Code-Based Zero-Knowledge Proofs, Signatures and More

Kirill Morozov
UNT

6th Code-Based Cryptography Workshop (CBC 2018)
Davie, Florida, USA
April 6, 2018
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:
  - Confirmersig. and securityof CFSscheme
• Other recent results
• Concluding remarks

“9am is too late at night even for me.”
– Response of an owl person to an invitation
(Traveling mathematical anecdote)
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 \((\text{rank}(G) = k)\)

• \(C = \{mG : m \in F_2^k\}\)

Example: Take \(m=(1 \ 1 \ 0 \ 0)\), \(G := \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}\)

The codeword \(x = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ + & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ = & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}\)

\(\text{Rem.}: \) Obtaining \(m\) from \(x\) is easy!
Minimum distance $d$

- **Hamming weight**
  \[ w_H(x) := \# \text{ non-zero entries of } x \in \mathbb{F}_2^n \]

- **Hamming distance**
  \[ d_H(x,y) := w_H(y-x) = w_H(y \oplus x) = w_H(y+x) = \# \text{ positions where } y \text{ and } x \text{ differ} \]

- **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)

---

**Example:**

\[
\begin{align*}
x &= (0 \ 1 \ 0 \ 0 \ 1 \ 1) \\
w_H(x) &= 3 \\
y &= (1 \ 1 \ 0 \ 0 \ 1 \ 0) \\
d_H(x,y) &= 2
\end{align*}
\]
General Decoding Problem

Input: \((G,y,t)\), where \(G \leftarrow_R \{ F_2^{k \times n} \mid \text{rank}(G)=k \}\),
\(e \leftarrow_R \{ F_2^n \mid w_H(e)=t \}\), \(m \in F_2^k\), \(y=mG+e\)

Output: \((m,e)\)

General Decoding is NP-complete
How to choose “t”?  

• Input \((G,y,t)\), output \((m,e)\)  
• Make the solution “e” unique!  
• \(\Rightarrow\) (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

• \( q \geq 2, \ \forall 0 \leq d/n < 1-1/q, \ 0 < \varepsilon \leq 1-H_q(d/n) \)

• \( \exists \) 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”

\[ \text{Codewords of } C \]

\[ c = x_0 + e \]

\[ \text{Weight} \]

\[ n \]

\[ \text{GV}(n,k) \]

\[ t \]

\[ F_{2^n} \]
How about adding a trapdoor?

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

\[
m = G^k \quad \text{(Random code)}
\]

\[
x = m + e \quad \text{with } w_H(e) = t
\]

\[
y = G^k
\]

“codeword with errors”
Hard!
Nearest Codeword Decoding

- Correct arbitrary pattern of "t" errors \( \leq \) min. dist. \( d(C) \geq 2t+1 \)
- Example: \( c = x_0 + e = (1 1 0 0 0 1 1) + (0 1 0 0 0 0 0) = (1 0 0 0 0 1 1) \)
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.

\[
m = \begin{array}{l}
g \end{array}
\]
\[
ty = \begin{array}{l}
y \end{array}
\]
\[w_H(e) = t\]

Hard! [McEliece ‘78]
Public-Key Encryption

Public database:
- Alice:
- Bob:
- Charlie:

Triple of alg.:
Key generation
Encryption
Decryption

Plaintext $m$  Ciphertext $c$  

Encryption

Decryption

Public key: $pk$

Secret key: $sk$
McEliece PKE: Key Generation

[McEliece, DSN Progress Report’78]

Matrices / vectors are over $F_2$ (binary)

Secret key

Non-singular matrix

$S_{k \times k}$  

$n \times n$ matrix

Generator matrix of a non-singular 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

$G_{k \times n}^{\text{pub}}$

$= SGP$

Random permutation matrix

Goppa Distinguishing Problem: Distinguish $G_{\text{pub}}$ from random code
McEliece PKE: Encryption and Decryption

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

- **plaintext**: $m \in \{0,1\}^k$
- **ciphertext**: $c = mG^{pub} + e$
- $w_H(e) = t$ (weight of error vector)

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

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

• By definition: $H \in \mathbb{F}_2^{n-k \times n} : C = \{ x \in \mathbb{F}_2^n : Hx^T=0 \}$
• $H$ 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)

\[
G := \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}
\quad H := \begin{pmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1
\end{pmatrix}
\]

(n-k)-bit vector called a “syndrome”
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**
A “dual” version of the McEliece encryption

• Syndrome decoding: Input: (H,s,t), output: e
• [Niederreiter ‘86]:
  • e → m (via encoding of messages as t-weight vectors)
  • s → c
  • H → 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

All the matrices and vectors are over $F_2$ (binary)

Secret key

Non-singular matrix

$M_{n-k} \rightarrow \{n-k\}$

$H_{n-k} \rightarrow \{n-k\}$

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

$H_{n-k}^{\text{pub}} \rightarrow \{n-k\}$

$= MHP$

Random permutation matrix

$P_{n} \rightarrow n$

$= P_{n}$

[Niederreiter, Problems of Control and Information Theory ’86]
Niederreiter: En(de)cryption

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

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. $m^T = P^{-1}Pm^T$

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

- $n - k$ columns of $H^{\text{pub}}$
- $n$ columns of $H^{\text{pub}}$
- $m^T$ columns

plaintext

$n$ columns

$ciphertext$

“syndrome of $m$”

$n - k$ columns

$w_H(m) = t$
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

\[
\begin{align*}
\text{Adv} &\quad \text{(pk,sk)} \\
\text{Challenger} & \\
\{0,1\} & \rightarrow b \\
\text{Enc}_{pk}(m_0, R) & \rightarrow c \\
\text{Enc}_{pk}(m_1, R) & \\
\text{c = Enc}_{pk}(m_b, R) &
\end{align*}
\]

If winning probability is negligible, then the PKE is semantically secure or indistinguishable under chosen plaintext attack (IND-CPA)
What messages would you choose?

- $\text{Enc}_{\text{ME}}(m_0)$ must be easy to distinguish from $\text{Enc}_{\text{ME}}(m_1)$
- How about $m_0 = 0^k$?
- $\Rightarrow w_H(\text{Enc}_{\text{ME}}(m_0)) = t$
- While for $m_0 \neq 0^k$, $w_H(\text{Enc}_{\text{ME}}(m_0)) \geq t+1$
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 \( m \neq m' \), 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

- [Nojima, Kobara, M, Imai’08]: If $G_{\text{pub}}$ is indistinguishable from random
  $\Rightarrow rG_0 + e$ is pseudorandom (using [Katz, Shin ’06])

- $\Rightarrow$ 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

Without random oracles
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

**Sender**

\[ a \]

\[ \text{com}(a) \] \rightarrow \text{Commit} \rightarrow \text{Open} \rightarrow \text{Receiver}

**Receiver**

\[ \text{com}(a) \]

\[ \{\text{Accept, Reject}\} \]

**Hiding:** Receiver cannot learn the value

**Binding:** Sender cannot open any other value

**Correctness:** Honest sender is always accepts
Security of commitment schemes

• 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]

To the best of my knowledge
Learning parity with noise (LPN) problem

• Secret $m \in \{0,1\}^k$
• Sample: $(g, <g,m>)+e,$
  where $g \in \{0,1\}^k$, $<\cdot,\cdot>$ – scalar product, $e \in \text{Ber}(\tau)$, $0 < \tau < 0.5$
• Input: “n” samples ; Output: “m”

\[ m \in \{0,1\}^k \]

\[ (g, <g,m>)+e, \]
where $g \in \{0,1\}^k$, $<\cdot,\cdot>$ – scalar product, $e \in \text{Ber}(\tau)$, $0 < \tau < 0.5$

• “g” → column of $G$, “e” with Bernoulli distribution

\[ \begin{bmatrix} m \\ k \end{bmatrix} \begin{bmatrix} G \\ k \end{bmatrix} = \begin{bmatrix} n \\ k \end{bmatrix} \begin{bmatrix} x \\ + \\ e \end{bmatrix} = \begin{bmatrix} y \\ \text{"codeword with errors"} \end{bmatrix} \]
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 same as Gen. Dec.
Jain et al’s commitment (Asiacrypt ‘12)

- **Commit**: \(\text{com}(m) = (r|m)G + e = c\), \(t < GV(n,k)/2\)
- **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”

• \(\Rightarrow (r_1+r_2|m_1+m_2)G = e_1+e_2\) – impossible
  since \(w_H(e_1+e_2) < GV(n,k)\) \(\Rightarrow\) cannot be a codeword

• Statistical = unconditional with negligible error

• Quiz: How to make it perfect? ( = unconditional with zero 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)

- **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
• $pk = \text{public data} , \ sk = \text{witness}$
• Recent: Privacy and anonymity in cryptocurrencies (Z-Cash, ...)

Very high-level idea – Example

• 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 : $c_0 = \text{com(masking)}$, $c_1 = \text{com(masked no.)}$
• 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 \(\leq\) “masking AND masked no. \(\Rightarrow\) secret no.”
  – But P never reveals both!
• Zero-knowledge \(\leq\) It is possible to generate the protocol transcript without interacting with P – simulation:
  1) Masking is random
  2) Masked no. is “random looking” \(\Rightarrow\) 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)

• 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

- 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

Prover
Witness: (r,m,e)

Verifier
Public data: (G,y)

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

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

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

**Prover**

Secret key (witness): $((r|m),e)$

\[ \text{s.t. } r,m \in \{0,1\}^{k/2}, \ e \in \{0,1\}^n, \ w_H(e)=t \]

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

\[ C_0 = \text{Com}(\pi'=\pi, t_0=vG+f) \]
\[ C_1 = \text{Com}(t_1=\pi(f)) \]
\[ C_2 = \text{Com}(t_2=\pi(f+e)) \]

**Verifier**

Public key: $(y,t)$

\[ \text{s.t. } c=(r|m)G+e, \ t=w_H(e) \]

2. $b \leftarrow_R \{0,1,2\}$
Jain et al.'s Protocol (2|2)

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

Verifier

Prover

1. $t_0, t_1$

Accept iff $t_0 + \pi^{-1}(t_1) \in \text{Code}(G)$

2. $t_0, t_2$

Accept iff $t_0 + \pi^{-1}(t_2) + c \in \text{Code}(G)$

3. $t_1, t_2$

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

$C_0 = \text{Com}(\pi' = \pi, t_0 = vG + f)$

$C_1 = \text{Com}(t_1 = \pi(f))$

$C_2 = \text{Com}(t_2 = \pi(f+e))$

- $\pi'$
- $\pi$
- $vG + f$
- $\pi(f)$
- $\pi(f+e)$
- $t_0$
- $t_1$
- $t_2$
- $w_H$
- $\pi^{-1}$
- $\text{Code}(G)$
Why Soundness Error is 2/3?

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

Example:
• \( C_0 = \text{Com}(\pi' = \pi, t_0 = vG + f + c) \)
• \( C_1 = \text{Com}(t_1 = \pi(f + c)) \)
• \( C_2 = \text{Com}(t_2 = \pi(f)) \)

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

• Correctness:
  • b=0: $t_0 + \pi^{-1}(t_1) = (vG + f) + \pi^{-1}(\pi(f)) = vG \in \text{Code}(G)$
  • b=1: $t_0 + \pi^{-1}(t_2) + c = (vG + f) + \pi^{-1}(\pi(f + e)) + (r|m)G + e = (v + (r|m))G \in \text{Code}(G)$
  • 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:

• => $\pi^{-1}(t_1 + t_2) + c \in \text{Code}(G)$

• => $c = (r' | m')G + \pi^{-1}(t_1 + t_2)$, where $(r' | m')$ is computed by elementary linear algebra

• ... for b=2: $w_H(t_1 + t_2) = t \Rightarrow ( (r' | m') , \pi^{-1}(t_1 + t_2) )$ is a valid witness

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

for b=1: Accept iff $t_0 + \pi^{-1} (t_2) + c \in \text{Code}(G)$
Proof of plaintext knowledge (PPK)

Prover P

\[ c = \text{Enc}_{pk}(m,R) \]

Verifier V

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

(Does not know sk!)

• Proposed in [Aumann, Rabin ‘99 (manuscript)]: generic const. (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: $G \rightarrow G_{\text{pub}}$ is ok? A: Yes.

  Actually, indist. of $G_{\text{pub}}$ is not needed for the ZK proof, only for security of the Rand. McEliece

• Q: $(pk,sk) \leftarrow \text{KeyGen}$ in id. scheme
  but $(c;(r,m,e))$ can be generated maliciously
  (Although not $G_{\text{pub}}$ 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!
Simple example: Proof of equality

• Verifiable encryption with respect to equality relation

\[ m, \ c = (r \| m)G^{\text{pub}} + e \]

Verifier V

Proof: “This \( m \) is inside \( c \), encrypted on \( \text{pk} \)”

(Does not know \( \text{sk} \)!

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

Verifiable encryption refs:
[Asokan, Shoup, Waidner ‘98]
[Camenisch, Damgård ‘00]
[Camenisch, Shoup ‘03]
Verifiable McEliece PKE (1|3)

Informally: Proof that “c” is correctly formed (also PPK)

Note: rank($G_{pub}$) = $k$ is a necessary condition

Prover

1. $c = (r | m)G_{pub} + e, m'$

Verifier

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

It won’t work for Rand. Niederreiter!

• Informally: Proof that “c” is correctly formed (also PPK)
• Note: rank($G_{pub}$) = $k$ is a necessary condition

[Hu, M, Takagi: AsiaCCS’2013]
Verifiable McEliece PKE (2|3)

Prover

1. $c' \leftarrow c \oplus m'G_1 = rG_0 \oplus e$

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
• \( \Rightarrow e + (m'\oplus m)G_1 \geq t+1 \) (error vector of weight \( \geq t+1 \))
• \( \Rightarrow \text{rejected (by the soundness property)} \)
Proof of inequality

**Prover P**

\[ m', \quad c = (r \Vert m)G^{\text{pub}} + e \]

**Verifier V**

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

- Public data: c, G^{\text{pub}}, m’
- Witness: (r, e) : \text{w}_H(e)=t \text{ and } c = (r \Vert m)G^{\text{pub}} + e

(Does not know sk!)
Proof of Inequality (cont.)

Informally: Proof that “c” is correctly formed

Encryption

Prover

1. \( m', \ c = (r | m)G^{\text{pub}} + e \)

Verifier

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

• Informally: Proof that “c” is correctly formed
Proof of Inequality (cont.)

**Prover 2.** \( c' \leftarrow c + m'G_1 \)

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)
• $w_H((m'+m)G_1+e) = t$ => rejected (by the soundness property)
• Remark: We assume $\text{rank}(G^{\text{pub}}) = k$
Application: Designated confirmor signature

• Let’s recap digital signatures...
Digital Signature

Triple of algs.: KeyGen
Signing
Verification

Signer

Verifier

Public key: pk
Secret key: sk

Message

Can’t forge a valid \( \sigma \)
Undeniable signatures

• Controlled verification. Modern application: Privacy/anonymity?
• Designated confirmer sig.: Confirm/Deny delegated to a “confirmer”

Chaum: [Crypto ‘89] [Eurocrypt ‘94]
DCS: “Signature of Encryption”

• Generic framework by [ElAimani: Provsec’10]
• Key generation: \((pks, sks)\) – key pair of SEUF-CMA signature
  \((pk_c, sk_c)\) – key pair of IND-CPA public key encryption

\[
m, \text{DCSig}(m) = (c, \sigma),
\]
where \(c = \text{Enc}(m, pk_c)\)

\[
\sigma = \text{Sig}(c, sk_s)
\]

Confirmation

Proof that \(c\) contains \(m\)

Denial

Proof that \(c\) does not contain \(m\)

1. Verify the signature \(\sigma\)
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

- **Message**: $m$
- **DCSign(m)**: $c = \text{Enc}(m, pk_C)$, $\sigma = \text{Sig}(c, sk_S)$

2. **Proof: “c contains m”**

1. **Verify**

- “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 $\leq$ Proofs are ZK
- Challenge: To construct Confirmation/Denial
Convertibility: Intuition

2. Proof: “c contains m”
1. Verify

- 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” valid signature
SEUF code-based signature?

- 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

To the best of my understanding, as $\exists$ different replies that lead to successful verification
Full Domain Hash (FDH) Signatures

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

\[ f(TDP) \]
\[ h(m) \]
\[ \text{random oracle} \]

\[ m \in \{0,1\}^* \]
\[ h : \{0,1\}^* \rightarrow D \]
Syndrome Decoding Problem (reminder)

• Input: \((H,s,t)\) s.t.
  • \(H \in \mathbb{F}_2^{(n-k) \times n}\), a parity-check matrix of a random \([n,k]\) code
  • \(s \in \mathbb{F}_2^{n-k}\), a syndrome
  • \(t\) - an integer

• Find: \(e \in \mathbb{F}_2^n\), \(w_H(e) \leq t\) s.t. \(s = He^T\)

• NP-complete

\[ n \quad e^T \quad = \quad s \quad n-k \]
\[ n-k \quad H \quad n \]

Hard!
BTW: A variant of the subset sum problem

\[
\begin{align*}
    H &= \langle 1, 1, 1, 1, 1 \rangle ^T \\
    &= h_1 + \cdots + h_i + \cdots + h_j + \cdots = s_{n-k}
\end{align*}
\]
Embedding a trapdoor a la Niederreiter

• Input: \((H^{\text{pub}}, s, t)\)

• **Find:** \(x \in F_2^n\), where \(w_H(x) \leq t\)
  
  s.t. \(s = H^{\text{pub}}x^T\)

• Choose \(H^{\text{pub}} = MH'P:\)
  
  • \(H' \in F_2^{n-k \times n}\) – 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. \(\text{rank}(M) = k\)
  
  • \(P \in F_2^{n \times n}\) – permutation matrix

• And set the **trapdoor** as \((M, H', P)\)
  
  • Its knowledge allows one to efficiently decode \(H^{\text{pub}}\)
    and to compute \(x\) from \((s, H^{\text{pub}})\)

\[
\begin{bmatrix}
  H_{\text{pub}} \\
  \vdots \\
  s
\end{bmatrix} = \begin{bmatrix}
  x^T \\
  \vdots \\
  1
\end{bmatrix}
\]

...unless decoding algorithm for \(H^{\text{pub}}\) is known

Hard!..
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)
• \( \text{Pr}(\text{decodable}) = \frac{N_{\text{dec}}}{N_{\text{tot}}} \)
  
  \[ N_{\text{dec}} = \sum_{i=1}^{t} \binom{n}{i} \approx \binom{n}{t} \approx \frac{n^t}{t!} \]
  
  \[ N_{\text{tot}} = 2^{n-k} = 2^{tm} = n^t \]
  
  \[ \Rightarrow \text{Pr}(\text{decodable}) \approx \frac{1}{t!} \]
  
  \[ \Rightarrow t \downarrow \Rightarrow k \text{ is close to } n \text{ i.e. “high-rate” codes} \]
Construction

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

[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]
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 ≈ n/2) are okay for now
Solution: Drop the GD assumption! ^_^

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

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

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

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

- \( \Rightarrow \) 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

<table>
<thead>
<tr>
<th>Code-based PKE</th>
<th>Random Oracle model</th>
<th>Standard model</th>
</tr>
</thead>
<tbody>
<tr>
<td>IK-CPA</td>
<td></td>
<td>Yamakawa et al. ’07</td>
</tr>
<tr>
<td>IK-CCA2</td>
<td>Yamakawa et al. ’07</td>
<td>Our result</td>
</tr>
</tbody>
</table>

- 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
Thank you very much for your attention!