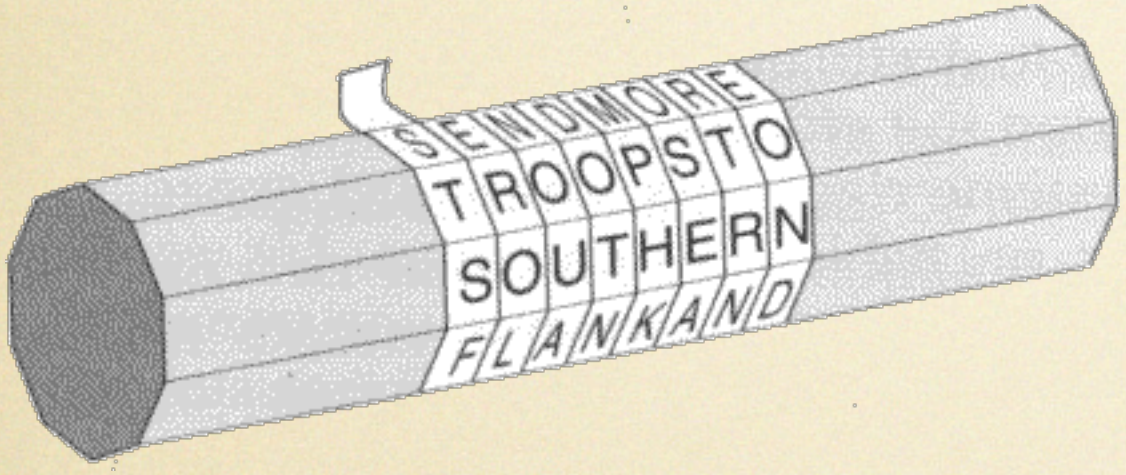
Cryptography Research
Directions and Challenges

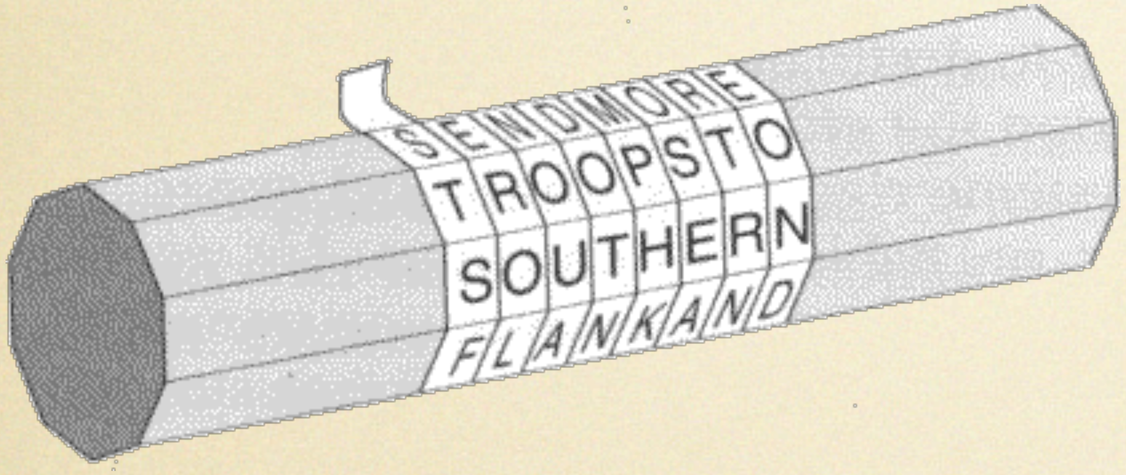
Aggelos Kiayias
University of Athens

Cryptography?

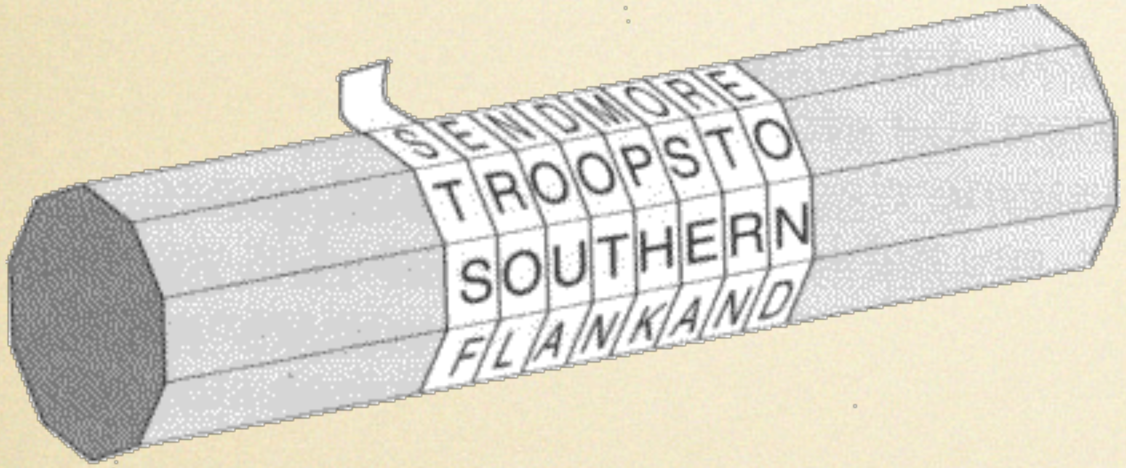
Cryptography?



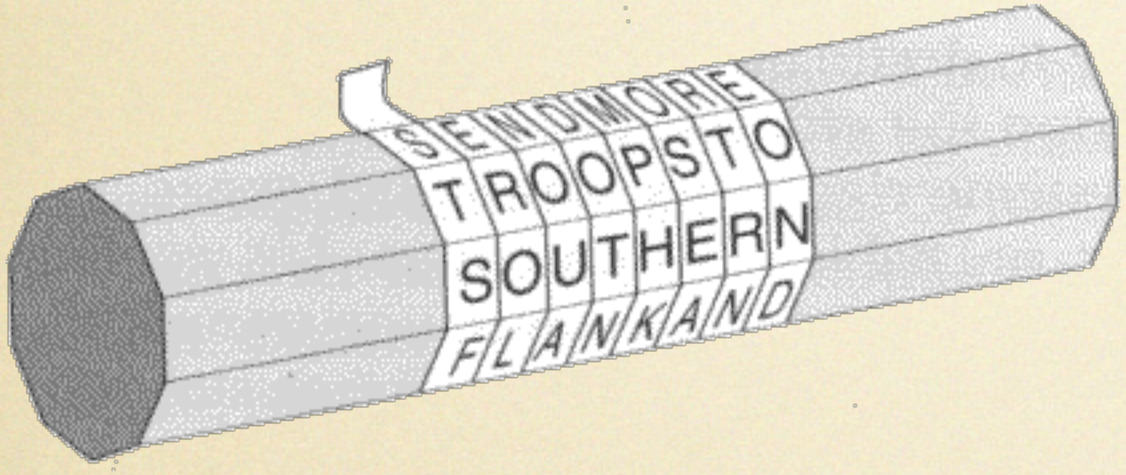
Cryptography?



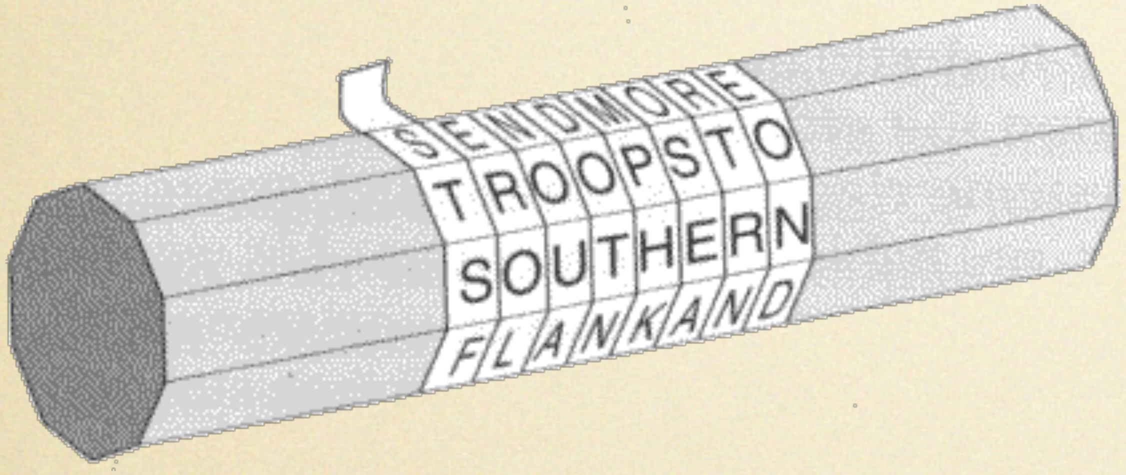
Cryptography?

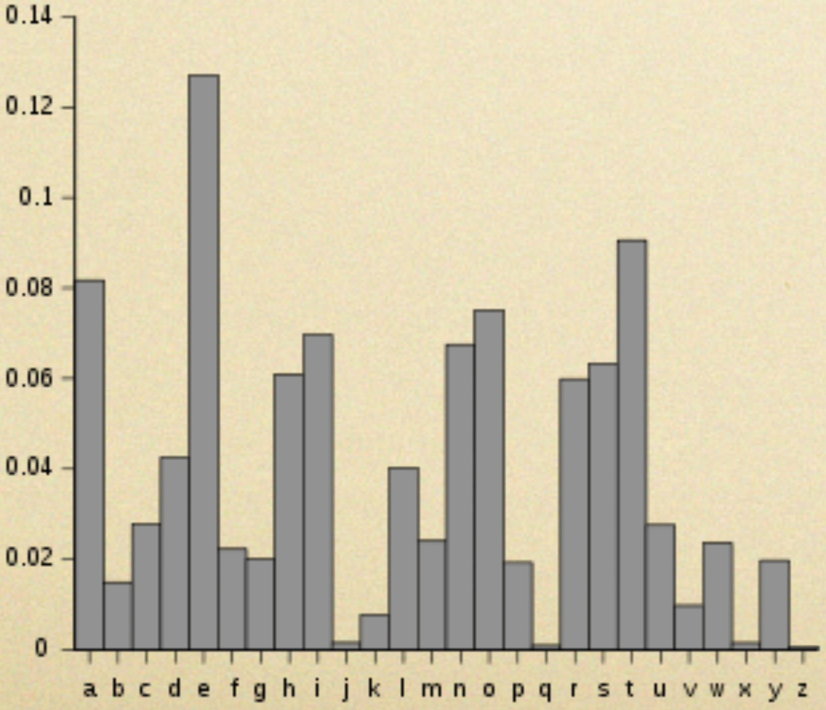


Cryptography?

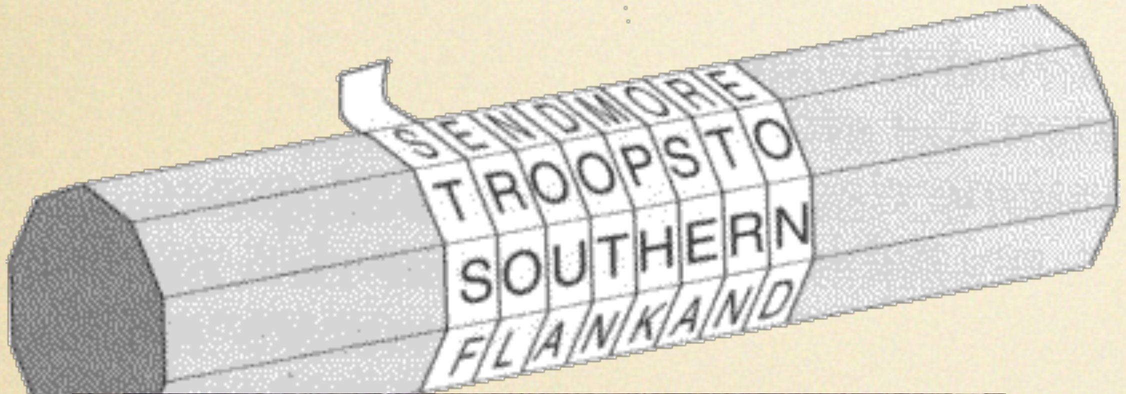


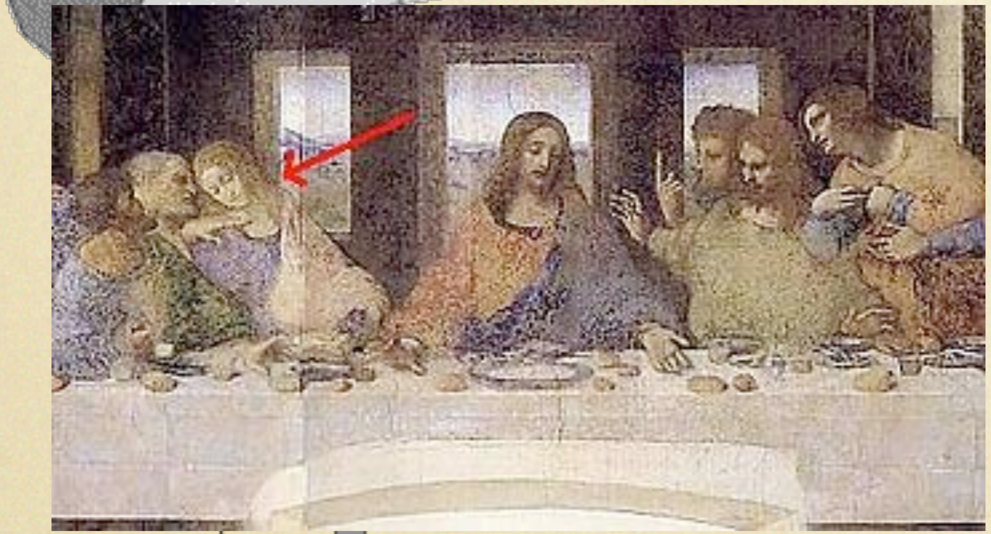
Cryptography?

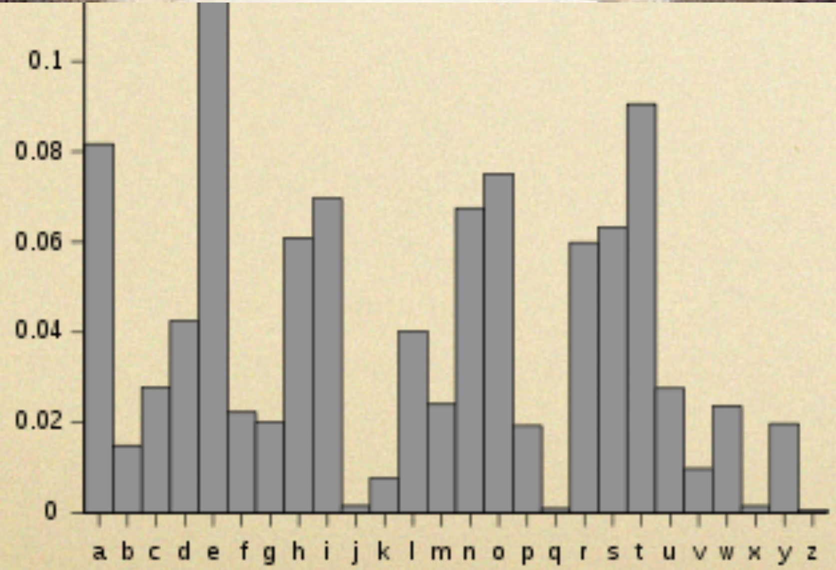




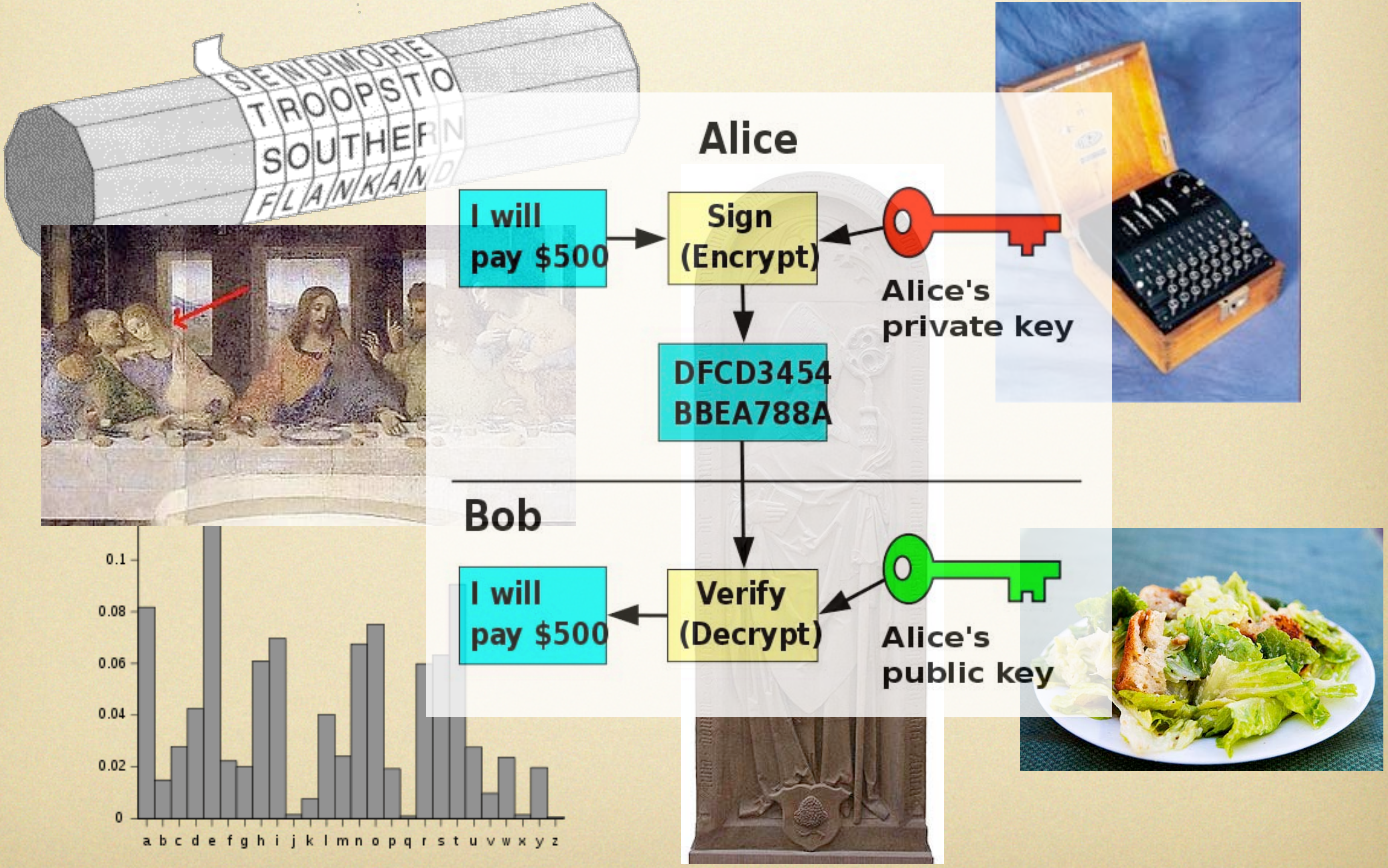
Cryptography?







Cryptography?



What is Cryptography?

A **r** t

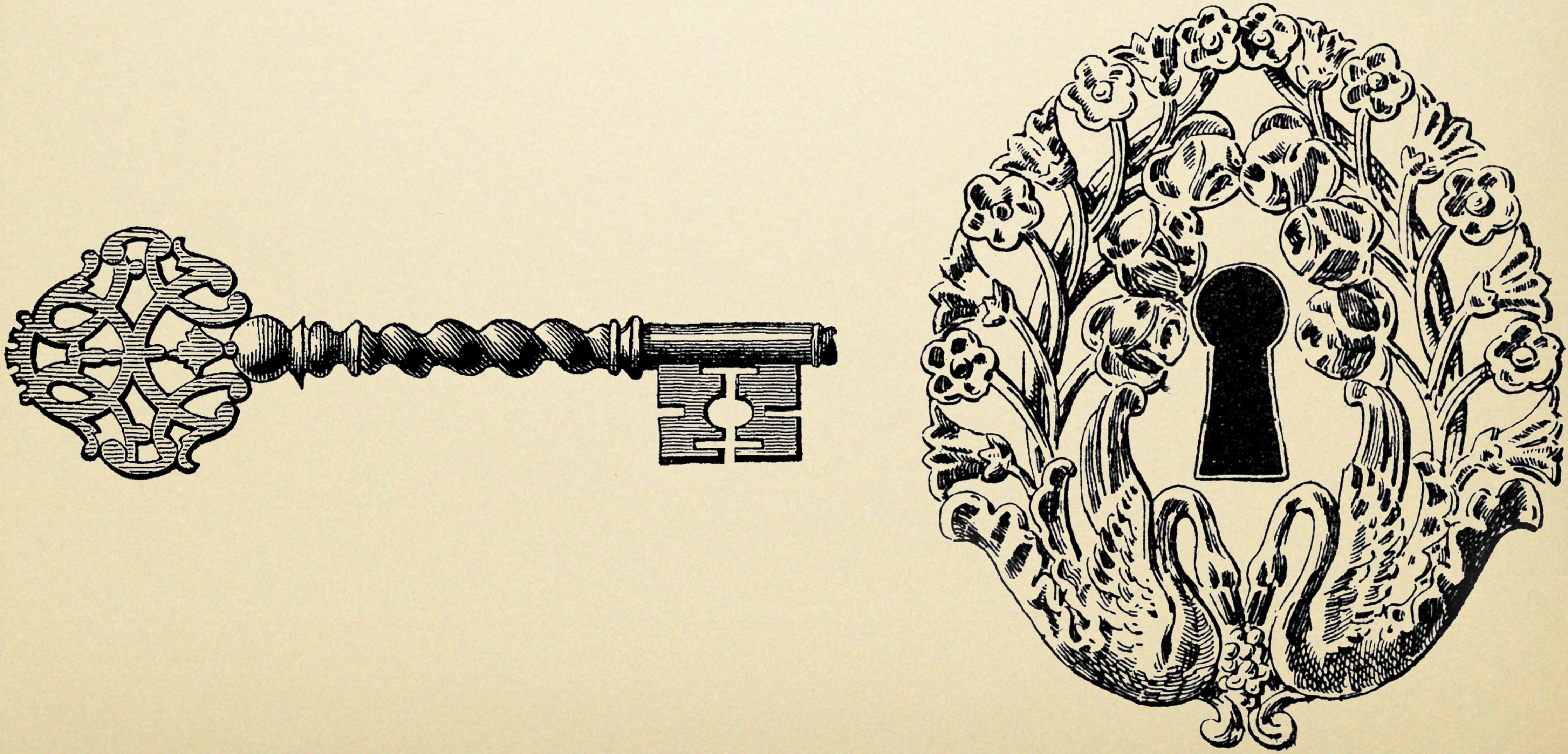
of secret writing

Wh t is C ptograp ?

A **r** t

of s t writi

Cryptography **reincarnated**



General Setting

- Consider a set of parties (>1)
  - Each may have some *input*.
  - Each wishes to a sample a specific *output distribution / functionality*.
  - They can communicate following some prescribed *mode of interaction*.

Modeling

- The parties' strategies are algorithmic.
- The course of their interaction is mediated by an external controller.

Adversity

- Parties can turn adversarial and may:
  - Engage in additional non-prescribed interactions between them.
  - Follow different algorithmic strategies.
  - Refuse to participate.

Adversity vs. Trust

- Total honesty is rare (and uninteresting)
- Total adversity is rare (and uninteresting)
- More common / interesting : a mixture of adversity and honesty subject to a certain trust configuration.
  - Note : honest parties' expectations may change depending on the level of adversity.

Example: Fair Exchange of Secrets

secretA

secretB

Alice

Bob

secretB

secretA

Trust Configuration

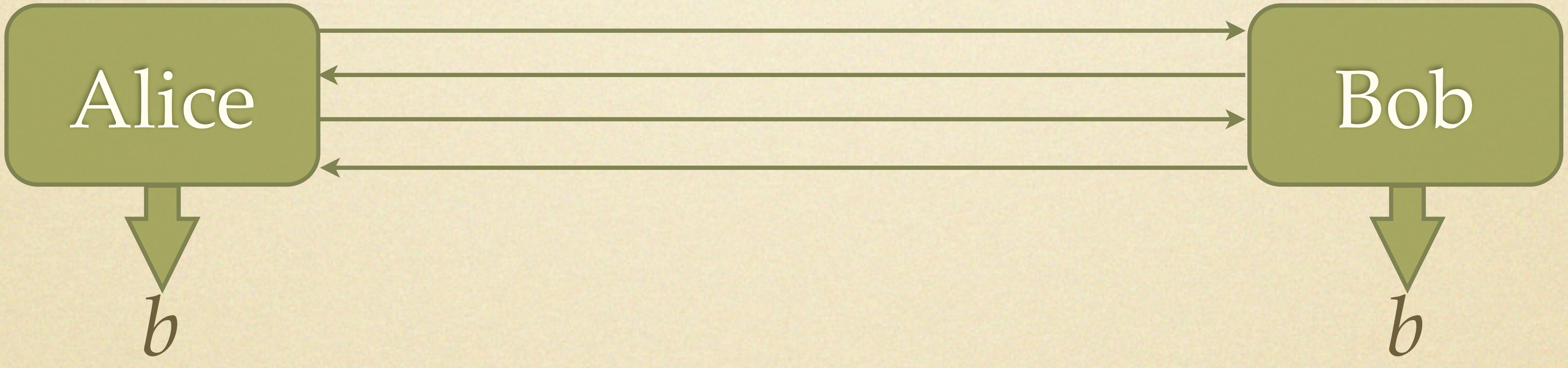
- Alice and Bob can both write messages to each that are delivered.
- If Alice is adversarial, there is no way she obtains output before Bob obtains output.
- If Bob is adversarial, there is no way he obtains output before Alice obtains output.

Trust Configuration

- Alice and Bob can both write messages to each that are delivered.
- If Alice is adversarial, there is no way she obtains output before Bob obtains output.
- If Bob is adversarial, there is no way he obtains output before Alice obtains output.

Observe: this *is* a cryptographic problem - but it has no obvious reliance of encryption or signatures.

Example: Coin Flipping



Alice

Bob

b

b

b is a uniformly distributed bit

Trust Configuration

- Alice and Bob can both write messages to each that are delivered.
- If Alice is adversarial, there is no way she can bias Bob’s output.
- If Bob is adversarial, there is no way he can bias Alice’s output.

Trust Configuration

- Alice and Bob can both write messages to each that are delivered.
- If Alice is adversarial, there is no way she can bias Bob’s output.
- If Bob is adversarial, there is no way he can bias Alice’s output.

Observe: again this *is* a cryptographic problem - but it has no obvious secrecy requirements

the cryptographic problem

- Consider
  (1) a functionality of interest.
  (2) a certain trust configuration.
- **Prove a theorem stating that :** honest parties can reach successfully the evaluation of the functionality given the trust configuration, in spite the presence of adversity.

3 important cryptographic objectives

3 important cryptographic objectives

- Depending on the occasion we may wish to ensure that adversity will not disrupt:

3 important cryptographic objectives

- Depending on the occasion we may wish to ensure that adversity will not disrupt:
    - Integrity. the ability of honest parties to obtain their (properly distributed) output.

3 important cryptographic objectives

- Depending on the occasion we may wish to ensure that adversity will not disrupt:
  - Integrity. the ability of honest parties to obtain their (properly distributed) output.
  - Secrecy. the honest parties private inputs will remain hidden from the adversaries.

3 important cryptographic objectives

- Depending on the occasion we may wish to ensure that adversity will not disrupt:
    - Integrity. the ability of honest parties to obtain their (properly distributed) output.
    - Secrecy. the honest parties private inputs will remain hidden from the adversaries.
    - Fairness. the honest parties are denied output while the adversarial ones do obtain.

3 important cryptographic objectives

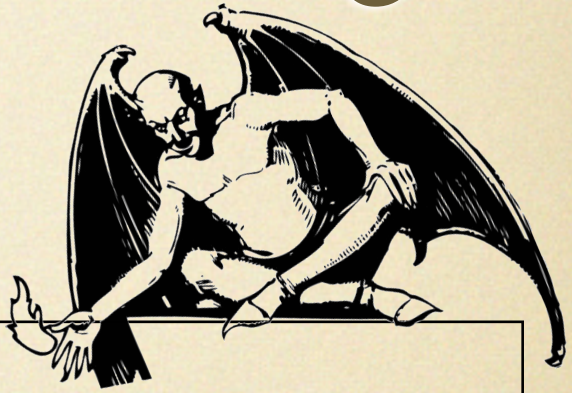
- Depending on the occasion we may wish to ensure that adversity will not disrupt:
  - Integrity. the ability of honest parties to obtain their (properly distributed) output.
  - Secrecy. the honest parties private inputs will remain hidden from the adversaries.
  - Fairness. the honest parties are denied output while the adversarial ones do obtain.

3 important cryptographic objectives

there are more!

- Depending on the occasion we may wish to ensure that adversity will not disrupt:
  - Integrity. the ability of honest parties to obtain their (properly distributed) output.
  - Secrecy. the honest parties private inputs will remain hidden from the adversaries.
  - Fairness. the honest parties are denied output while the adversarial ones do obtain.

Formalizing Security



- The simulation paradigm:
  - prove that **the whole view** of the adversaries *can be simulated* without access to resources that are unavailable to adversarial parties.

cryptography ...redefined

cryptography …redefined

Cryptography *is* a CS discipline that applies mathematics/ statistics, algorithms and computational complexity **to solve problems of trust** between two or more parties.

Cryptographic Proofs

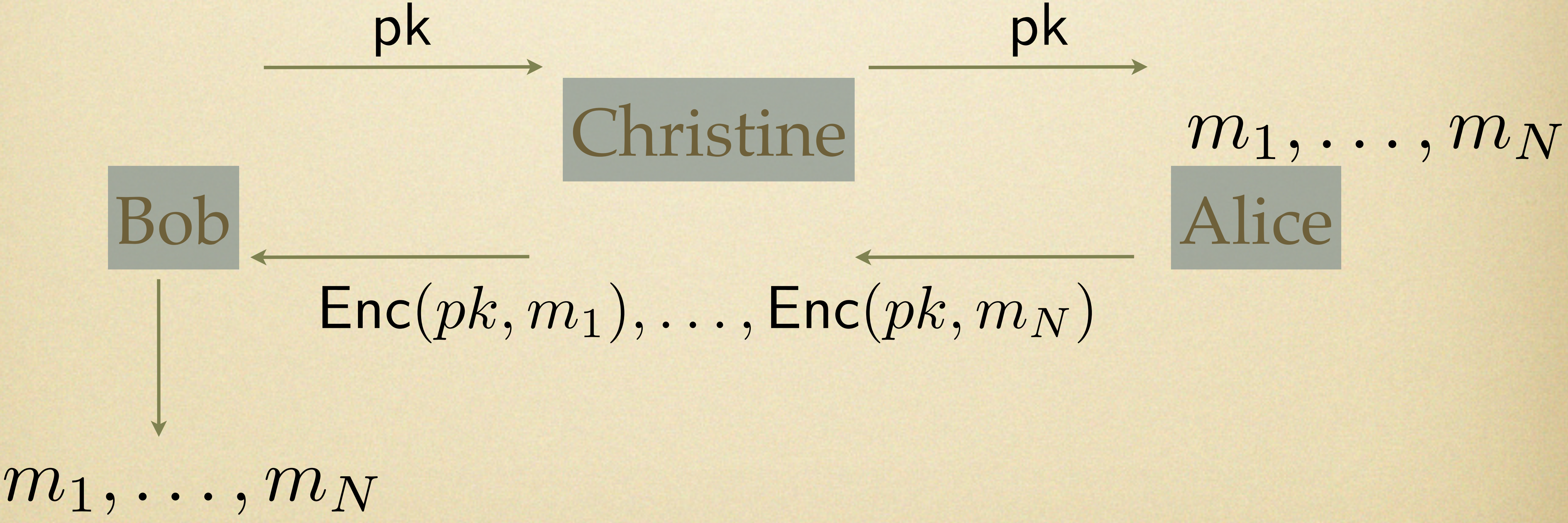
Example: a secure channel.

- Three parties: Alice, Bob, Christine.
- Mode of interaction : Alice wishes to send an unlimited number of private messages to Bob. The only way to communicate is through Christine.
- Trust model : Christine will always deliver Alice and Bob's messages but she cannot be trusted not to read them.

Using PK Encryption

KeyGen, Enc, Dec

$\text{KeyGen} \rightarrow (\mathsf{pk}, \mathsf{sk})$



pk

pk

Christine

$m_1, \ldots, m_N$

Bob

Alice

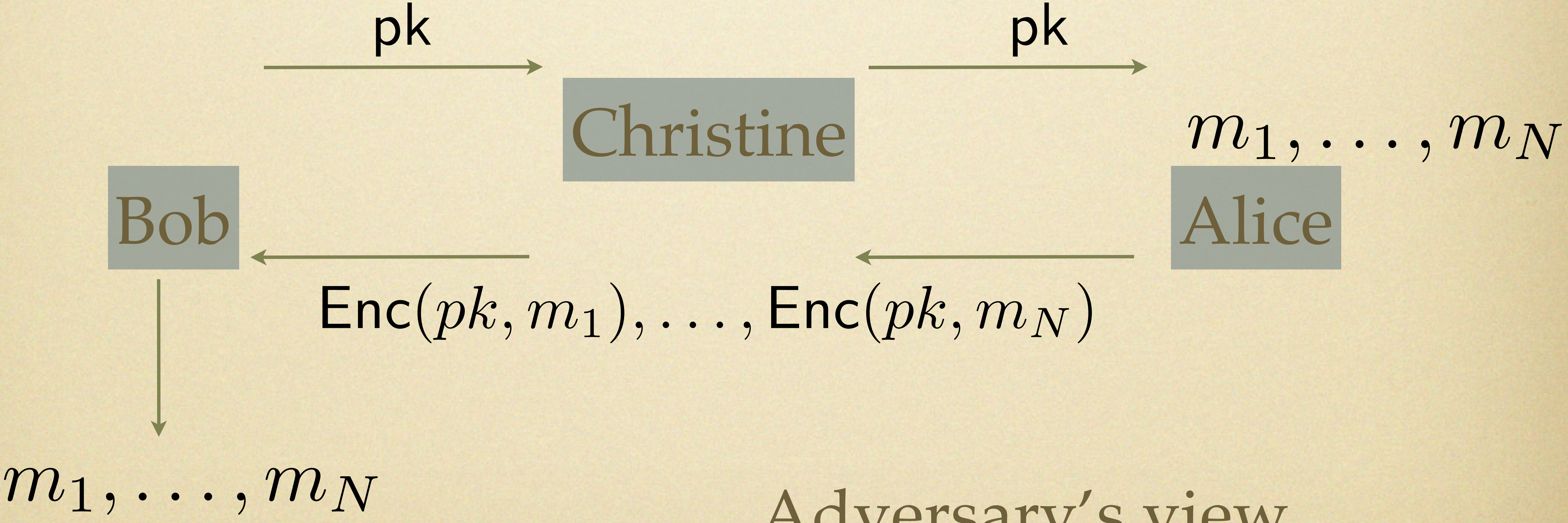
$\mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_N)$

$m_1, \ldots, m_N$

Using PK Encryption

KeyGen, Enc, Dec

$\text{KeyGen} \rightarrow (\mathsf{pk}, \mathsf{sk})$



pk

pk

Christine

$m_1, \ldots, m_N$

Bob

Alice

$\mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_N)$

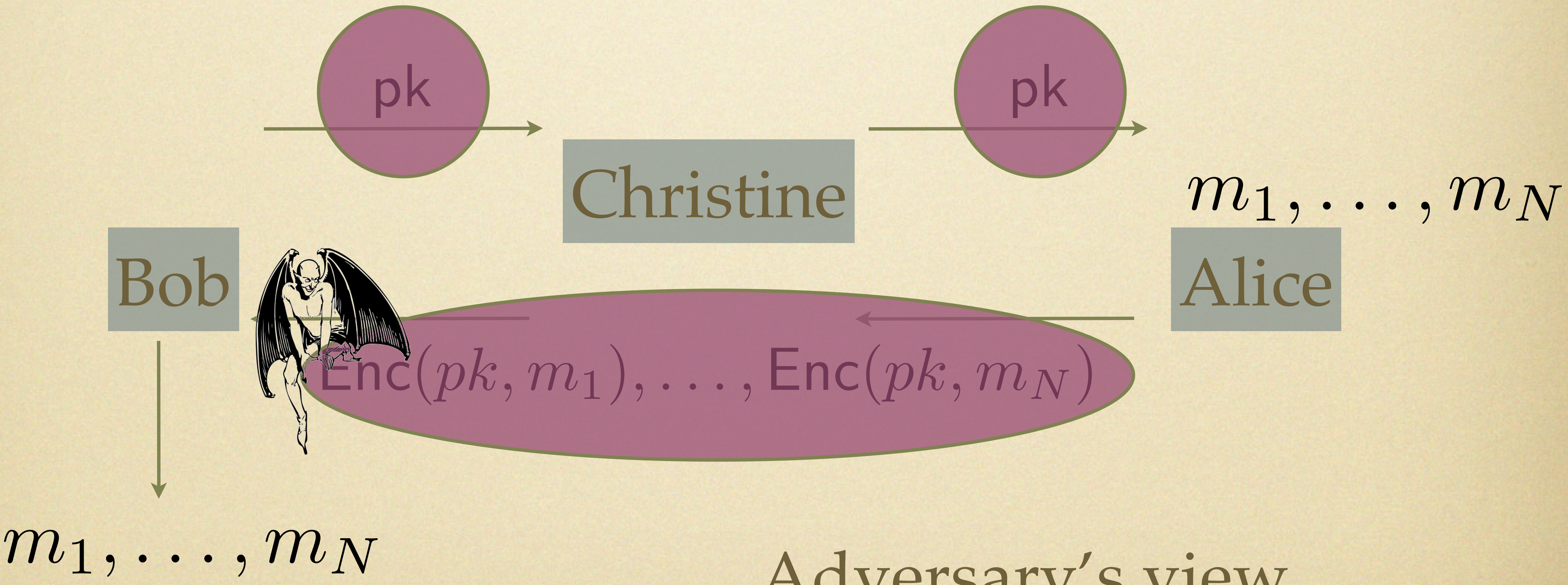
$m_1, \ldots, m_N$

Adversary's view

Using PK Encryption

KeyGen, Enc, Dec

$\text{KeyGen} \to (\mathsf{pk}, \mathsf{sk})$



pk

pk

Christine

$m_1, \ldots, m_N$

Bob

Alice

$\mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_N)$

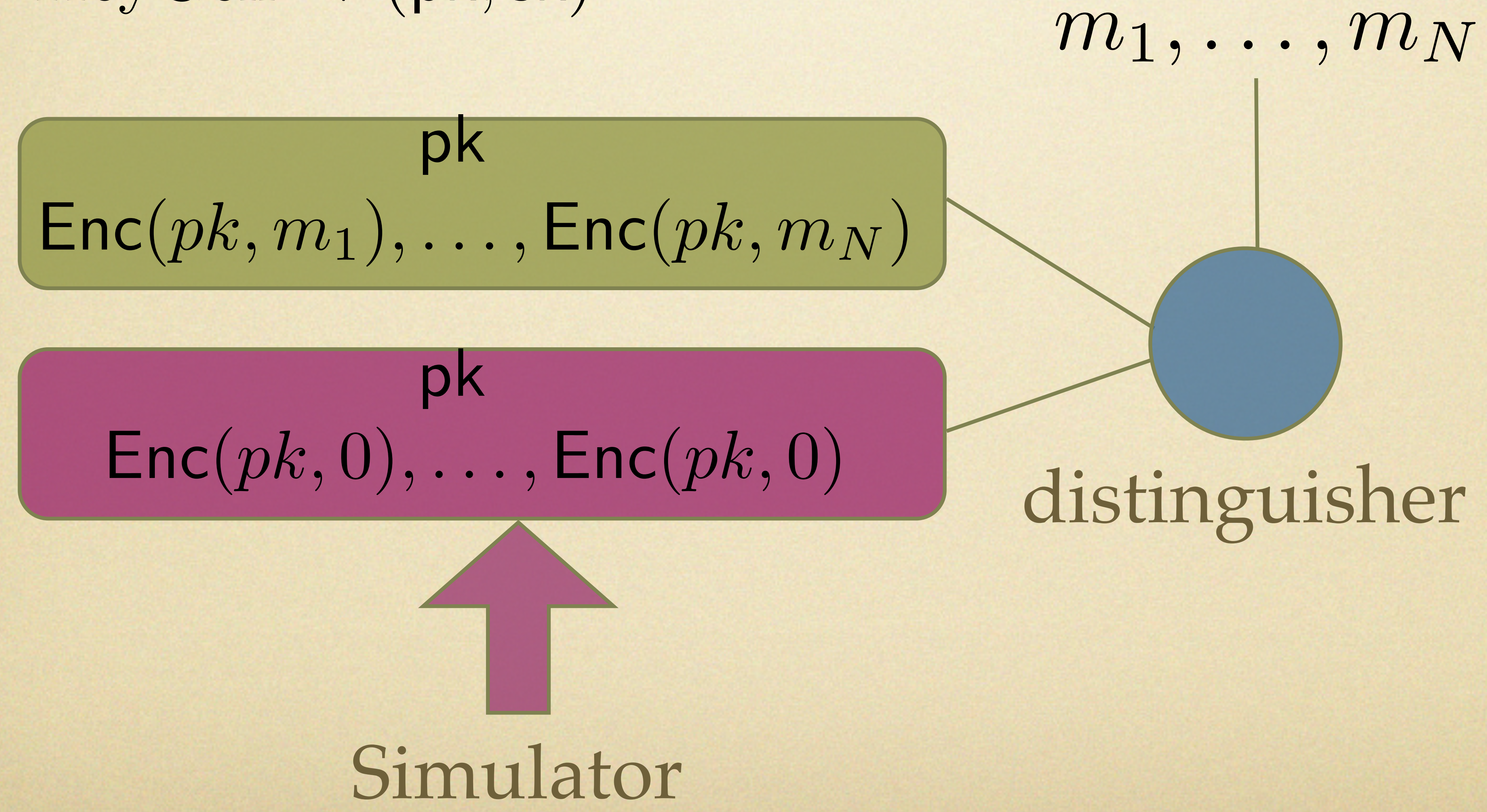
$m_1, \ldots, m_N$

Adversary's view

Using PK Encryption

KeyGen, Enc, Dec

$\mathrm{KeyGen} \rightarrow (\mathsf{pk}, \mathsf{sk})$



$m_1, \ldots, m_N$

pk

$\mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_N)$

pk

$\mathsf{Enc}(pk, 0), \ldots, \mathsf{Enc}(pk, 0)$

distinguisher

Simulator

Hybrid Argument

pk

$\langle \mathsf{Enc}(pk, m_1), \ldots, \mathsf{Enc}(pk, m_i), \mathsf{Enc}(pk, m_{i+1}), \ldots, \mathsf{Enc}(pk, m_N)$

pk

$\langle \mathsf{Enc}(pk, 0), \ldots, \mathsf{Enc}(pk, 0), \mathsf{Enc}(pk, m_{i+1}), \ldots, \mathsf{Enc}(pk, m_N) \rangle$

Any distinguishing advantage ε between the extremes will translate to a distinguishing advantage of ε/N between hybrids, something that yields a ciphertext distinguisher:

$$\langle m, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m) \rangle \approx \langle m, \mathsf{pk}, \mathsf{Enc}(pk, 0) \rangle$$

Trapdoor Functions

Trapdoor One Way Function

ParGen

$\langle e, d \rangle$

$$f_e : \{0,1\}^n \to Y$$

$$f_d : Y \to \{0,1\}^n$$

"trapdoorness" $\forall x : f_d(f_e(x)) = x$

"one-wayness" $Pr[A(f_e(x) = x] = \mathsf{negl}$

RSA

$$f_{e,N}(x) = x^e \bmod N$$

$$f_{d,N}(y) = y^d \bmod N$$

$$e \cdot d = 1 \bmod \phi(N)$$

GPV

$\langle \mathbf{A}, \mathbf{S} \rangle$ short basis for / orthogonal lattice

$$f_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}$$

Hardcore Bits

(for any one-way function)

random mapping :

$r \in \{0,1\}^n \quad \langle e, r, x \rangle \to \langle e, r, f_e(x) \rangle$

Hardcore Bit

$$B(r, x) = r \odot x = \sum_{i=1}^{n} r_i \cdot x_i \bmod 2$$

Goldreich-Levin Theorem. Given an oracle to B that works with probability $1/2 + \epsilon$ *f* can be inverted with probability 1/2 in time

$$O(n^3 \epsilon^{-4})$$

Realizing PK Encryption

$\langle e, d \rangle$: public-key and secret-key

Encryption of a bit m :

$\langle r, f_e(x), (r \odot x) \oplus m \rangle$

Decryption of a ciphertext $\langle r, y, c \rangle$

$c \oplus (f_d(y) \odot r)$

Security Proof, 1

$$\langle m, \mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m)\rangle \approx \langle m, \mathsf{pk}, \mathsf{Enc}(pk, 0)\rangle$$

$$\langle m, e, r, f_e(x), (r \odot x) \oplus 1\rangle \approx \langle m, e, r, f_e(x), (r \odot x)\rangle$$

Observe that the existence of a distinguisher between the two distributions can be used to build a predicate *B* guessing the hardcore bit.

E.g. , if *D* biases to the left with distance ε, then

$$D(m, e, r, y, b) \oplus b$$

predicts the hardcore bit

Security Proof, 2

Given a distinguisher for the simulation of N messages with advantage α

hybrid argument

We obtain a ciphertext distinguisher with probability α/N

A ciphertext distinguisher yields a hardcore bit predictor with α/N

G-L theorem

An algorithm inverting f running in time $O(n^3 N \alpha^{-1})$

Parameterization

- Suppose we want "security" of 80 bits and the ability to send up to 2^{20} messages.
- Suppose that the best algorithm inverting *f* has time-complexity $2^{\sqrt{n}}$

Then we should choose parameters:

$$3 \log n + 20 + 80 < \sqrt{n}$$

so that our reduction complexity becomes less than the best algorithm and hence impossible

$n \approx 20436$ bits

QUESTION #1

Tight Reductions

- Most reductions of relevant constructions are non-tight.
- Obtaining lower bound arguments on tightness is an open question in most cases.

Possible Targets

- Building Public-Key encryption from a given trapdoor function.
- Building Digital Signatures and PRG's from a given one-way function.
    - even for specific assumptions : e.g., obtain Public-Key encryption under RSA in the standard model

QUESTION #2

Trapdoor Functions

- We showed that trapdoor functions imply public-key encryption.
  Security was shown in the "indistinguishability" sense.
  - Reverse question is open : does secure public-key encryption imply trapdoor functions? [BHSV98] show in RO model.
  - Other examples of trapdoor functions?

QUESTION #3

Versatile Encryption

In a typical encryption correctness is supposed to work as follows:

$$\forall m : \mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$$

In **versatile encryption** we have the ability to generate secret-keys such that:

$$\forall V, m : \mathsf{Dec}(sk_V, \mathsf{Enc}(pk, m)) = V(m)$$

A Trivial Solution

Consider $V_1, \ldots, V_n$ functions

$(pk_1, sk_1), \ldots, (pk_n, sk_n)$

$\mathsf{Enc}(pk, m) = \langle \mathsf{Enc}(pk_i, V_i(m)) \rangle_{i=1}^{n}$

Note that with homomorphic encryption we can transform $\mathsf{Enc}(pk, m)$ to $\mathsf{Enc}(pk, V(m))$

However it is unclear how to obtain the appropriate secret-keys.

QUESTION #4

Broadcast Encryption

$\langle pk, sk_1, \ldots, sk_n \rangle$

$\mathsf{Enc}(pk, m, R)$ $R \subseteq \{1, \ldots, n\}$

is decryptable only by the set

$\{1, \ldots, n\} \setminus R$

Currently unknown how to obtain sublinear parameters (only known constant ciphertext schemes are based on elliptic curves)

Anonymous Broadcast Encryption is also open.

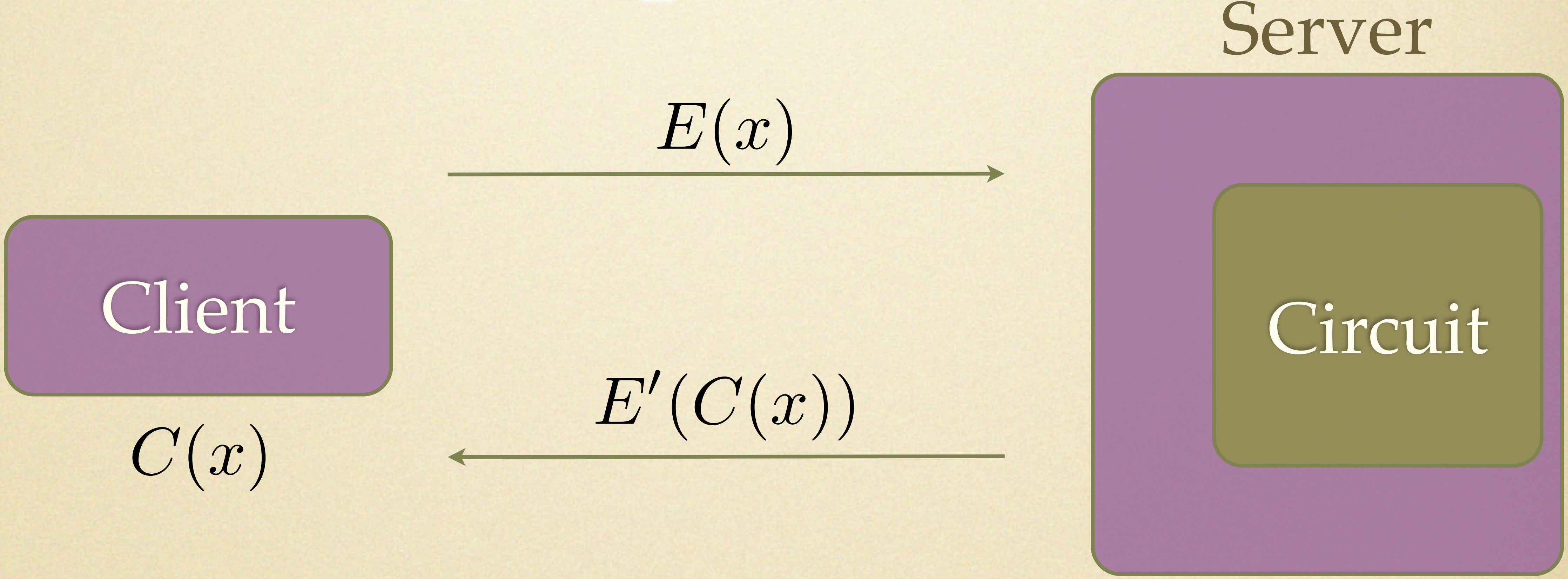
QUESTION #5

Verifiable Computation

- Can you *delegate* computation to a server so that :

1. The server cannot cheat you.
2. The server cannot learn your data.

How to delegate computation

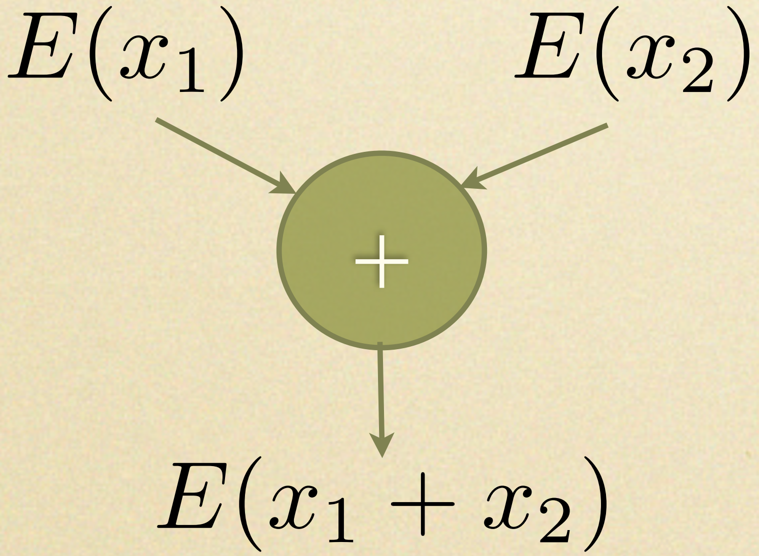


The client wants to ensure that the server performs the computation properly (without repeating the computation). + *overall communication should be* $O(|x| + |C(x)|)$

Fully Homomorphic Encryption

Gentry'09

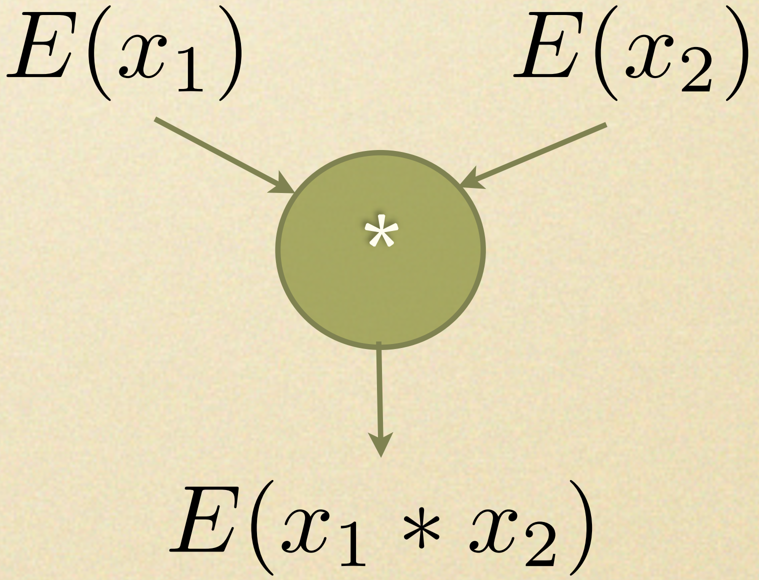
- A type of public-key encryption that allows oblivious computation over ciphertext



$E(x_1)$ $E(x_2)$

$+$

$E(x_1 + x_2)$



$E(x_1)$ $E(x_2)$

$*$

$E(x_1 * x_2)$

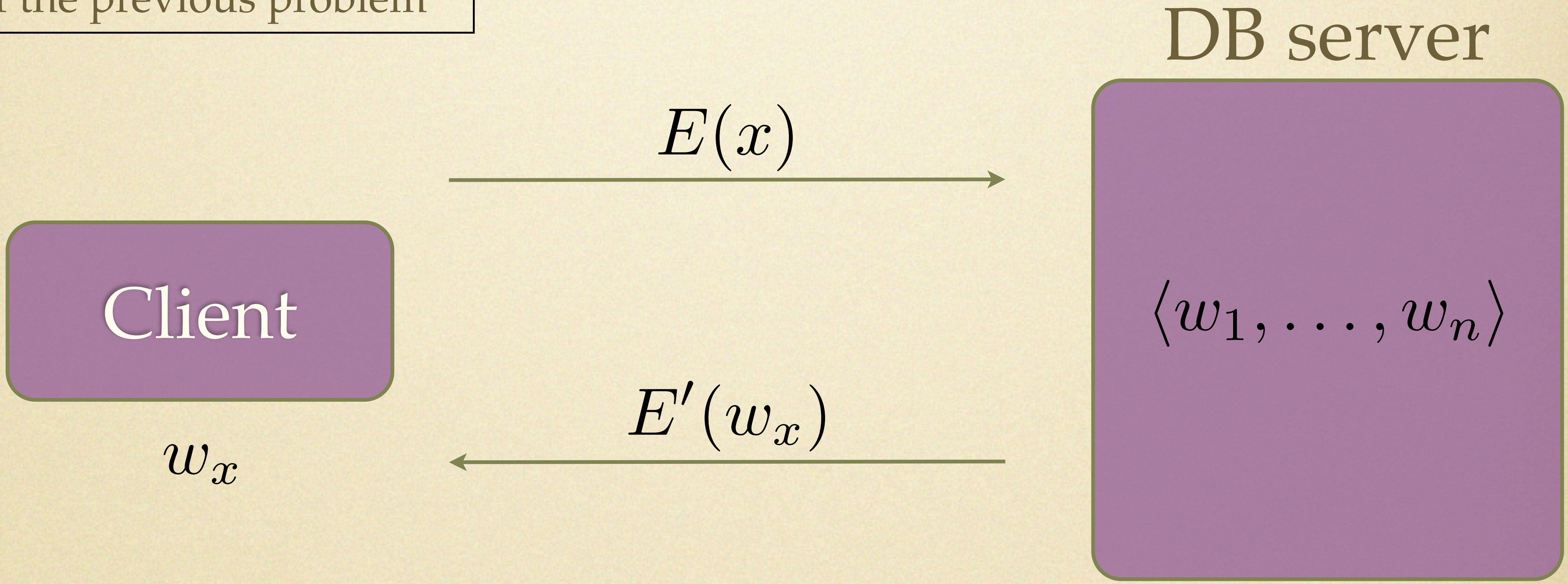
This can be combined with PCP (probabilistically checkable proofs) to provide a (plausibility-type) solution.

Efficient ZK's

- Note that PCP's do not readily yield an *efficient* way to construct zero-knowledge proofs.
  (due to the fact the length of the proof itself might be large)
  - [Killian] : collision resistance hashing => short commit to the PCP proof and then open selectively.
- [GKR08] show ZK-*proofs* with communication quasi-linear in witness length for **NC** verifiable **NP**-languages.
- [Lipmaa11] show sublinear **non-interactive** ZK arguments for all **NP**-languages using bilinear maps using results from additive combinatorics.

Private Information Retrieval
(PIR)

can be seen as a special case of the previous problem



DB server

$E(x)$

Client

$\langle w_1, \ldots, w_n \rangle$

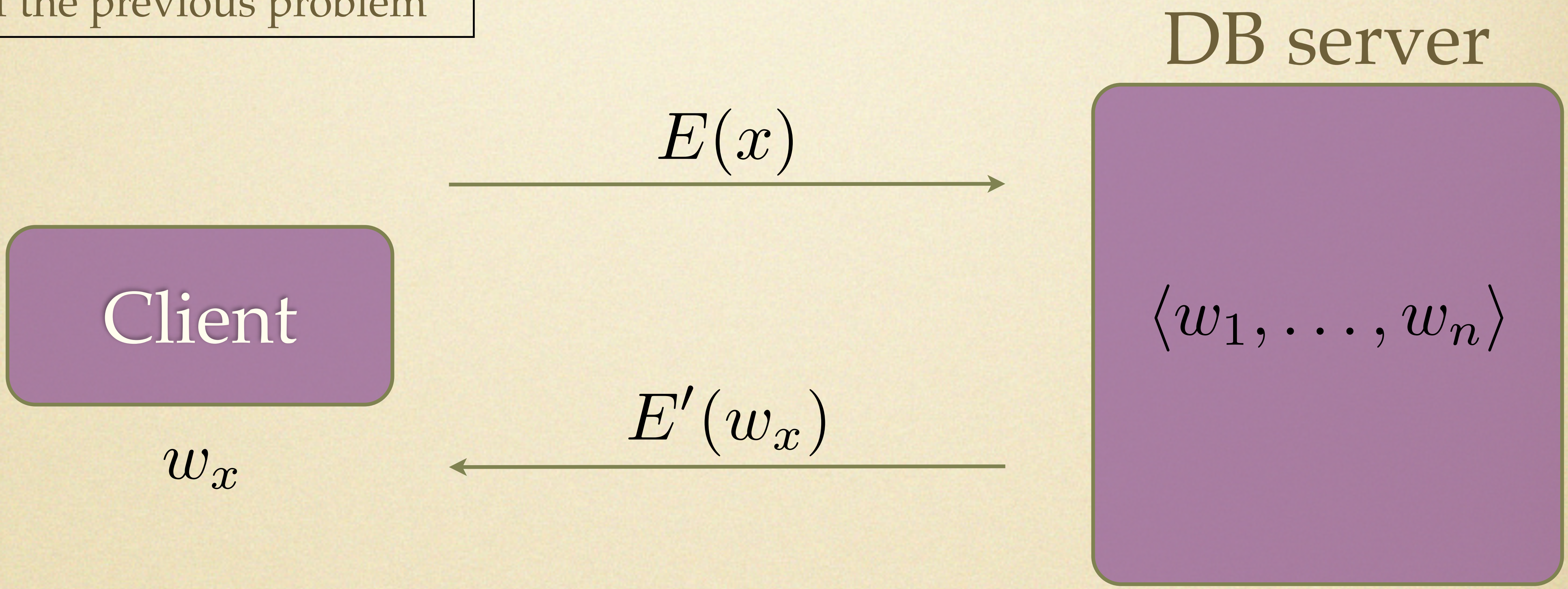
$E'(w_x)$

w_x

Private Information Retrieval

(PIR)

can be seen as a special case of the previous problem



DB server

$E(x)$

Client

$\langle w_1, \ldots, w_n \rangle$

$E'(w_x)$

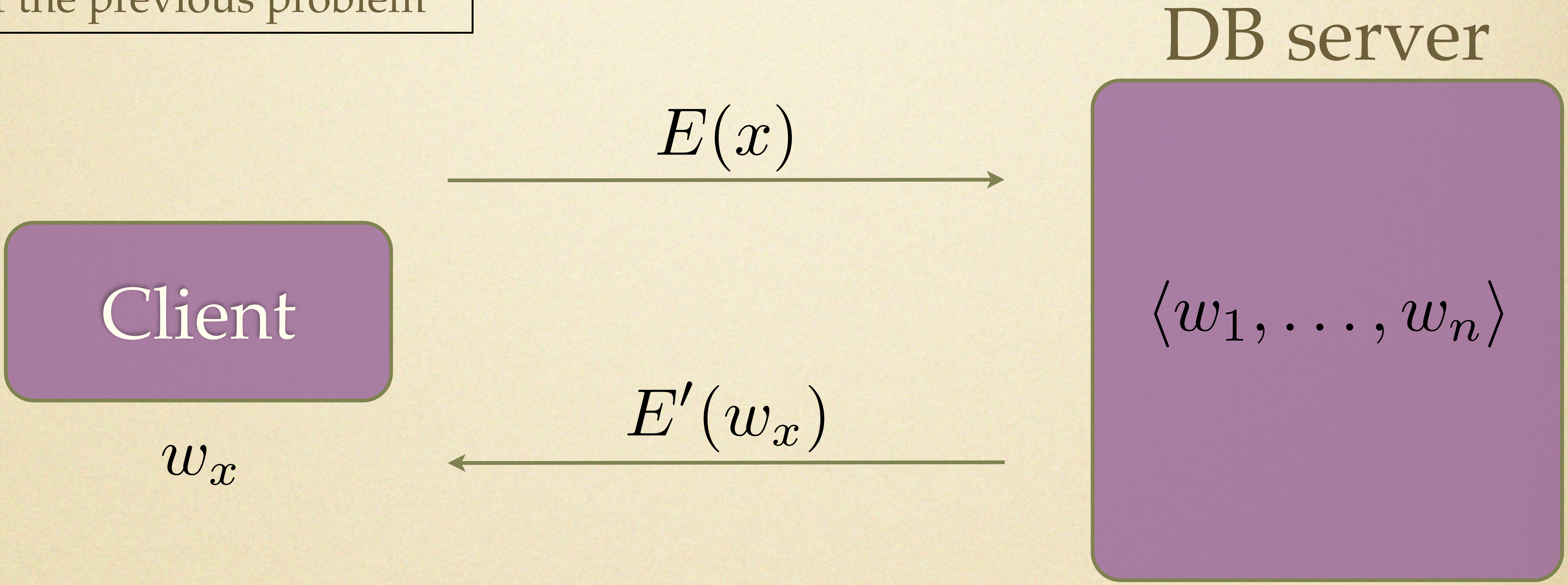
w_x

Currently there are explicit solutions with $O(\log^2 n)$

Private Information Retrieval

(PIR)

can be seen as a special case of the previous problem



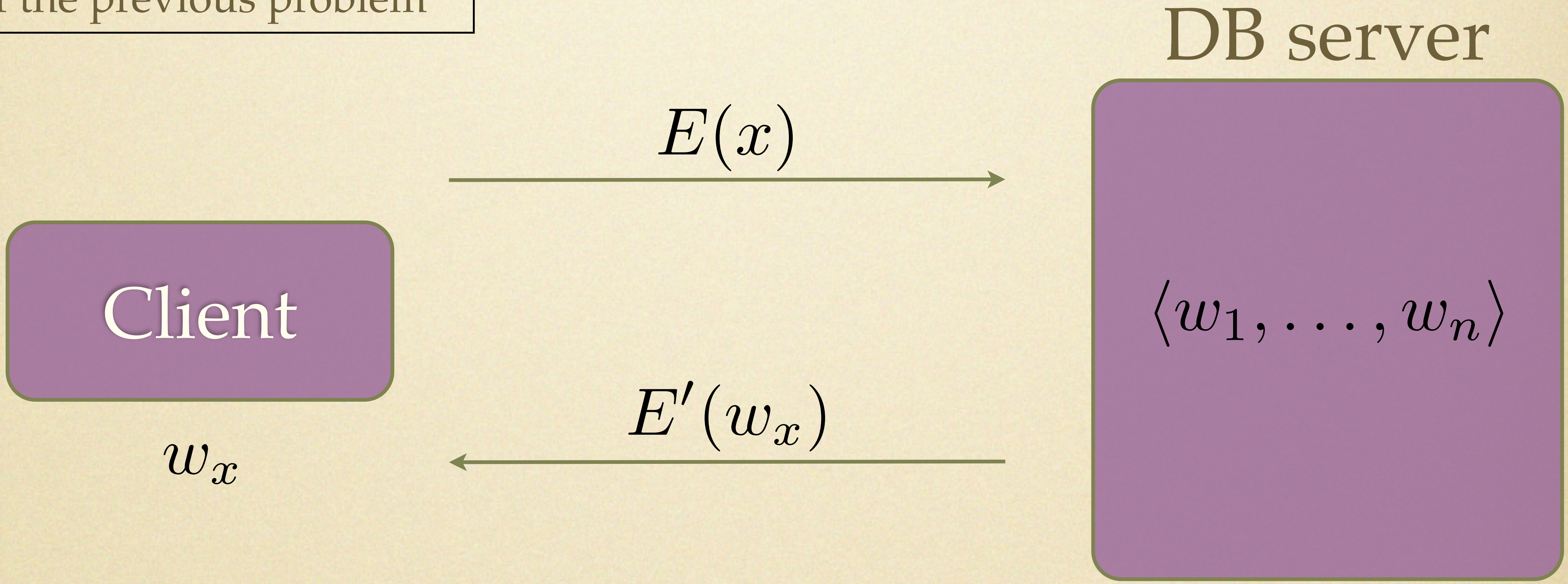
Currently there are explicit solutions with $O(\log^2 n)$

Practical complexity nowhere near “real efficiency”

Private Information Retrieval

(PIR)

can be seen as a special case of the previous problem



DB server

$E(x)$

Client

$\langle w_1, \ldots, w_n \rangle$

$E'(w_x)$

w_x

Currently there are explicit solutions with $O(\log^2 n)$

Practical complexity nowhere near “real efficiency”

[HHS08] show that all trapdoor permutation constructions would incur $\Omega(n)$ complexity

PIR

PIR

- How to minimize server computation?

PIR

- How to minimize server computation?
- FHE implies logarithmic communication. Are there any other logarithmic constructions without FHE?

PIR

- How to minimize server computation?
- FHE implies logarithmic communication. Are there any other logarithmic constructions without FHE?
- What are useful relaxations of privacy ?

PIR

- How to minimize server computation?
- FHE implies logarithmic communication. Are there any other logarithmic constructions without FHE?
- What are useful relaxations of privacy ?
- What is the simplest property we can add to trapdoor permutations so that we break the linear lower bound barrier for PIR?

QUESTION #6

Leakage / Tamper resilience

- Cryptographic implementation may be:
  - prone to *leakage* (side-channels).
  - prone to tampering / faults.
- Due to those issues previous security arguments collapse.
- The restatement of all cryptographic problems in this light is a current major undertaking.

Symmetric Cryptography

Symmetric Cryptography

- In symmetric cryptogaphy *efficiency* is the prime resource.

Symmetric Cryptography

- In symmetric cryptogaphy *efficiency* is the prime resource.
- We are interested in *linear* algorithms with *very small* constants.

Symmetric Cryptography

- In symmetric cryptogaphy *efficiency* is the prime resource.
- We are interested in *linear* algorithms with *very small* constants.
- Despite many years of attempts complexity-theoretic treatment of security is still unsuccessful.

Symmetric Cryptography

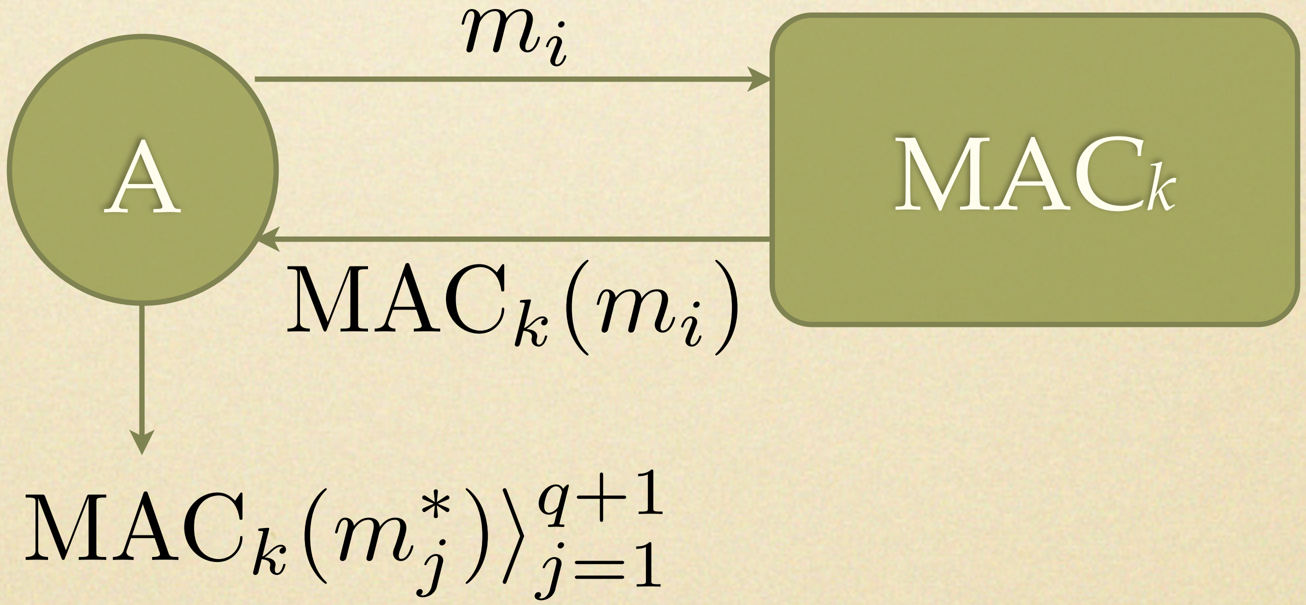
- In symmetric cryptogaphy *efficiency* is the prime resource.
- We are interested in *linear* algorithms with *very small* constants.
- Despite many years of attempts complexity-theoretic treatment of security is still unsuccessful.
- Crypto primitive design remains *black magic.*

QUESTION #7

Foundations of Symmetric Crypto

- Security is defined through complex interactions.

Example: security of MACs



A

MAC$_k$

m_i

$\mathrm{MAC}_k(m_i)$

$\langle m_j^*, \mathrm{MAC}_k(m_j^*)\rangle_{j=1}^{q+1}$

Currently : any proof of security of (*efficient*) MACs is based on a *non-falsifiable assumption*.

Falsifiable Assumptions

[Naor 2003]

Typical structure of cryptographic theorems

$$\mathbf{A} \implies S \text{ is secure}$$

A desired form for the assumption is:

$$\mathbf{A} : \forall \text{ PPT } T : Pr[Q(x, T(x))] = \mathbf{negl}$$

Where Q is a poly-time predicate

such assumptions are *falsifiable.*

cf. $$\mathbf{A} : \forall \text{ PPT } T : Pr[Q(x, T^{O(x,\cdot)}(x))] = \mathbf{negl}$$

Founding Symmetric Cryptography

- Is it possible to obtain constructions for all basic symmetric cryptography primitives with security based on falsifiable assumptions?
  - message authentication codes.
  - encryption.
  - collision resistance hashing.

QUESTION #8

Cryptographic Relations

QUESTION #8

Cryptographic Relations

- Given *primitive* X can one construct *primitive* Y?

QUESTION #8

Cryptographic Relations

- Given *primitive* X can one construct *primitive* Y?
- Celebrated known results:

QUESTION #8

Cryptographic Relations

- Given *primitive* X can one construct *primitive* Y?
- Celebrated known results:
    - Trapdoor functions imply PK encryption.

QUESTION #8

Cryptographic Relations

- Given *primitive* X can one construct *primitive* Y?
- Celebrated known results:
  - Trapdoor functions imply PK encryption.
  - One-way functions imply digital signatures [optimal reduction still open]

QUESTION #8

Cryptographic Relations

- Given *primitive* X can one construct *primitive* Y?
- Celebrated known results:
  - Trapdoor functions imply PK encryption.
  - One-way functions imply digital signatures [optimal reduction still open]
  - One-way functions **do not** imply key-agreement (*black-box separation:* there exists an oracle relative to which OWP exist but KA is impossible)

QUESTION #9

Computational complexity of Cryptographic Assumptions

- Currently there is a wide array of cryptographic assumptions used for arguing security of various constructions.

Understanding their complexity is essential for choosing parameters in the real-world.

Algorithmic Questions

Algorithmic Questions

1. How hard is discrete-logarithm over elliptic curves? currently (Joux-Vitse, Eurocrypt 2012 *best paper*) made the first application of subexponential techniques to DLP over a certain type of curves.

Algorithmic Questions

1. How hard is discrete-logarithm over elliptic curves? currently (Joux-Vitse, Eurocrypt 2012 *best paper*) made the first application of subexponential techniques to DLP over a certain type of curves.

2. What is the relation between RSA and factoring ?

Algorithmic Questions

1. How hard is discrete-logarithm over elliptic curves? currently (Joux-Vitse, Eurocrypt 2012 *best paper*) made the first application of subexponential techniques to DLP over a certain type of curves.

2. What is the relation between RSA and factoring ? Aggarwal Maurer (Eurocrypt 2009) show they are *generically* equivalent.

Algorithmic Questions

1. How hard is discrete-logarithm over elliptic curves? currently (Joux-Vitse, Eurocrypt 2012 *best paper*) made the first application of subexponential techniques to DLP over a certain type of curves.

2. What is the relation between RSA and factoring ? Aggarwal Maurer (Eurocrypt 2009) show they are *generically* equivalent.

Algorithmic Questions

1. How hard is discrete-logarithm over elliptic curves? currently (Joux-Vitse, Eurocrypt 2012 *best paper*) made the first application of subexponential techniques to DLP over a certain type of curves.

2. What is the relation between RSA and factoring ? Aggarwal Maurer (Eurocrypt 2009) show they are *generically* equivalent.

3. What is the exact relation of the learning with errors problem (LWE) and the shortest independent vectors problem (SIVP) ? (Regev 2005 show they are *quantumly* equivalent).

QUESTION #10

reduce/expand the 5 worlds

R. Impagliazzo

Pessiland

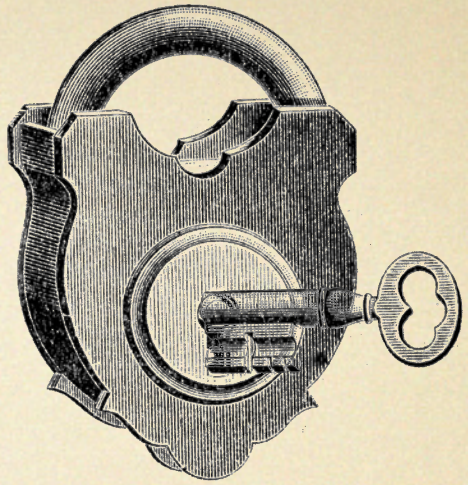
Heuristica

Minicrypt

Algorithmica

Cryptomania

Cryptography



- ... has rapidly expanded and evolved in the last 36 years enriching itself with various areas of mathematics, statistics, CS theory and algorithms.
- ... problems are firmly grounded on real-world problems and security needs.
- ... is intricately connected with the most fundamental problems of CS theory.
- ... puts to (*sometimes surprising*) use many techniques and concepts that before remained purely theoretical or seemingly unrelated.

EUROCRYPT 2013

May 26-30, 2013

- Biggest Cryptography conference outside the USA.
- The flagship conference of the International Association of Cryptologic Research.

for more information

CRYPTO.SEC

cryptography.security
@university-of-athens

κρυπτογραφία - ασφάλεια
CRYPTOSEC
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

http://crypto.di.uoa.gr

funded
Ph.D. positions
are available