Overview of COD's Research: Coding & Cryptography

Antonia Wachter-Zeh

Rudolf Mößbauer Tenure Track Assistant Professor
Professorship for Coding for Communications and Data Storage (COD)

Industry Day 2019

COD Group

Currently **9 researchers** (1 professor, 1 postdoc, 7 doctoral students)

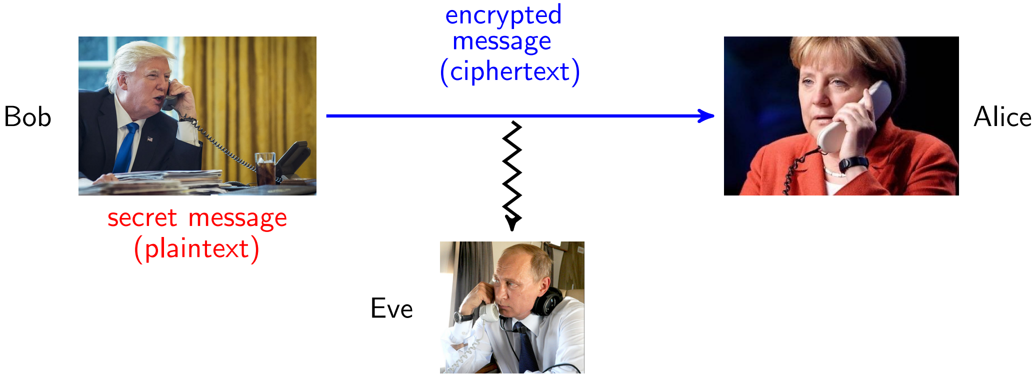
- **Research areas:** Coding for storage (distributed storage, memories), network coding, PQ cryptography, private information retrieval
- **Main funding:** DFG, ERC

Outline

Outline

1. Post-Quantum Cryptography
   - Public-Key Cryptography
   - Post-Quantum Cryptography
   - New Code-Based Cryptosystems
2. Further Current Research
   - Coding for Distributed Data Storage
   - Coding for DNA Storage
   - Private Information Retrieval
   - Network Coding

Basic Encryption Model



- Bob: wants to transmit a secret message to Alice
- Eve: wants to get this secret message (but should not)
- Alice: decrypts the ciphertext and obtains the secret message

Quantum Computers, Shor's Algorithm & Grover's Algorithm

[Image source: GEO 05/2018]

- Many qubits needed to correct errors in the computation process
- Size of current quantum computers still far from being useful!

- **Shor**'s quantum algorithm can find the prime factorization of any positive integer efficiently

$\Longrightarrow$ That would break classical public-key systems (RSA, ElGamal,...)
code-based & lattice-based crypto remain secure!

- **Grover**'s quantum algorithm: efficient root finding

$\Longrightarrow$ key size of symmetric systems has to be doubled

Quantum Computers, Shor's Algorithm & Grover's Algorithm

[Image source: GEO 05/2018]

- Many qubits needed to correct errors in the computation process
- Size of current quantum computers still far from being useful!

- **Shor**'s quantum algorithm can find the prime factorization of any positive integer efficiently

$\Longrightarrow$ That would break classical public-key systems (RSA, ElGamal,...)
code-based & lattice-based crypto remain secure!

- **Grover**'s quantum algorithm: efficient root finding

$\Longrightarrow$ key size of symmetric systems has to be doubled

Quantum Computers, Shor's Algorithm & Grover's Algorithm

[Image source: GEO 05/2018]

- Many qubits needed to correct errors in the computation process
- Size of current quantum computers still far from being useful!

- **Shor**'s quantum algorithm can find the prime factorization of any positive integer efficiently

$\Longrightarrow$ That would break classical public-key systems (RSA, ElGamal,...)
code-based & lattice-based crypto remain secure!

- **Grover**'s quantum algorithm: efficient root finding

$\Longrightarrow$ key size of symmetric systems has to be doubled

A New Rank-Metric Cryptosystem of Small Key Size[5]

Our Results

- Code-based PKC based on the hardness of list decoding Gabidulin codes[1]
- New observation: public key of FL system[2] is corrupted codeword of interleaved Gabidulin code

$\Longrightarrow$ Prevent attacks: use keys where decoder fails

- Security analysis: security level not decreased
- Small resulting key size & decryption guarantee

Ongoing work:

- Hardware implementation & side-channel attacks[3]
- Investigation of weak public keys[4]

[1] Wachter-Zeh, "Bounds on list decoding rank-metric codes," T-IT 2013
[2] Faure, Loidreau, "A new public-key cryptosystem based on the problem of reconstr. *p*-poly.," DCC 2006
[3] Ongoing work together with TUM-SEC
[4] Jerkovits, Bartz, "Weak keys in the Faure–Loidreau cryptosystem," 2019
[5] Wachter-Zeh, Puchinger, Renner, "Repairing the Faure–Loidreau public-key cryptosystem," ISIT 2018

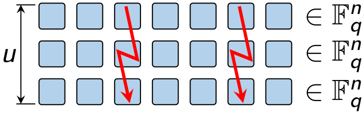
Comparison to Goppa codes[6], Loidreau[7], QC-MDPC[8], LRPC[9]

Method	q	u	k	n	m	w	Security level	Rate	Key size
McEliece	2		1436	1876	11		80.04	0.77	78.98 KB
Loidreau	2		32	50	50		80.93	0.64	3.60 KB
New System	2	3	31	61	61	16	90.00	0.46	1.86 KB
QC-MDPC	2		4801	9602			80.00	0.50	0.60 KB
LRPC	2		37	74	41		80.00	0.50	0.19 KB
McEliece	2		2482	3262	12		128.02	0.76	242.00 KB
Loidreau	2		40	64	96		139.75	0.63	11.52 KB
New System	2	3	31	62	62	17	131.99	0.45	1.92 KB
QC-MDPC	2		9857	19714			128.00	0.50	1.23 KB
LRPC	2		47	94	47		128.00	0.50	0.30 KB
McEliece	2		5318	7008	13		257.47	0.76	1123.43 KB
Loidreau	2		80	120	128		261.00	0.67	51.20 KB
New System	2	4	48	83	83	21	256.99	0.53	4.31 KB
QC-MDPC	2		32771	65542			256.00	0.50	4.10 KB

[6] Barbier, Barreto, "Key reduction of McEliece's cryptosystem using list decoding," ISIT 2011
[7] Loidreau, "A new rank metric code based encryption scheme," PQCrypto 2017
[8] Misoczki et al., "MDPC-McEliece: New McEliece variants from MDPC codes," ISIT 2013
[9] Gaborit et al., "Low rank parity check codes and their application to cryptography," WCC 2013

McEliece Public-Key System Based on Interleaved Goppa Codes[10]

- Consider multiple ciphertexts:
  $\mathbf{c}_i = \mathbf{m}_i \cdot \mathbf{G}_{\text{pub}} + \mathbf{e}_i,\ i = 1, \dots, u$
- Choose errors as burst
- Adapt classical attacks



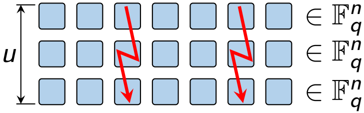
$\Longrightarrow$ **Decoding radius**: $t_{\text{pub}} = \frac{u}{u+1} \cdot \frac{q}{q-1} \cdot r$ (w.h.p.) instead of $t = \frac{q}{q-1} \cdot \frac{r}{2}$
(for wild interleaved Goppa codes with $d \geq \frac{q}{q-1} \cdot r + 1$)

Security level [bits]	q	m	Method	r	n	k	t (u, t_{pub}, d_E)	R	Key size [Bytes]
128	3	8	unique decoding	84	3004	2332	63	0.78	310 476
			interleaved		2586	1914	$(7, 110, 70)$	0.74	**254 824**
	5	5	unique decoding	100	2342	1842	62	0.79	267 312
			interleaved		1593	1093	$(8, 111, 83)$	0.69	**158 617**
256	5	5	unique decoding	204	4617	3597	128	0.78	1 064 877
			interleaved		3533	2513	$(7, 223, 156)$	0.71	**743 964**

[10] Holzbaur, Liu, Puchinger, Wachter-Zeh, "On decoding and applications of interleaved Goppa codes," 2019

McEliece Public-Key System Based on Interleaved Goppa Codes[10]

- Consider multiple ciphertexts: $\mathbf{c}_i = \mathbf{m}_i \cdot \mathbf{G}_{\text{pub}} + \mathbf{e}_i, \ i = 1, \dots, u$
- Choose errors as burst
- Adapt classical attacks



$\Longrightarrow$ **Decoding radius**: $t_{\text{pub}} = \frac{u}{u+1} \cdot \frac{q}{q-1} \cdot r$ (w.h.p.) instead of $t = \frac{q}{q-1} \cdot \frac{r}{2}$
(for wild interleaved Goppa codes with $d \geq \frac{q}{q-1} \cdot r + 1$)

Security level [bits]	q	m	Method	r	n	k	t (u, t_{pub}, d_E)	R	Key size [Bytes]
128	3	8	unique decoding	84	3004	2332	63	0.78	310 476
			interleaved		2586	1914	$(7, 110, 70)$	0.74	**254 824**
	5	5	unique decoding	100	2342	1842	62	0.79	267 312
			interleaved		1593	1093	$(8, 111, 83)$	0.69	**158 617**
256	5	5	unique decoding	204	4617	3597	128	0.78	1 064 877
			interleaved		3533	2513	$(7, 223, 156)$	0.71	**743 964**

[10] Holzbaur, Liu, Puchinger, Wachter-Zeh, "On decoding and applications of interleaved Goppa codes," 2019

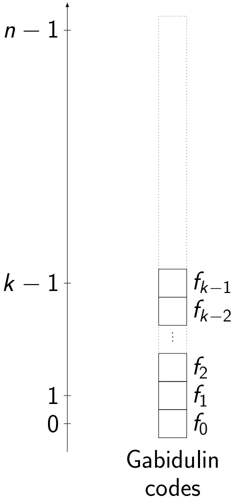
McEliece Public-Key System Based on Twisted Gabidulin Codes[14]

GPT cryptosystem[11]:

- McEliece based on Gabidulin codes
- Broken by Overbeck's attack using: $\dim(\mathcal{G} + \mathcal{G}^q + \ldots \mathcal{G}^{q^i}) = \min\{k+i, n\}$
- Loidreau[12] unbroken, but larger key size

$\Longrightarrow$ **Twisted** Gabidulin codes[13] in McEliece:

- Key sizes approximately half of Loidreau's
- No efficient decoder known (yet)
- Distinguisher: $\dim(\mathcal{G}_t + \mathcal{G}_t^q + \ldots \mathcal{G}_t^{q^i}) = \min\{k-1+(i+1)(\ell+1), n\}$ but no explicit attack



[11] Gabidulin, Paramonovo, Tretjakov, "Ideals over a non-commutative ring & application in cryptology," 1991
[12] Loidreau, "A new rank metric code based encryption scheme," PQCrypto 2017
[13] Sheekey, "A new family of linear maximum rank distance codes," AMC 2016
[14] Puchinger, Renner, Wachter-Zeh, "Twisted Gabidulin codes in the GPT cryptosystem," ACCT 2018

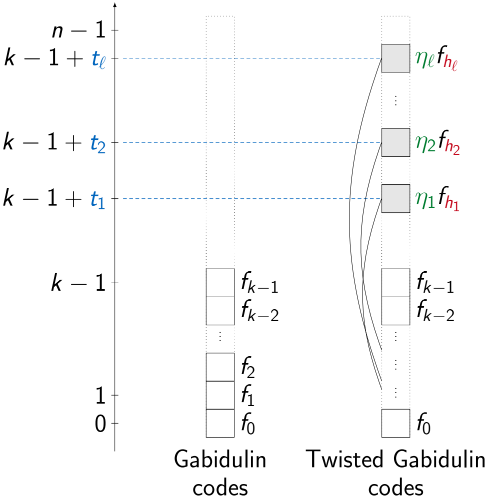
McEliece Public-Key System Based on Twisted Gabidulin Codes[14]

GPT cryptosystem[11]:

- McEliece based on Gabidulin codes
- Broken by Overbeck's attack using: $\dim(\mathcal{G} + \mathcal{G}^q + \ldots \mathcal{G}^{q^i}) = \min\{k+i, n\}$
- Loidreau[12] unbroken, but larger key size

$\Longrightarrow$ **Twisted** Gabidulin codes[13] in McEliece:

- Key sizes approximately half of Loidreau's
- No efficient decoder known (yet)
- Distinguisher: $\dim(\mathcal{G}_t + \mathcal{G}_t^q + \ldots \mathcal{G}_t^{q^i}) = \min\{k-1+(i+1)(\ell+1), n\}$ but no explicit attack

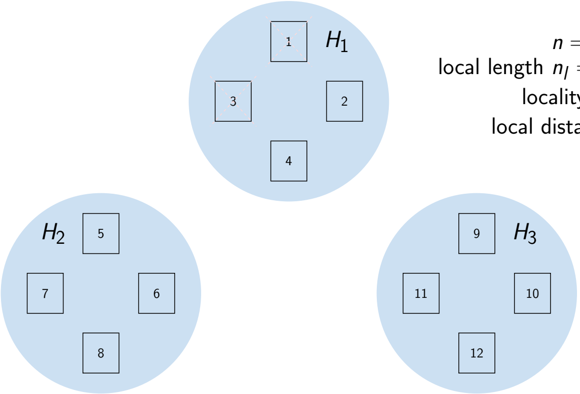


[11] Gabidulin, Paramonovo, Tretjakov, "Ideals over a non-commutative ring & application in cryptology," 1991
[12] Loidreau, "A new rank metric code based encryption scheme," PQCrypto 2017
[13] Sheekey, "A new family of linear maximum rank distance codes," AMC 2016
[14] Puchinger, Renner, Wachter-Zeh, "Twisted Gabidulin codes in the GPT cryptosystem," ACCT 2018

Outline

Coding for Distributed Data Storage: Locally Repairable Codes

Locality: number of servers (symbols) needed to repair a failed server (erased symbol)



$n = 12$

local length $n_l = r + \delta - 1 = 4$

locality $r = 2$

local distance $\rho = 3$

Our Research: (list) decoding algorithms[15,16], investigation of good codes[17]

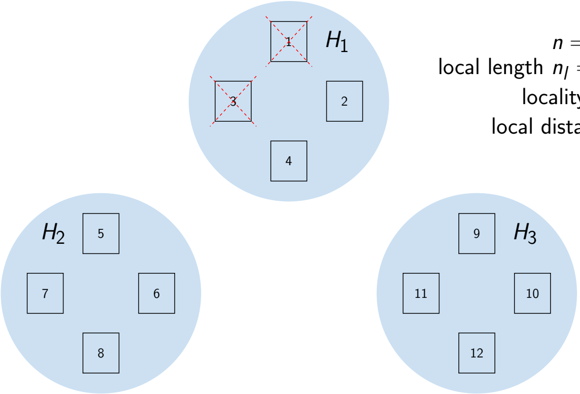
[15] Holzbaur, Wachter-Zeh, "List decoding of locally repairable codes," ISIT 2018

[16] Holzbaur, Puchinger, Wachter-Zeh, "Error Decoding of Locally Repairable and PMDS Codes," ITW 2019

[17] Holzbaur, Freij-Hollanti, Wachter-Zeh, "Cyclic Codes with Locality and Availability," 2019

Coding for Distributed Data Storage: Locally Repairable Codes

Locality: number of servers (symbols) needed to repair a failed server (erased symbol)



$n = 12$

local length $n_l = r + \delta - 1 = 4$

locality $r = 2$

local distance $\rho = 3$

Our Research: (list) decoding algorithms[15,16], investigation of good codes[17]

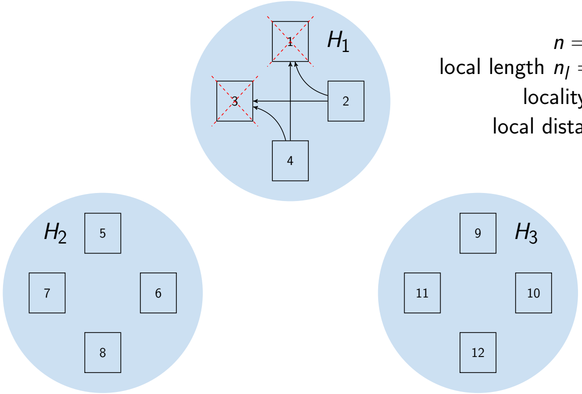
[15] Holzbaur, Wachter-Zeh, "List decoding of locally repairable codes," ISIT 2018

[16] Holzbaur, Puchinger, Wachter-Zeh, "Error Decoding of Locally Repairable and PMDS Codes," ITW 2019

[17] Holzbaur, Freij-Hollanti, Wachter-Zeh, "Cyclic Codes with Locality and Availability," 2019

Coding for Distributed Data Storage: Locally Repairable Codes

Locality: number of servers (symbols) needed to repair a failed server (erased symbol)



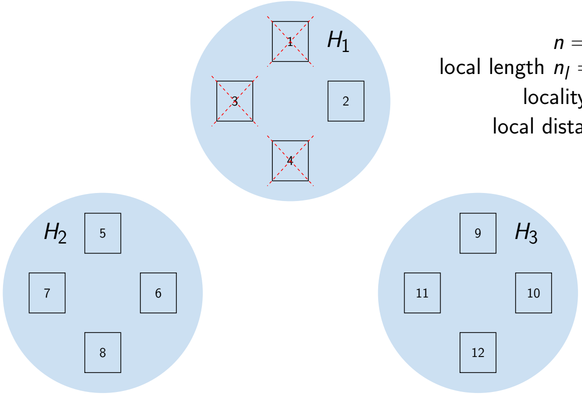
$n = 12$
local length $n_l = r + \delta - 1 = 4$
locality $r = 2$
local distance $\rho = 3$

Our Research: (list) decoding algorithms[15,16], investigation of good codes[17]

[15] Holzbaur, Wachter-Zeh, "List decoding of locally repairable codes," ISIT 2018
[16] Holzbaur, Puchinger, Wachter-Zeh, "Error Decoding of Locally Repairable and PMDS Codes," ITW 2019
[17] Holzbaur, Freij-Hollanti, Wachter-Zeh, "Cyclic Codes with Locality and Availability," 2019

Coding for Distributed Data Storage: Locally Repairable Codes

Locality: number of servers (symbols) needed to repair a failed server (erased symbol)



$n = 12$

local length $n_l = r + \delta - 1 = 4$

locality $r = 2$

local distance $\rho = 3$

Our Research: (list) decoding algorithms[15,16], investigation of good codes[17]

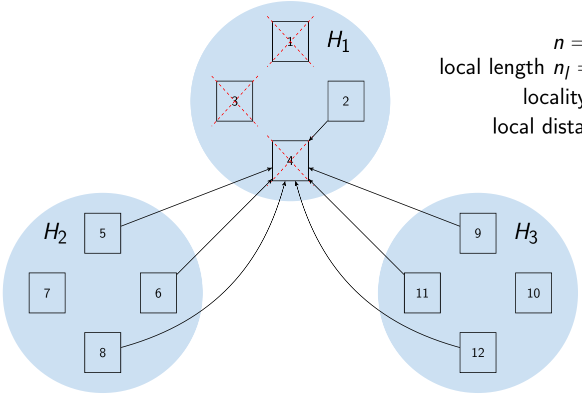
[15] Holzbaur, Wachter-Zeh, "List decoding of locally repairable codes," ISIT 2018

[16] Holzbaur, Puchinger, Wachter-Zeh, "Error Decoding of Locally Repairable and PMDS Codes," ITW 2019

[17] Holzbaur, Freij-Hollanti, Wachter-Zeh, "Cyclic Codes with Locality and Availability," 2019

Coding for Distributed Data Storage: Locally Repairable Codes

Locality: number of servers (symbols) needed to repair a failed server (erased symbol)



$n = 12$
local length $n_l = r + \delta - 1 = 4$
locality $r = 2$
local distance $\rho = 3$

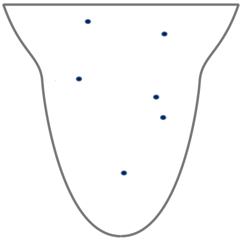
Our Research: (list) decoding algorithms[15,16], investigation of good codes[17]

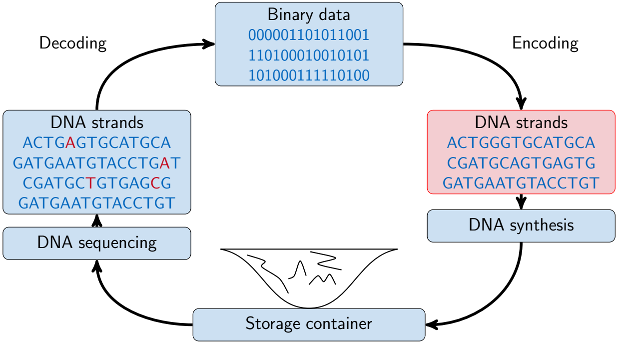
[15] Holzbaur, Wachter-Zeh, "List decoding of locally repairable codes," ISIT 2018

[16] Holzbaur, Puchinger, Wachter-Zeh, "Error Decoding of Locally Repairable and PMDS Codes," ITW 2019

[17] Holzbaur, Freij-Hollanti, Wachter-Zeh, "Cyclic Codes with Locality and Availability," 2019

Coding for DNA Storage: Channel Model





- Channel input: Sequences to be stored

Our Research: codes for insertions/deletions[18], duplications[19], ... and coding over sets[20] with insertions/deletions and substitutions

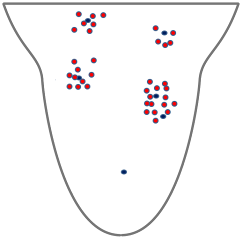
---

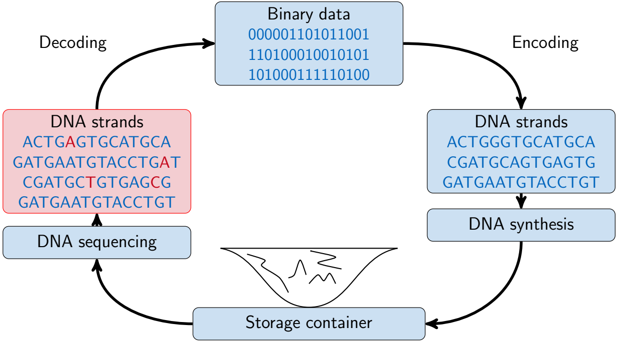
[18] Wachter-Zeh, "List decoding of insertions and deletions," T-IT 2018

[19] Lenz, Jünger, Wachter-Zeh, "Duplication-correcting codes," DCC 2018

[20] Lenz, Siegel, Wachter-Zeh, Yaakobi, "Coding over sets for DNA storage," ISIT 2018

Coding for DNA Storage: Channel Model



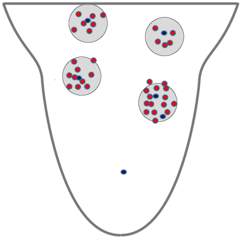


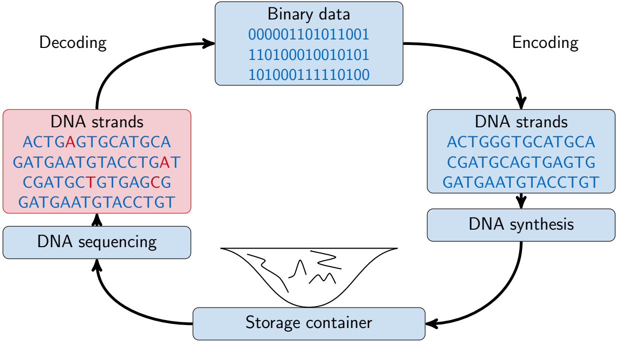
- Received sequences

Our Research: codes for insertions/deletions[18], duplications[19], ... and coding over sets[20] with insertions/deletions and substitutions

[18] Wachter-Zeh, "List decoding of insertions and deletions," T-IT 2018
[19] Lenz, Jünger, Wachter-Zeh, "Duplication-correcting codes," DCC 2018
[20] Lenz, Siegel, Wachter-Zeh, Yaakobi, "Coding over sets for DNA storage," ISIT 2018

Coding for DNA Storage: Channel Model





- Clusters around received sequences

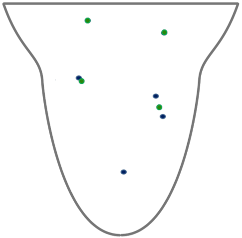
Our Research: codes for insertions/deletions[18], duplications[19], ... and coding over sets[20] with insertions/deletions and substitutions

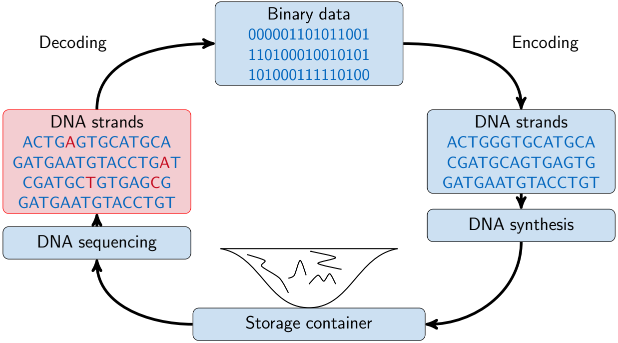
[18]Wachter-Zeh, "List decoding of insertions and deletions," T-IT 2018

[19]Lenz, Jünger, Wachter-Zeh, "Duplication-correcting codes," DCC 2018

[20]Lenz, Siegel, Wachter-Zeh, Yaakobi, "Coding over sets for DNA storage," ISIT 2018

Coding for DNA Storage: Channel Model





- Channel output: Reconstructed sequences

Our Research: codes for insertions/deletions[18], duplications[19], ... and coding over sets[20] with insertions/deletions and substitutions

[18] Wachter-Zeh, "List decoding of insertions and deletions," T-IT 2018
[19] Lenz, Jünger, Wachter-Zeh, "Duplication-correcting codes," DCC 2018
[20] Lenz, Siegel, Wachter-Zeh, Yaakobi, "Coding over sets for DNA storage," ISIT 2018

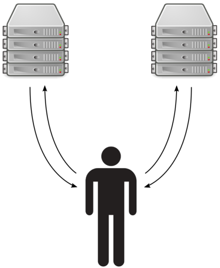
Private Information Retrieval

Goal

Retrieve a file from a public database or distributed storage system without revealing the index of the file.

Protocol:

1. Query: The user sends a query to each server
2. Response: The servers respond according to the received queries
3. Decoding: The user retrieves the desired file from the responses

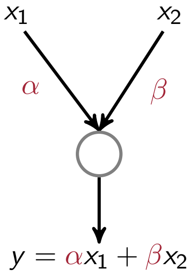


Our Research: Privacy for streaming[21], PIR over networks[22]

[21] Holzbaur, Freij-Hollanti, Wachter-Zeh, Hollanti, "Private streaming with convolutional codes," ITW 2018
[22] Tajeddine, Wachter-Zeh, Hollanti, "Private information retrieval over networks," For. & Security 2019

Network Coding: Alphabet Size

Task: find coefficients at the nodes s.t. each receiver obtains its requested packets.



- **Scalar network coding**:
  scalars over field of size q_s
  $\rightsquigarrow$ for each coefficient: q_s possibilities

- **Vector network coding** of dimension t:
  $t \times t$ matrices over field of size q
  $\rightsquigarrow$ for each coefficient: q^{t^2} possibilities

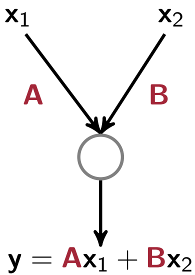
For equivalent field sizes ($q_s = q^t$), vector network coding offers more freedom!

- Gap: $q_s - q^t \geq q^{\left(1-\frac{1}{\ell}\right)t^2+o(t)}$ (for any $\ell \geq 2$)[23]
- Upper bound on the number of nodes in the middle layer of subnetworks of combination networks[24]

[23] Etzion, Wachter-Zeh, "Vector network coding outperforms scalar network coding," T-IT 2018
[24] Cai, Etzion, Schwartz, Wachter-Zeh, "Network coding solutions for the combination network," 2019

Network Coding: Alphabet Size

Task: find coefficients at the nodes s.t. each receiver obtains its requested packets.



- **Scalar network coding**:
  scalars over field of size q_s
  $\rightsquigarrow$ for each coefficient: q_s possibilities

- **Vector network coding** of dimension t:
  $t \times t$ matrices over field of size q
  $\rightsquigarrow$ for each coefficient: q^{t^2} possibilities

For equivalent field sizes ($q_s = q^t$), vector network coding offers more freedom!

- Gap: $q_s - q^t \geq q^{(1-\frac{1}{\ell})t^2+o(t)}$ (for any $\ell \geq 2$)[23]
- Upper bound on the number of nodes in the middle layer of subnetworks of combination networks[24]

[23] Etzion, Wachter-Zeh, "Vector network coding outperforms scalar network coding," T-IT 2018
[24] Cai, Etzion, Schwartz, Wachter-Zeh, "Network coding solutions for the combination network," 2019

Thank you...

...for your attention!

Questions?

Thanks for the financial support to:

DFG Deutsche Forschungsgemeinschaft

IAS TUM Institute for Advanced Study

erc European Research Council

Thanks for the collaboration to (alphabtical order):
Han Cai (BGU), Tuvi Etzion (Technion), Ragnar Freij-Hollanti (Aalto), Lukas Holzbaur (TUM), Camilla Hollanti (Aalto), Andreas Lenz (TUM), Lia Liu (TUM), Sven Puchinger (TUM), Julian Renner (TUM), Moshe Schwartz (BGU), Paul Siegel (UCSD), Razan Tajeddine (Aalto), Eitan Yaakobi (Technion)