H^{pub},s,t)
- Find: $x \in F_2^n$, where $w_H(x) \leq t$ s.t. $s=H^{pub}x^T$
- Choose $H^{pub} = MH'P$:
  - $H' \in F_2^{n-k \times n}$ – parity-check matrix of a binary irreducible Goppa code correcting $\leq t$ errors
  - $M \in F_2^{n-k \times n-k}$ – invertible i.e. rank(M)=k
  - $P \in F_2^{n \times n}$ – permutation matrix
- And set the trapdoor as (M,H',P)
  - Its knowledge allows one to efficiently decode H^{pub} and to compute x from (s,H^{pub})



…unless decoding algorithm for H^{pub} is known

Hashing to the range?

- Hashing = sampling a random syndrome
- Decodable? I.e. corresponding to the vector of weight $\leq t$?
- Take $n=2^m$, and $k=n-tm$ (due to Goppa code)
- Pr(decodable) = N_{dec} / N_{tot}
- $N_{dec} = \sum_{i=1}^{t} \binom{n}{i} \approx \binom{n}{t} \approx \frac{n^t}{t!}$
- $N_{tot} = 2^{n-k} = 2^{tm} = n^t$
- => Pr(decodable) $\approx \frac{1}{t!}$
- => t $\downarrow\downarrow$ => k is close to n i.e. "high-rate" codes

Construction

[Courtois, Finiasz, Sendrier: Asiacrypt'01]
[Dallot '07] – random counter
[MRSX '18] – hash with pk

Generic defense from
key substitution attacks (KSA)
by [Menezes, Smart '04]

- Secret key: (M,H',P) and Dec_H – the decoding alg. of H'
- Public key: $H^{pub} = MH'P$
- Message: $m \in \{0,1\}^*$
- Sign:
  1. $i \leftarrow_R \{1,\ldots,2^{n-k}\}$
  2. $x' \leftarrow Dec_{H'}(\, M^{-1} h(m|pk|i)\,)$, for the random oracle $h : \{0,1\}^* \times \{1,\ldots,2^{n-k}\} \rightarrow F_2^{n-k}$
  3. If $Dec_{H'}$ fails to return x', go to 1
  4. Output $sign(m) := (i,x'P)$
- Verify: $s' \leftarrow H^{pub}\,(x'P)^T$ and $s \leftarrow h(m|pk|i)$
  - Output 1 if $s'=s$, and 0 otherwise

Discussion of security proof

- [Dallot: WEWoRC'07]: CFS scheme is EUF-CMA under hardness of SD and Goppa Dist. in ROM
- Purpose: To "go back" to the (general) SD problem, even though having a trapdoor
- [Faugere, Gauthier-Umaña, Otamani, Perret, Tillich: ITW'11]: Distinguisher for the "high-rate" Goppa codes
  - Poly-time for the parameters relevant to CFS
  - Although typical parameters for PKE ($k \approx n/2$) are okay for now

Solution: Drop the GD assumption! ^_^

- Consider the Permuted Goppa Synd. Decoding (PGSD) problem (CFS-parametrized Niederreiter problem):
- Input: (H^{pub},s,t), where $H^{pub} \leftarrow \mathrm{KeyGen}_{CFS}(1^{\kappa})$, $t=t(\kappa)$
- Find: $x \in F_2^n$, $w_H(x) \leq t$ s.t. $s=H^{pub}x^T$

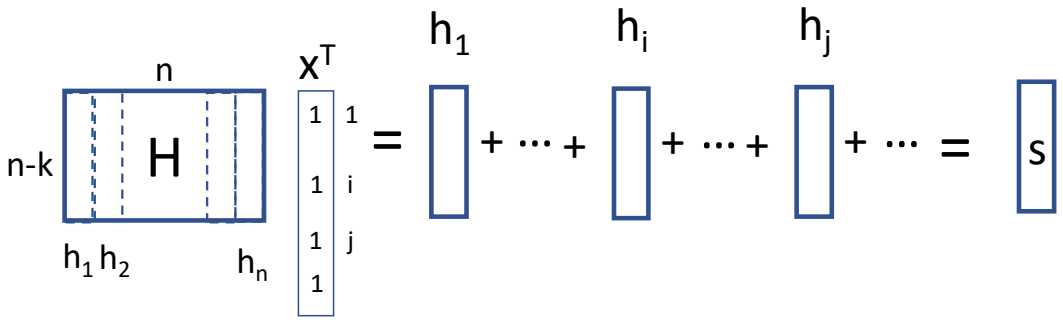
- Pro: Attacker actually faces the PGSD problem!
  - No poly-time attack (whether classical or quantum) is known
- Contra: Less general than SD

[M, Roy, Steinwandt, Xu: Open Mathematics ‘18]

Our Result

- CFS signature variant is SEUF-CMA and KSA-secure under hardness of PGSD in ROM

Key substitution attack



- Strong-key substitution attack – come up with an “alternative” public key pk’, for which a given signature verifies
- Weak-key SA – as above plus sk’ for pk’
- [Dou, Chen, Zhang ‘12]: CFS is susceptible to both WKSA and SKSA
- Intuition for SKSA: Not all columns of H are used for verification
- => another key may work as well

RaCoSS – Random-code-based signature scheme

- [Roy, M, Fukushima, Kiyomoto, Takagi '17]
  - Submitted to first round of NIST PQC Competition
- Code-based id. scheme and signature via Fiat-Shamir with abort
- "FS with abort" is due to [Lyubashevsky: Asiacrypt '09]

RaCoSS – Current status

- Attack on original parameters published in Nov 2017
- Updated secure parameters will be published soon,

  but the keys and signature sizes are terabytes
- Switch to quasi-cyclic (QC) codes will reduce the above sizes to megabytes
- Number of signatures (life-time of the key) may be limited
- Design improvements needed to shift from theoretical to practical security

Anonymity for code-based encryption

Code-based PKE	Random Oracle model	Standard model
IK-CPA		Yamakawa et al. ’07
IK-CCA2	Yamakawa et al. ’07	**Our result**

- IK-CCA2 secure code-based PKE in the standard model [Yoshida, M, Tanaka: PQCrypto ‘17]

Framework for efficient adaptively secure UC oblivious transfer (OT) in ROM

- [Barreto, David, Dowsley, M, Nascimento, Crypto ePrint '17]

  https://eprint.iacr.org/2017/993
- Efficient universally composable (UC) protocol for OT secure against active adaptive adversaries from any OW-CPA secure PKE with certain properties in ROM
- Admits: Low-noise LPN, McEliece, QC-MDPC, and CDH assumpt.
- The first UC-secure OT protocols based on coding assumptions to achieve: 1) adaptive security, 2) low round complexity, 3) low communication and computational complexities

Overview of Code-Based Primitives

- Public-key encryption: 👍
- Signature: 👍
- Commitments: 👍
- ZK proofs: 👍
- Identification schemes: 👍
- Oblivious transfer: 👍
- Leakage-resilient PKE: 👍
- Identity-based encryption: 👍
- Homomorphic encryption: Fully hom.: 👎
  Additively hom.: **?**

Concluding remarks

We have:

- Good understanding of coding assumptions and underlying mathematical problems
- Basic primitives for building code-based cryptographic protocols via generic constructions

We need:

- To study concrete security of existing primitives
- Advanced protocols for privacy / anonymity and computations on encrypted data

Code-based crypto needs you!

UNT®

UNIVERSITY
OF NORTH TEXAS®

EST. 1890

Thank you very much for your attention!