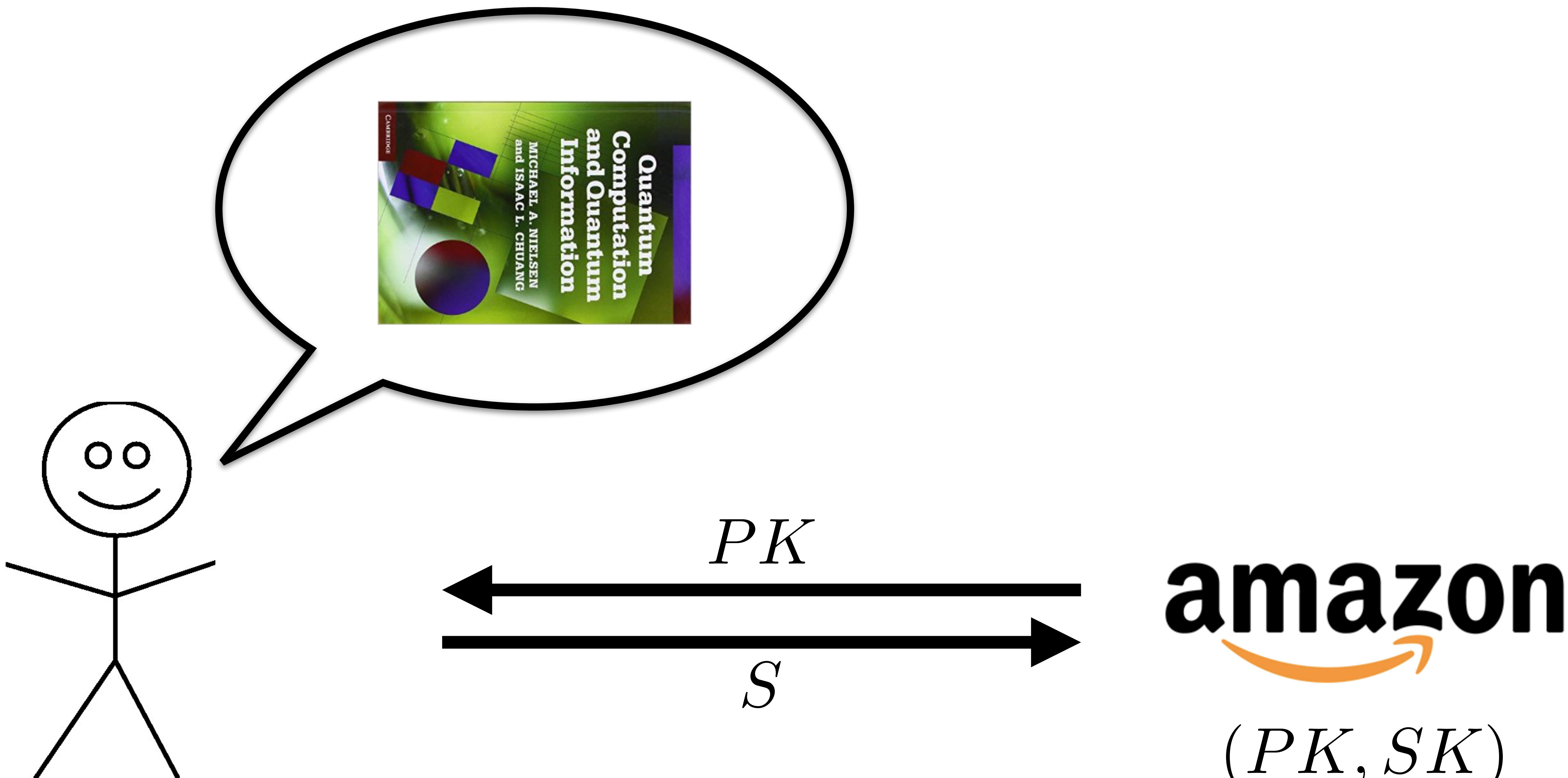
Quantum Computation & Cryptography
Day 3

Post-quantum cryptography

Andru Gheorghiu

More about public-key cryptography



$S = Enc(PK, \text{Card details})$ $\quad$ $Dec(SK, S) \rightarrow \text{Card details}$

Let's be a bit more precise

More about public-key cryptography

$$KeyGen : \mathcal{S} \to \mathcal{K} \times \mathcal{K}$$

$$KeyGen(seed) = (PK, SK)$$

$$Enc : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$$

Not necessarily a function (might use randomness)

$$Dec : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$$

Properties we want

$KeyGen, Enc, Dec$ computable in (classical) polynomial time

$\forall (PK, SK) \in Range(KeyGen),$

$$\forall M \in \mathcal{M}, Dec(SK, Enc(PK, M)) = M$$

More about public-key cryptography

$KeyGen, Enc, Dec$ computable in (classical) polynomial time

$$\forall (PK, SK) \in Range(KeyGen),$$

$$\forall M \in \mathcal{M}, Dec(SK, Enc(PK, M)) = M$$

Denote as PPM the set of probabilistic poly-time machines/algorithms

Given any two messages M_1 and M_2 it must be that

$$\forall A \in PPM$$

$$|Pr[A(PK, Enc(PK, M_1)) = 1] - Pr[A(PK, Enc(PK, M_2)) = 1]|$$

$$\leq \text{small}$$

More about public-key cryptography

$KeyGen, Enc, Dec$ computable in (classical) polynomial time

$$\forall (PK, SK) \in Range(KeyGen),$$

$$\forall M \in \mathcal{M}, Dec(SK, Enc(PK, M)) = M$$

Denote as PPM the set of probabilistic
poly-time machines/algorithms

Given any two messages M_1 and M_2 it must be that

$$\forall A \in PPM$$

$$Pr[A(PK, Enc(PK, M_1)) = 1] \approx Pr[A(PK, Enc(PK, M_2)) = 1]$$

Computational (semantic) security

More about public-key cryptography

These properties can be achieved with
trapdoor one-way functions

One-way function

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

The function can be evaluated in polynomial-time

Hard to invert efficiently

$$\forall A \in PPM, Pr[A(f(x_1)) = 1] \approx Pr[A(f(x_2)) = 1]$$

What about the trapdoor?

More about public-key cryptography

Trapdoor one-way function

$$(f, T)\ where\ f : \mathcal{X} \rightarrow \mathcal{Y},\ T \in \mathcal{T},\ s.t.$$

f is a one-way function

There exists a PPM M such that

$$\forall y \in Range(f), M(T, y) = x, f(x) = y$$

Trapdoor information allows you to
invert the function efficiently

$$(f, T) \rightarrow (PK, SK)$$

$$Enc(PK, \cdot) = f(\cdot)$$
$$Dec(SK, \cdot) = M(T, \cdot)$$

More about public-key cryptography

Do such functions exist?

We think so, but there is no proof

$$f(x, n, l) = x^l \ mod \ n$$

$$n = p \cdot q, l \text{ co-prime with } (p-1)(q-1)$$

This is the RSA function

For l=2, can be shown that inverting f is equivalent to factoring n

No known poly-time classical algorithm

But there is Shor's algorithm

Complexity theory

One-way functions are based on **NP** problems

A problem is in **NP** iff the solution can be checked in classical polynomial time

Example

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Complexity theory

One-way functions are based on **NP** problems

A problem is in **NP** iff the solution can be checked in classical polynomial time

Example

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Complexity theory

One-way functions are based on **NP** problems

A problem is in **NP** iff the solution can be checked in classical polynomial time

Recall that **P** is the class of problems whose solution can be found in classical polynomial time

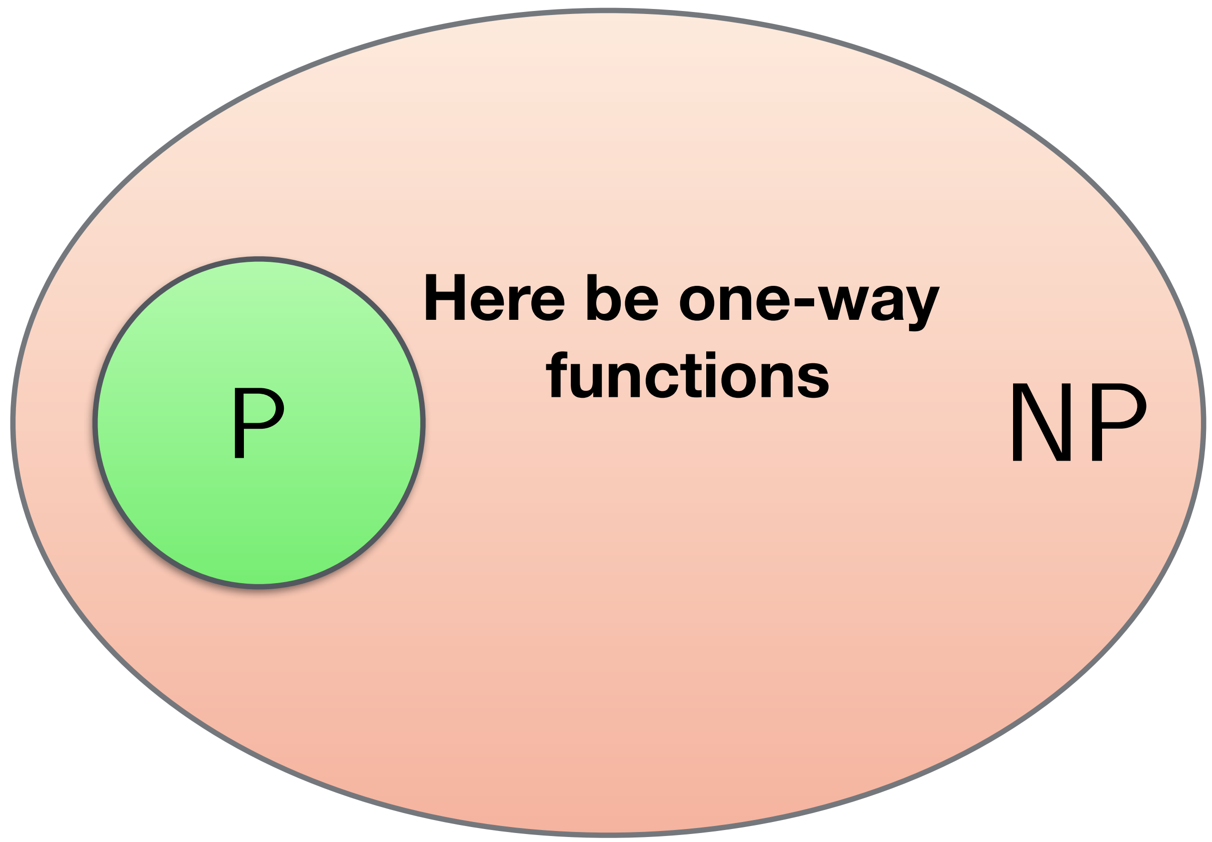
Clearly $\mathsf{P} \subseteq \mathsf{NP}$

The million dollar question

$$\mathsf{P} \stackrel{?}{=} \mathsf{NP}$$

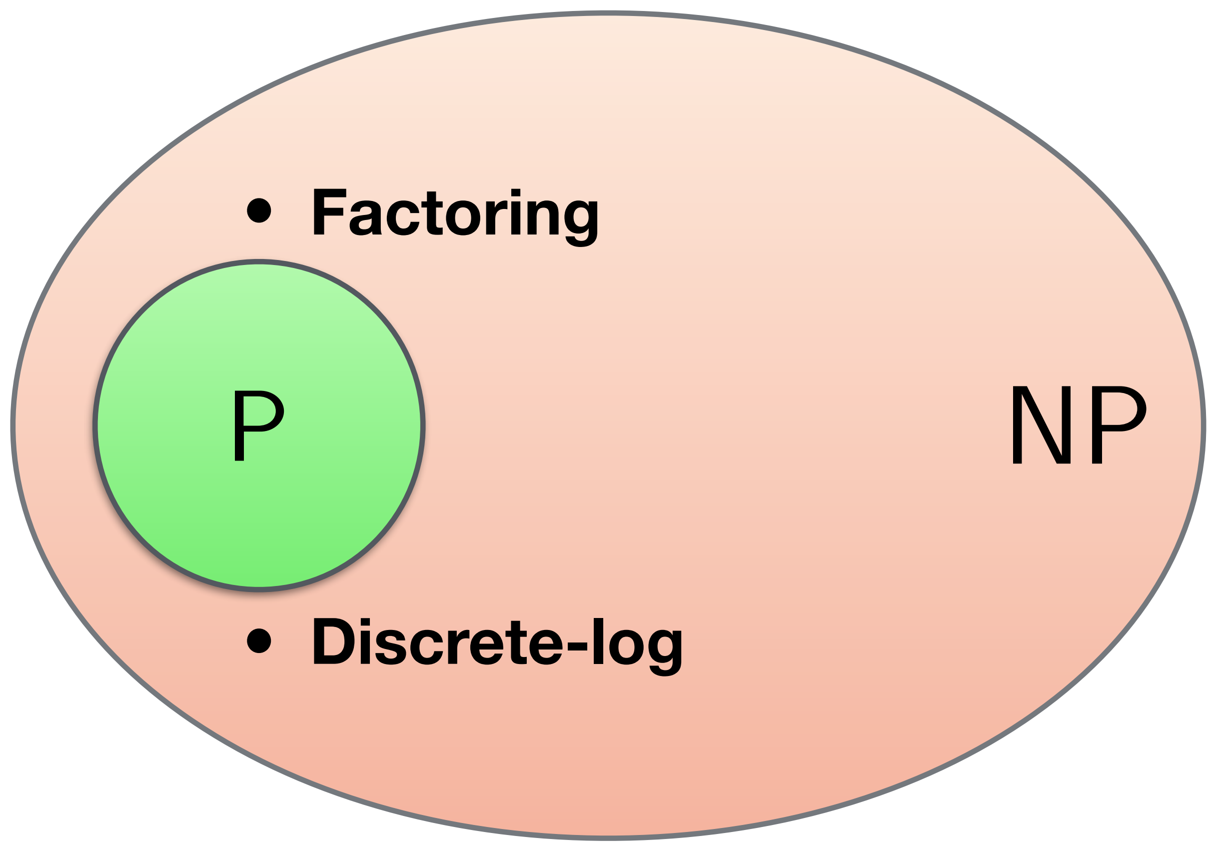
Complexity theory

Conjectured relationship between classes



Complexity theory

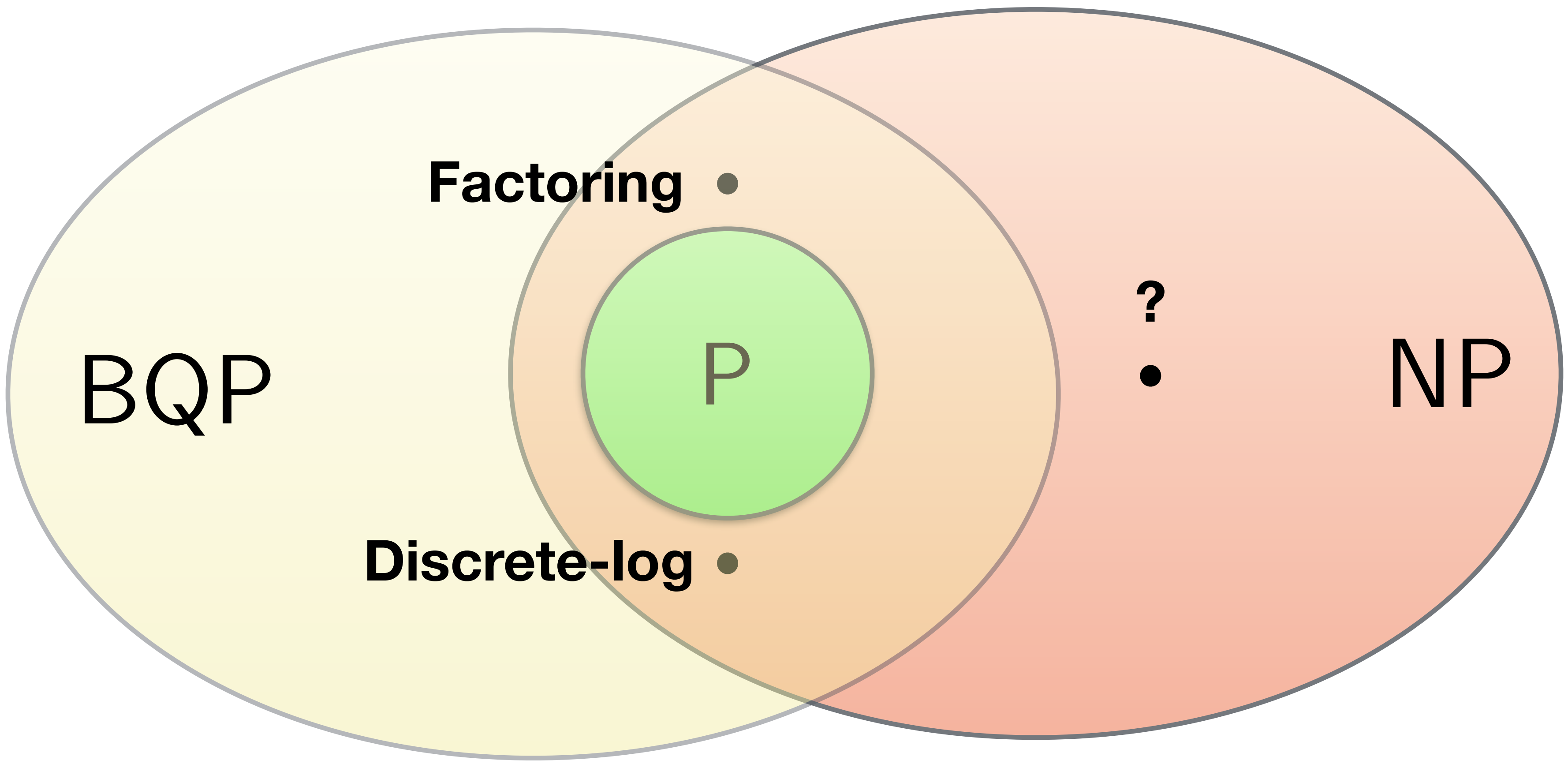
Conjectured relationship between classes



What about quantum computations (**BQP**)?

Complexity theory

Conjectured relationship between classes

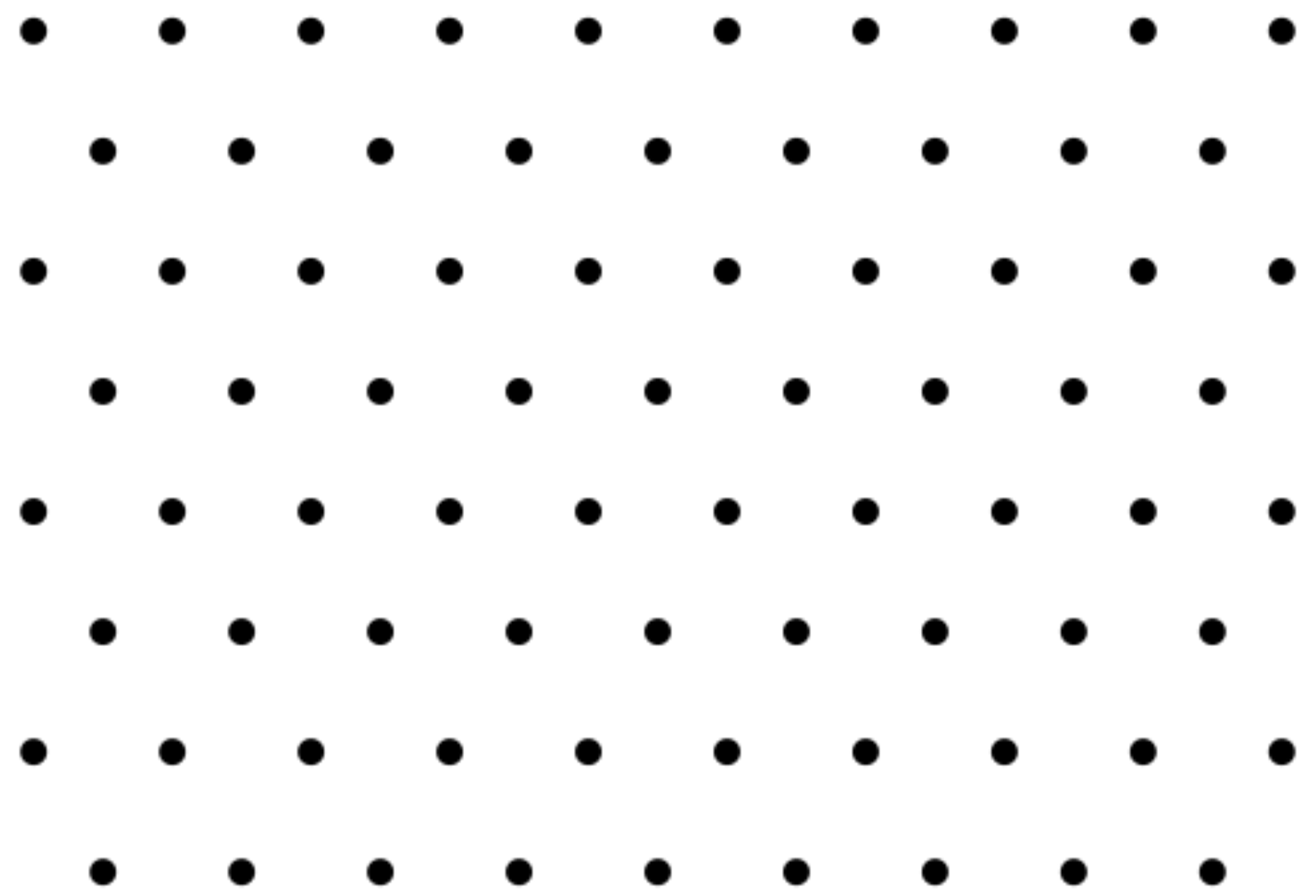


Can we find one-way functions that are hard to invert for quantum computers as well?

Lattice problems

Lattices

What is a lattice?



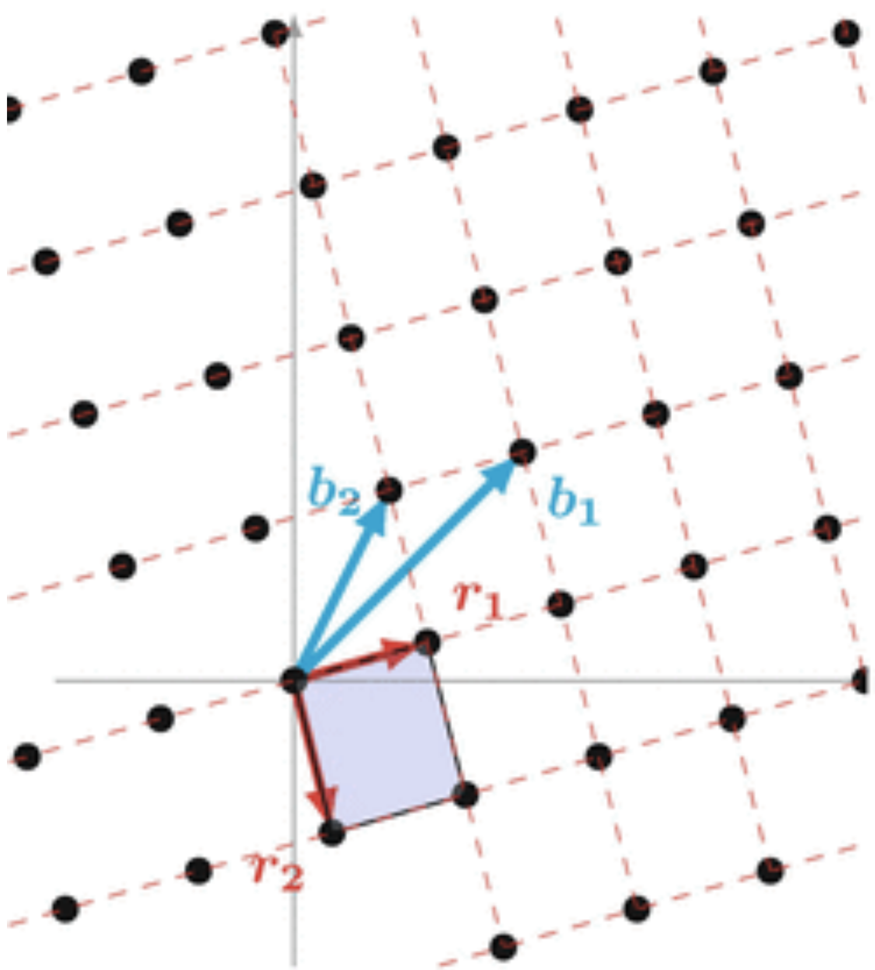
Essentially a discrete vector space

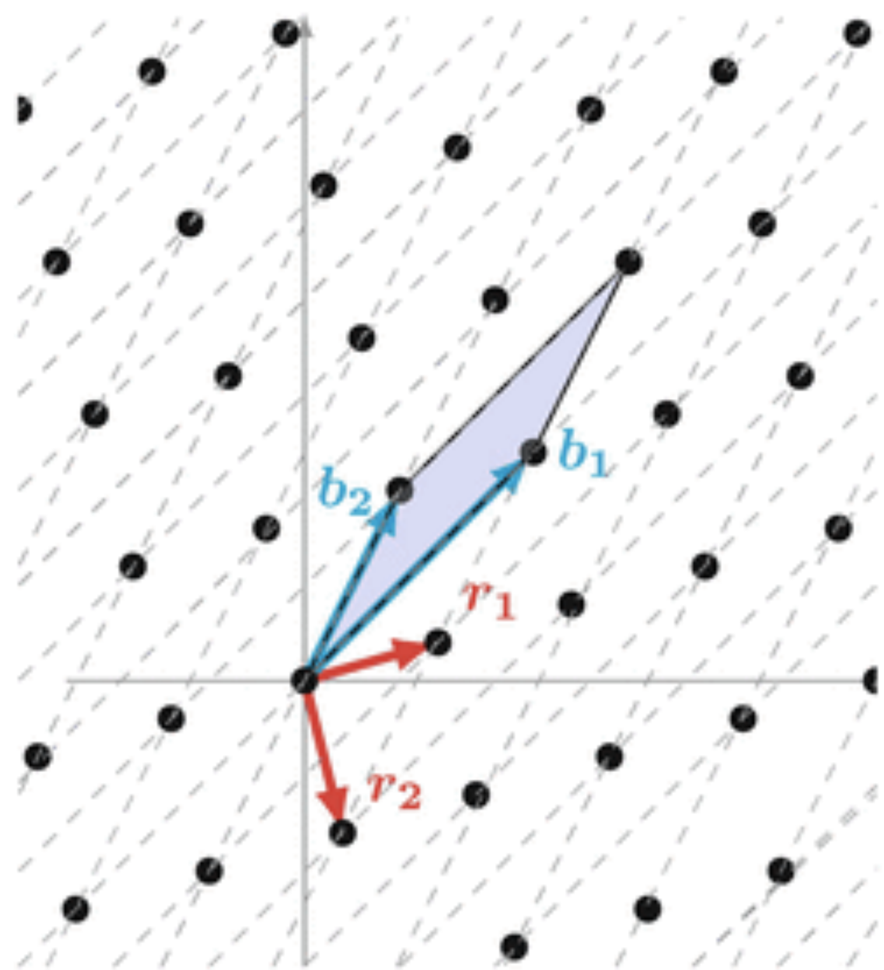
Lattices

Let $\mathbf{B} = \{v_1, v_2, ...v_n\}$ be a basis of $\mathbb{R}^n$

Then, a lattice is the following

$$\mathcal{L}(\mathbf{B}) = \{a_1v_1 + a_2v_2 + ... + a_nv_n | a_1, a_2, ...a_n \in \mathbb{Z}\}$$





Lattice problems

Input: $\mathbf{B} = \{v_i\}_{i \leq n}$

Output: $\{a_i\}_{i \leq n} \quad a_i \in \mathbb{Z}$

Such that for $w = \sum_{i=1}^{n} a_i v_i$ it must be that

$$w \neq 0$$

w is (one of) the shortest vector(s) in $\mathbf{B}$

Lattice problems

Input: $\mathbf{B} = \{v_i\}_{i \leq n}$

Output: $\{a_i\}_{i \leq n} \quad a_i \in \mathbb{Z}$

Such that for $w = \sum_{i=1}^{n} a_i v_i$ it must be that

$$w \neq 0 \qquad ||w|| = l_{min}$$

Lattice problems

Input: $\mathbf{B} = \{v_i\}_{i \leq n}$

Output: $\{a_i\}_{i \leq n} \quad a_i \in \mathbb{Z}$

Such that for $w = \sum_{i=1}^{n} a_i v_i$ it must be that

$$w \neq 0 \qquad ||w|| = l_{min}$$

Shortest Vector Problem (SVP)

If you can solve it in poly-time,
you can solve any **NP** problem in poly-time

Lattice problems

Input: $\mathbf{B} = \{v_i\}_{i \leq n}$

Output: $\{a_i\}_{i \leq n} \quad a_i \in \mathbb{Z}$

Such that for $w = \sum_{i=1}^{n} a_i v_i$ it must be that

$$w \neq 0 \qquad ||w|| = l_{min}$$

Shortest Vector Problem (SVP)

Problem is **NP**-hard

There's also a similar problem called
Closest Vector Problem (CVP)

Lattice problems

$$SVP_\gamma,\ \gamma \geq 1$$

Input: $\mathbf{B} = \{v_i\}_{i \leq n}$

Output: $\{a_i\}_{i \leq n} \quad a_i \in \mathbb{Z}$

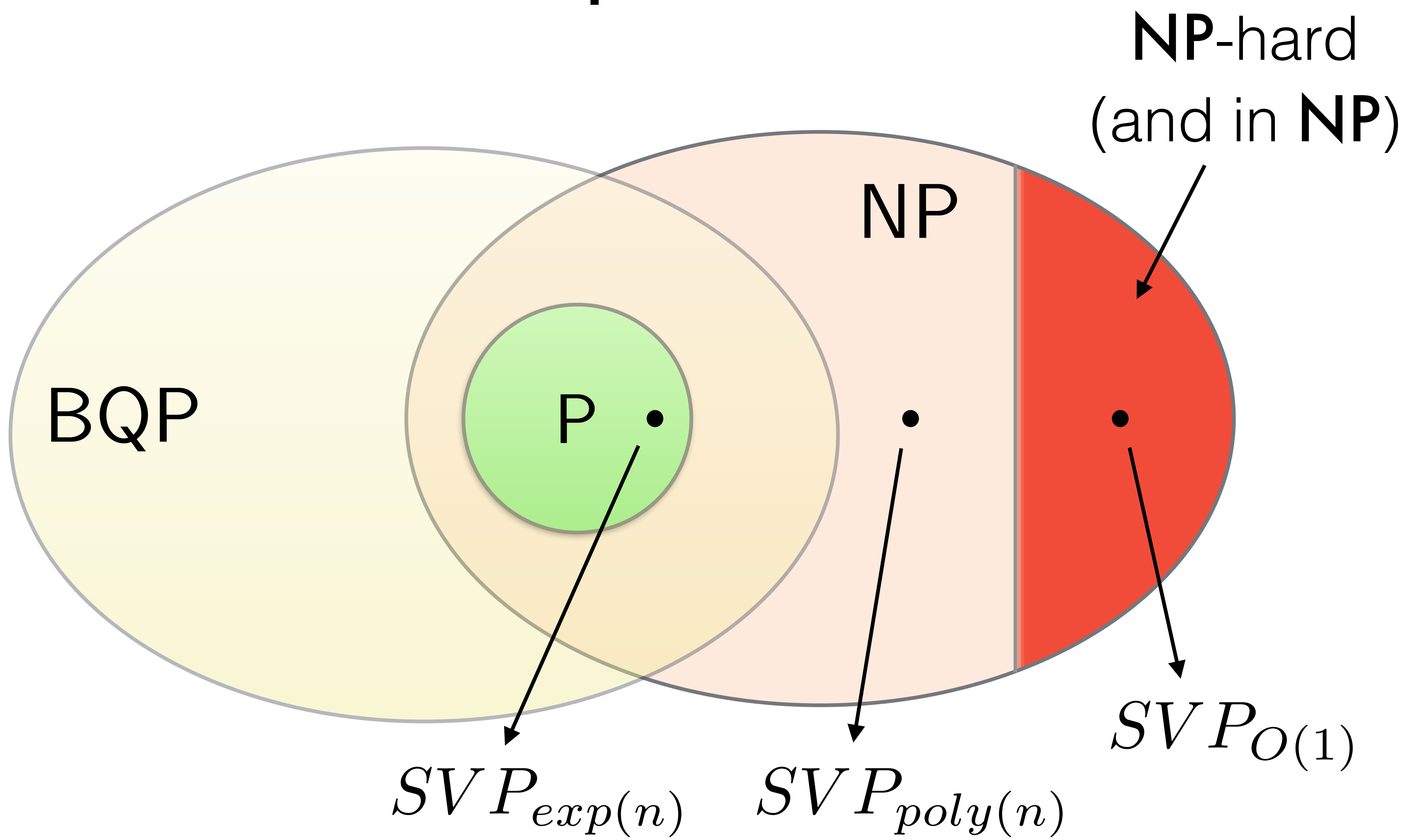
Such that for $w = \sum_{i=1}^{n} a_i v_i$ it must be that

$$w \neq 0 \qquad ||w|| \leq \gamma \cdot l_{min}$$

For constant γ this is still **NP**-hard*

For $\gamma = poly(n)$ best algorithms require $2^{O(n)}$ time and space

Lattice problems



$SVP_{poly(n)}$ seems like a good candidate for post-quantum crypto!

Average case vs. worst case

Complexities we've mentioned refer to worst case

In practice we care about **average case**

Expected running time averaged over all inputs

Why average case?

Instances we generate are usually random

Might be difficult to find the worst case instances

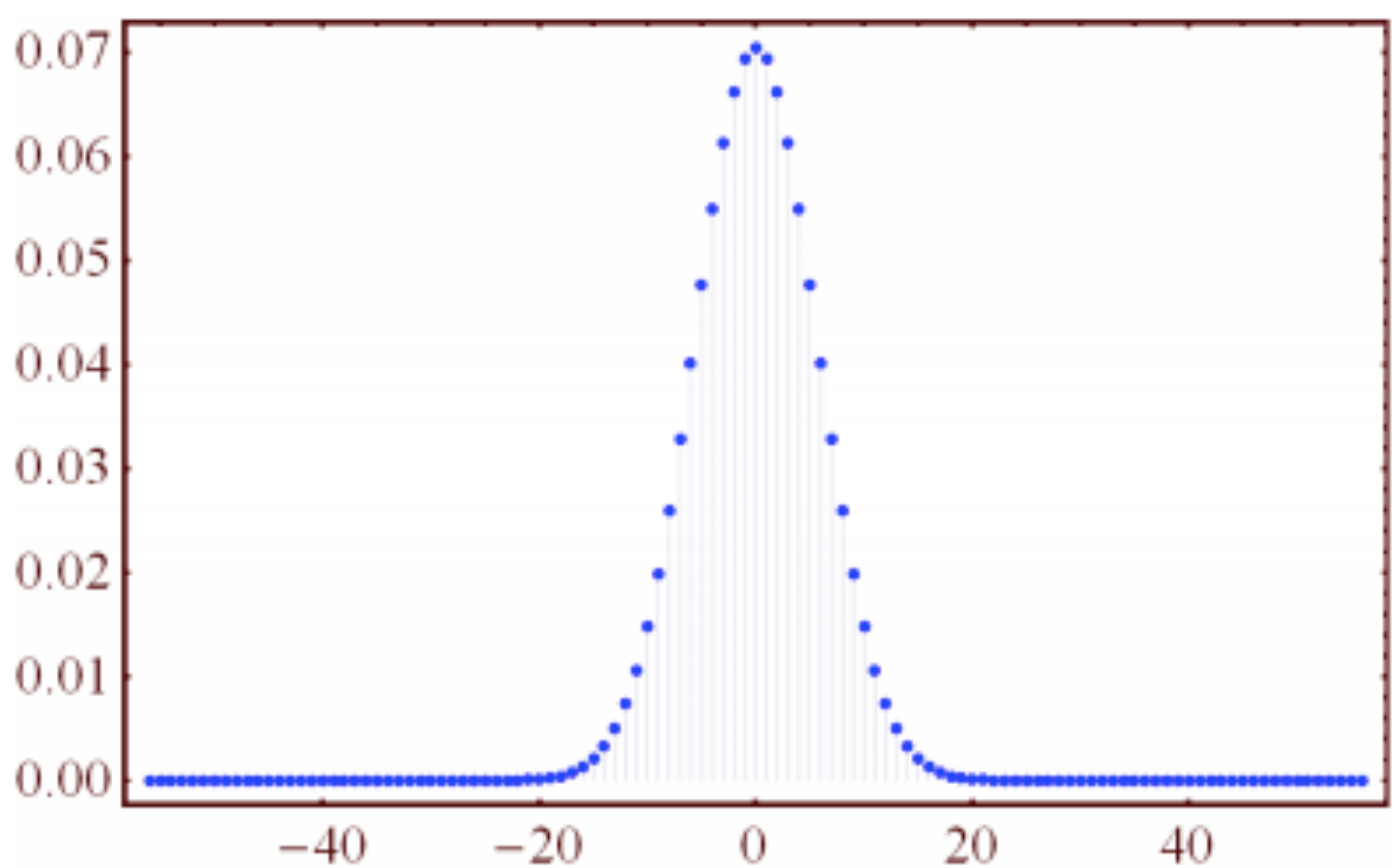
Learning With Errors (LWE)

Parameters

Dimension $n, q = poly(n), m = O(n\, log(q)), \mathcal{E}$ error distribution

$$\mathcal{E} : \{0, 1, ...q-1\} \to [0, 1] \qquad \sum_x \mathcal{E}(x) = 1 \qquad \sqrt{n} \le std(\mathcal{E}) \ll q$$

Error distribution is typically a discrete Gaussian



Learning With Errors (LWE)

Parameters

Dimension $n, q = poly(n), m = O(n\ log(q)), \mathcal{E}$ error distribution

$$\mathcal{E} : \{0, 1, ...q - 1\} \to [0, 1] \qquad \sum_x \mathcal{E}(x) = 1 \qquad \sqrt{n} \leq std(\mathcal{E}) \ll q$$

Error distribution is typically a discrete Gaussian

Problem

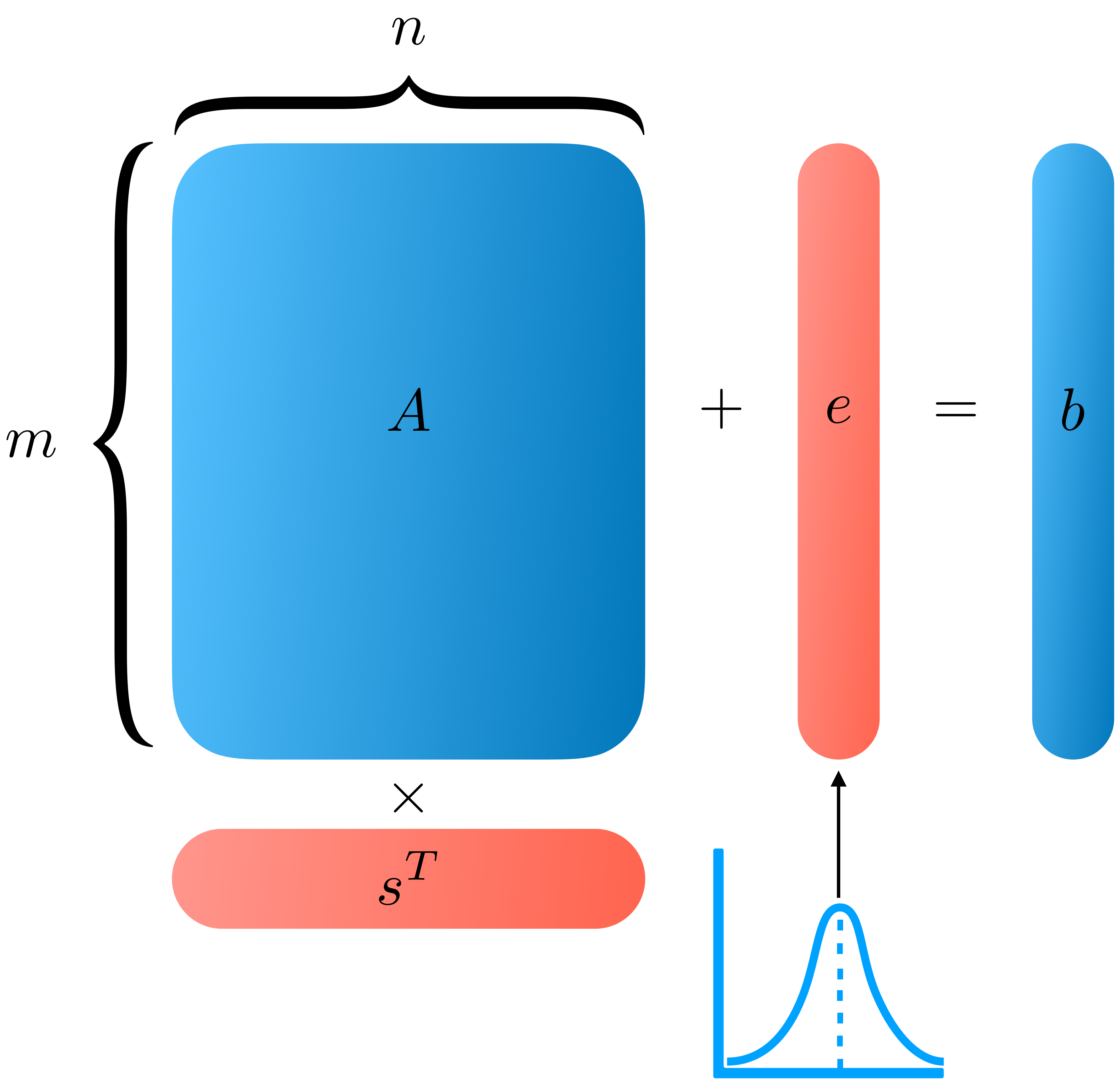
Let $e \leftarrow_R \mathcal{E}^m$

Input: $A \in \mathbb{Z}_q^{m \times n}, b \in \mathbb{Z}_q^m, \mathcal{E}$

$$b = (As + e)\ mod\ q$$

Output: $s \in \mathbb{Z}_q^n$

Learning With Errors (LWE)



Learning With Errors (LWE)

Example from O. Regev's survey paper
https://cims.nyu.edu/~regev/papers/lwesurvey.pdf

$$14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17}$$
$$13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17}$$
$$6s_1 + 10s_2 + 13s_3 + 1s_4 \approx 3 \pmod{17}$$
$$10s_1 + 4s_2 + 12s_3 + 16s_4 \approx 12 \pmod{17}$$
$$9s_1 + 5s_2 + 9s_3 + 6s_4 \approx 9 \pmod{17}$$
$$3s_1 + 6s_2 + 4s_3 + 5s_4 \approx 16 \pmod{17}$$
$$\vdots$$
$$6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17}$$

Say the error is ± 1

Learning With Errors (LWE)

Example from O. Regev's survey paper
https://cims.nyu.edu/~regev/papers/lwesurvey.pdf

$$14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 \pmod{17}$$
$$13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 \pmod{17}$$
$$6s_1 + 10s_2 + 13s_3 + 1s_4 \approx 3 \pmod{17}$$
$$10s_1 + 4s_2 + 12s_3 + 16s_4 \approx 12 \pmod{17}$$
$$9s_1 + 5s_2 + 9s_3 + 6s_4 \approx 9 \pmod{17}$$
$$3s_1 + 6s_2 + 4s_3 + 5s_4 \approx 16 \pmod{17}$$
$$\vdots$$
$$6s_1 + 7s_2 + 16s_3 + 2s_4 \approx 3 \pmod{17}$$

Say the error is ± 1

$$s = (0, 13, 9, 11)$$

Learning With Errors (LWE)

What does LWE have to do with lattices?

$$SVP_{poly(n)} \leq LWE$$

$\uparrow$

Quantum

$$\text{worst case } SVP_{poly(n)} \leq_Q \text{ average case } LWE$$

LWE can be used for many crypto applications
(PK crypto, SK crypto, signatures, hash functions etc)

$$\text{worst case } SVP_{poly(n)} \leq_Q \text{ average case } LWE \leq \text{ crypto}$$

Learning With Errors (LWE)

Public-key crypto based on LWE

$$KeyGen(1^n)$$

Samples A, s uniformly at random and e from error distribution

$$KeyGen(1^n) \to (PK, SK)$$

$$SK = s, PK = (A, b = As + e)$$

Assume message, M, is one bit

$$Enc(PK, M) \to (u, v)$$

Sample a *short* vector $r \in \{0,1\}^m$

$$u = r^T A, \quad v = \langle r, b \rangle + M \cdot \lceil q/2 \rceil$$

Learning With Errors (LWE)

$$SK = s, PK = (A, b = As + e)$$

$$Enc(PK, M) \rightarrow (u, v)$$

Sample a *short* vector $r \in \{0,1\}^m$

$$u = r^T A, \quad v = \langle r, b \rangle + M \cdot \lceil q/2 \rceil$$

$$Dec(SK, (u, v)) \rightarrow M'$$

$$M' = \begin{cases} 0, & \text{if } ||v - \langle u, s \rangle|| \leq q/4 \\ 1, & \text{otherwise} \end{cases}$$

Learning With Errors (LWE)

Some real parameters

$$n = 64, q = 251, m = 1024$$

Then A will have **65536** elements

Each element requires 8 bits to store

The public key will be at least 64KB!

Can LWE be made to have smaller keys?

Yes

Ring Learning With Errors (R-LWE)

$$R_q = \mathbb{Z}_q/(x^n + 1)$$

Polynomials of degree at most n, with coefficients mod q

$$a, b, s, e \in R_q$$

such that

$$b = a \cdot s + e$$

e is drawn from some gaussian distribution over polynomials

Find s!

$$n = 64, q = 251$$

Keys require only 64 bytes

Ring Learning With Errors (R-LWE)

worst case $SVP_{poly(n)}$ on ideal lattices $\leq_Q$ average case $R - LWE \leq crypto$

$$\mathcal{I} \subseteq \mathbb{Z}/(x^n + 1)$$

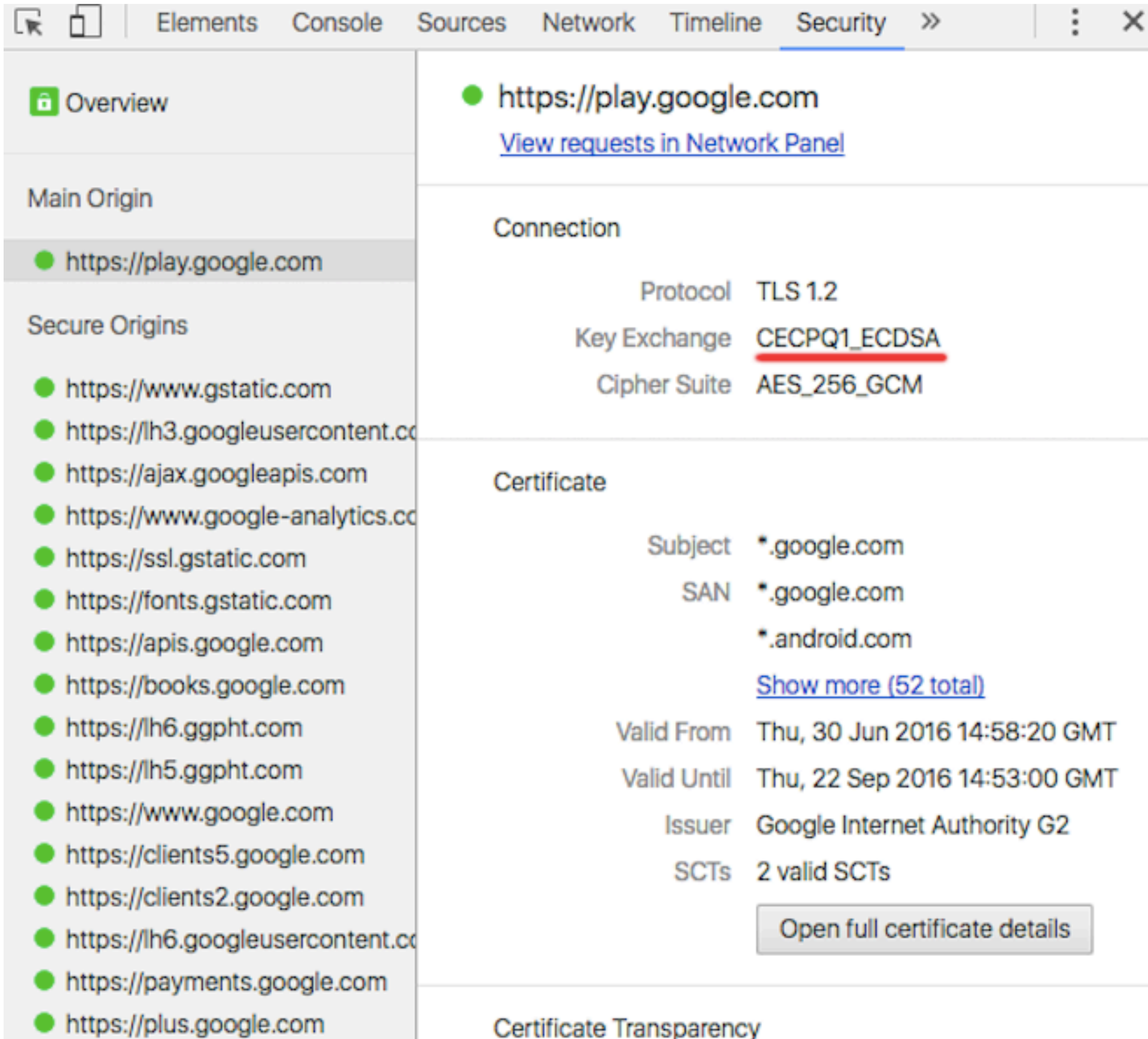
$$(\mathcal{I}, +) \text{ subgroup of } \mathbb{Z}/(x^n + 1)$$

Ideal lattice problem still seem hard

But research is ongoing

A new hope (2015)

A new hope


Elements
Console
Sources
Network
Timeline
Security
Overview
Main Origin
https://play.google.com
Secure Origins
https://www.gstatic.com
https://lh3.googleusercontent.co
https://ajax.googleapis.com
https://www.google-analytics.co
https://ssl.gstatic.com
https://fonts.gstatic.com
https://apis.google.com
https://books.google.com
https://lh6.ggpht.com
https://lh5.ggpht.com
https://www.google.com
https://clients5.google.com
https://clients2.google.com
https://lh6.googleusercontent.co
https://payments.google.com
https://plus.google.com
https://play.google.com
View requests in Network Panel
Connection
Protocol TLS 1.2
Key Exchange CECPQ1_ECDSA
Cipher Suite AES_256_GCM
Certificate
Subject *.google.com
SAN *.google.com
*.android.com
Show more (52 total)
Valid From Thu, 30 Jun 2016 14:58:20 GMT
Valid Until Thu, 22 Sep 2016 14:53:00 GMT
Issuer Google Internet Authority G2
SCTs 2 valid SCTs
Open full certificate details
Certificate Transparency

Another implementation

A version of LWE with many many optimisations

It’s plain LWE not R-LWE

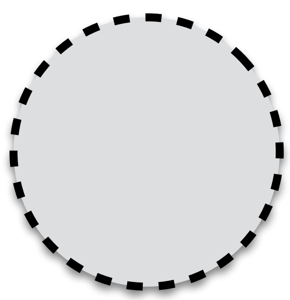
Frodo (2016)

Only 2x slower than ECDH

Fully homomorphic encryption

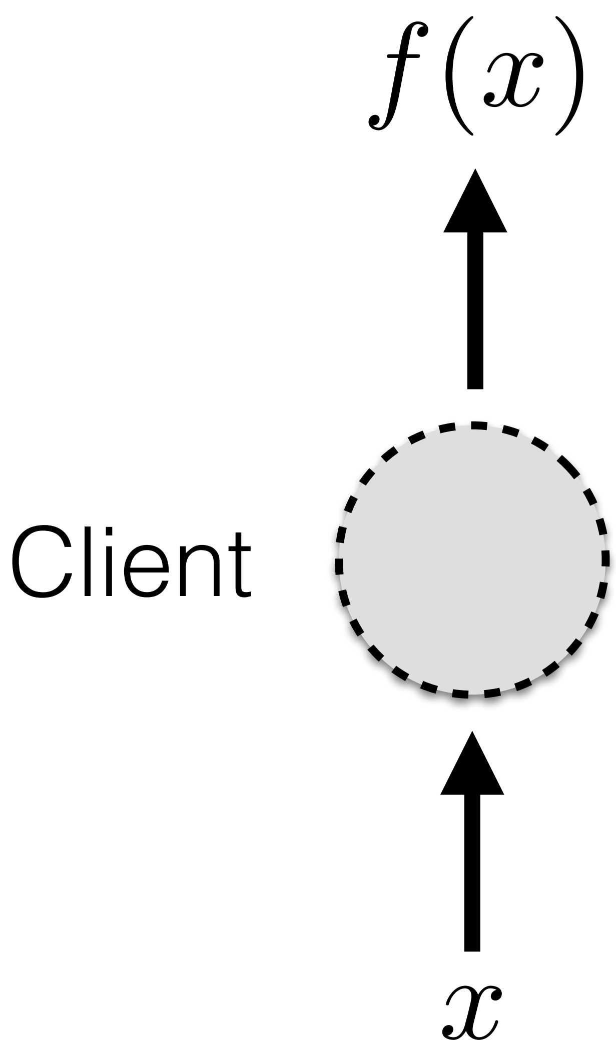
$$f : \{0,1\}^m \rightarrow \{0,1\}^n$$

Client

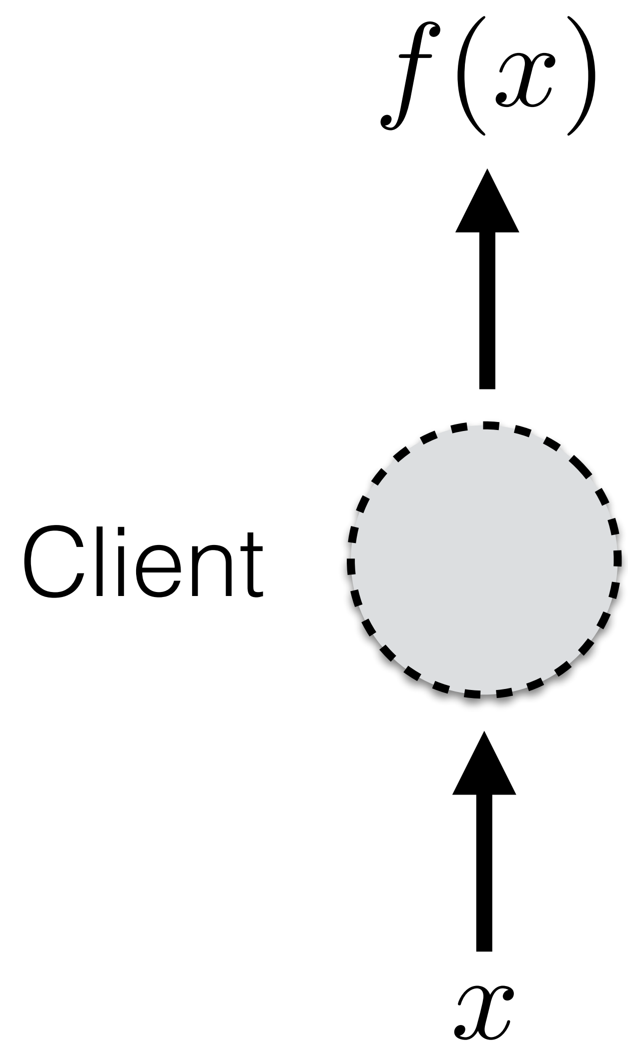


Fully homomorphic encryption

$$f : \{0,1\}^m \rightarrow \{0,1\}^n$$



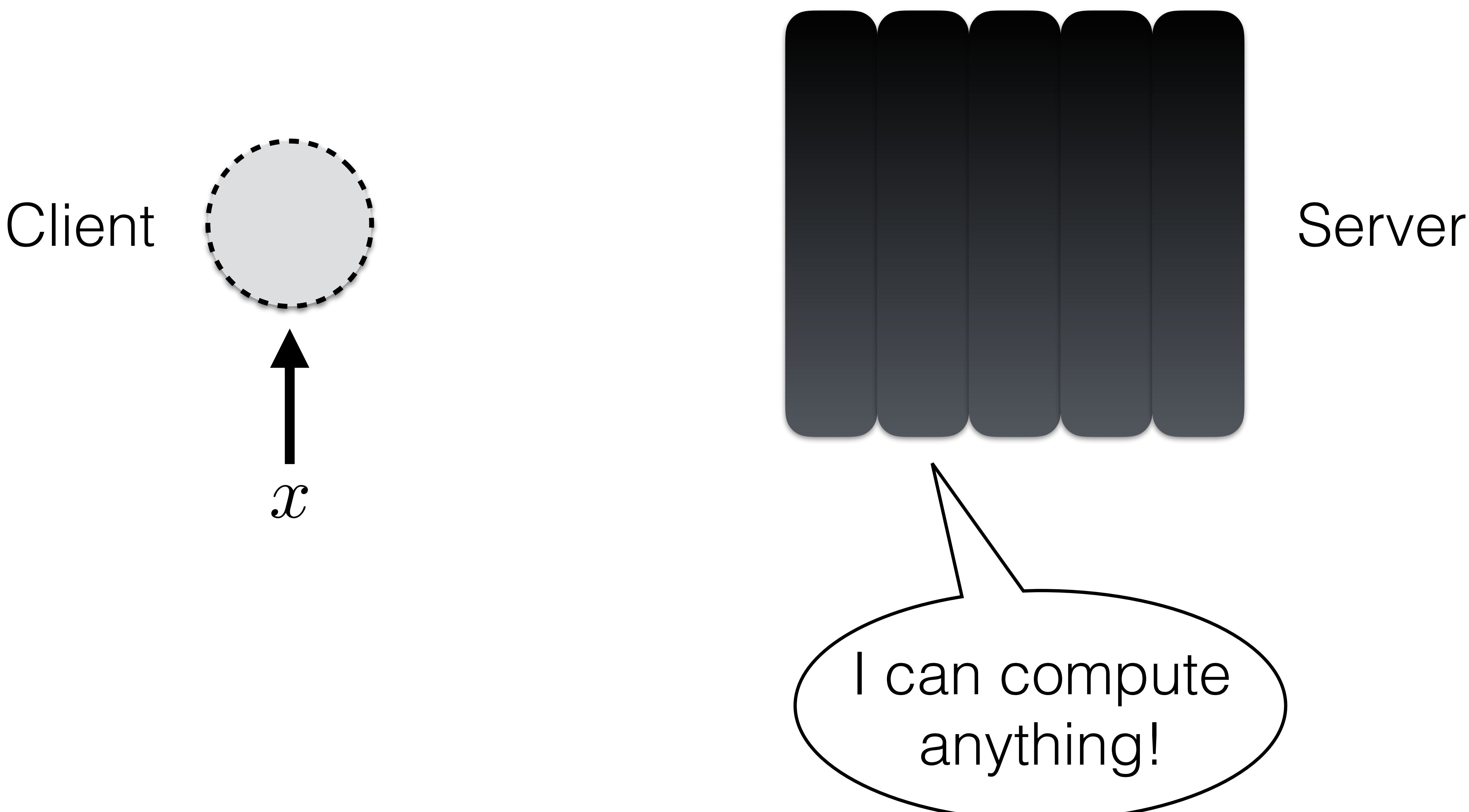
Fully homomorphic encryption



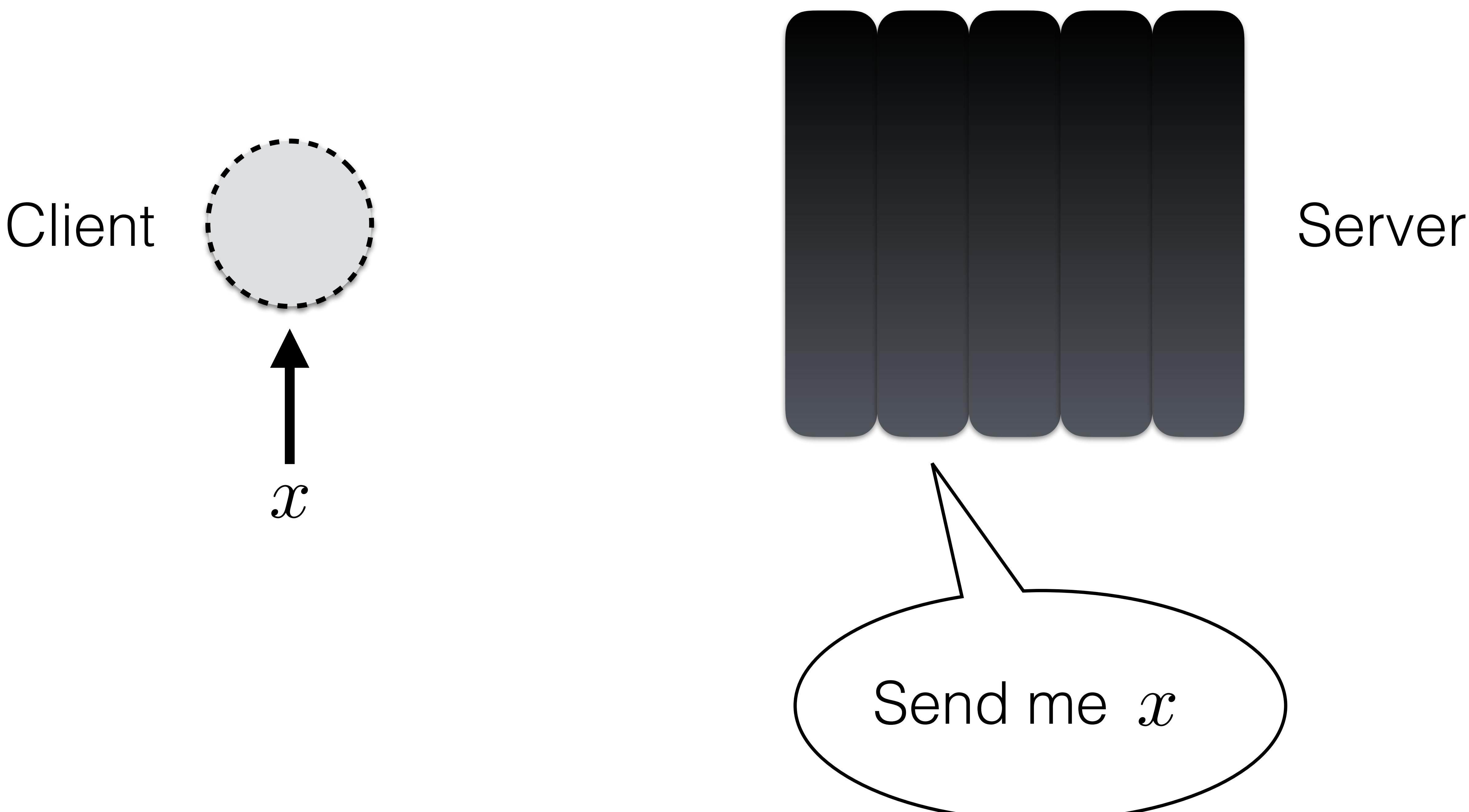
Suppose f is too hard for the client to compute

Fully homomorphic encryption

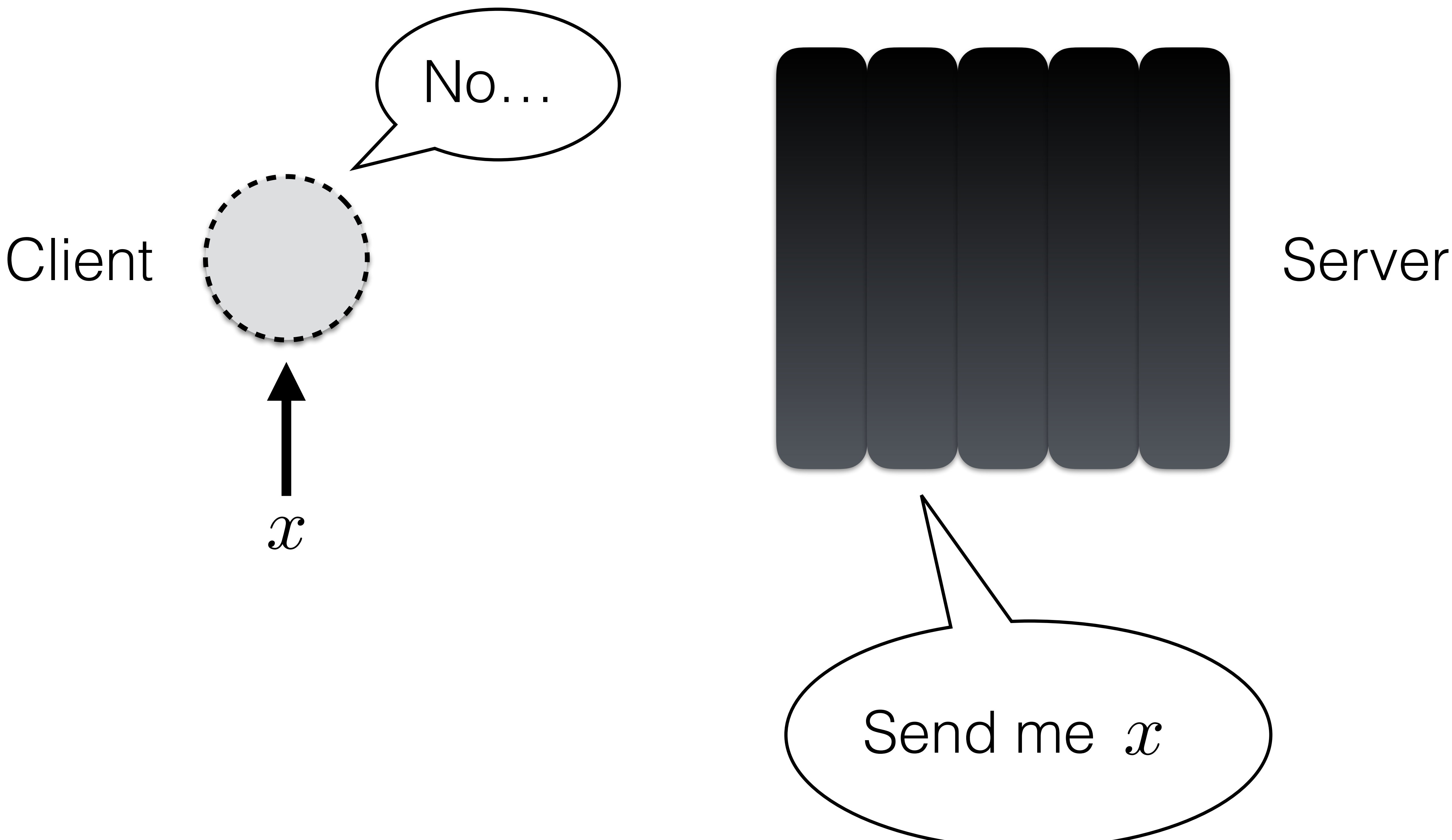
Fully homomorphic encryption



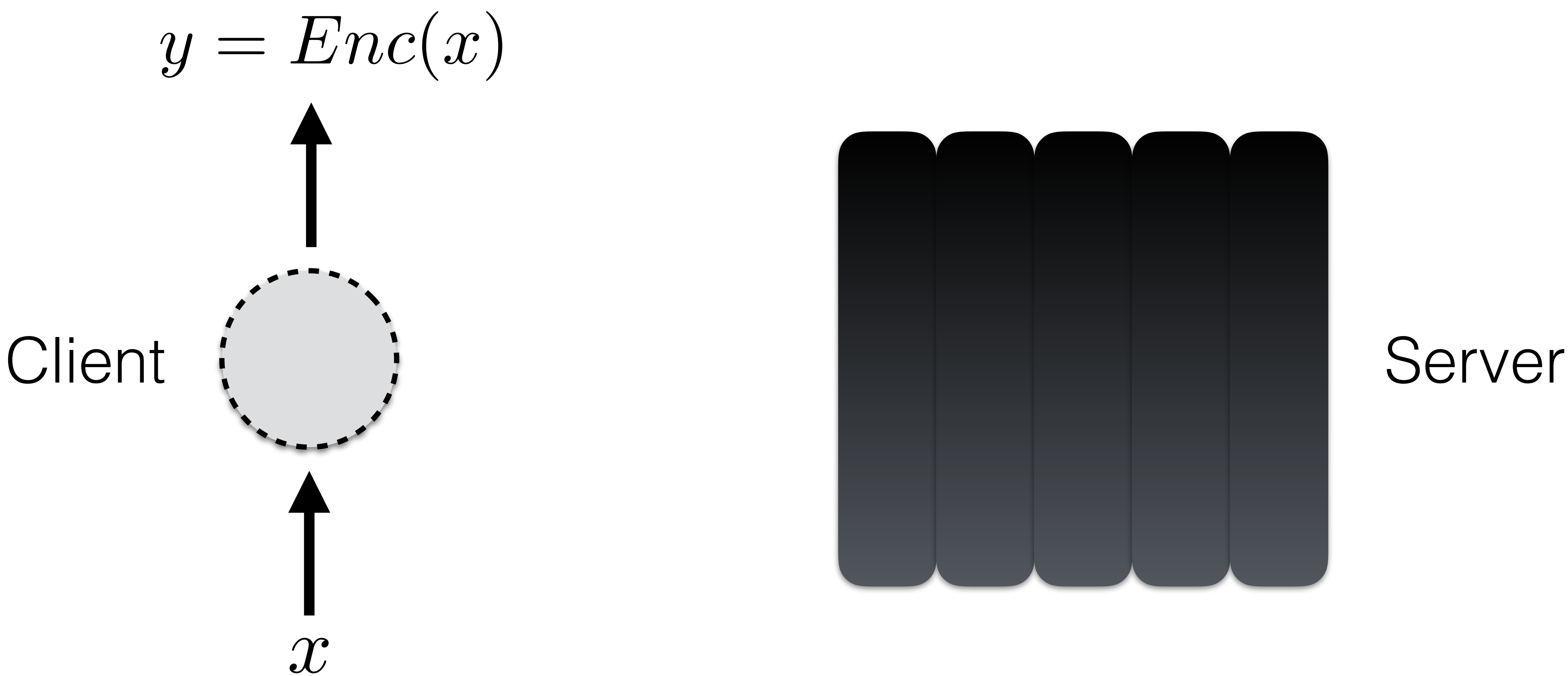
Fully homomorphic encryption



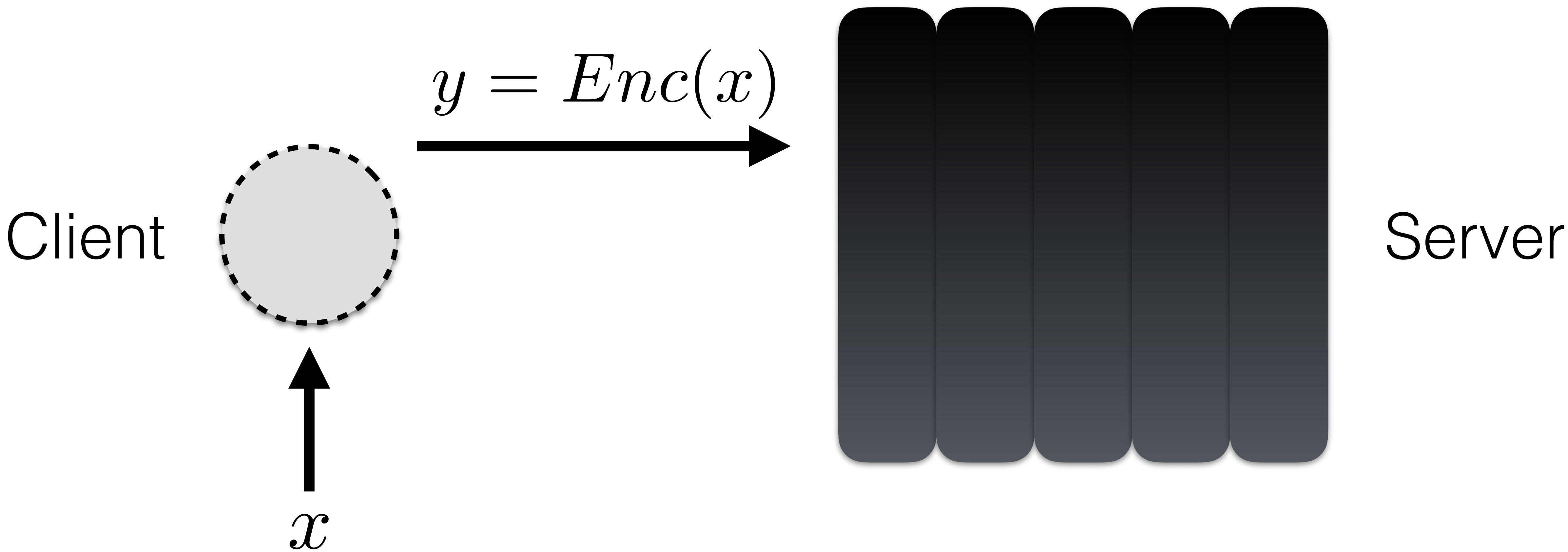
Fully homomorphic encryption



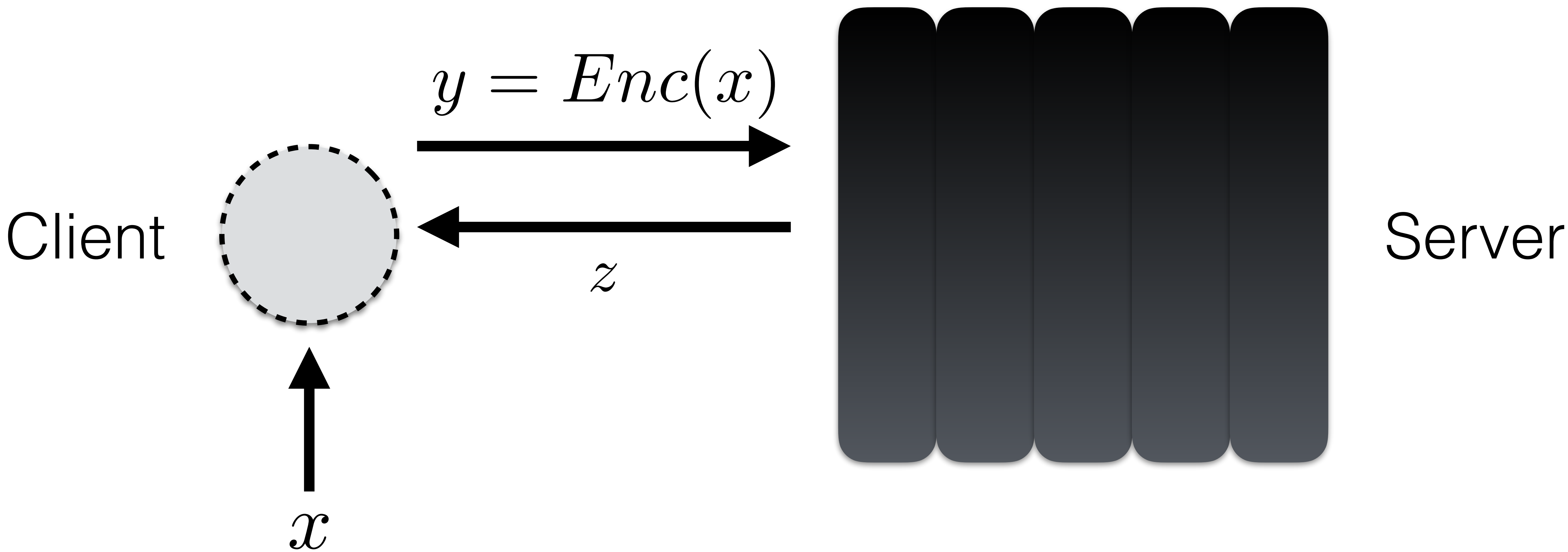
Fully homomorphic encryption



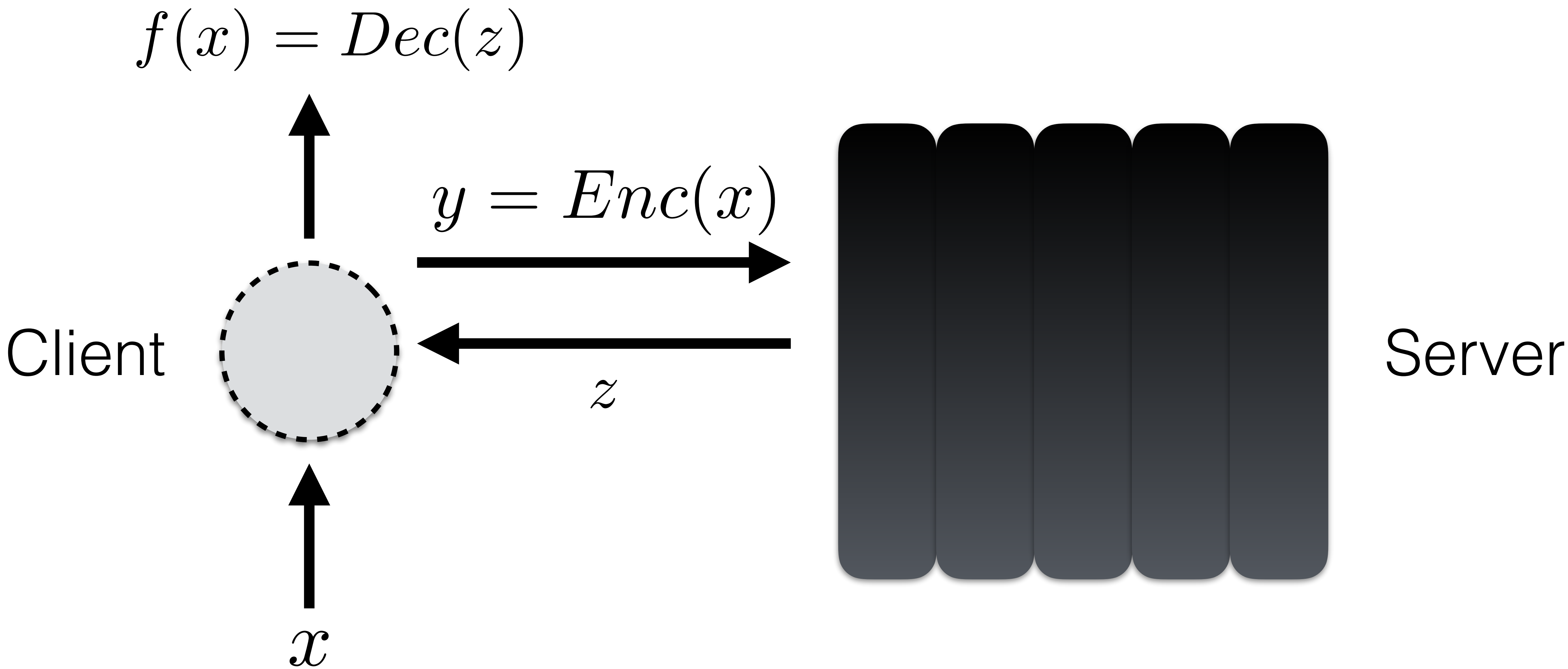
Fully homomorphic encryption



Fully homomorphic encryption



Fully homomorphic encryption



Enc, Dec should be efficient for the client

Can be done with LWE!

Efficiency of Enc, Dec independent of efficiency of f

Check out

https://github.com/shaih/HElib

References and resources

Semantic security

https://en.wikipedia.org/wiki/Semantic_security

https://lucatrevisan.wordpress.com/2009/01/22/cs276-lecture-2-semantic-security/

Crypto references

http://theory.stanford.edu/~trevisan/books/crypto.pdf

https://www.amazon.com/Introduction-Modern-Cryptography-Principles-Protocols/dp/1584885513

https://crypto.stanford.edu/~dabo/cryptobook/

Scott Aaronson's survey on P vs NP

https://www.scottaaronson.com/papers/pnp.pdf

References and resources

Complexity and quantum computing

https://www.scottaaronson.com/democritus/lec10.html

Lattice problems and LWE

https://www.youtube.com/watch?v=FVFw_qb1ZkY

https://www.youtube.com/watch?v=Fp-IiVpgDlc

https://cims.nyu.edu/~regev/papers/qcrypto.pdf

https://en.wikipedia.org/wiki/Learning_with_errors

Reductions and crypto protocols based on LWE

https://people.csail.mit.edu/vinodv/6876-Fall2015/L13.pdf

Fully homomorphic encryption

https://www.youtube.com/watch?v=O8IvJAIvGJo

https://en.wikipedia.org/wiki/Homomorphic_encryption

https://crypto.stanford.edu/craig/craig-thesis.pdf

https://people.csail.mit.edu/vinodv/6876-Fall2015/L14.pdf