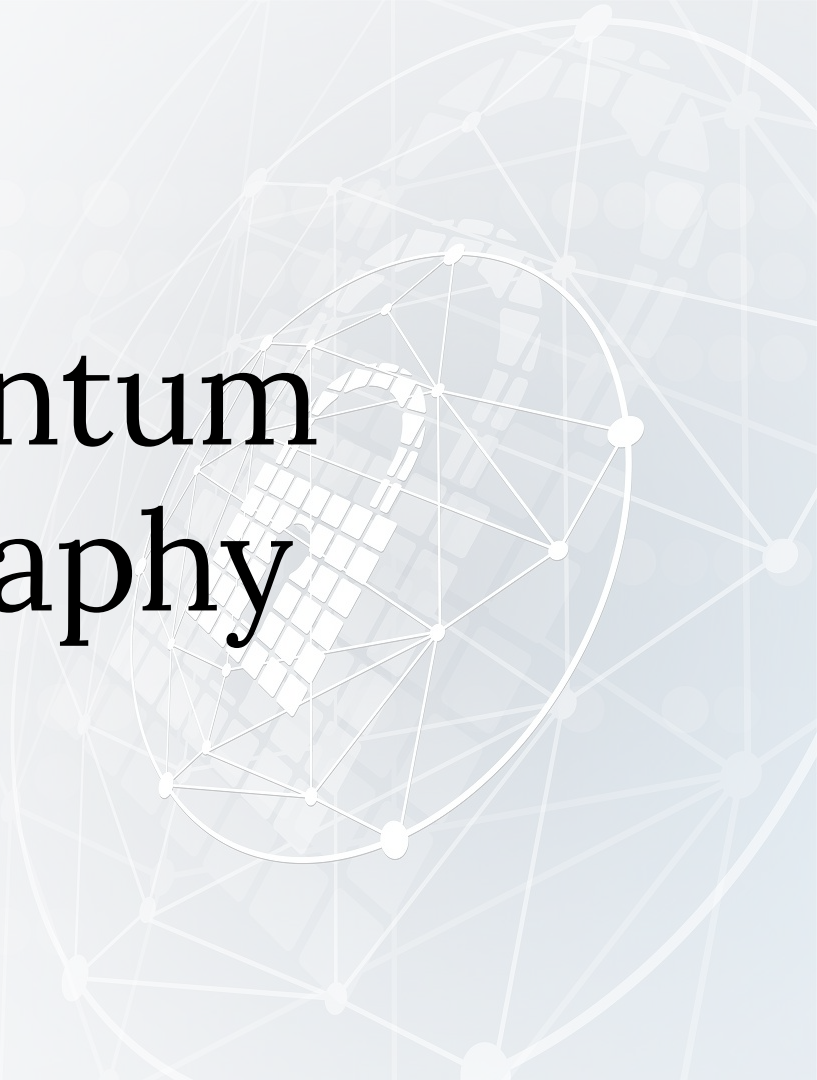
Post-quantum cryptography



Sirius, 2020

Project team

- Зоркин Александр (ТГУ, 2 курс)
  - Wrote almost all code and performed simulation modeling
  - Contacts: @somnoynadno
- Ким Раиса (МФТИ, 3 курс)
  - Performed cryptographic attacks on McEliece cryptosystem
  - Contacts: @rachel16_16
- Иванова Светлана (УрФУ, 2 курс)
  - Added all math models and approximations
  - Contacts: @amoniaka_knabino

Public repo

https://github.com/somnoynadno/McEliece

Contents

Code-based cryptography

Code-based cryptography fundamentals

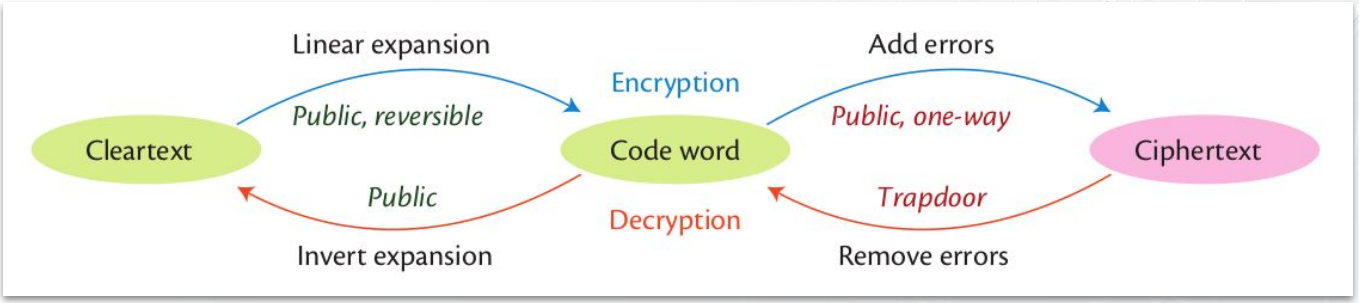
- **Main idea:** let attacker solve NP-hard problem (arbitrary code decoding)
  - Let's hide structure of code with well-known decoding algorithm with some linear operations (like permutations) and get **G'**
  - Use **G'** to encode word with some error vector
  - If we know how to decode codeword and which operations we did with **G**, we can easily extract this word back
  - For attacker this code seems **arbitrary**
- **McEliece** and **Niederreiter** cryptosystems are most popular in code-based cryptography
  - Their **equivalence** is proved
  - For such cryptosystem we can use **any linear code** with *effective* decoding algorithm
  - This cryptosystems are *persistent* under quantum computer
  - Main flaw: large key sizes (*e.g. 13 Mb*)

McEliece cryptosystem: keys

- Suppose G is a generation (k, n)-matrix for a (n, k)-linear code correction t errors with well-known decoding algorithm
- Pick random non-singular (k, k)-matrix S
- Pick random permutation (n, n)-matrix P
- **Public key** is a pair (SGP, t)
- **Private key** is a tuple (S, G, P)

McEliece cryptosystem: encryption

- Encrypt message **m** (with t and SGP=G')
  - Pick random error vector **e** with length n and weight w <= t
  - Encode message like **c = mG' + e**
- Decrypt ciphertext **c** (with S, G and P):
  - Compute **c' = cP^{-1}**
  - Decode **c'** with known decoding algorithm -> get **m'**
  - Compute **m = m'S^{-1}**



LDPC-based McEliece cryptosystem

LDPC code can be a good base for a McEliece cryptosystem because of:

1. LDPC structure is **easy to hide**
2. Some **fast decoding** algorithms exist
3. **Compact** QC-LDPC can **reduce key size** (*can be 1760 times smaller!*)

Disadvantages are:

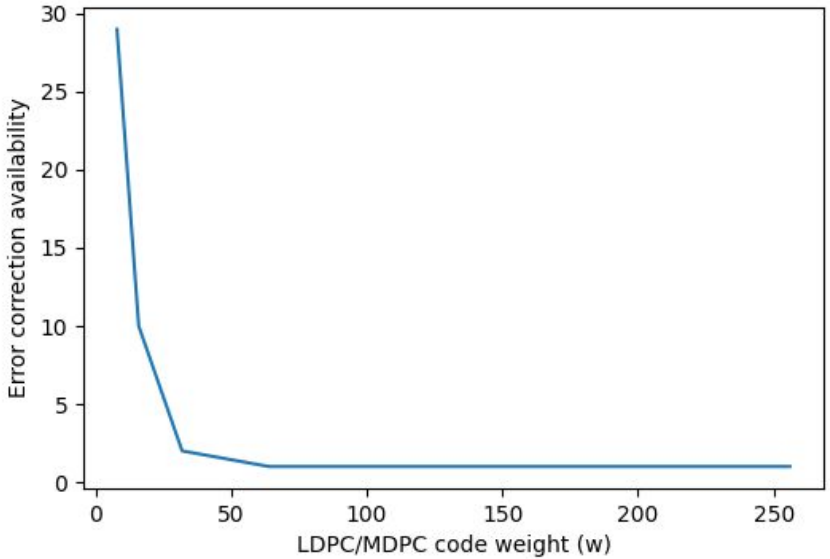
1. **Error correction availability** of randomly generated LDPC **is unknown**
   a. *Density evolution* is used for long-length codes
   b. *Finite-length analysis* is used for short-length codes
2. LDPC code suffers from a **dual code** and **density reduction** attacks
3. Need to adopt **very large code** to reach sufficient security level

LDPC codes error correction availability comparison

Simulation modeling details

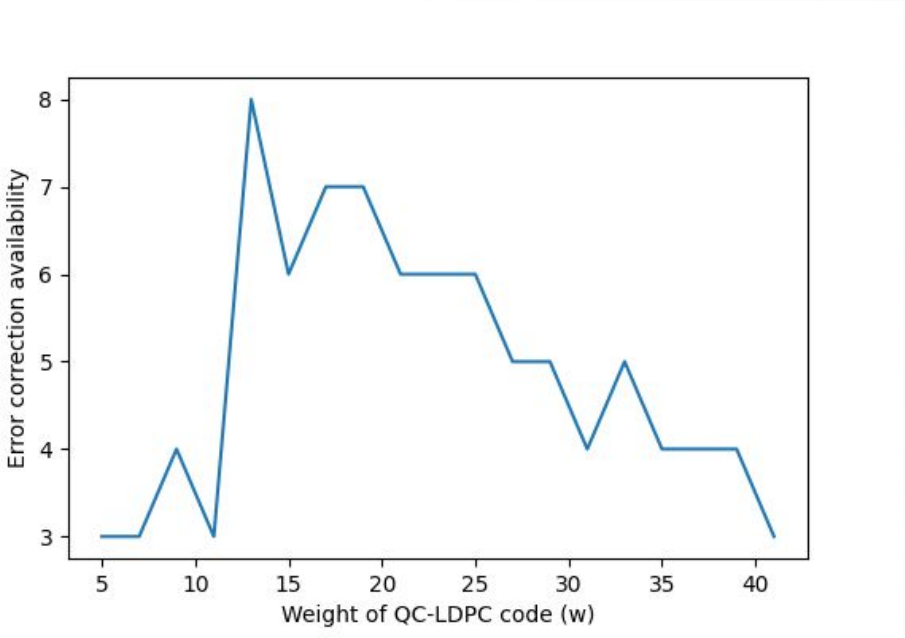
- Decoding algorithm: **bit-flipping**
  - Hard-decision decoding
  - 200 maximum iterations
  - Confidence interval: 97%
- LDPC generation algorithm:
  - **Gallagher's** algorithm for regular codes
  - **MacKey's** algorithm for irregular codes
- Naming nuances:
  - Codes with w <= 10 called **LDPC**
  - All heavier codes are **MDPC** (w ~ $\sqrt{n \cdot log(n)}$)
  - Codes based on *circulants* are **QC-MDPC/QC-LDPC**

LDPC/MDPC row weight estimation



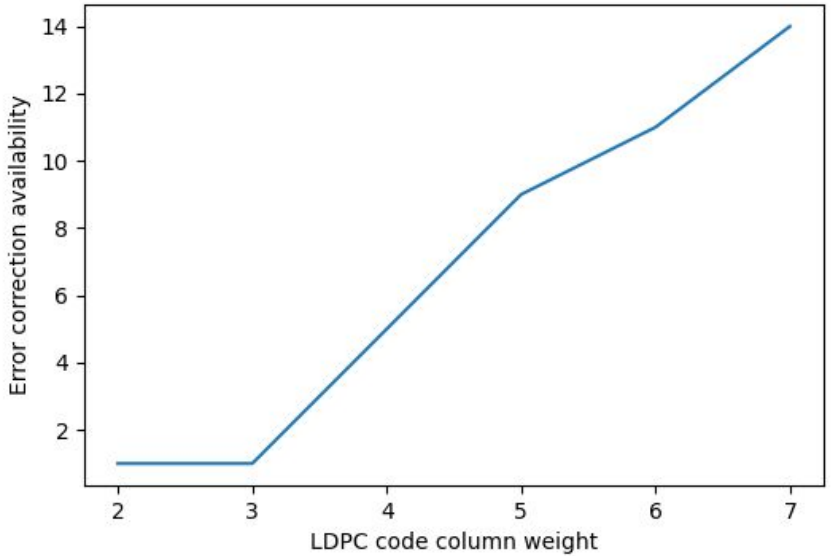
- n = 512, $w_c = 6$
- w_r goes by 2^i
- Conclusion: sparse LDPC matrix allows to fix more code errors

QC-LDPC row weight estimation



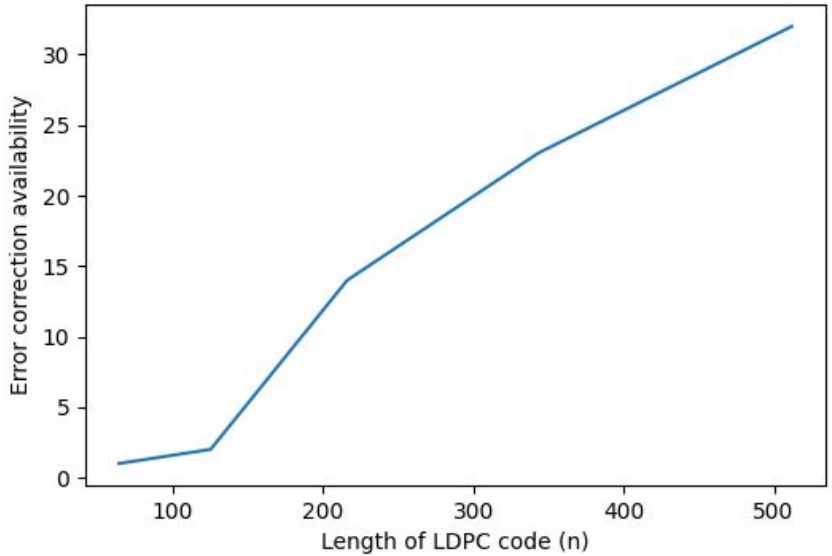
- n = 302, p = 151
- Conclusion: QC-LDPC codes are worse in error correction and need more code weight than LDPC

Regular LDPC column weight estimation



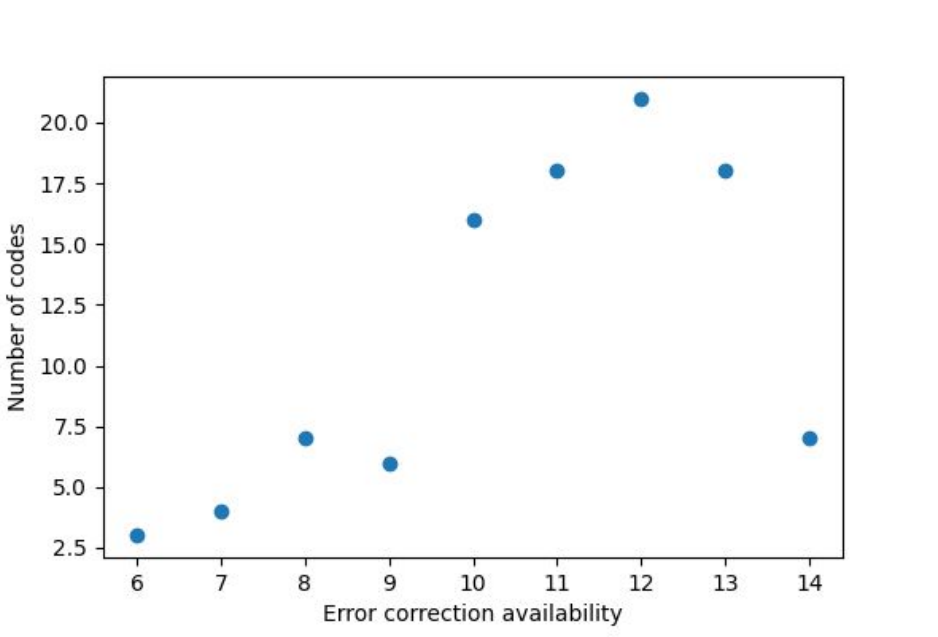
- n = 256, w_r = 8
- Conclusion: bigger column weight allows to correct more errors, but decreases code rate
- Dependency close to linear

Regular LDPC code length estimation



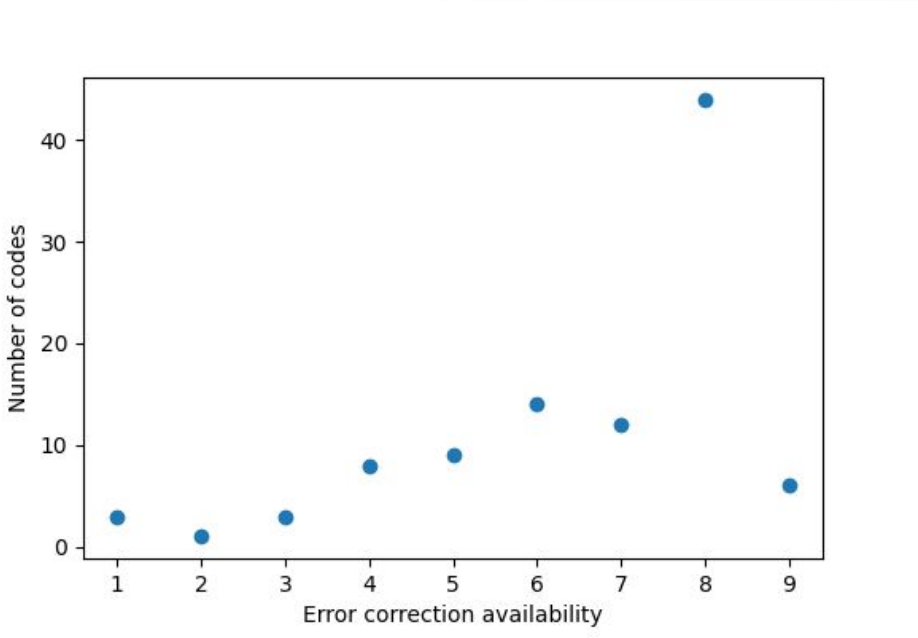
- $w_r = n^{-1/3}$, $w_c = w_r - 1$
- Conclusion: evidently, bigger length allows to fix more errors
- Dependency close to linear

Regular LDPC **t** parameter distribution



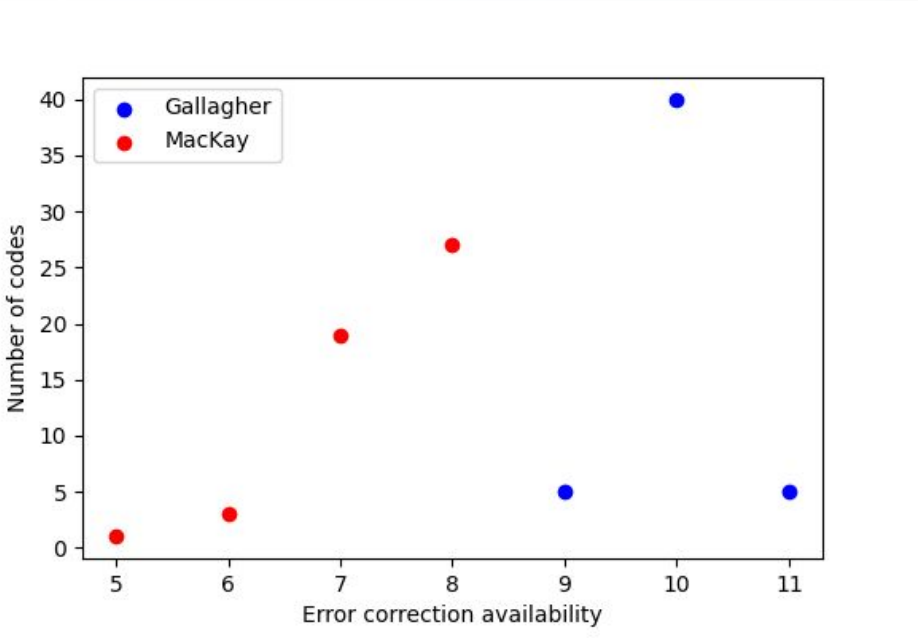
- n = 300, w_r = 6, w_c = 4
- 100 iterations
- E(X) = mean(X) = 11, std(X) = 2
- Conclusion: error amount looks like log-normal distributed (not proven)

QC-LDPC **t** parameter distribution



- n = 302, p = 151, w = 13
- 100 iterations
- E(X) = 6.65, std(X) = 1.9
- Conclusion: QC-LDPC can correct less errors than LDPC with the same parameters

Regular/irregular MDPC comparison



- n = 266, w = 14
- 50 iterations for each
- $E(X_{reg}) = 10$, $E(X_{irreg}) = 7.44$
- Conclusion: regular code is better with small **n** parameter, irregular codes can be better on a big very sparse matrix

Common attacks on McEliece cryptosystem

Common attacks on LDPC-based McEliece

- Attacks on arbitrary linear code
  - Bruteforce attack
  - Stern attack
  - ISD attack
- Attacks on LDPC algebraic structure
  - Density reduction attack
  - Dual code attack
  - Attack on circulants (for QC-LDPC case)
- Side-channel attacks
  - Attack based on decoding time

That's why MDPC code is preferable!

Level of security

Definition: **n-bit security level** means that the attacker would have to perform **2^n operations** to break cryptosystem with such parameters.

MDPC generated with parameters **n = 9602** and **w = 90** can obtain the security level of **80 bits**.

For comparison: classical McEliece cryptosystem with (1024, 524, 101)-Goppa code can guarantee ***50 bits*** *security level.*

Bruteforce Attack

If Alice add **exactly t errors** to her n-codeword after encoding, Eva has two possible ways:

1. Try all 2^k words (maximum likelihood decoding).
2. Perform $\binom{n}{t}$ operations (*upper bound*) to find exact error vector and then attack arbitrary linear code with Information Set Decoding (would be very fast, because Eva now handle codeword without any errors).

Multiple Encryption Attack

Improper cryptosystem usage can cause Bruteforce Attack acceleration.

For example, encrypting the **same message** with the **same key** is **danger**. It can reduce possible brute area up to $\binom{2t}{t}$ codes (*two messages, best case, upper bound*).

```
      c = 0100110101011011101001000110110101
c + e1  = 0100110001011011101101000110100101
c + e2  = 0110110111011111101000000110110101
   diff = 01?0110?01011?11101?0?0001101?0101
```

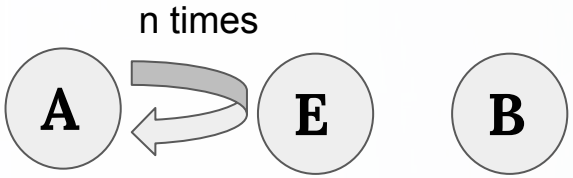
Eva knows exact error positions. Now she need to guess t bits on that positions while the syndrome is not equal to zero.

Interesting remark: so far encryption of blocks with **small entropy** (like images) might be insecure: hacker could extract some information about plaintext in this case. Also proven that multiple encryption decreases block entropy and make **ISD attack** easier.

Man In The Middle Attack

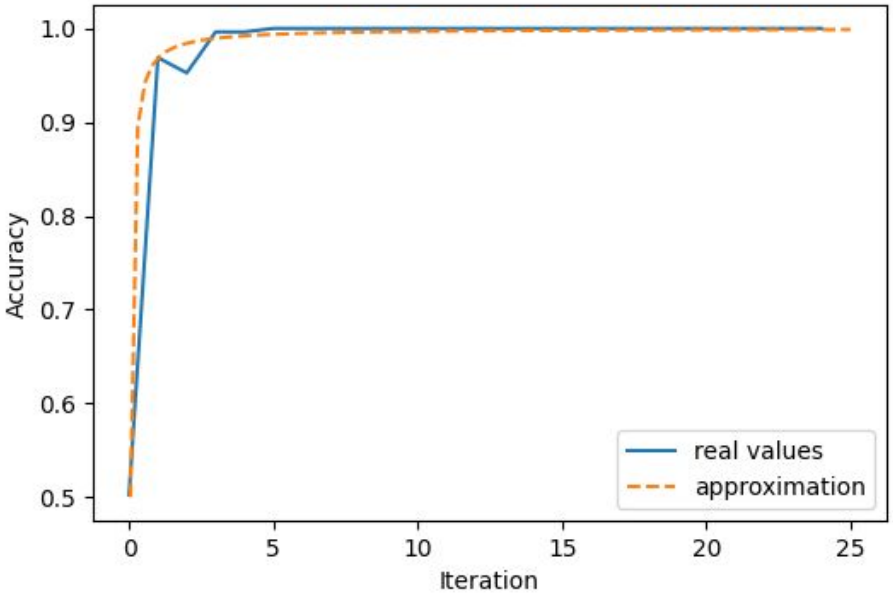
Suppose that Bob can discard messages from Alice if he can't decode them (*that happens sometimes*). Then Alice send exact the same message encrypted with the same key (*clarification*: **G'** stays the same, **e** is always different).

If Eva perform **MITM** attack (Alice think that Eva is Bob) and get **n** messages from Alice (saying "*Hey, I can't decode it, send me once more*"), she can **recover the cleartext unambiguously** with $n \to \infty$ and some probabilistic assumptions.




```
        c = 01001101010110111010010001101101 01
c + e1  = 01001100101101111011010001101001 01
c + e2  = 01101101110111111101000001101101 01
c + e3  = 01001101110110111011010011101101 00
.....................................................
guess   = 01001101010110111010010001101101 01
```

MITM Attack simulation results



- Arbitrary linear code
- n = 800, t = 30
- Accuracy formula: $(n-e_{guessed})/n$
- Conclusions:
  - **error positions could be uniquely determined in 6 iterations** (*in average*)
  - convergence is guaranteed by a law of large numbers

Reaction Attack

Described on previous slides attack on message (aka *reaction attack*) can be improved and be able to **recover private key** to decrypt any ciphertext.

After 356 million different ciphertext observations entire code distance spectrum could be determined and private key recovered for MDPC with n=9602, w=90, t=84.

This attack could be performed in a few minutes.

Security level falls from 80 to 28 bits.

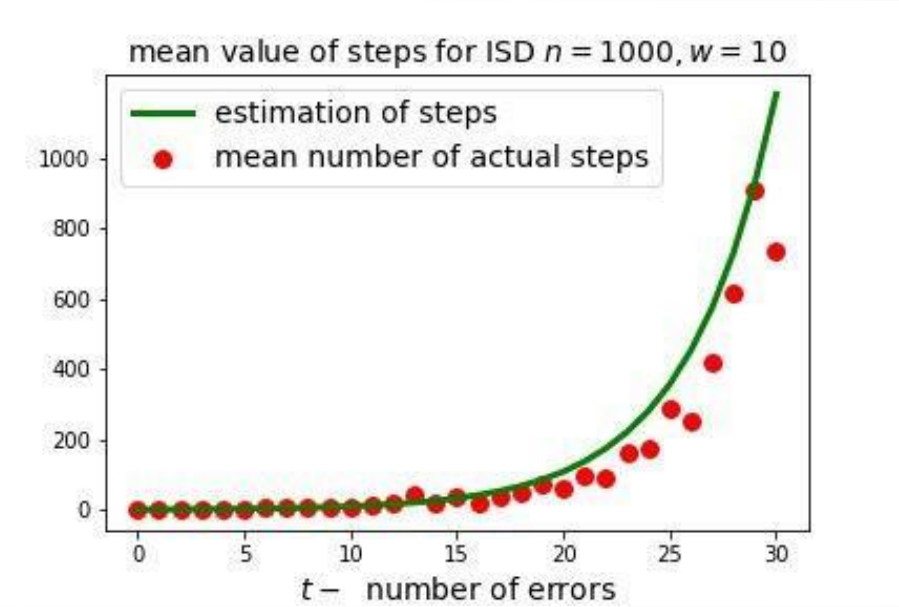
Information Set Decoding Attack

Information Set Decoding (ISD) is one of the best tools to crack code-based cryptosystems.

Following algorithm can extract message **m** with $\frac{\binom{n}{t}}{\binom{n-k}{t}}$ iterations (*in average*):

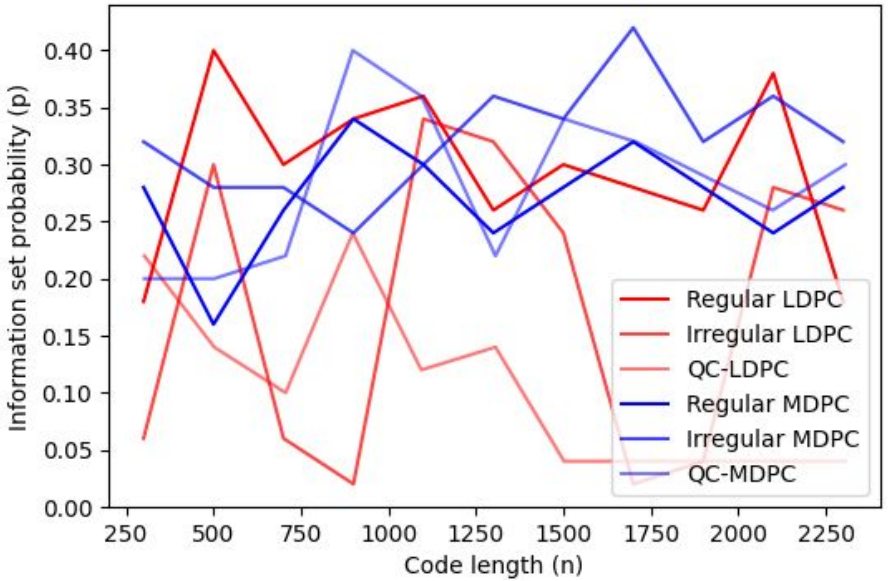
1. Pick random information set ***I*** from {1, 2, …, n}
2. If $\mathbf{x_I}$ does not contain errors, $x_I = m_I G_I + e_I$ (*explanation*: G_I, m_I and e_I contains indices only from ***I***)
3. If $wt(x + x_I G_I^{-1} G) = t$, thus x_I does not contain errors and $wt(e_I) = 0$. In this case $\mathbf{m} = \mathbf{x_I G_I}$. In other way go to step 1.

ISD Attack simulation results



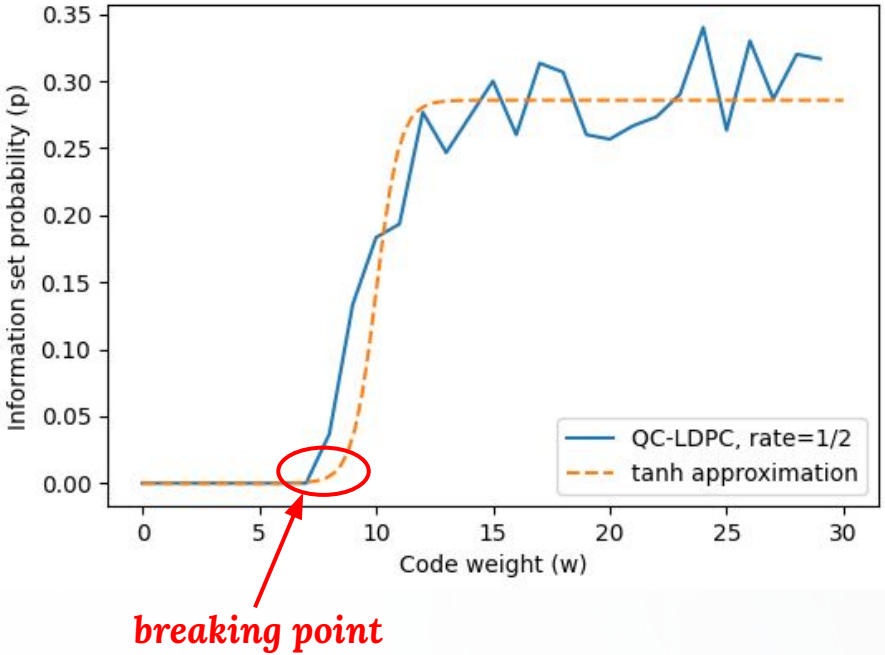
- LDPC code rate: 1/3
- Security level: 19 bits
- <u>Conclusions:</u>
  - attack complexity grows exponentially with number of errors in linear code
  - Time ~ number of steps

Information set probability in arbitrary G



- Parameters:
  - LDPC w = 10
  - MDPC w = $\sqrt{n \cdot log(n)}$
  - LDPC code rate ~ ⅕
  - QC-LDPC code rate = ½
- Conclusions:
  - Every **4th** MDPC subset is information set (*in average*)
  - Every **25th** LDPC subset is information set (*for large* ***n***)
  - Information sets amount *might be* correlated with **w**

Information sets/code weight correlation



- QC-LDPC, n = 502, p = 251
- Conclusions:
  - Hard to extract information set from a very sparse matrix
  - After some threshold **w'**, valid information set could be extracted with **p ~ 0.28**
  - **Some stochastic formula** connecting n, k, w and amount of informations sets **could exist**

Breaking point searching: math approach

We suggest simple formula to find "*breaking point*" with good probability:

- Let's determine random event **X** like "subset matrix is not full-rank because of 2 or more rows containing only zeros"
- Hence, $P(X) = \sum_{i=2}^{k} p_0^i (1-p_0)^{k-i} \binom{k}{i}$, where $p_0 = (\frac{n-w}{n})^k, n \to \infty$
- If for some **w'** P(X) < 0.99, "*breaking point*" found successfully

Summary

1. LDPC codes are much better in error correction, but worse in security level than MDPC codes. For cryptography **picking bigger MDPC is preferable** than picking smaller LDPC instead.
2. **Probability of picking an information set** in arbitrary linear code with known parameter w **can be estimated**. There *could be* a probabilistic formula to approximate it. We suggested good formula to find "*breaking point*" in that unknown function.
3. Code-based cryptosystems could be **completely destroyed** under MITM Attack in conjunction with Reaction Attack.

References

1. Modern Coding Theory (Tom Richardson, Rudiger Urbanke)
2. A Public-Key Cryptosystem Based On Algebraic Coding Theory (R. J. McEliece)
3. An Introduction to Low-Density Parity Check Codes (Daniel J. Costello, Jr.)
4. Construction Of LDPC Codes Using Randomly Permutated Copies Of Parity Check Matrix (Amr Yehia Lulu)
5. On the Usage of LDPC Codes in the McEliece Cryptosystem (Marco Baldi)
6. LDPC codes in the McEliece cryptosystem: attacks and countermeasures (Marco Baldi)
7. QC-LDPC Code-Based Cryptography (Marco Baldi)
8. A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors (Qian Guo, Thomas Johansson, and Paul Stankovski)
9. MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes (Rafael Misoczki and Jean-Pierre Tillich and Nicolas Sendrier and Paulo S. L. M. Barreto)
10. Information-set decoding for linear codes over F_q (Christiane Peters)