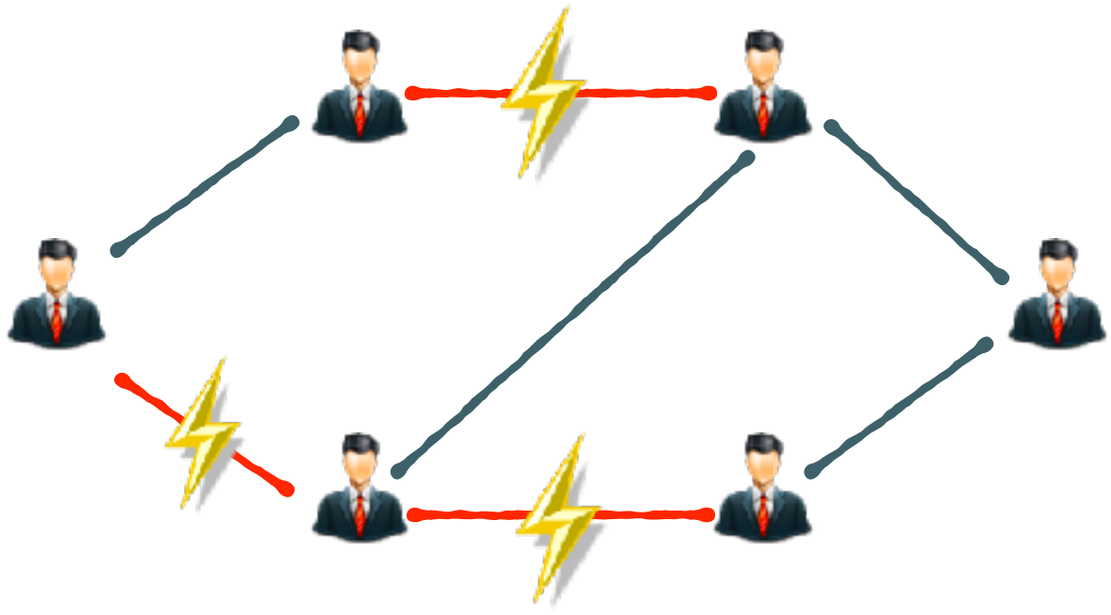
SECURE MULTIPARTY INTERACTIVE COMMUNICATION WITH UNKNOWN NOISE RATE

Abhinav Aggarwal
Varsha Dani, Thomas Hayes, Nico Döttling, Jared Saia

PROBLEM STATEMENT

Private channels



Adversary

Protocol π of unknown length

Noisefree channels ⟶ ?? ⟶ ***Protocol π'***

Bit flipping Adversary with unknown budget

Can π be compiled into π' such that :

- ***π' succeeds with high probability***
- ***π' has small bit overhead?***

OUR ASSUMPTIONS

- *π runs in an asynchronous model*
- *Private channels*
- *Unknown T and L*
- *Each user knows the number of users, n*
- *Instantaneous local computation*

OUR RESULT

VS

STATE OF THE ART

General Assumptions

- *Public channels* → *Private channels*
- *L, n and T known* → *L and T unknown*

UPPER BOUNDS

Jain, Abhishek, Yael Tauman Kalai, and Allison Bishop Lewko. **"Interactive coding for multiparty protocols."** *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*. ACM, 2015.

(1) ~~Fixed speaking order~~

Unknown amount of
(2) ~~Consider only Θ(1/n) fraction~~ of adversarial errors

(3) ~~Star network~~

Private
(4) ~~Public~~ channels

Their result : Constant blowup in communication rate

Our result : Within log factors of the optimal

LOWER BOUNDS

Braverman, Mark, et al. "**Constant-rate coding for multiparty interactive communication is impossible.**" *Electronic Colloquium on Computational Complexity (ECCC), TR15-197*. 2015.

Adversarial

~~*Stochastic noise, with prob $\varepsilon < 1/2$ of*~~ *bit flipping*

Their result : Blowup by a factor of (log n/log log n) in the number of bits sent

Our result : Within log factors of this lower bound

OUR ALGORITHM

OUR MAIN RESULT

Expected number of bits exchanged in π' :

$$\mathcal{O}\left((L+T)\log n\,(L+T)\right)$$

Within log factors of the optimal!!!

Probability of successful communication :

$$1-\mathcal{O}\left(1/n\right)$$

Success guaranteed with high probability!!!

MESSAGE EXCHANGE PROTOCOL (MEP)

Alice *Bob*

Hey! I have a message for you. What's your **key**?

My key is **k.**

The message is **m.** I know your key is **k**. Now you know it's me!

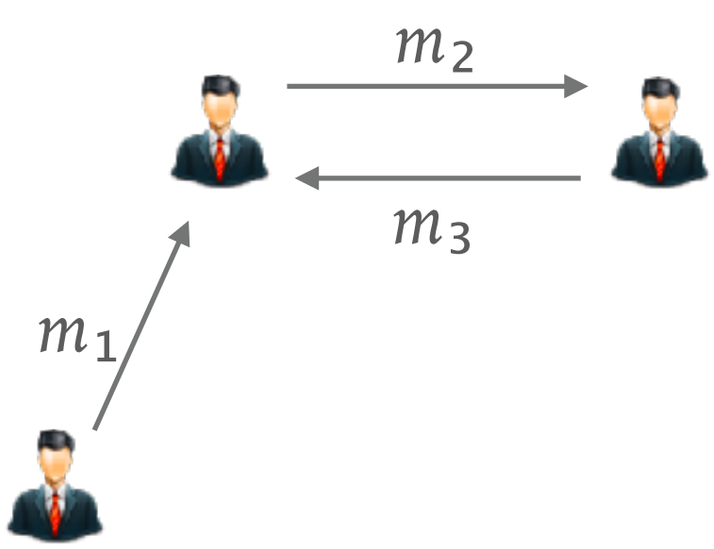
Terminate

Terminate

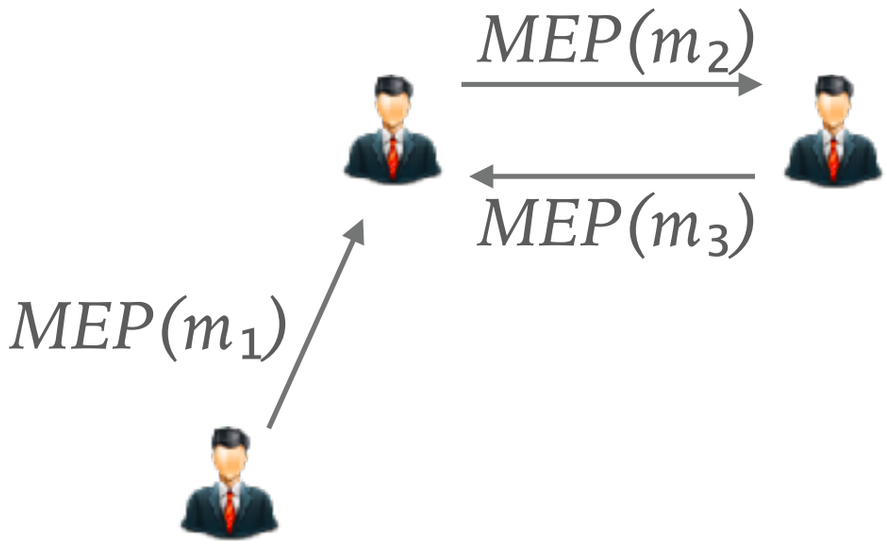
MESSAGE EXCHANGE PROTOCOL (MEP)

- *Neighbors talk via dedicated channels*
- *One message exchange protocol (MEP) per message in π*
- *Terminate upon hearing "silence"*

π :



π' :



SILENCE

A b-bit string on the channel is interpreted as silence if it contains fewer than (b/3) bit alternations.

0100000000100

(1/3)-ERROR CORRECTING CODES

- *Corrects at most a third of total bits*
- *Multiplicative blowup of at most 2*
- *Adversary pays Θ(message length) to corrupt*

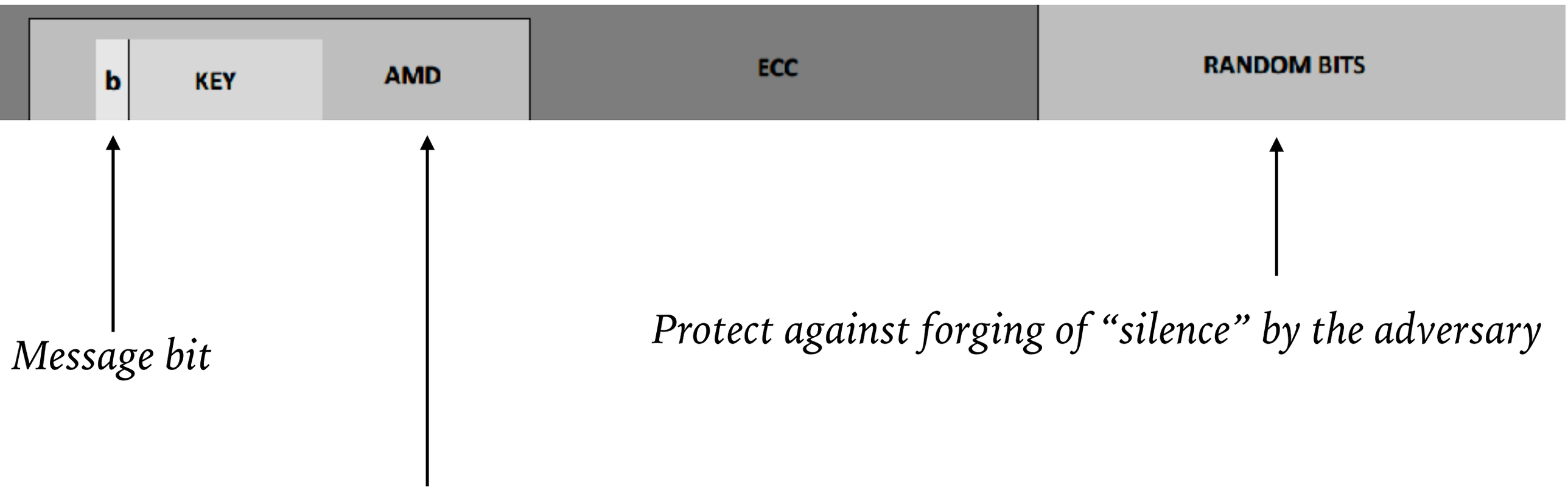
ALGEBRAIC MANIPULATION DETECTION CODES (AMD)

- *Enable detection of bit corruption*
- *Work only for private channels*
- *Encode a message m into a value m'*
- *Any bit flipping of m is detected with probability* $\geq 1-\delta$
- *For* $\delta \in (0,1/2)$, *produces codewords of length* $|m'| \leq |m| + O(\log(1/\delta))$

MESSAGE FORMAT

For authentication

For making corruption expensive



Message bit

For detection of bit corruption

Protect against forging of "silence" by the adversary

MESSAGE EXCHANGE PROTOCOL *(Bit corruption)*

Alice *Bob*

Hey! I have a message for you.
What's your **key**?

Message corrupted.

Hey! I have a message for you.
What's your **key**?

My key is **k**.

The message is **m.** I know your key is **k.** Now you know it's me!

Message corrupted.

Authentication failure! Please **resend**.

The message is **m.** I know your key is **k.** Now you know it's me!

Terminate

Terminate

ALICE' TERMINATION

Alice *Bob*

Hey! I have a message for you.
What's your **key**?

My key is **k**.

The message is **m.** I know your key is **k.** Now you know it's me!

Terminate

Delayed termination

RESEND

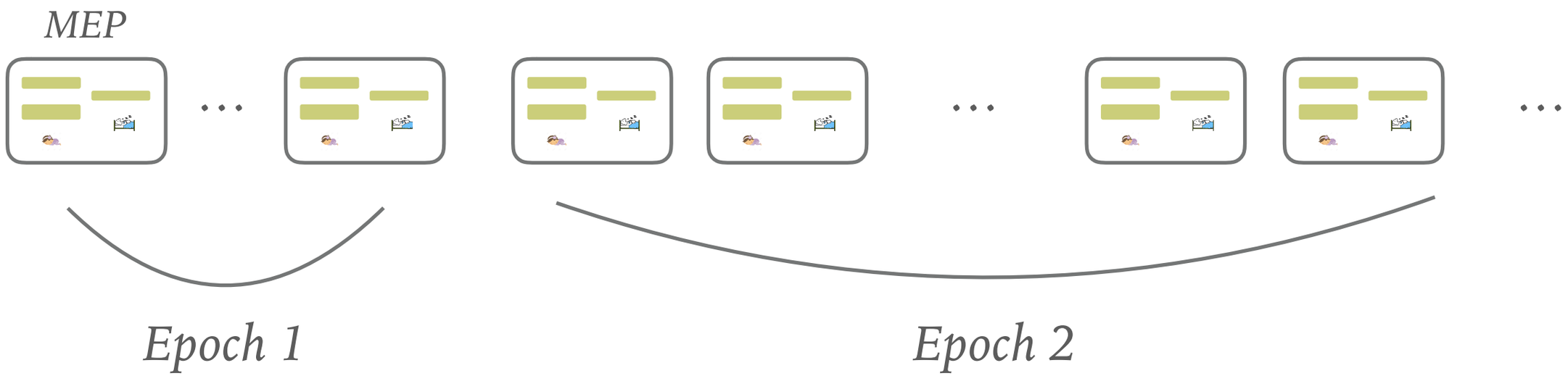
RESEND

The message is **m.** I know your key is **k.** Now you know it's me!

The message is **m.** I know your key is **k.** Now you know it's me!

EPOCHS

- *Our algorithm proceeds in epochs.*
- *Once an epoch is over, all unfinished MEPs rollback and new epoch is started.*
- *Security increased with epochs.*



EPOCHS

- *Max number of codewords in epoch* $j = O(2^j n^{10})$
- *AMD security in epoch* $j = O(4^{-j} n^{-14})$

Geometric increase

- *Key length in epoch* $j = O(j + \log n)$
- *Length of the codeword in epoch* $j = O(j + \log n)$

Linear increase

Total number of epochs before termination

$$\mathcal{O}(L + T)$$

FAILURE EVENTS

(Failure of AMD codes)

Alice *Bob*

Hey! I have a message for you.
What's your **key**?

My key is **k**.

The message is **m.** I know your key is **k.** Now you know it's me!

AMD codes compromised to produce a valid (m',k) pair

Terminate

Terminate

Terminates with knowledge of bits not in P

Prob. of any occurrence

$$\mathcal{O}\left(\frac{1}{n^2}\right)$$

FAILURE EVENTS

(Conversion to silence)

Alice *Bob*

Hey! I have a message for you.
What's your **key**?

My key is **k**.

The message is **m.** I know your key is **k.** Now you know it's me!

Message corrupted.

Authentication failure! Please **resend**.

Converted to silence.

Terminate

Prob. of any occurrence

$$\mathcal{O}\left(\frac{1}{n}\right)$$

Repeat

FORGING AN ENTIRE MEP

Alice | ***Adversary*** | *Bob*

Hey! I have a message for you.
What's your **key**?

My key is **k**.

Adversary guesses the key correctly

I know your key is **k.** Now you know it's me, Alice.

Send a million dollars to account XXX-XXX-XXX.

Terminate

Prob. of any occurrence

$$\mathcal{O}\left(\frac{1}{n}\right)$$

OUR MAIN RESULT

Expected number of bits exchanged in P' :

$$\mathcal{O}\left((L+T)\log n\,(L+T)\right)$$

Within log factors of the optimal!!!

Probability of successful communication :

$$1-\mathcal{O}\left(1/n\right)$$

Success guaranteed with high probability!!!

FUTURE WORK

- *How much privacy is necessary?*

- *Tighten analysis.*

- *Optimize for large scale practical applications like map reduce.*

OUR TEAM

Varsha Dani

Jared Saia

Tom Hayes

Nico Döttling

QUESTIONS??