MIT 6.875 & Berkeley CS276

Application: Privacy-preserving machine learning

Lecture 26

In this lecture

- Recording…
- Application of our various tools in this class
  - We will solve a real problem through MPC, ZK proofs, homomorphic encryption, commitments
    - In particular, the solution has to be practical
  - An example of how you might go about using the knowledge in this class for a real problem
- Leave time for ending remarks

Real problem:
The need for collaborative computation

Organizations often

wish to run a cross-organization joint computation

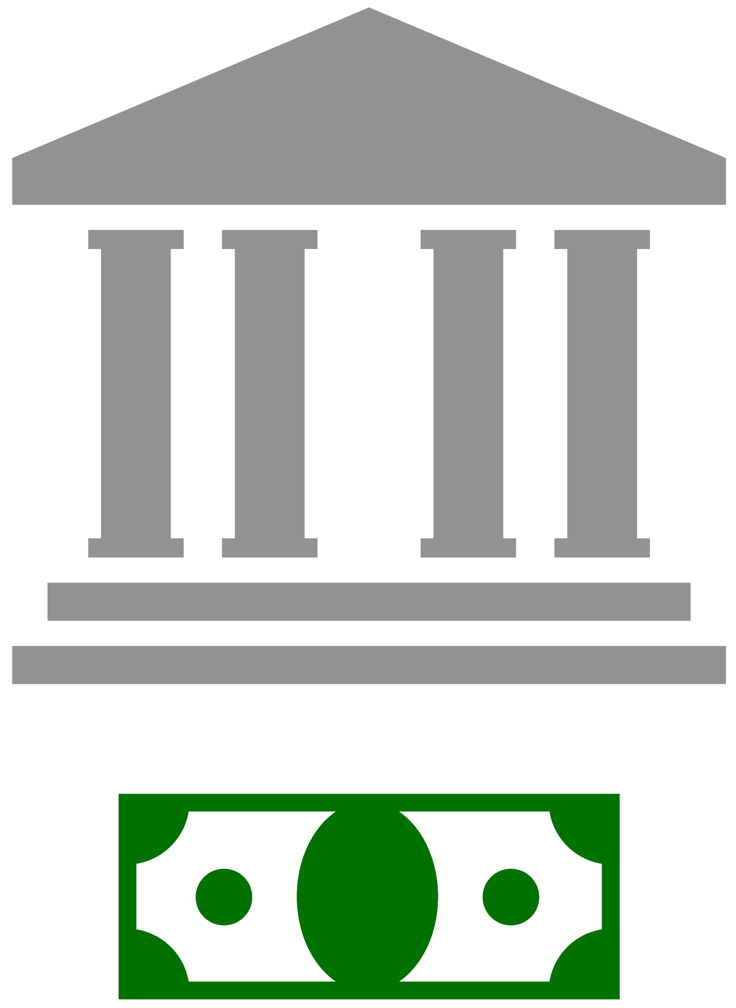
but

have sensitive data they cannot share

Anti-money laundering

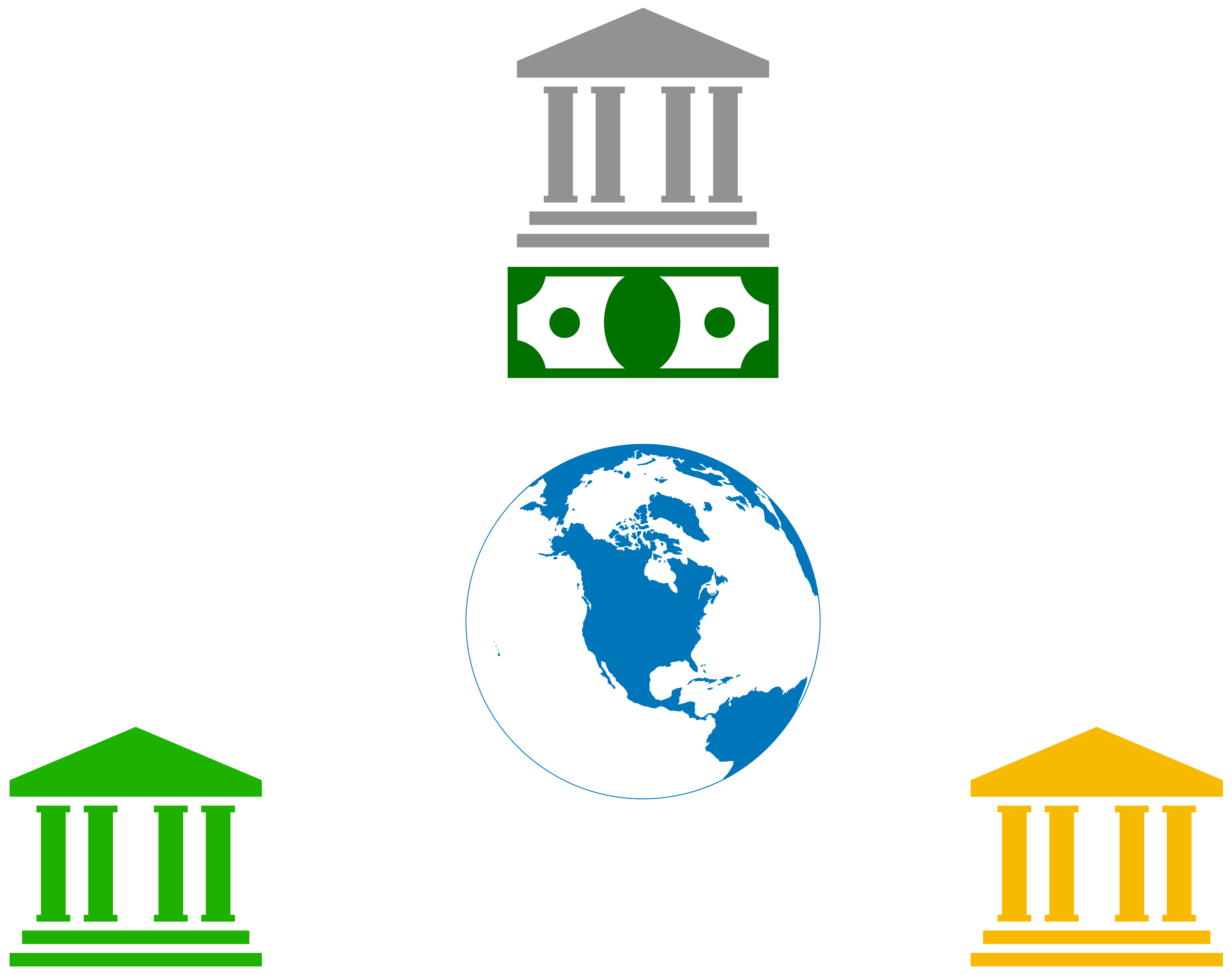
Anti-money laundering

- Banks want to detect money laundering



Anti-money laundering

- Banks want to detect money laundering
- Criminals conceal illegal activities across many banks



Anti-money laundering

- Banks want to detect money laundering
- Criminals conceal illegal activities across many banks

Anti-money laundering

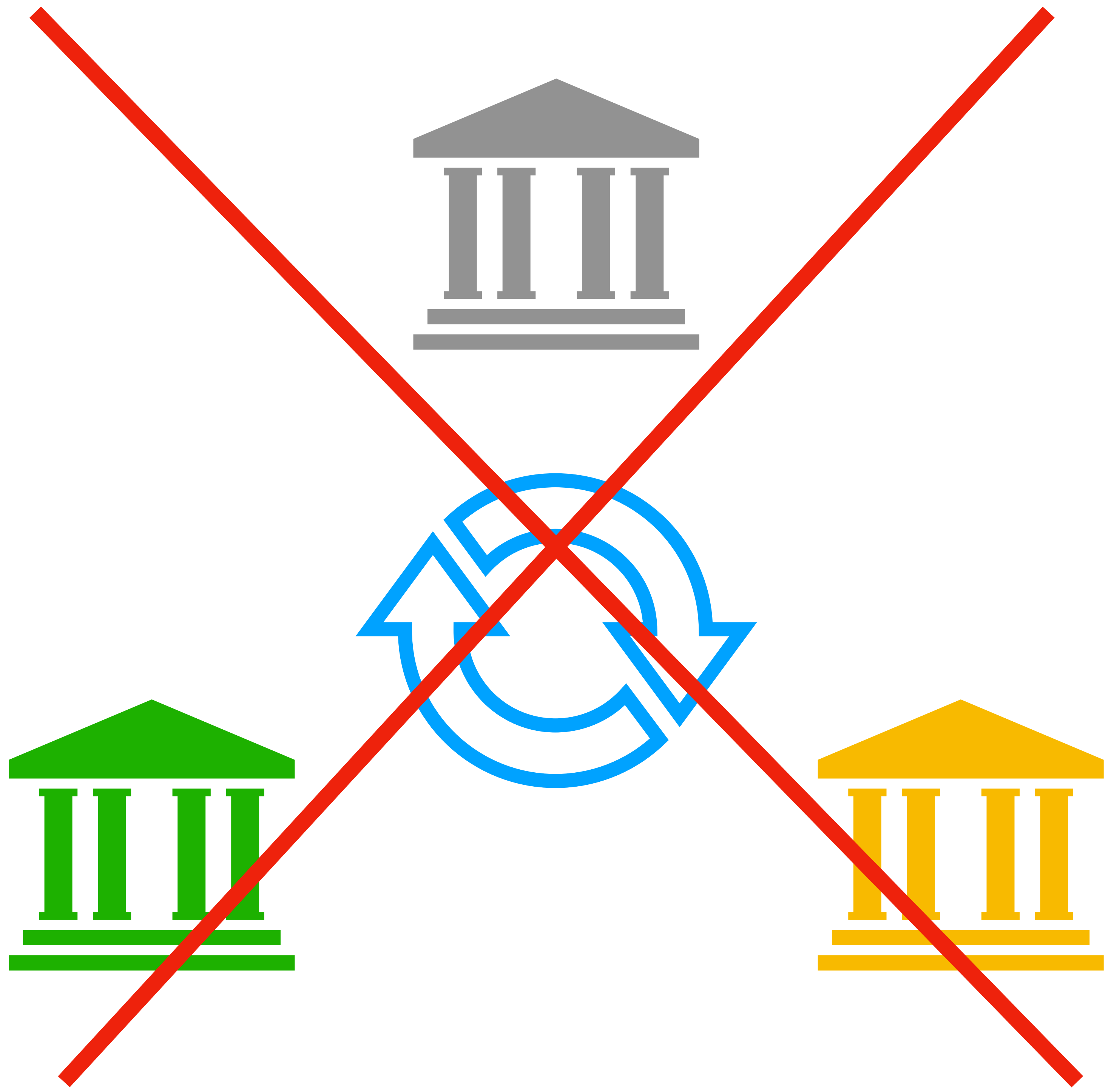
To detect money laundering, one needs to **learn** from data from multiple banks

Anti-money laundering

To detect money laundering, one needs to **learn** from data from multiple banks

but

cannot share data due to competition



Anti-money laundering

- An accurate result needs computation on data from multiple banks
- Cannot share data due to competition

> "So in the future, ***collaboration will be vital***: across the financial-services industry, government, and law enforcement. The ability to put together our data sets and collaborate on typologies of attack – and the use of both advanced-encryption methods and analytics methods to mine the data – ***will enhance yields by orders of magnitude***."
>
> – Chief Risk Officer of Scotiabank

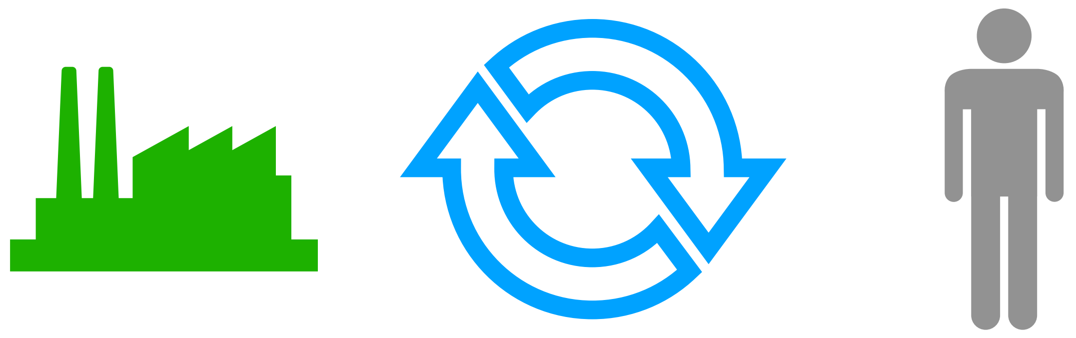
What tools cryptographic protocol solves this problem?

MPC!

Many other use cases:

Fraud & Human trafficking detection

Flu prediction



Nuclear facility auditing

Cloud-based video analysis for home intrusion

...

Helen

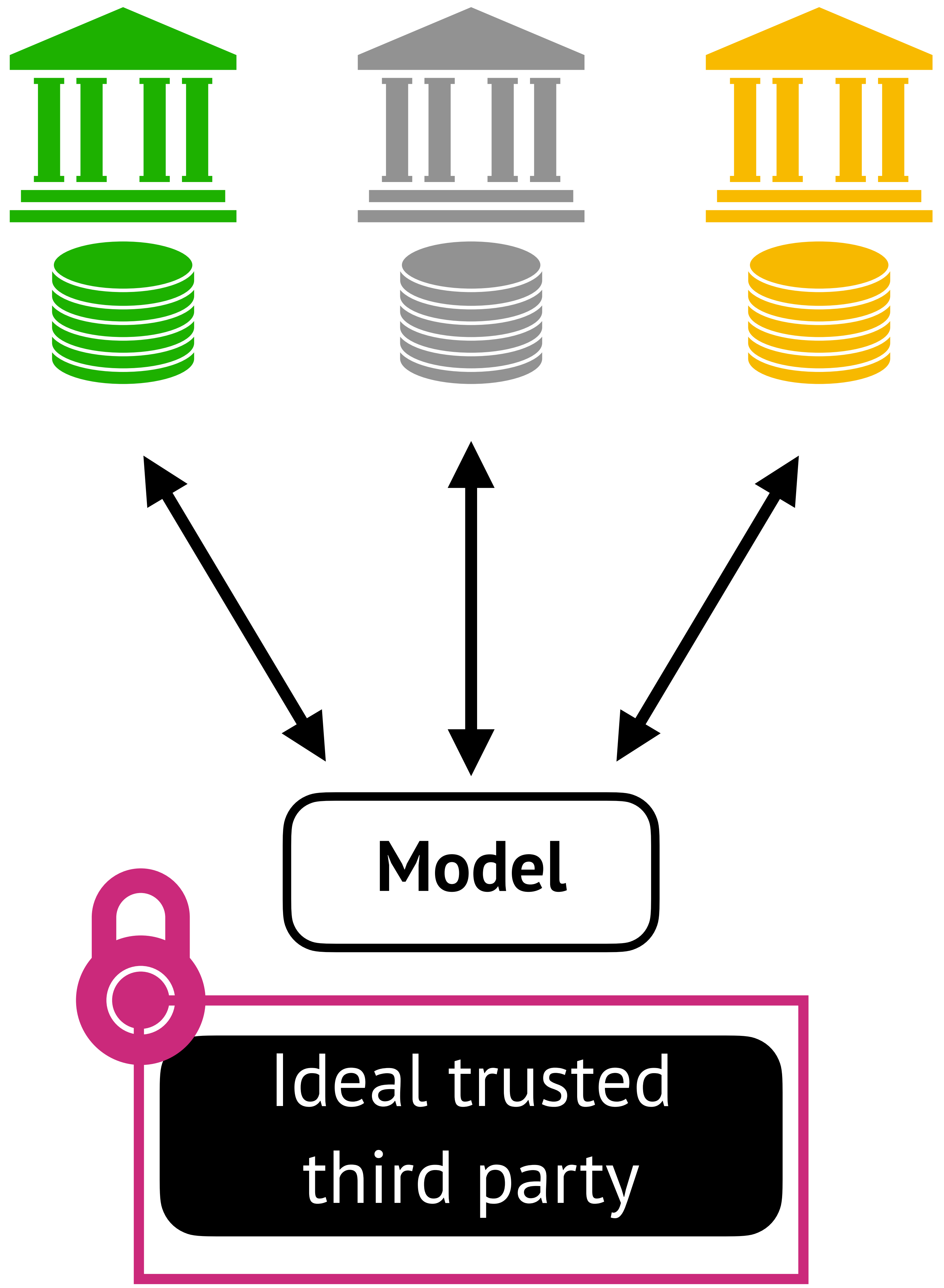
[ZPGS, IEEESP19]

Provides maliciously-secure MPC for collaboratively training regularized linear models

p-1 out of p parties are malicious:
each party need only trust itself

Scope of Helen

- Parties choose their inputs
  - Protection for poisoning attacks is complementary [JOBLNL18][CLLLS17]
- Final result is released to everyone
  - Privacy mechanisms for protecting against data leakage from the model, such as differential privacy, are complementary [SS18][CLKES18][INSTTW19]



Threat model

- Secure computation executed among the parties
- Attacker can compromise p - 1 out of p parties
- Protection against malicious attacker, where the attacker can deviate from the protocol
- Allows
  - parties to input data of their choice
  - parties to learn the final model

Challenge: generic MPC is expensive

For LASSO (a type of regularized linear model), SGD (stochastic gradient descent) for 4 parties, 100K samples per party, 90 features using SPDZ

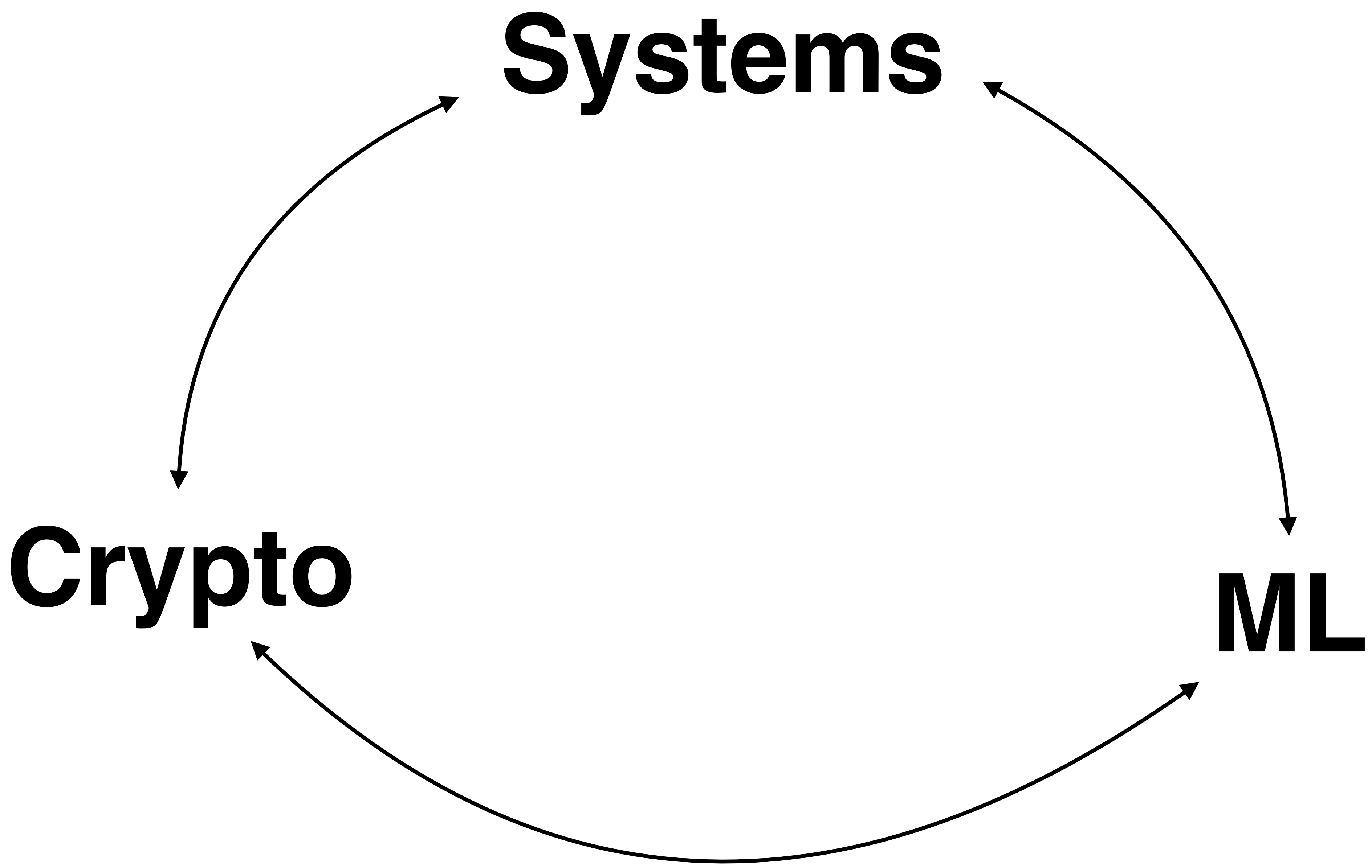
estimated **3 months** to train a model

In practice, we design MPC from scratch tailored to **a computation & a setting** for efficiency

Prior work

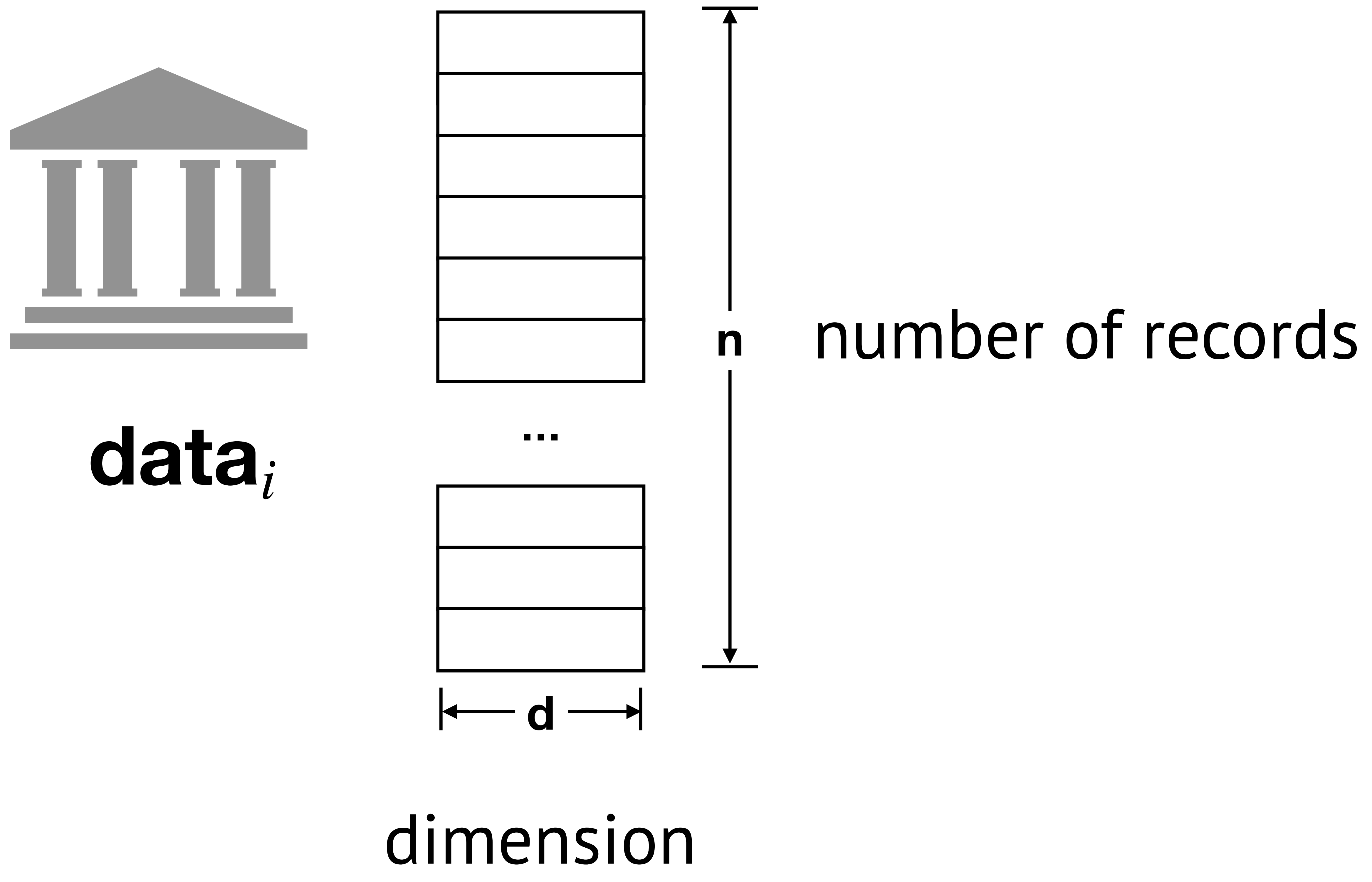
Work	Functionality	p-party? (p>2)	maliciously secure?
NWIJBT13	Ridge regression	✗	✗
HFN11	Linear regression	✓	✗
GSBRDZE16	Linear regression	✗	✗
CDNN15	Linear regression	✗	✗
GJJPY17	Ridge regression	✗	✗
AGSSTP18	Quadratic optimization	✗	✗
MZ17	Linear, logistic, deep learning	✗	✗

Helen: a synergy



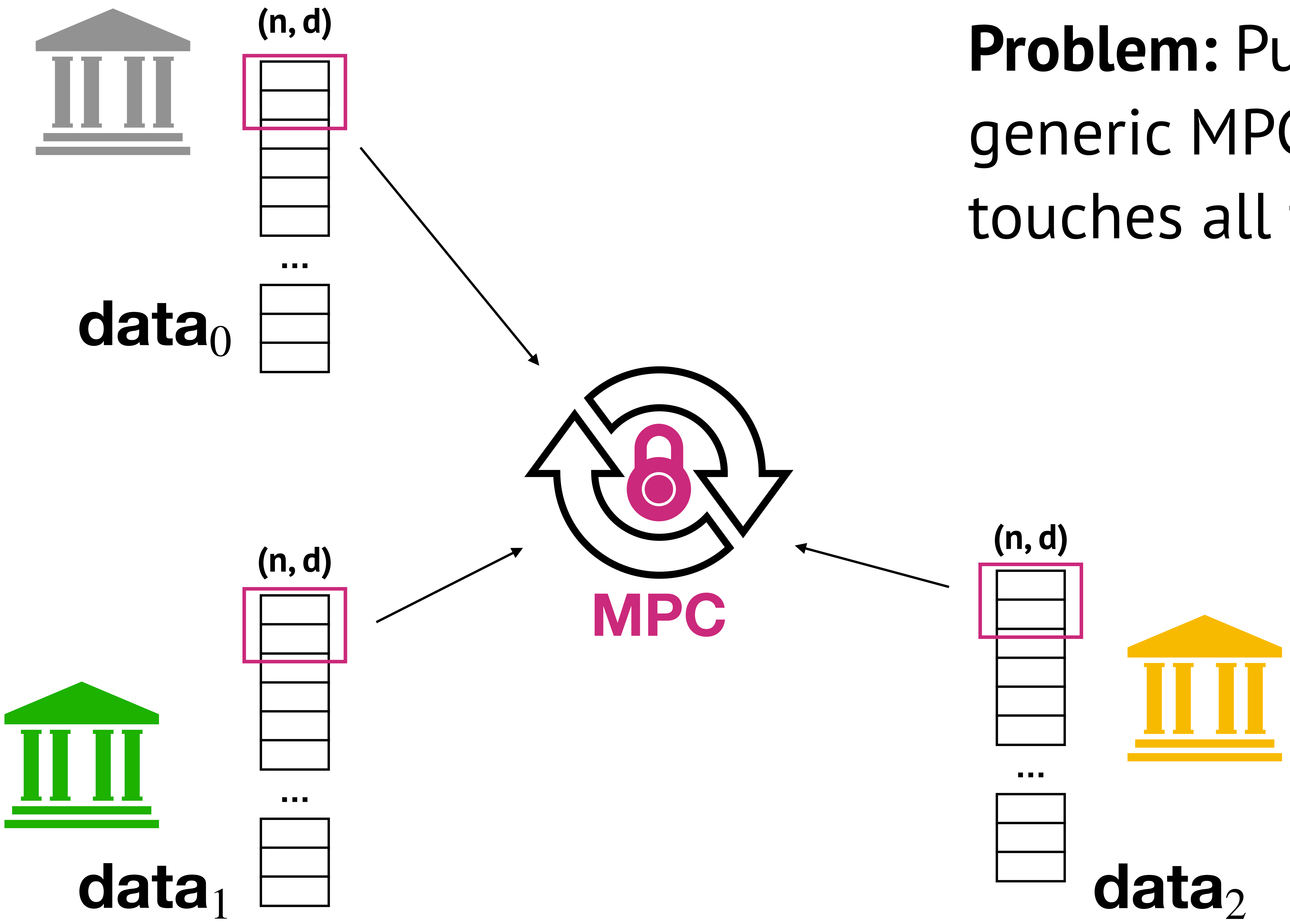
Generic MPC: 3 months ⟶ **Helen: < 3 hours**

Training input



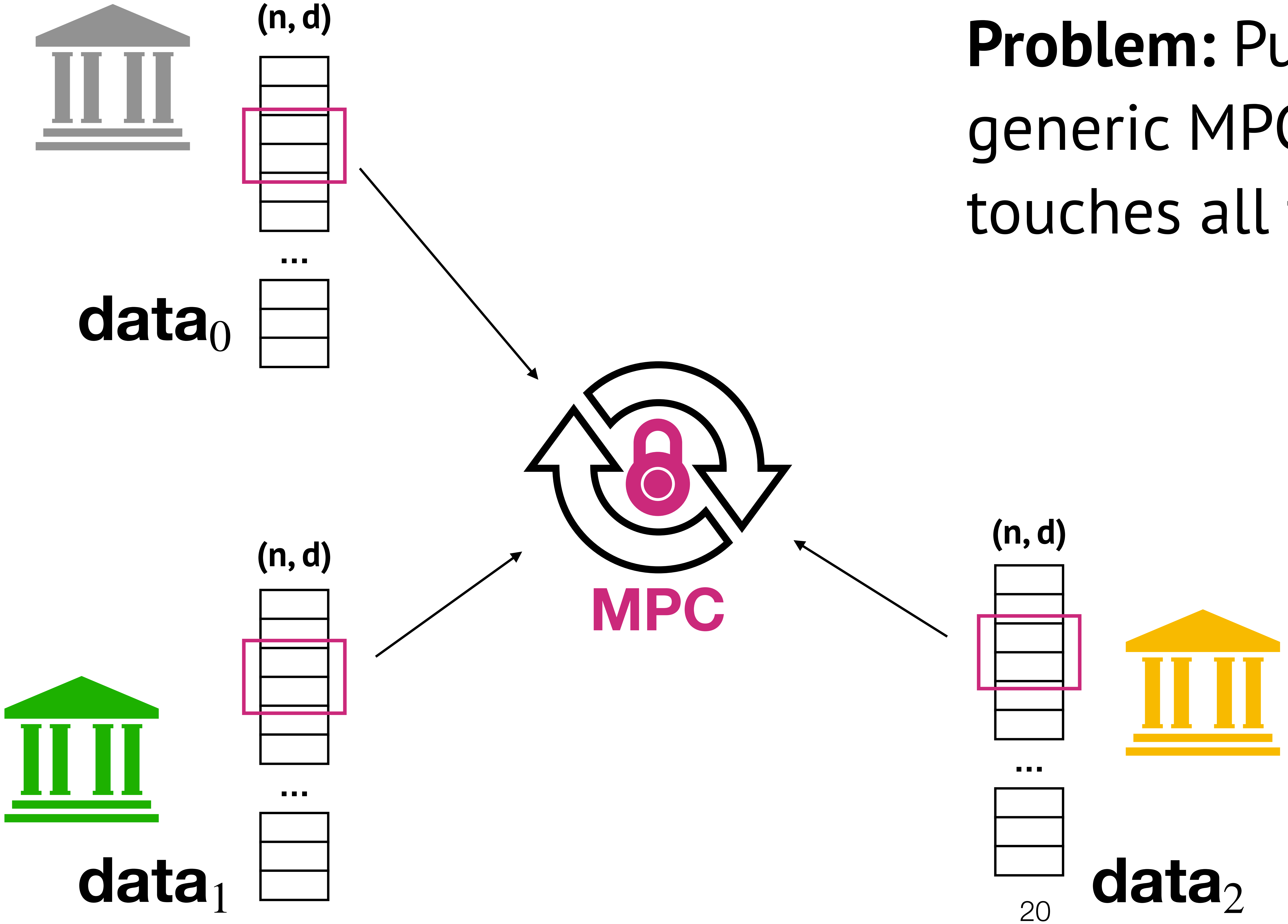
Usage scenario: n >> d

SGD is not scalable in MPC



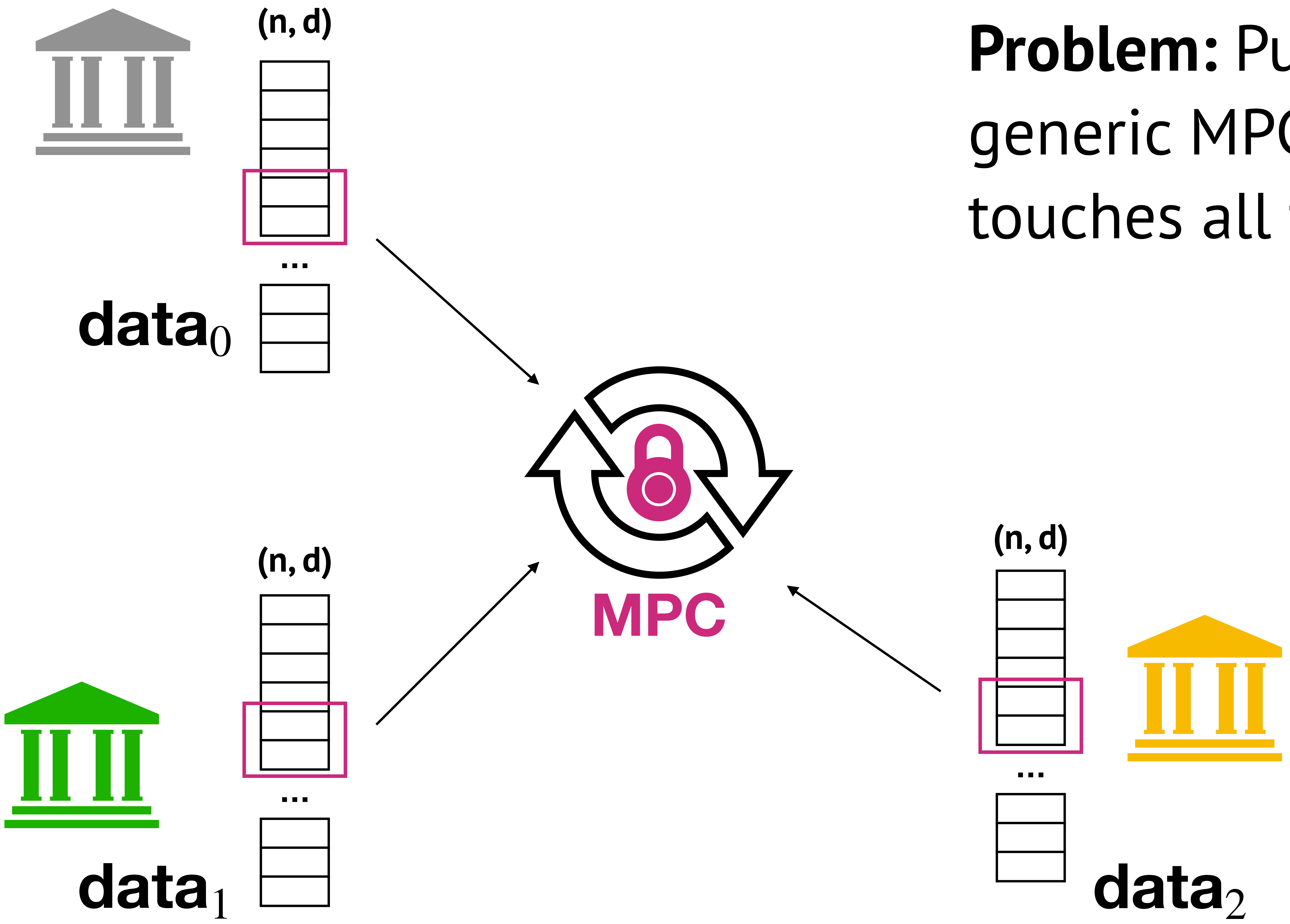
Problem: Putting SGD directly into a generic MPC is expensive because MPC touches all the data

SGD is not scalable in MPC



Problem: Putting SGD directly into a generic MPC is expensive because MPC touches all the data

SGD is not scalable in MPC



Problem: Putting SGD directly into a generic MPC is expensive because MPC touches all the data

Insight

Specialized protocol enables cryptographic computation to

scale independently of the number of records

while

maintaining the **same accuracy** and **security** guarantees

Technique #1

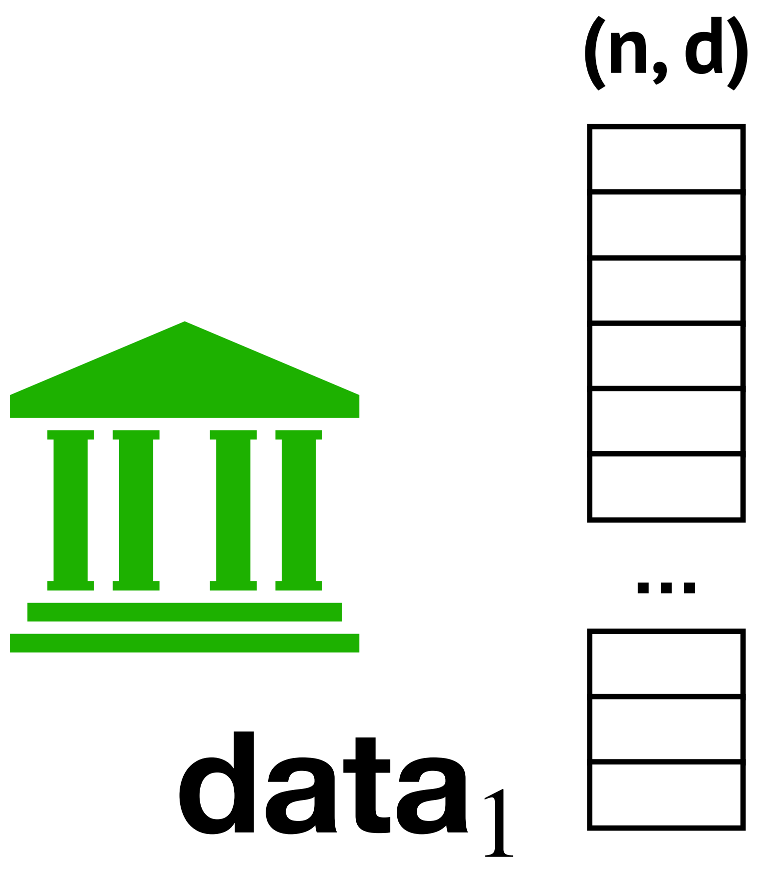
Algorithm (training) Data

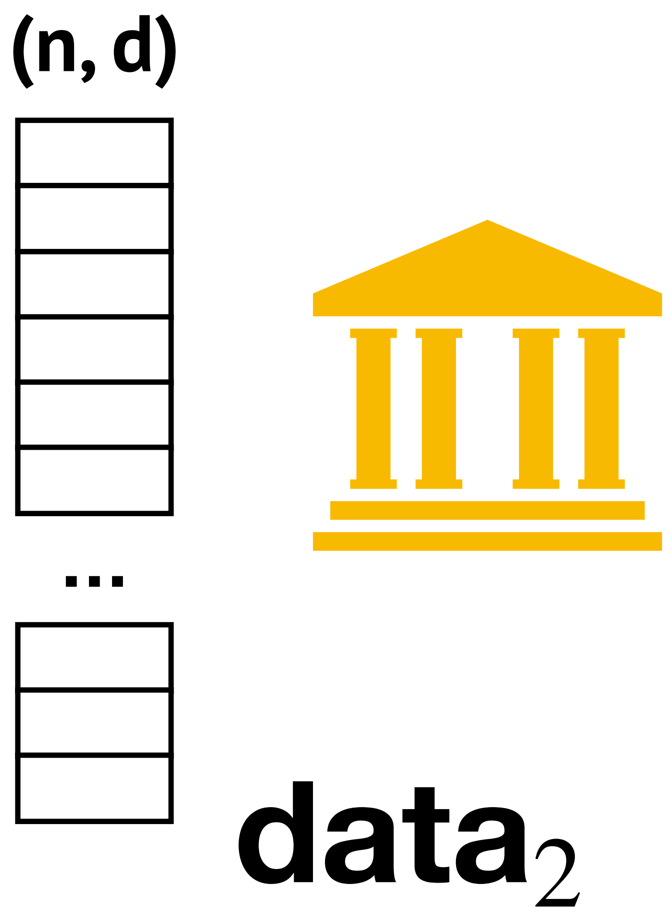
Alternative formulations

of the problem that make cryptographic computation more scalable

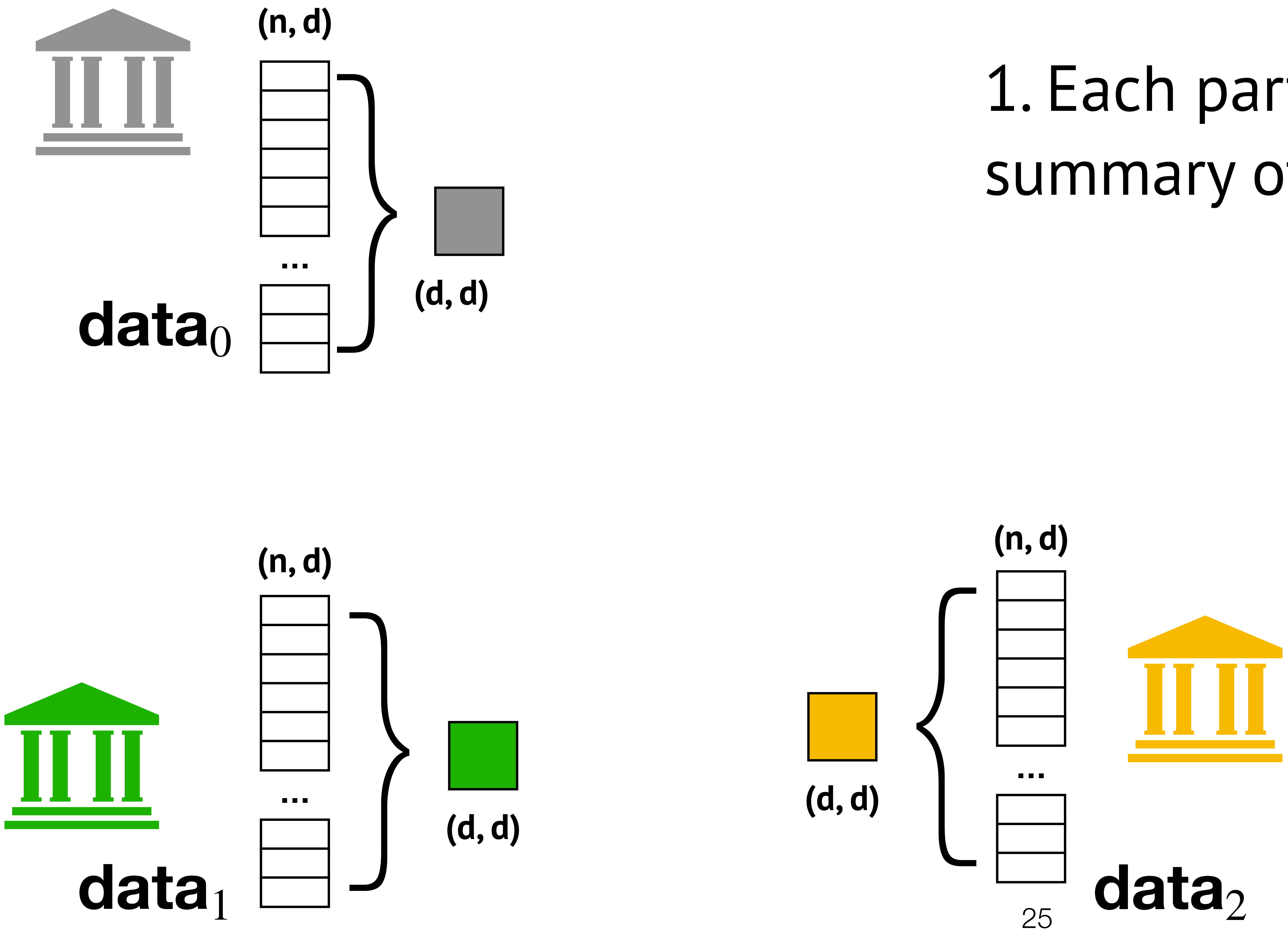
Alternative formulation of training

We identified ADMM, which allows iterative training on a small precomputed summary



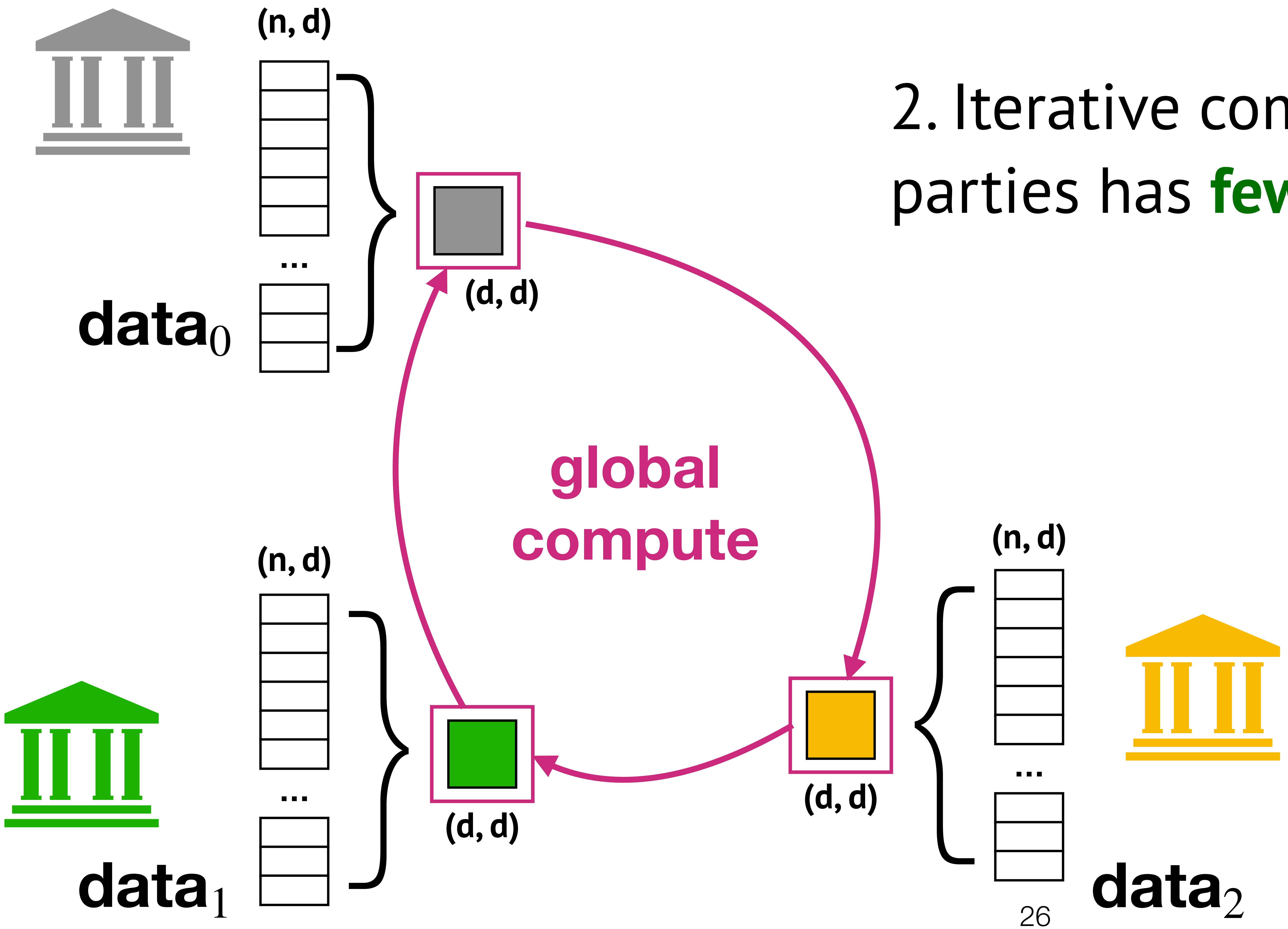


ADMM



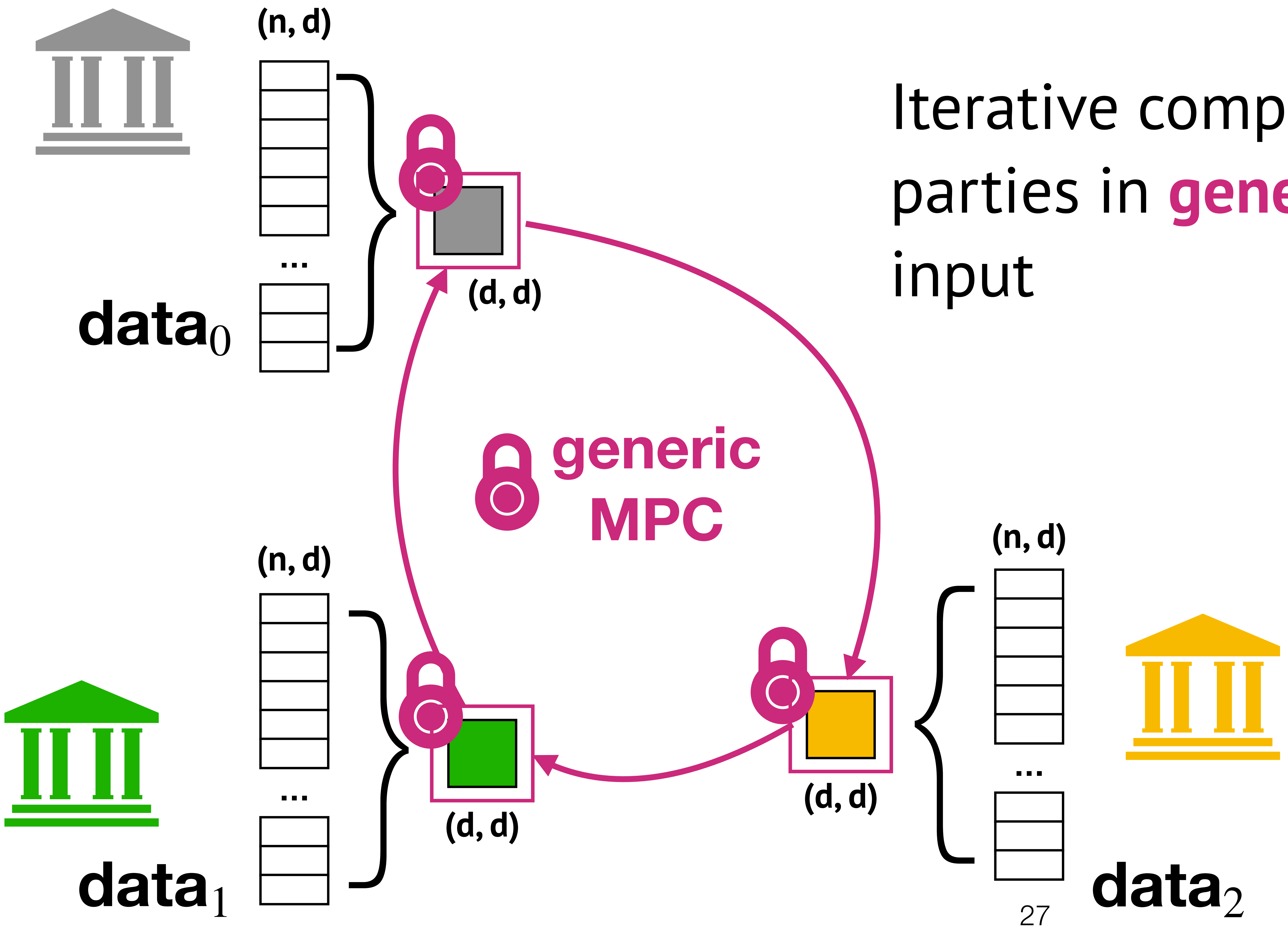
1. Each party precomputes a small summary of its input data in plaintext

ADMM



2. Iterative computation executed by all parties has **fewer steps** on **(d,d)** input

Strawman design



Iterative computation executed by all parties in **generic MPC**, **fewer** MPC on **(d,d)** input

LASSO in ADMM

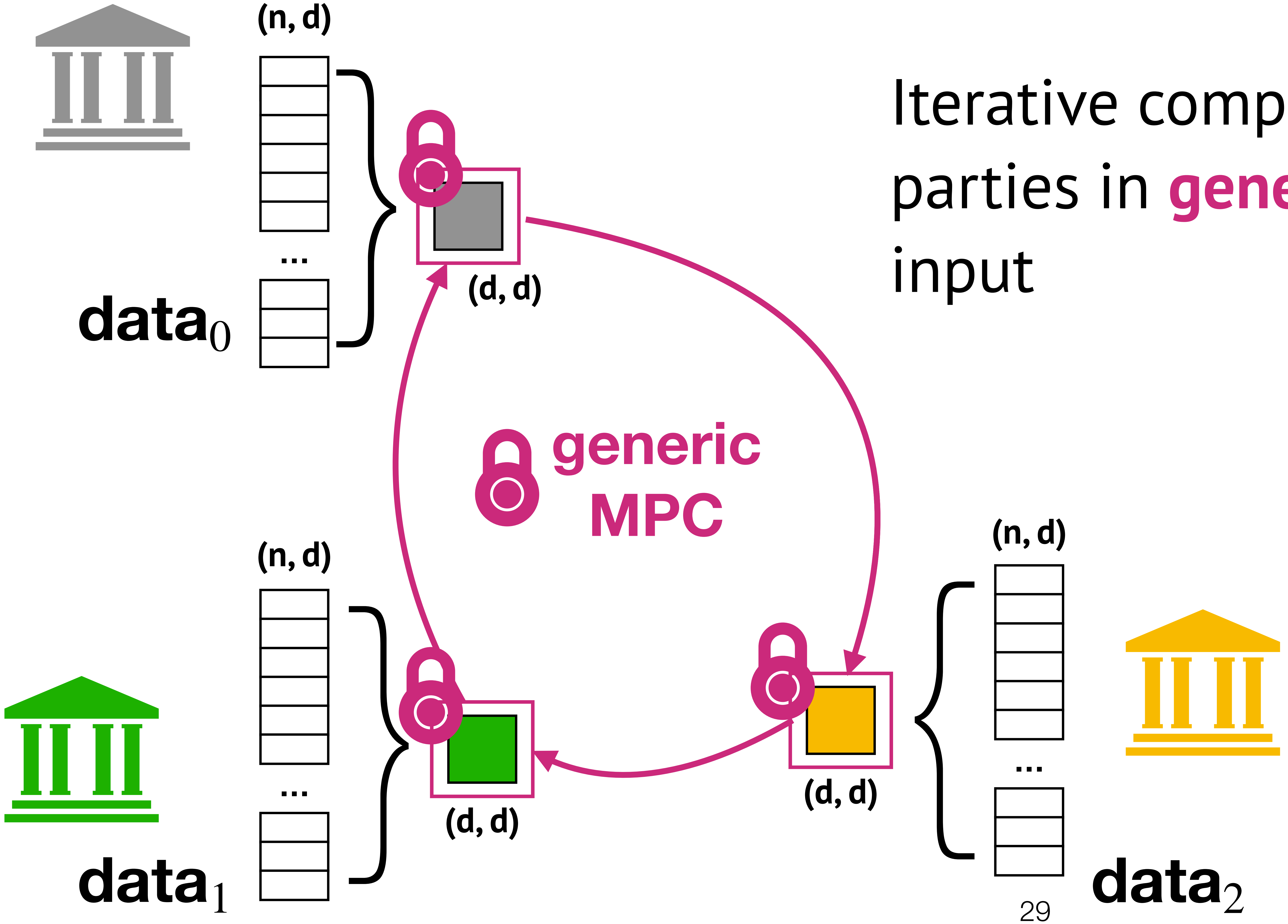
1. $\text{Summary}_i \leftarrow (\text{data}_i^T \text{data}_i + \rho I)^{-1}$
2. $\text{summary}_i \leftarrow \text{data}_i^T y_i$
3. $u^0, z^0, w^0 \leftarrow 0$
4. For $k = 1,$ **ITERS** $:$
   (a) $w_i^{k+1} \leftarrow \text{Summary}_i(\text{summary}_i + \rho(z^k - u_i^k))$
   (b) $z^{k+1} \leftarrow S_{\lambda/\rho p}(\frac{1}{p}\sum_{i=1}^{p}(w_i^{k+1} + u_i^k))$
   (c) $u_i^{k+1} \leftarrow u_i^k + w_i^{k+1} - z^{k+1}$

short (d,d)

fewer

global compute

Strawman design

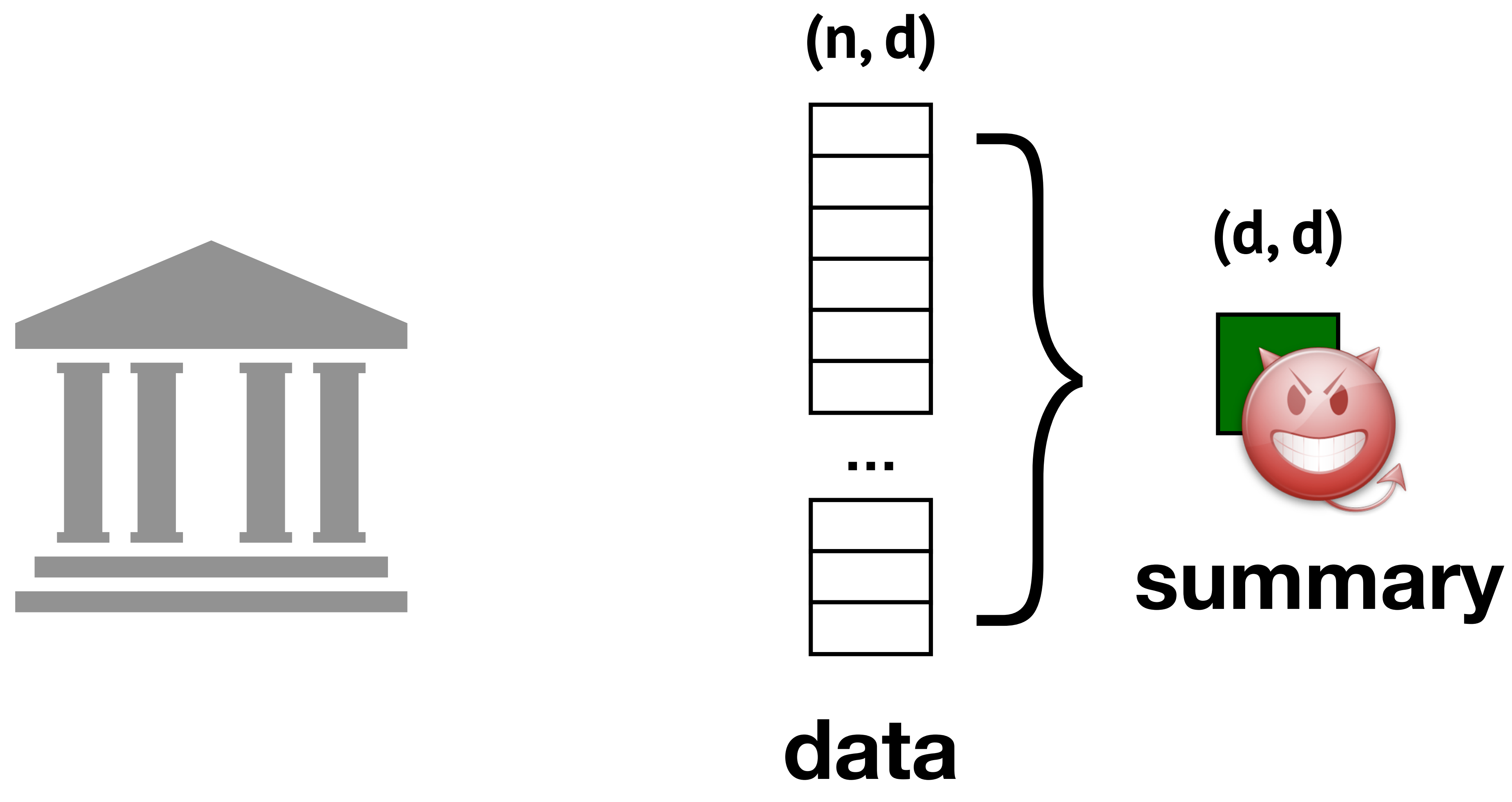


Iterative computation executed by all parties in **generic MPC**, **fewer** MPC on **(d,d)** input

Not yet secure!

Why?

Precomputation under malicious security

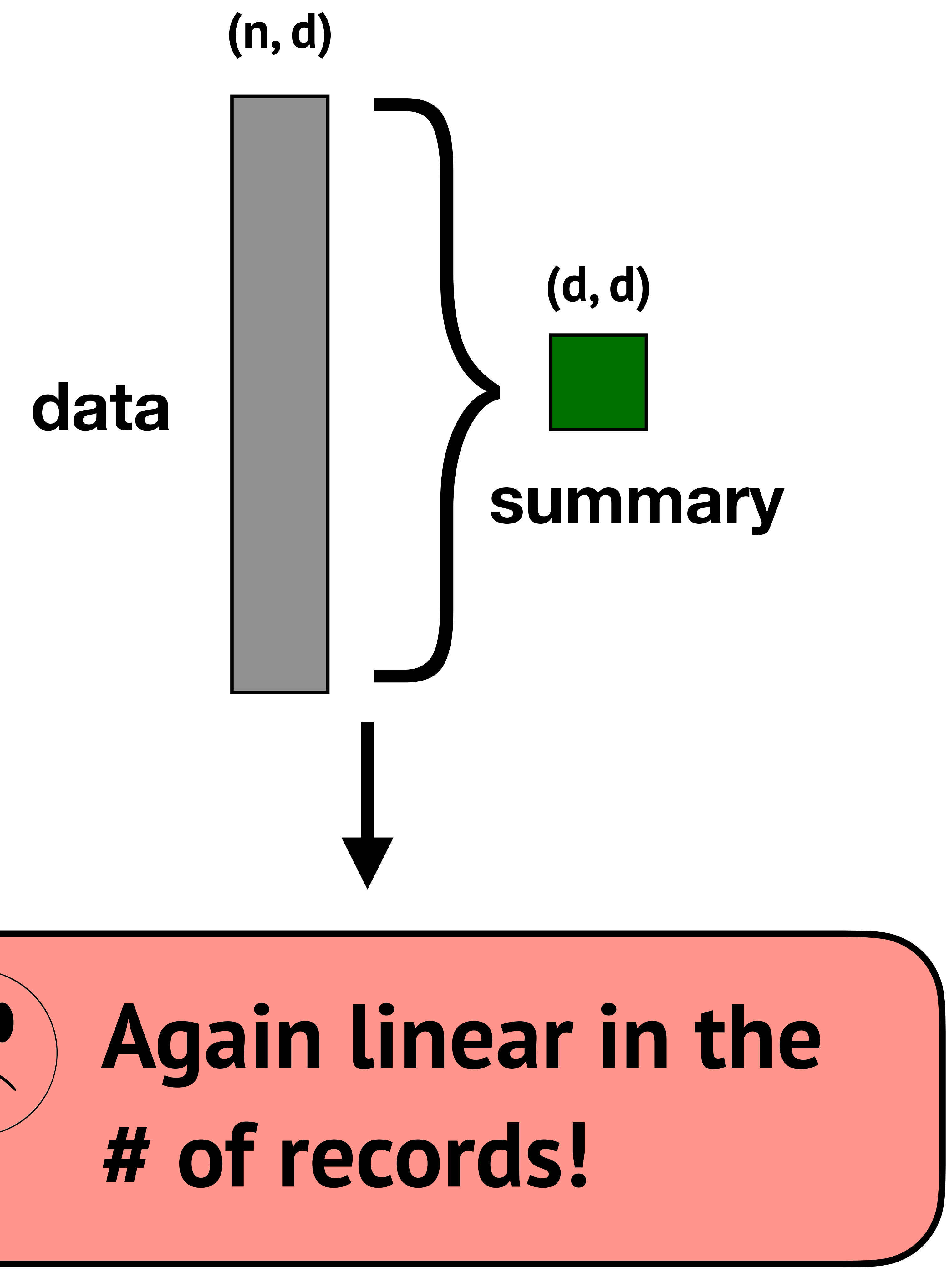


Problem: a wrong summary cannot be mapped to any valid inputs, violates ideal trusted third party model

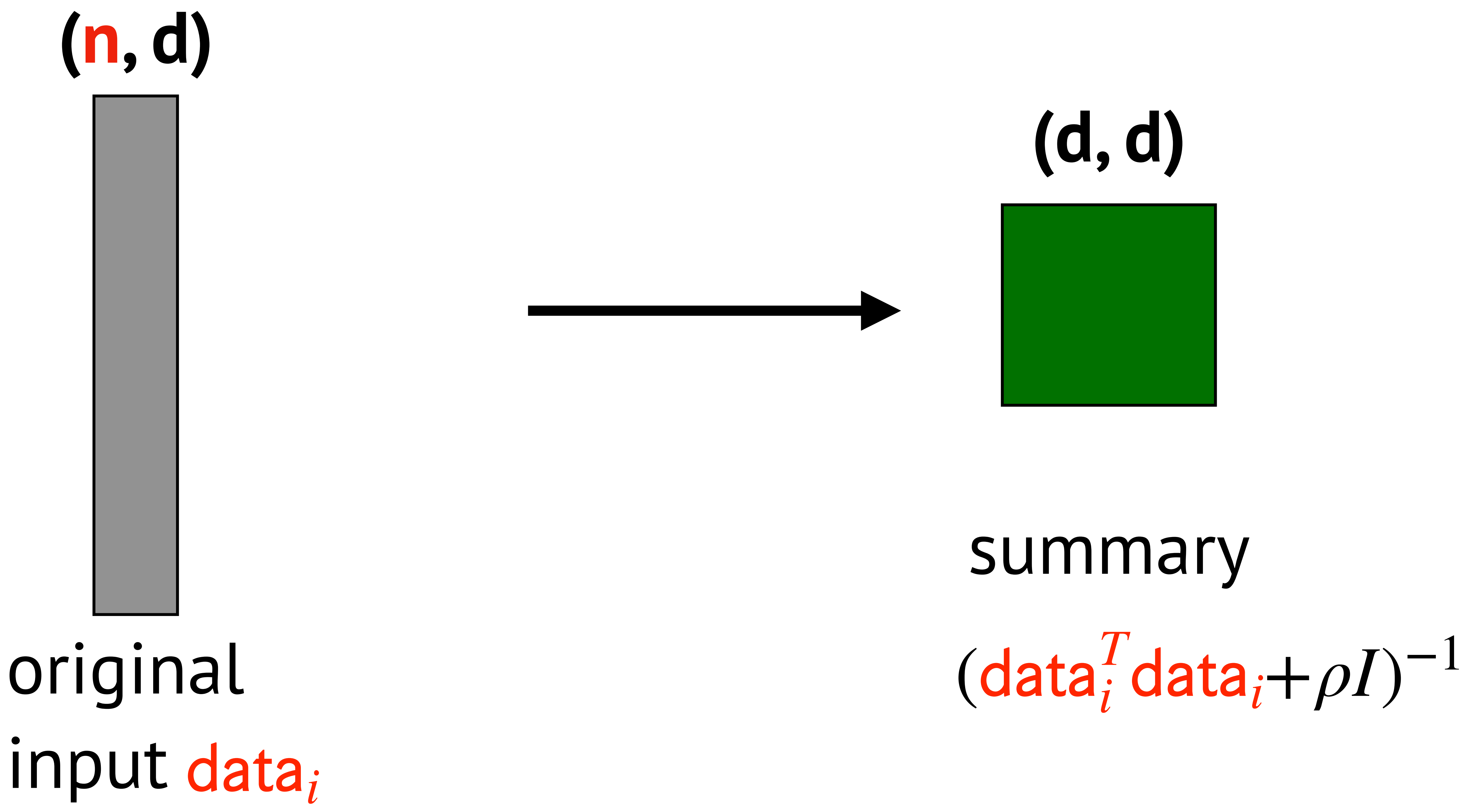
Ideas of what the attacker could achieve with this?

Proof of precomputation

We can have each party prove summary computation in zero-knowledge to the other

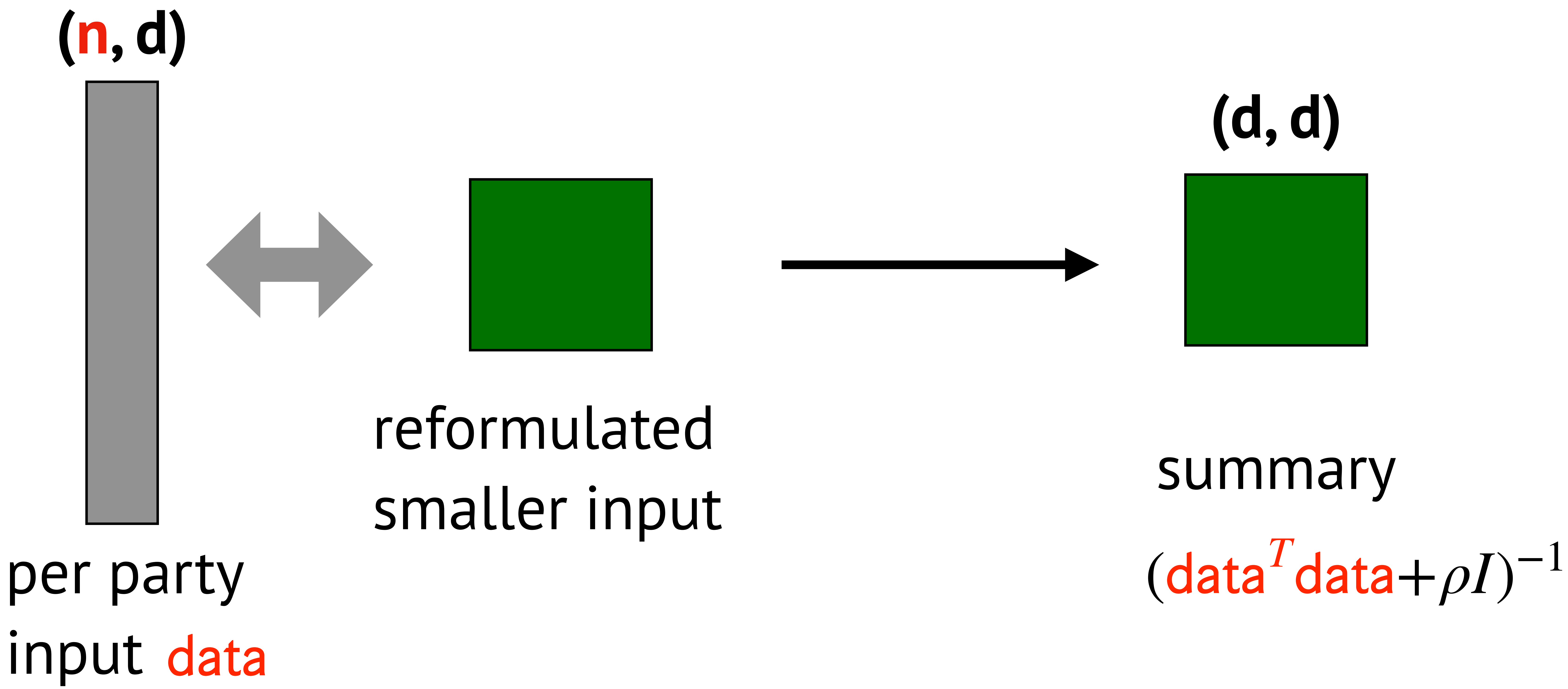


Alternative formulation of input data



Insight: find smaller inputs that preserve the summary

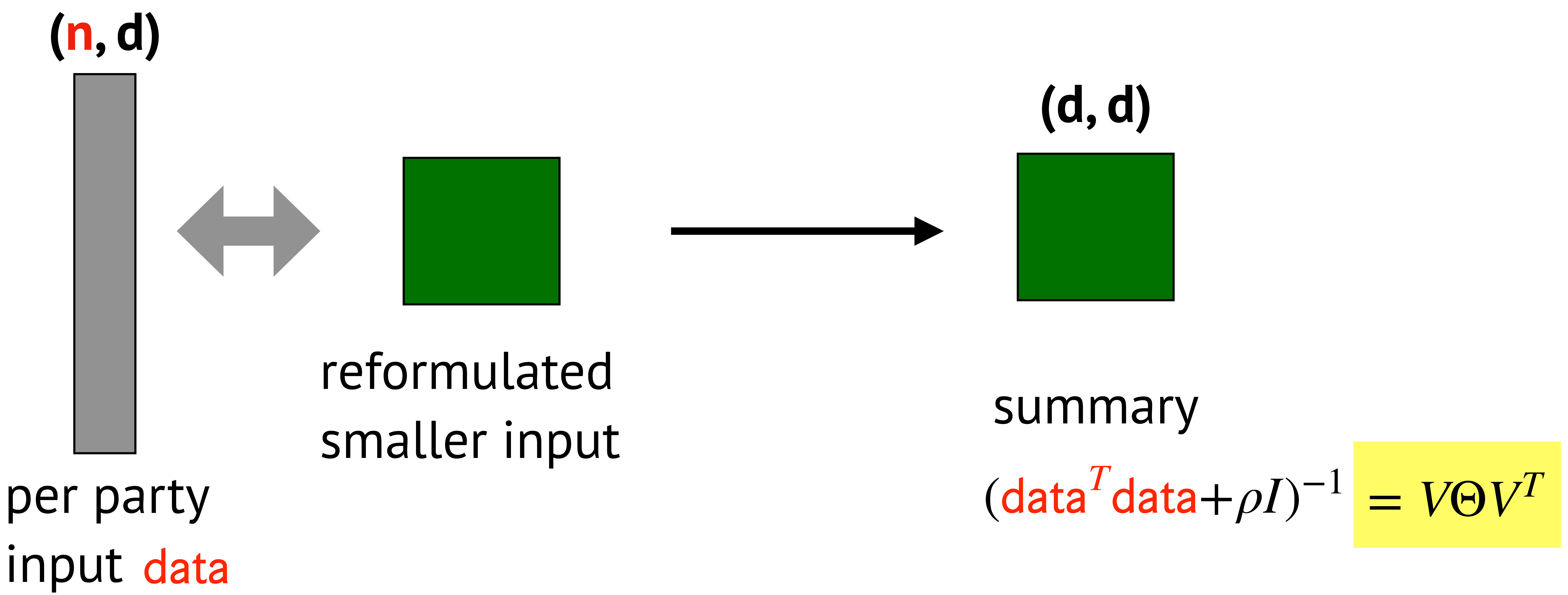
Alternative formulation of input data



Singular value decomposition says that $\exists U,\, V,\, \Gamma : \text{data} = U\Gamma V^T$ with $V \in \mathbb{R}^{d\times d}$

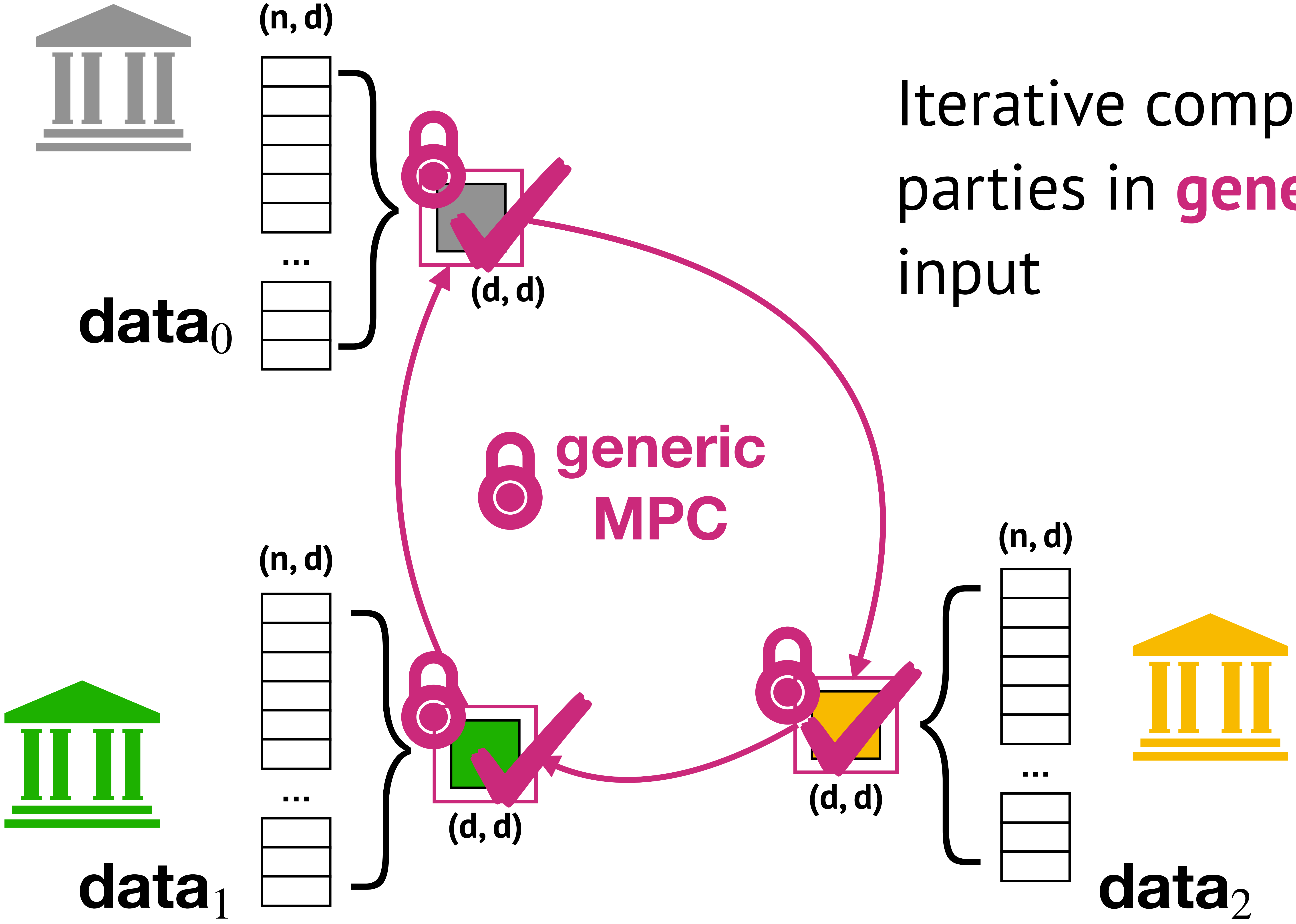
Turns out that $(\text{data}^T\text{data}+\rho I)^{-1} = V\Theta V^T$

Alternative formulation of input data



Each party proves in ZK that it knows V, Θ with certain properties from SVD such that summary $= V\Theta V^T$ ➡ **proof does not depend on n**

Strawman design 2



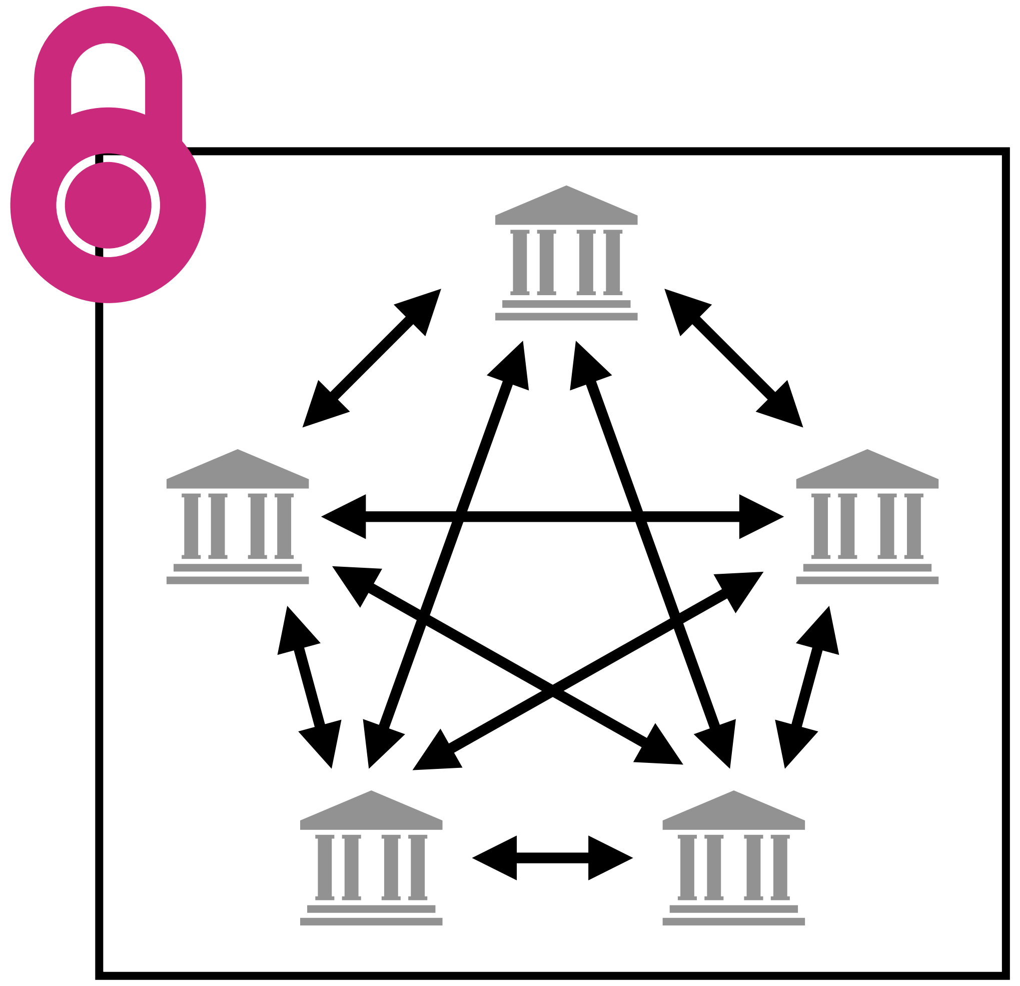
Iterative computation executed by all parties in **generic MPC**, **fewer** MPC on **(d,d)** input

Technique #2

Split secure computation into

Single party
plaintext computation
linear in n

Efficient single party
encrypted computation



Generic MPC

minimize

LASSO in ADMM

1. $\text{Summary}_i \leftarrow (\text{data}_i^T \text{data}_i + \rho I)^{-1}$
2. $\text{summary}_i \leftarrow \text{data}_i^T y_i$
3. $u^0, z^0, w^0 \leftarrow 0$
4. **For** $k = 1$, **ITERS** :

   (a) $w_i^{k+1} \leftarrow \text{Summary}_i(\text{summary}_i + \rho(z^k - u_i^k))$

   (b) $z^{k+1} \leftarrow S_{\lambda/\rho p}(\frac{1}{p}\sum_{i=1}^{p}(w_i^{k+1} + u_i^k))$

   (c) $u_i^{k+1} \leftarrow u_i^k + w_i^{k+1} - z^{k+1}$

generic MPC computation

secret MPC data

LASSO in ADMM

1. $\text{Summary}_i \leftarrow (\text{data}_i^T \text{data}_i + \rho I)^{-1}$
2. $\text{summary}_i \leftarrow \text{data}_i^T y_i$
3. $u^0, z^0, w^0 \leftarrow 0$
4. **For** $k = 1$, **ITERS** :

   (a) $w_i^{k+1} \leftarrow \text{Summary}_i(\text{summary}_i + \rho(z^k - u_i^k))$

   (b) $z^{k+1} \leftarrow S_{\lambda/\rho p}(\frac{1}{p}\sum_{i=1}^{p}(w_i^{k+1} + u_i^k))$

   (c) $u_i^{k+1} \leftarrow u_i^k + w_i^{k+1} - z^{k+1}$

linearly homomorphic encryption + custom ZK

generic MPC computation

secret MPC data

Technique #1:

Algorithm (training) Data

Alternative formulations so
cryptographic computation does not depend on the number of records

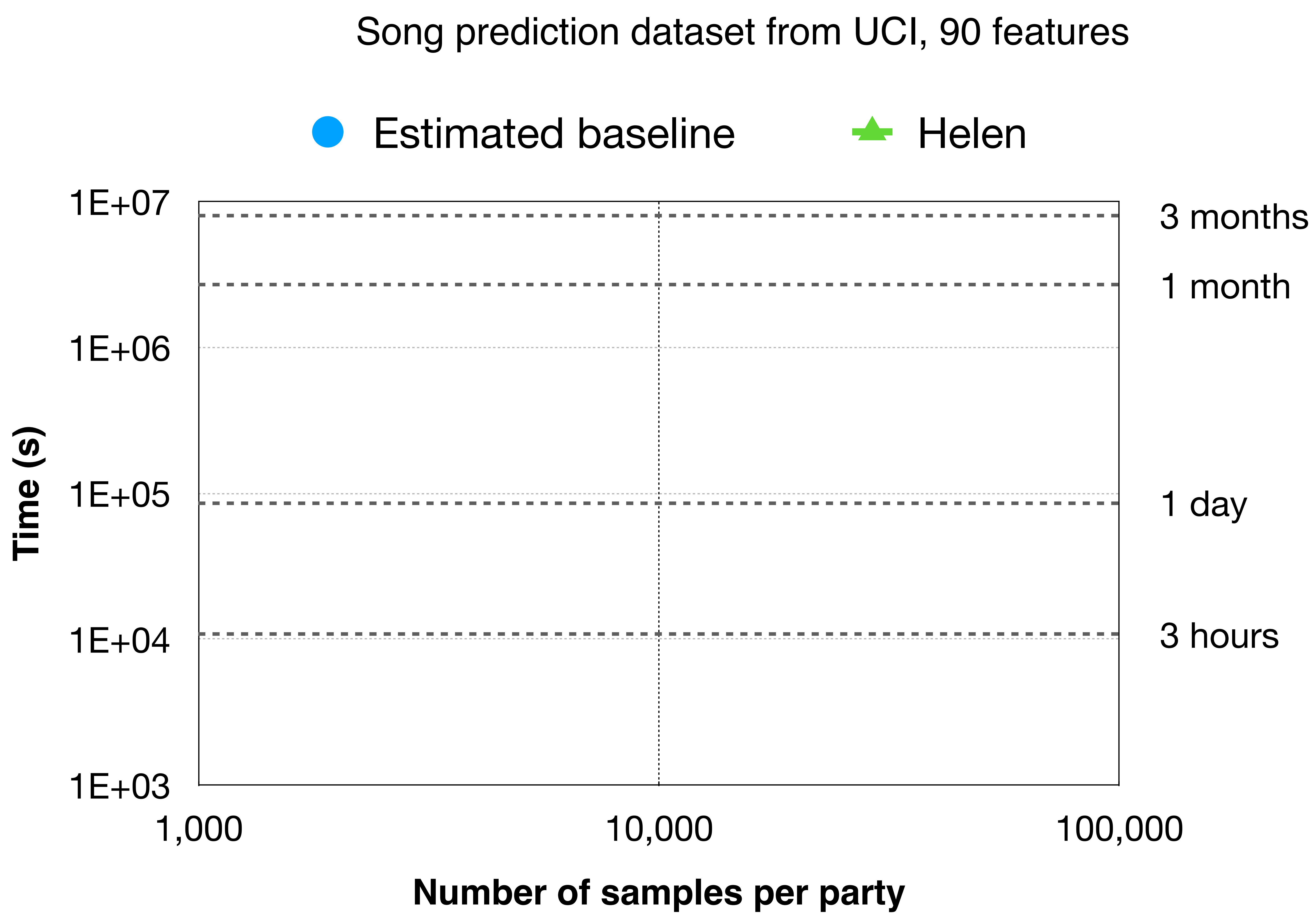
Technique #2:

split secure computation into local and global computation
to **minimize global generic MPC**

Evaluation

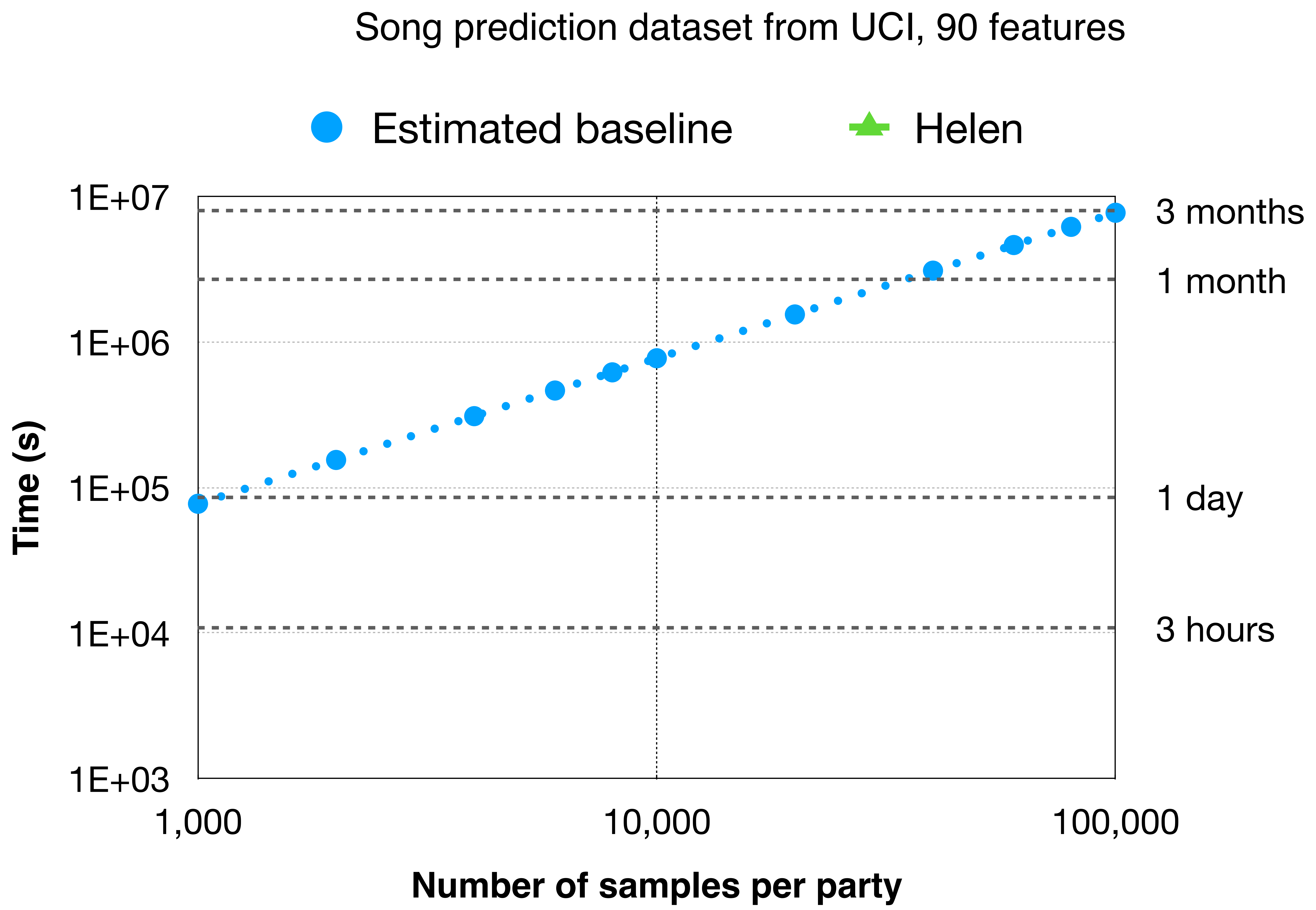
- Experiment setup
  - 4 parties: 4 r4.8xlarge machines on EC2. Two in Oregon and two in Northern Virginia
- Baseline is SGD implemented in SPDZ, a generic maliciously secure MPC platform
- ADMM converges within 10 iterations

Evaluation

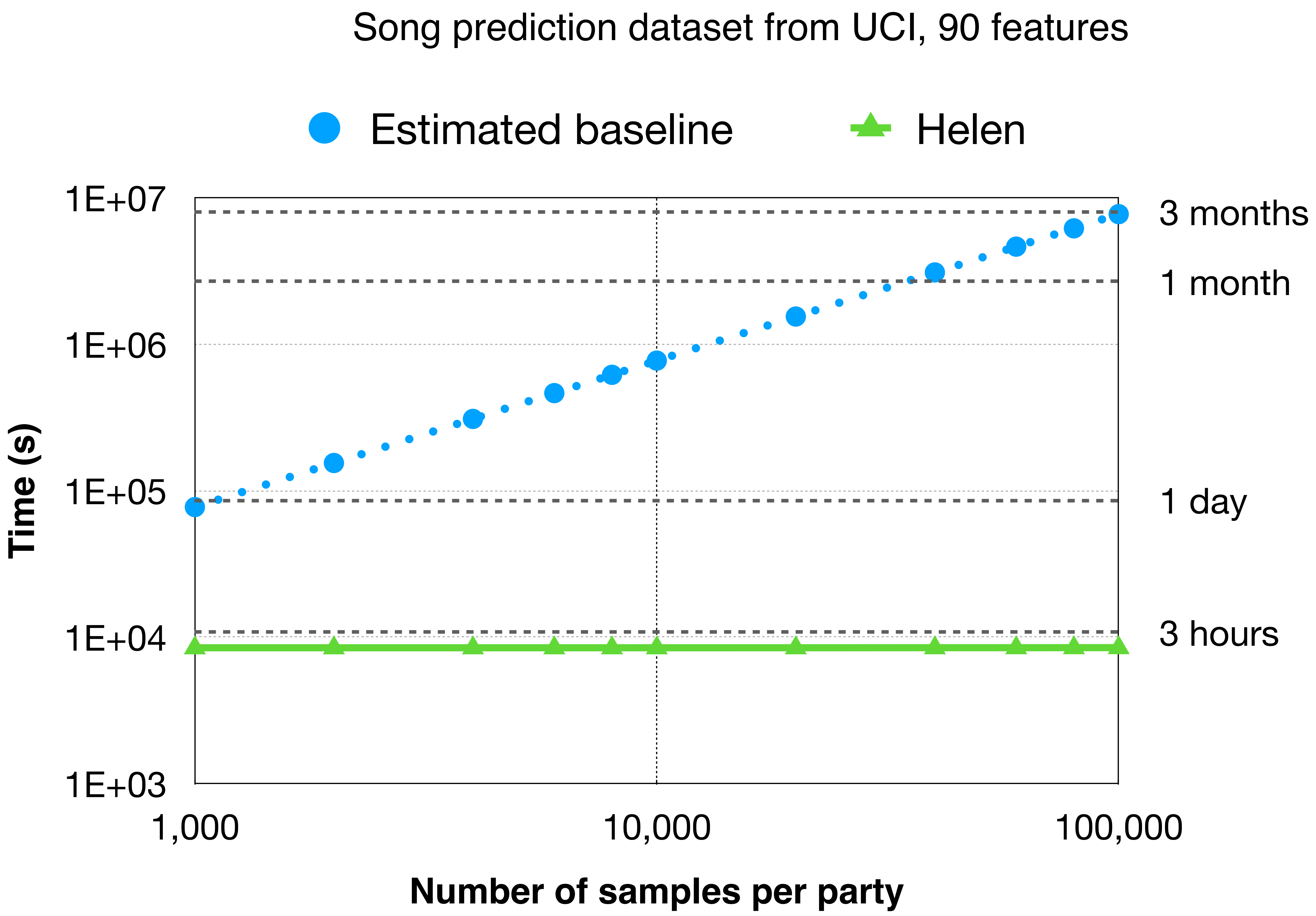


Song prediction dataset from UCI, 90 features
Estimated baseline
Helen
1E+07
1E+06
1E+05
1E+04
1E+03
Time (s)
3 months
1 month
1 day
3 hours
1,000
10,000
100,000
Number of samples per party

Evaluation

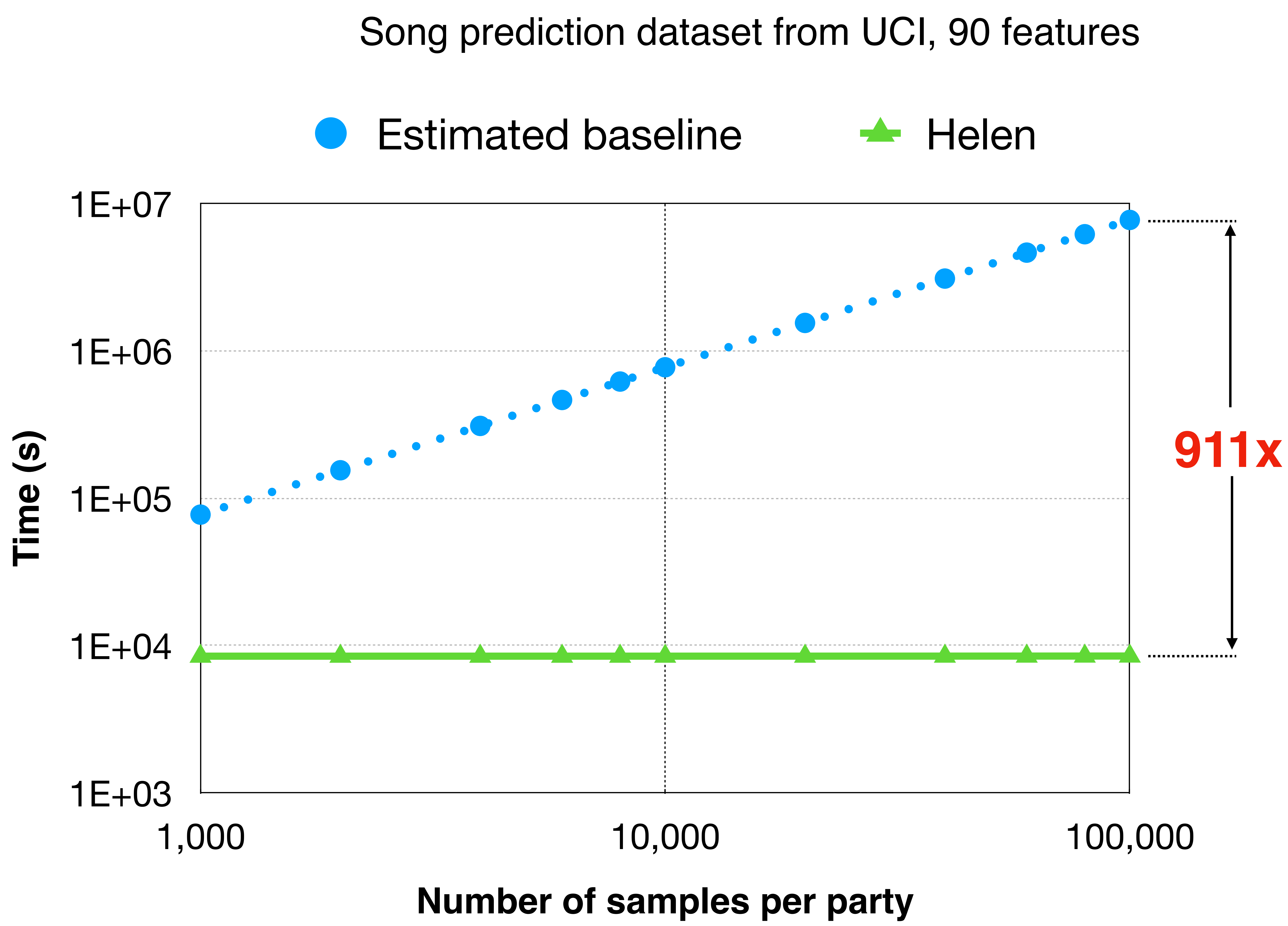

Song prediction dataset from UCI, 90 features
Estimated baseline
Helen
1E+07
1E+06
1E+05
1E+04
1E+03
Time (s)
3 months
1 month
1 day
3 hours
1,000
10,000
100,000
Number of samples per party

Evaluation



Song prediction dataset from UCI, 90 features
Estimated baseline
Helen
1E+07
1E+06
1E+05
1E+04
1E+03
Time (s)
3 months
1 month
1 day
3 hours
1,000
10,000
100,000
Number of samples per party

Evaluation



Song prediction dataset from UCI, 90 features
Estimated baseline
Helen
1E+07
1E+06
1E+05
1E+04
1E+03
Time (s)
1,000
10,000
100,000
Number of samples per party
911x

Helen summary

Provides maliciously-secure MPC for collaboratively training regularized linear models

Reduces state of the art by 3 orders of magnitude, making such training feasible for modest data sizes

Efficiency is achieved via a co-design of cryptography, systems, and ML

Ending remarks

- Future classes
- Please fill in course evals (Piazza links)
- Thanks