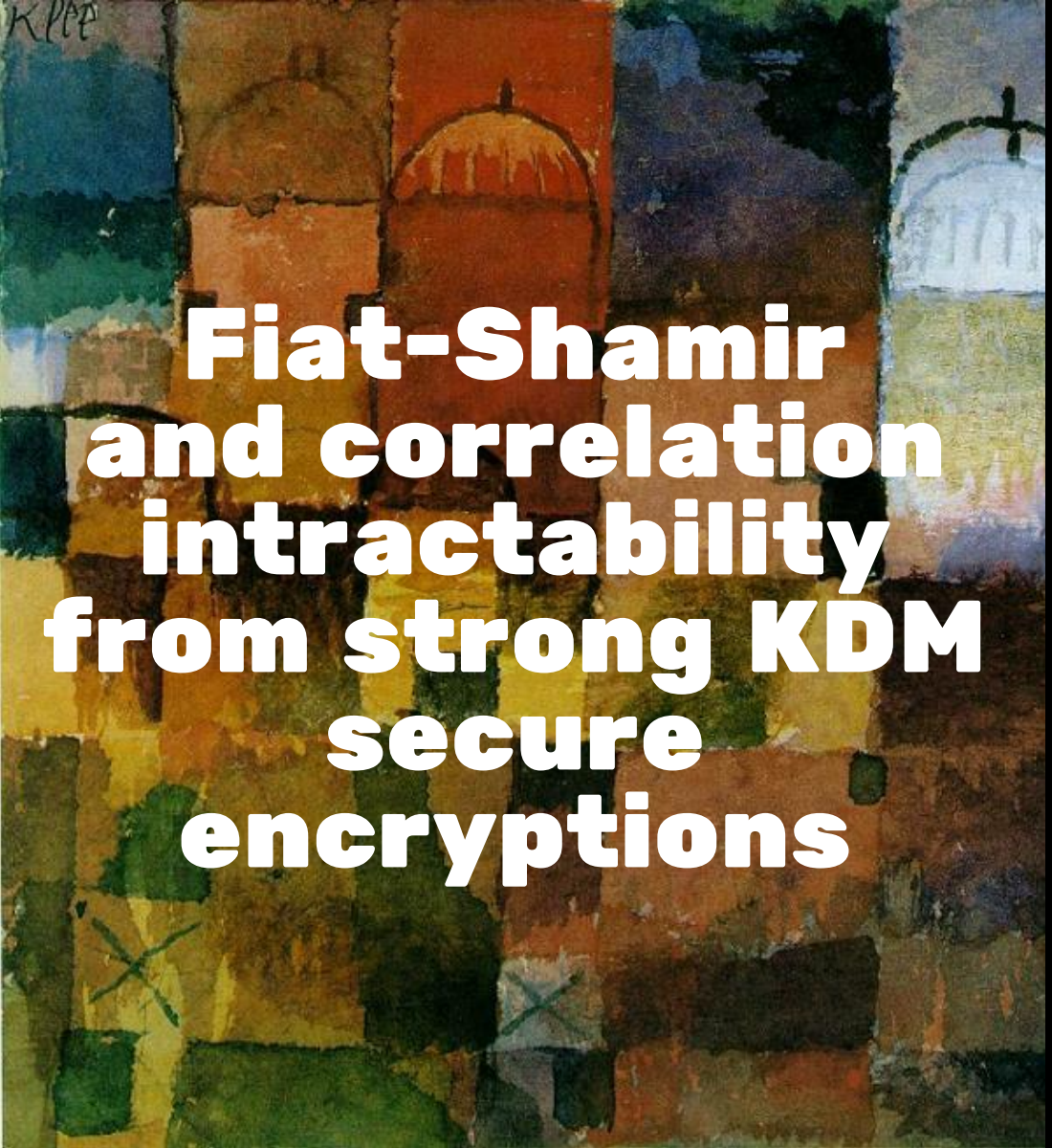
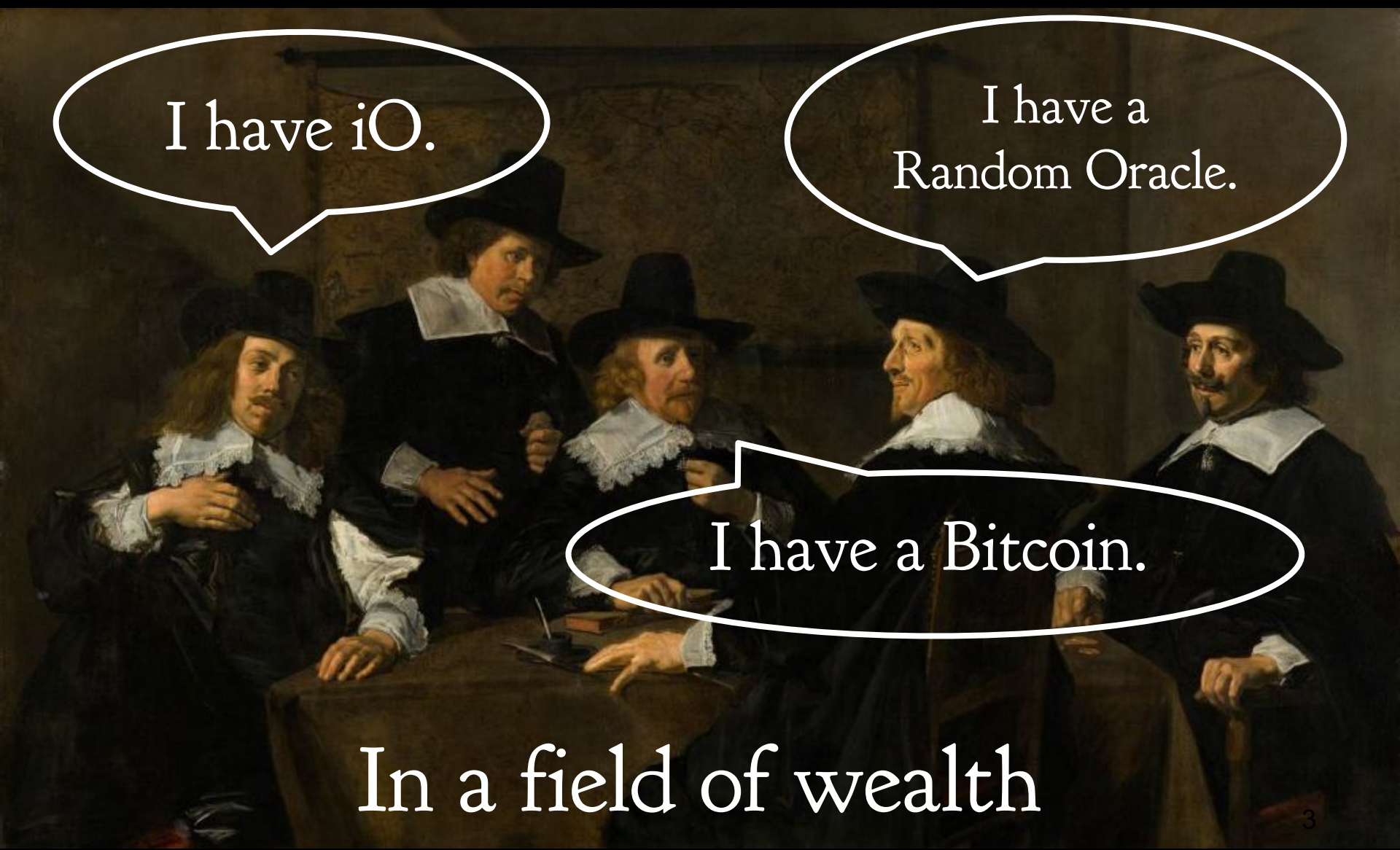


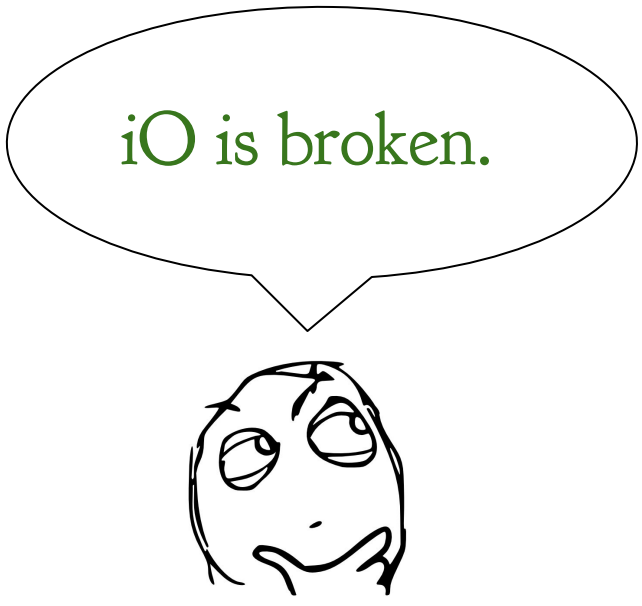
Fiat-Shamir and correlation intractability from strong KDM secure encryptions

Ran Canetti
Yilei Chen
Leonid Reyzin
Ron Rothblum

Trailer



In a field of wealth



They had nothing

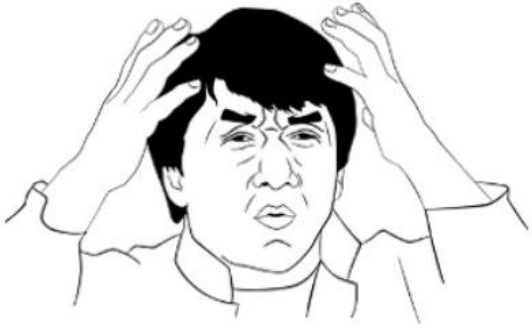
Equal number of Bitcoin for every member!

Feel free to assume iO.

Free access to Random oracle!

On the land of liberty

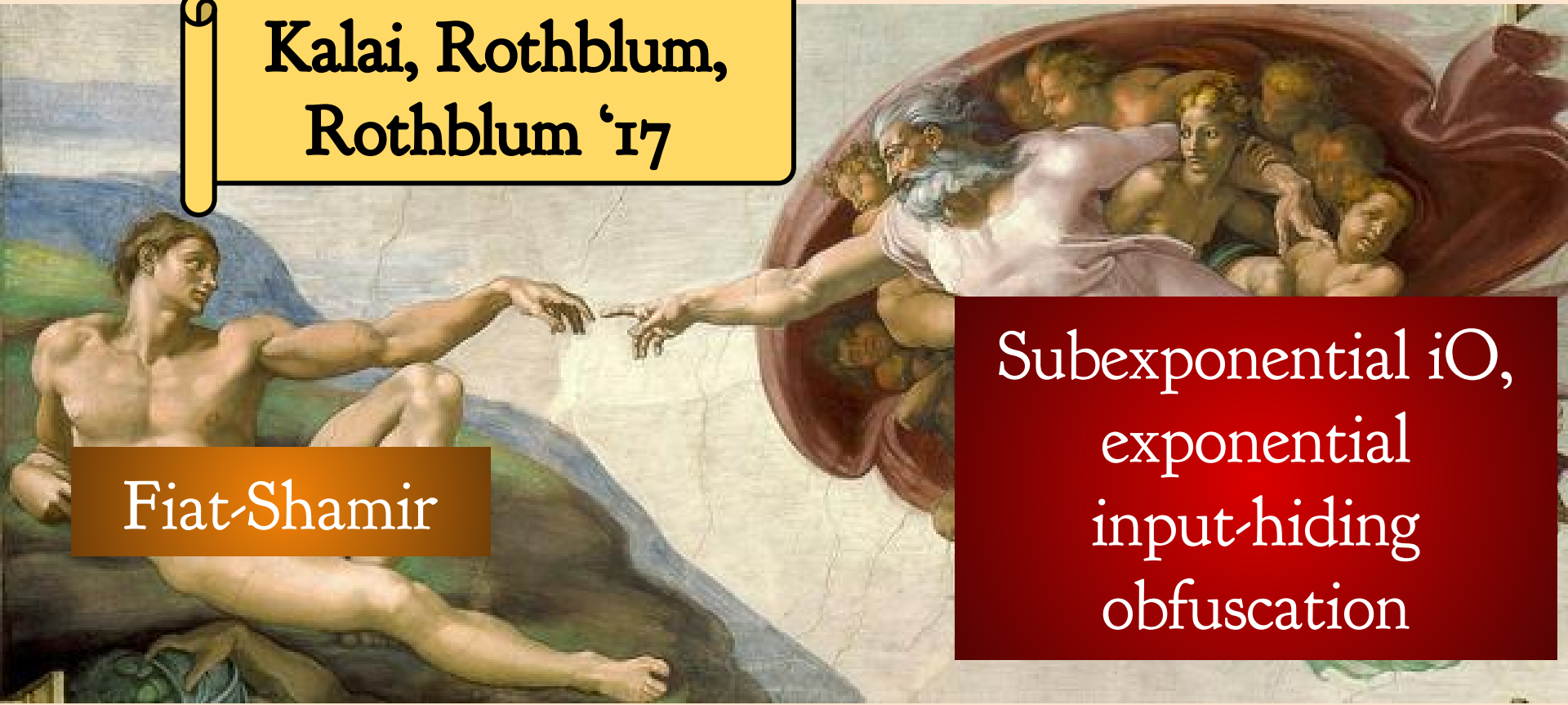
Practical Identity-Based Encryption without Random Oracles



Functional Encryption without Obfuscation

They are enslaved

At the moment of glory



At the moment of glory

Fiat-Shamir and correlation intractability

from

~~subexponentially secure iO and~~

~~exponentially secure input-hiding obfuscation~~

They raise,

Fiat-Shamir and correlation intractability

from

~~subexponentially secure iO and~~

~~exponentially secure input-hiding obfuscation~~

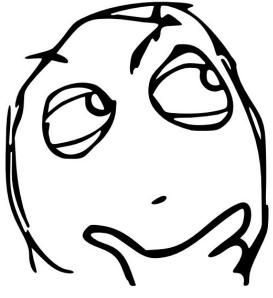
exponentially KDM secure encryption schemes

They raise, united

Directors

Ran Canetti
Yilei Chen
Leonid Reyzin
Ron Rothblum

How to capture the properties of a “good enough” cryptographic hash function?

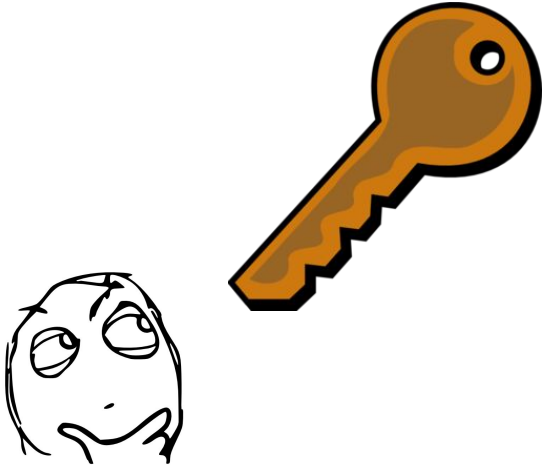


Fiat-Shamir ‘86, Bellare-Rogaway ‘93:

Can model cryptographic hash functions as “Random Oracles”

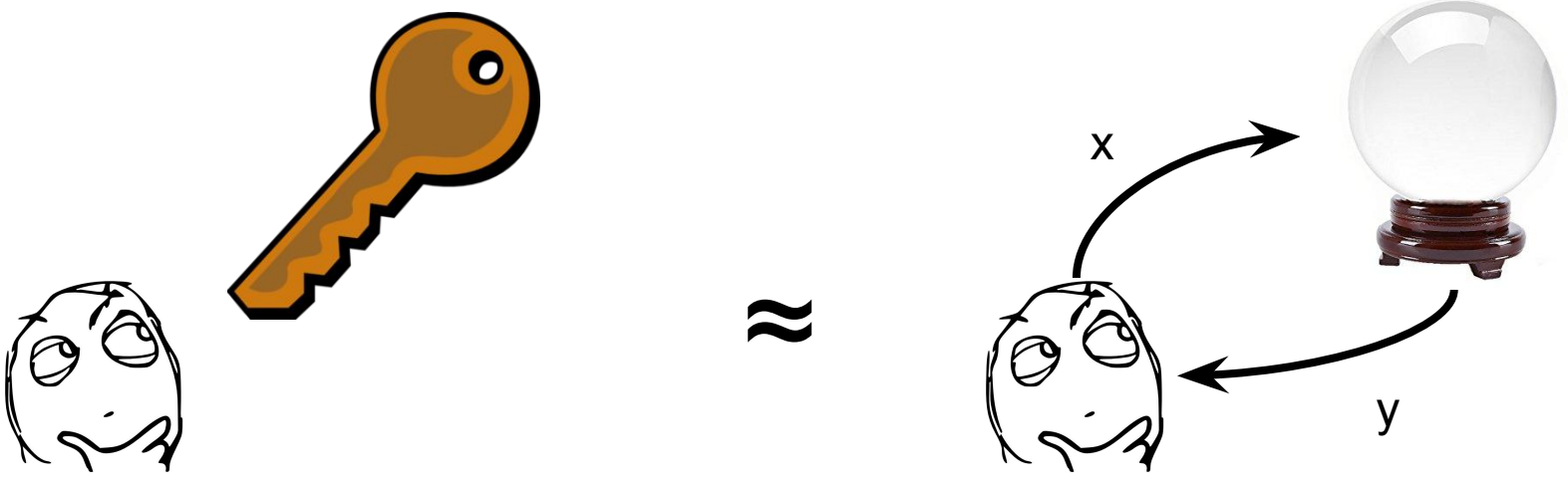
Fiat-Shamir ‘86, Bellare-Rogaway ‘93:

Can model cryptographic hash functions as “Random Oracles”



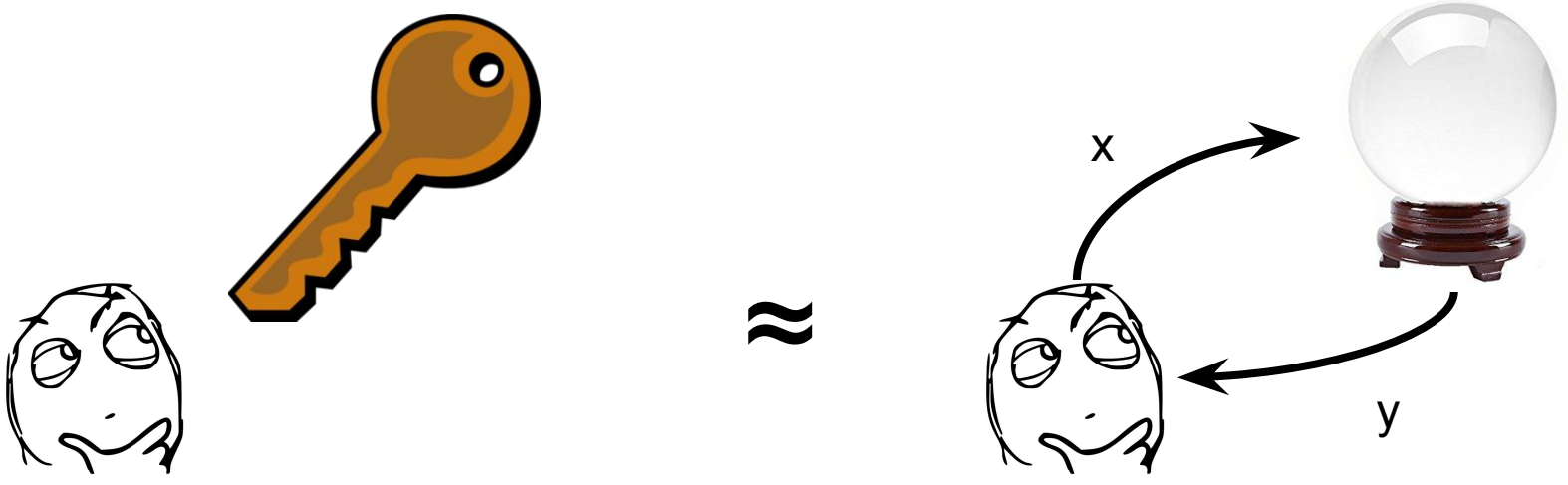
Fiat-Shamir ‘86, Bellare-Rogaway ‘93:

Can model cryptographic hash functions as “Random Oracles”



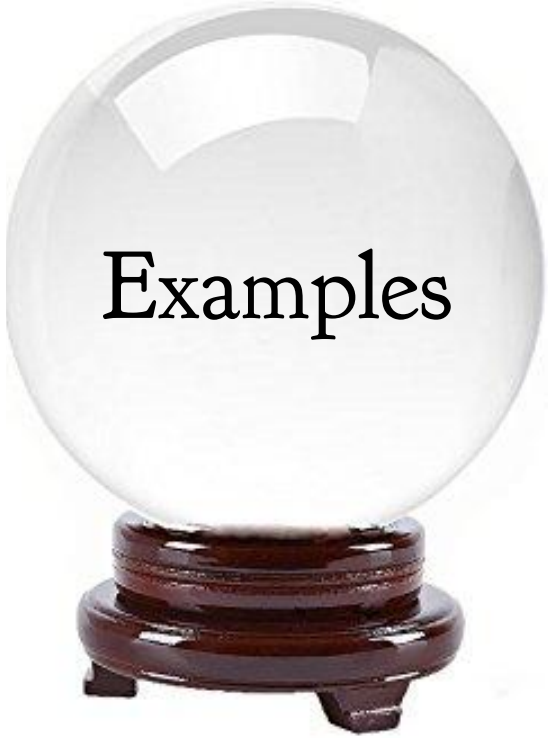
Fiat-Shamir ‘86, Bellare-Rogaway ‘93:

Can model cryptographic hash functions as “Random Oracles”



Build efficient crypto schemes (secure under heuristics):

- Efficient CCA secure encryptions
- Hash-and-sign paradigm
- Many applications


Examples

d

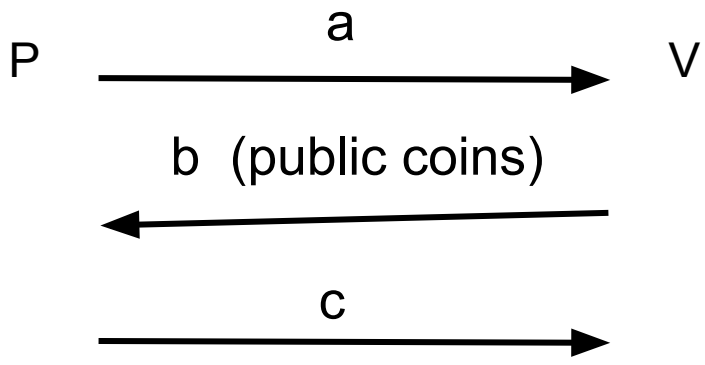
H(X)=000000000110….110010111101

Desired property:
Find an input X such that H(X) has a prefix of d 0s takes roughly 2^d steps.

Fiat, Shamir 86: Turning an interactive protocol => non-interactive

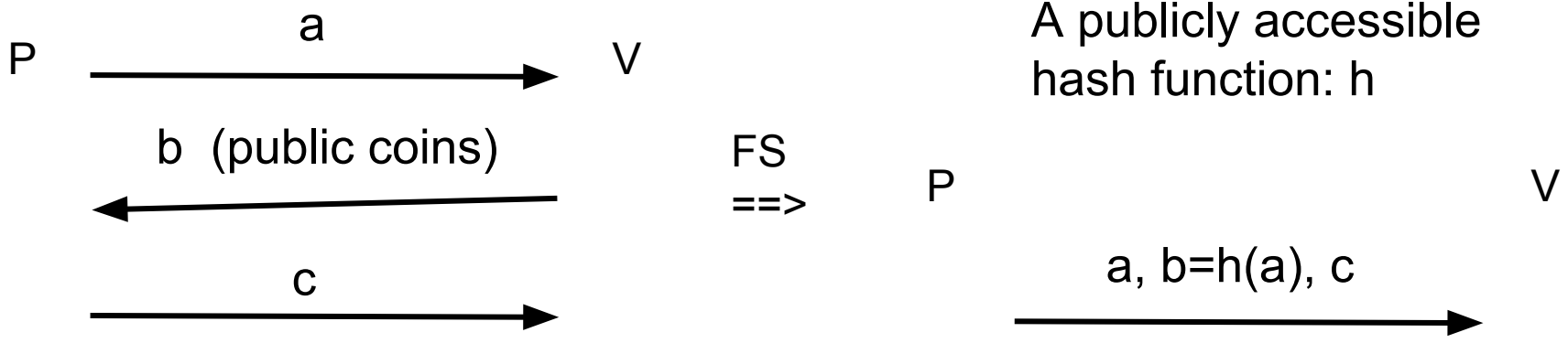
Fiat, Shamir 86: Turning an interactive protocol => non-interactive

An interactive protocol for a language L and an instance x:



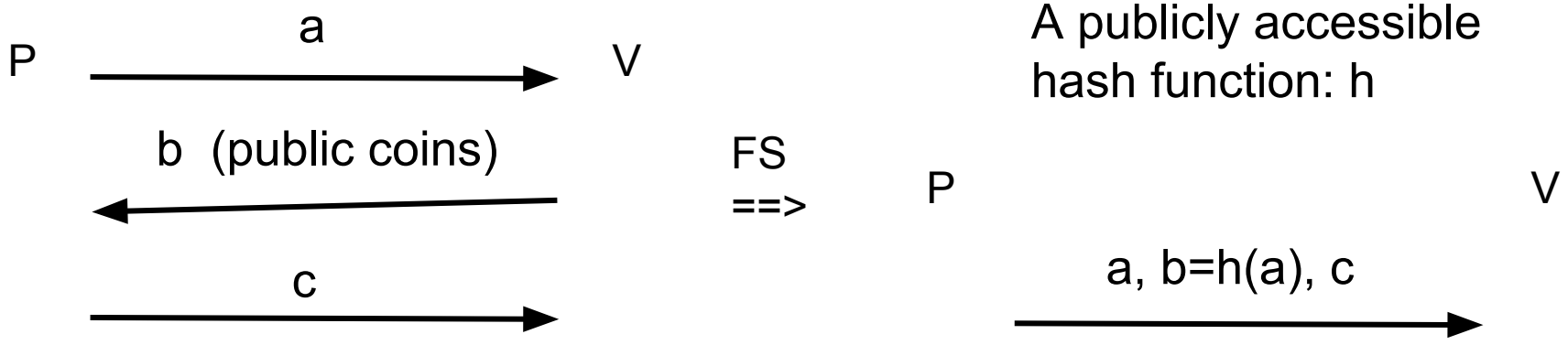
Fiat, Shamir 86: Turning an interactive protocol => non-interactive

An interactive protocol for a language L and an instance x:



Fiat, Shamir 86: Turning an interactive protocol => non-interactive

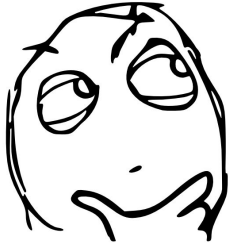
An interactive protocol for a language L and an instance x:



Goal: preserve the original properties of the protocol, e.g. completeness and soundness.

Fiat, Shamir

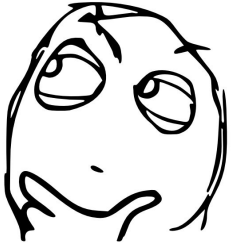
Does the Random oracle model oversimplifies the problem in the reality?



Fiat, Shamir

Does the Random oracle model oversimplifies the problem in the reality?

Can we define a more concrete property that captures what we want?



Today:

Correlation Intractability

"infeasibility of finding 'sparse' input-output relations"

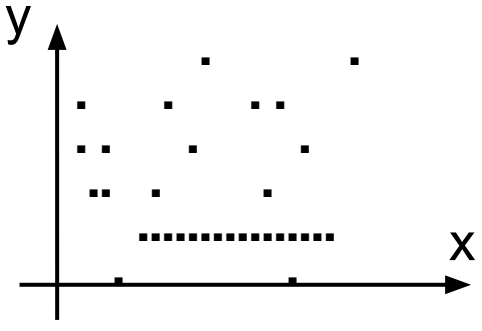
--- Canetti, Goldreich, Halevi 1998

Sparse Relations

“For each input (x),
the fraction of outputs (y) in the relation is negligible”

Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"



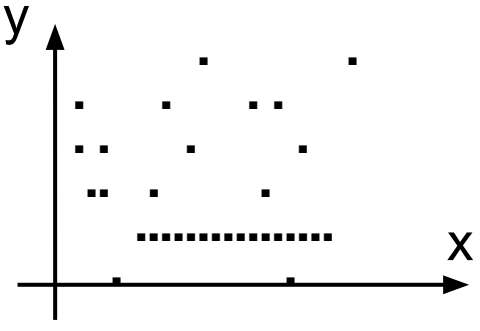
Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Implicitly: relations that are intractable on truly random functions

For all (non-uniform) p.p.t. Adversary:

$$Pr_{Adv,\ O}[\ Adv^{O} \rightarrow x: R(x, O(x))=1\ ] < negl.$$



Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Implicitly: relations that are intractable on truly random functions

For all (non-uniform) p.p.t. Adversary:

$$Pr_{Adv,\ O}[\ Adv^{O} \rightarrow x: R(x, O(x))=1\ ] < negl.$$

***Can naturally generalize to multi-input-output relations**

For all (non-uniform) p.p.t. Adversary:

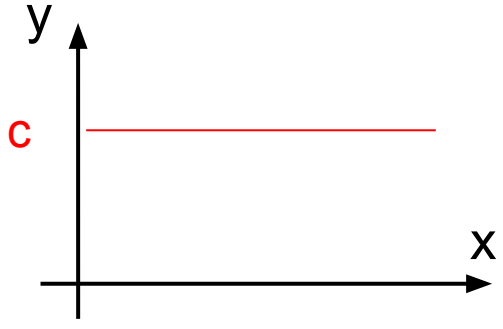
$$Pr_{Adv,\ O}[\ Adv^{O} \rightarrow x1, x2:\ \ R(x1, O(x1), x2, O(x2))=1\ ] < negl.$$

Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

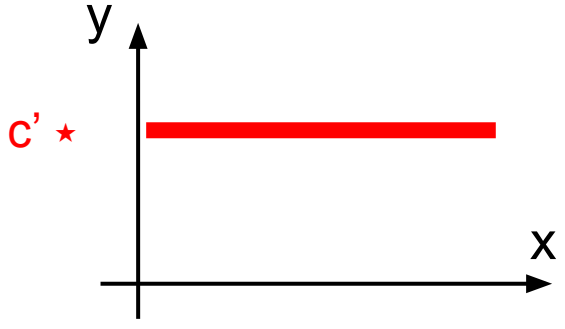


Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c
Partial constant relation: R(x, y) = 1, if the first half of y=c'



Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c
Partial constant relation: R(x, y) = 1, if the first half of y=c'

*Examples for interesting multiple-input-output relations

Collision relation: R(x1, y1, x2, y2) = 1, if y1=y2 and (not x1=x2)

Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Correlation intractability [Canetti, Goldreich, Halevi '98]

Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Correlation intractability [Canetti, Goldreich, Halevi '98]

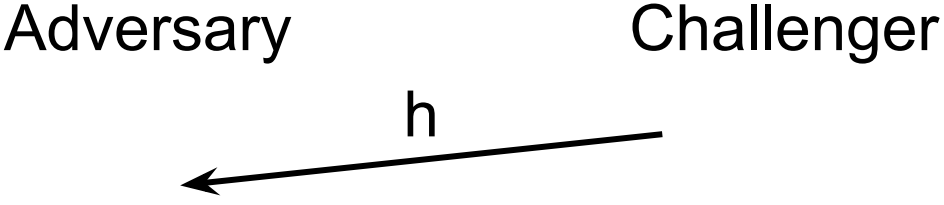
For all sparse relations R:

Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Correlation intractability [Canetti, Goldreich, Halevi '98]

For all sparse relations R:

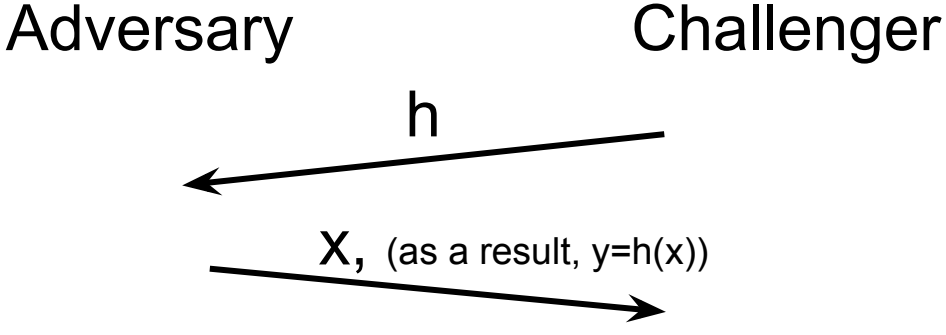


Sparse Relations

"For each input (x),
the fraction of outputs (y) in the relation is negligible"

Correlation intractability [Canetti, Goldreich, Halevi '98]

For all sparse relations R:



Adversary wins if R(x, h(x))=1

Examples: Interesting sparse relations

Constant relation: $R(x, y) = 1$, if $y=c$

Partial constant relation: $R(x, y) = 1$, if the first half of $y=c'$

Exercise: Prove CI w.r.t. simple relations from your favorite hash function.

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c’

Bitcoin

H(???...?)=000000….XYZ3d83h

H(???...?)=000000….XYZ3d83h

Correlation intractability [Canetti, Goldreich, Halevi 98]
For all relations of negligible density, all polynomial adversaries succeed with negligible probability.

H(???...?)=000000….XYZ3d83h

Correlation intractability [Canetti, Goldreich, Halevi 98]
For all relations of negligible density, all polynomial adversaries succeed with negligible probability.

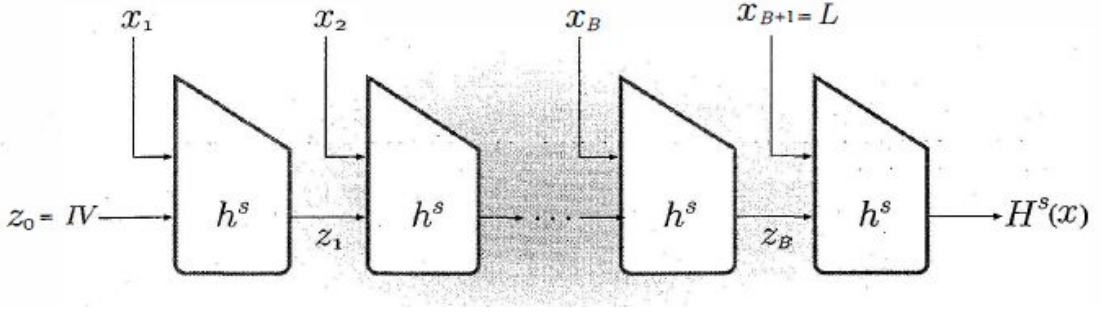
Quantitative correlation intractability [This work]
For all relations of density D, all adversaries running in time T succeed with probability f(D, T).

The smallest possible f to hope for: f(D,T) = DT

H(???...?)=000000….XYZ3d83h

In fact, SHA256 doesn’t satisfy the best-possible quantitative CI.

AsicBoost takes advantage of Merkle-Damgard to speed up bitcoin mining.



Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

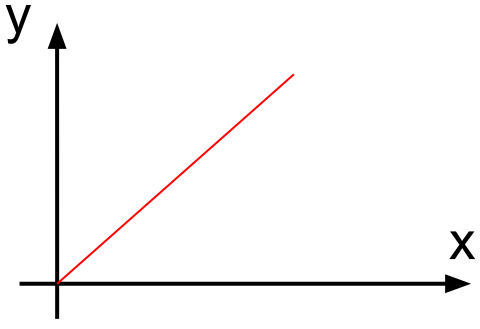
Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)



Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

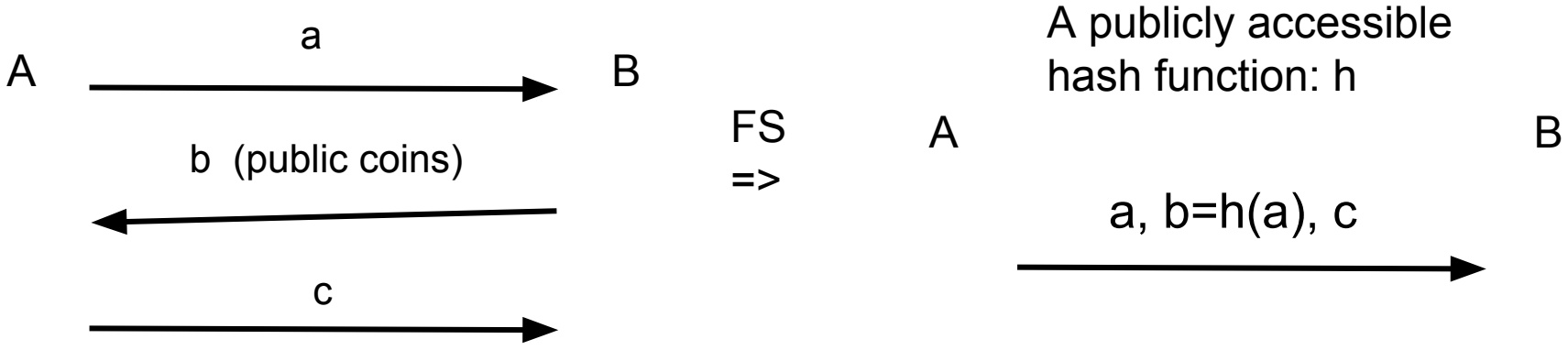
Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

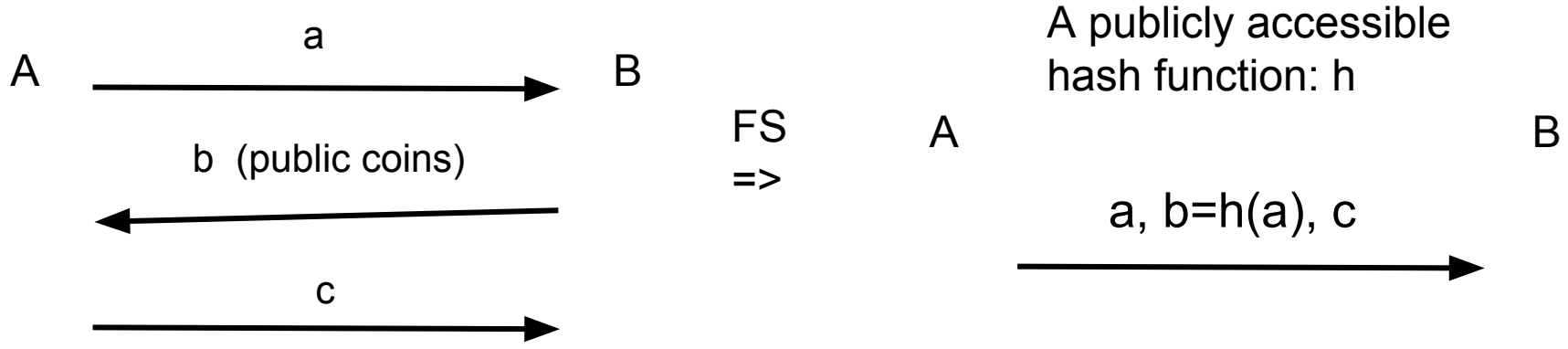
Fiat-Shamir: Long story ...

An interactive protocol for a language L and an instance x:



Fiat, Shamir 86: 3 round proof system => 1 round argument

An interactive protocol for a language L and an instance x:



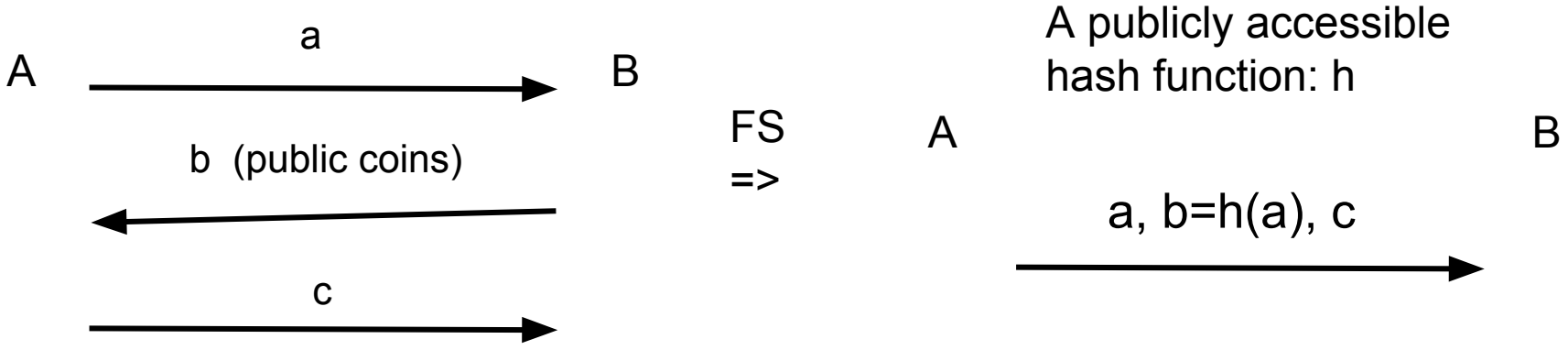
Fiat, Shamir 86: 3 round proof system => 1 round argument

Fiat, Shamir relation: [Hada, Tanaka 99, Dwork et al 03]

R(a, b)=1 if $x \notin L$ and $\exists$ c s.t. Verifier(x, a, b, c) accepts

R is a sparse relation following the soundness of the 3 round proof system.
The membership of R is typically not polynomial-time decidable due to "$\exists$ c".

An interactive protocol for a language L and an instance x:



Fiat, Shamir 86: 3 round proof system => 1 round argument

Difficulty to prove the Fiat-Shamir property:

[Goldwasser, Kalai ‘03] impossibility for arguments.
[Barak, Lindell, Vadhan ‘06] Define “Entropy preserving”, it implies FS for proofs.
[Dodis, Ristenpart, Vadhan ‘12] “Entropy preserving” is necessary for FS for proofs.
[Bitansky et al. ‘13] for proof systems, impossible from black-box reductions to falsifiable assumptions.

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

Fiat-Shamir for proofs: R(a, b) = 1 if $x \notin L$ and $\exists$ c s.t. Verifier(x, a, b, c)=1

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

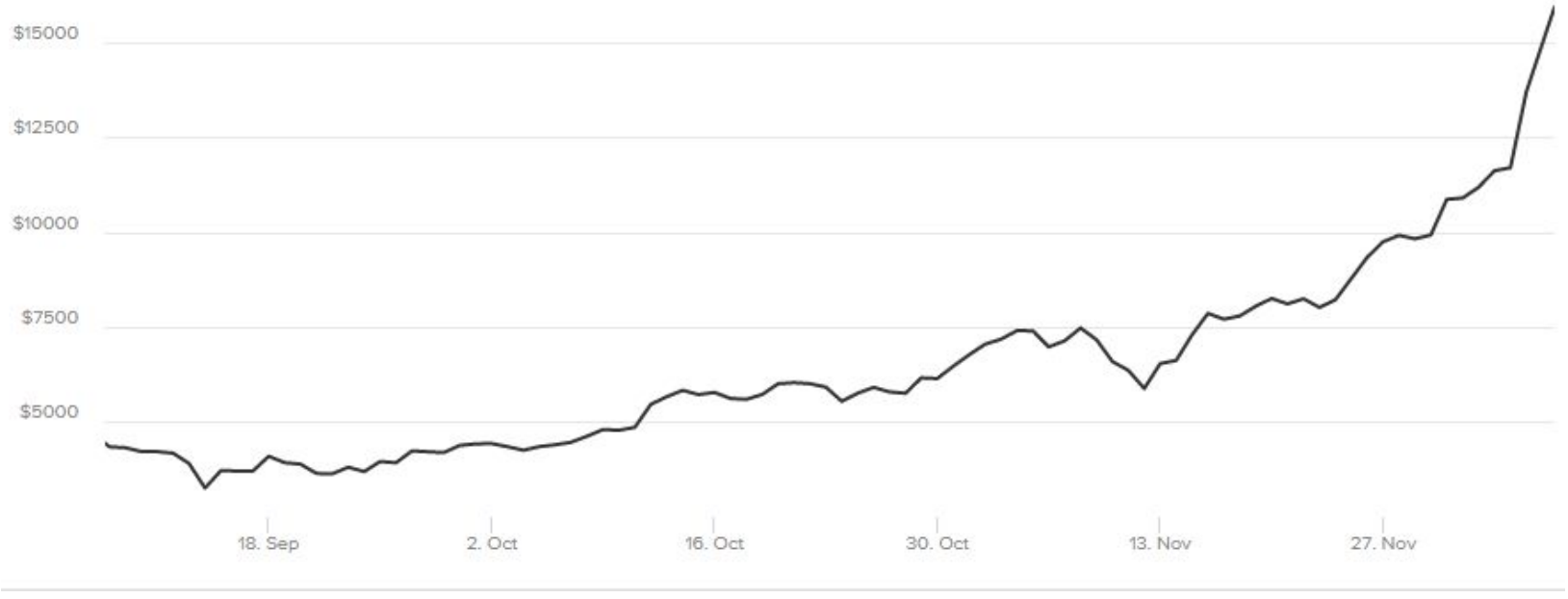
Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

Fiat-Shamir for proofs: R(a, b) = 1 if $x \notin L$ and $\exists$ c s.t. Verifier(x, a, b, c)=1

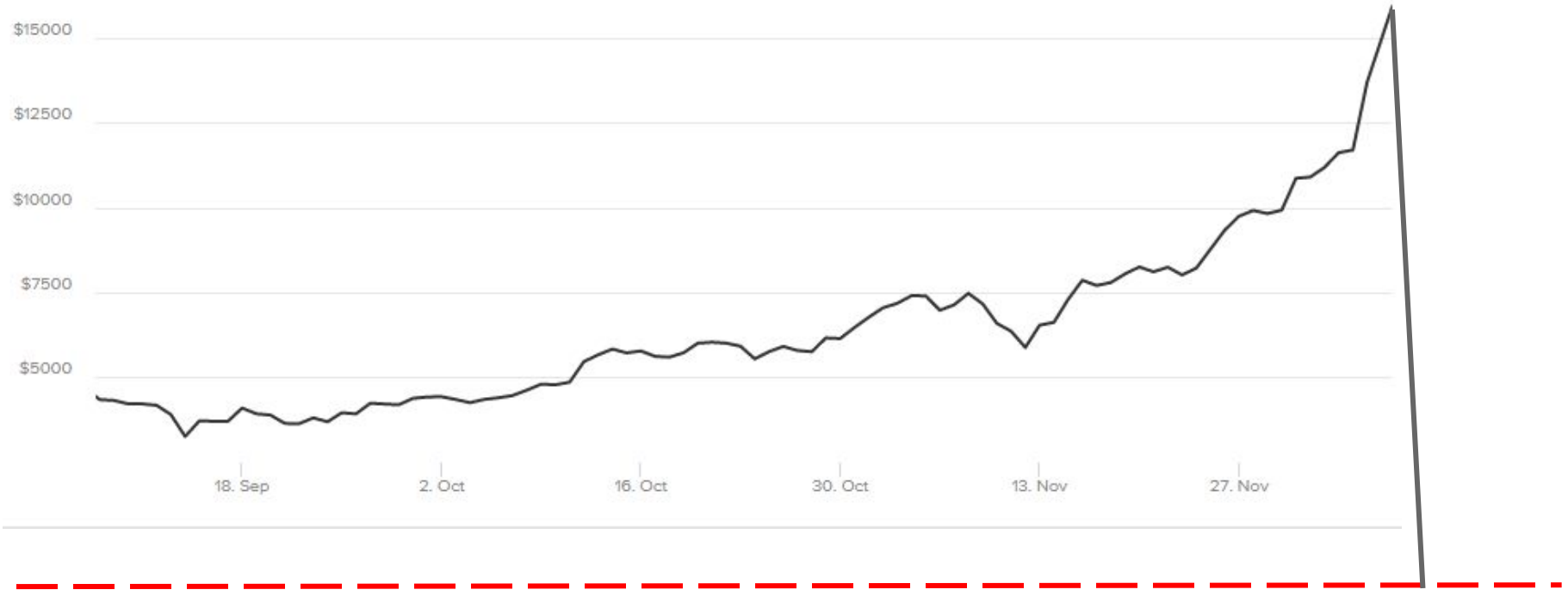
[CGH 98]

Correlation intractability is impossible to achieve

Bitcoin price
live

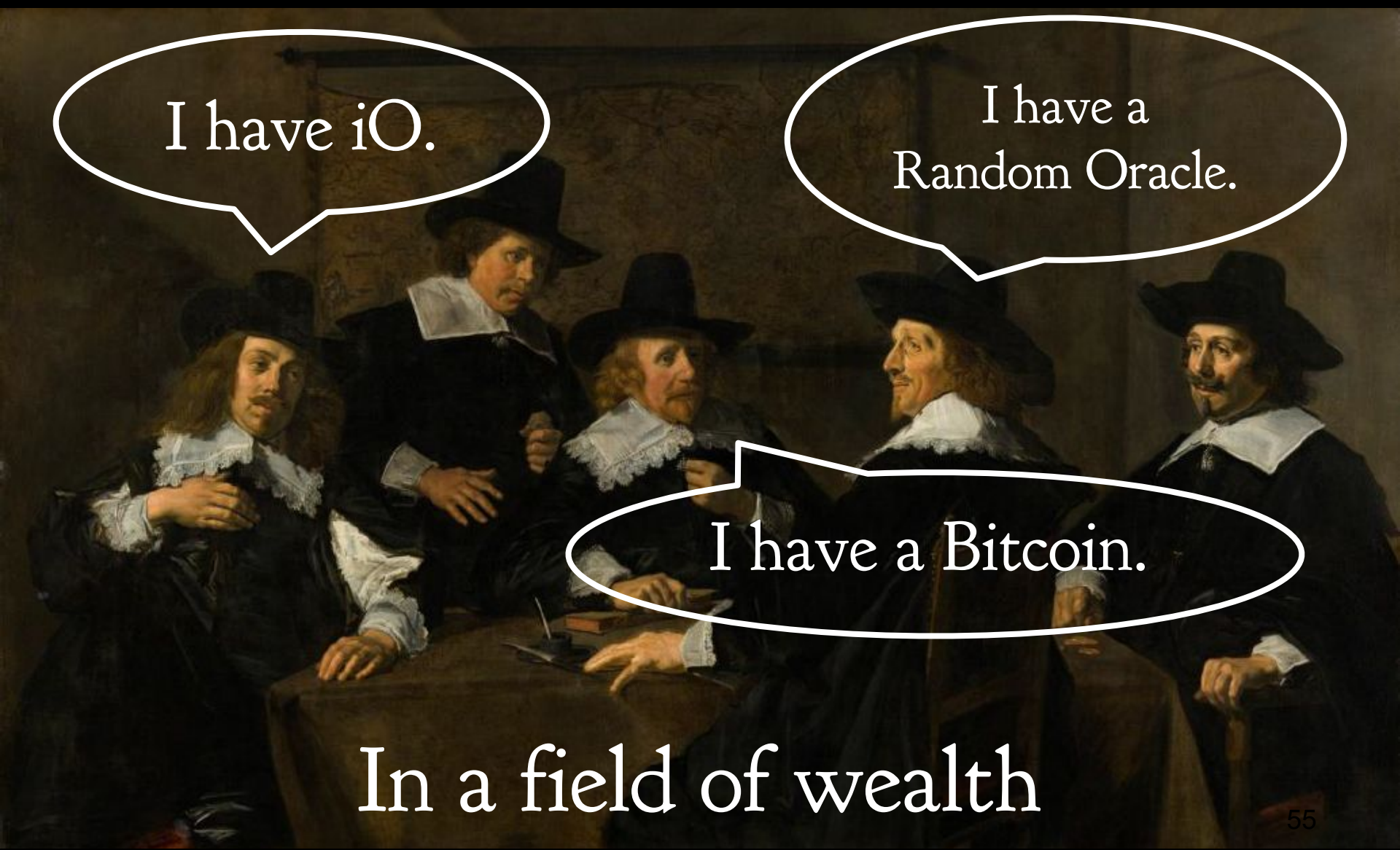


Bitcoin price
live

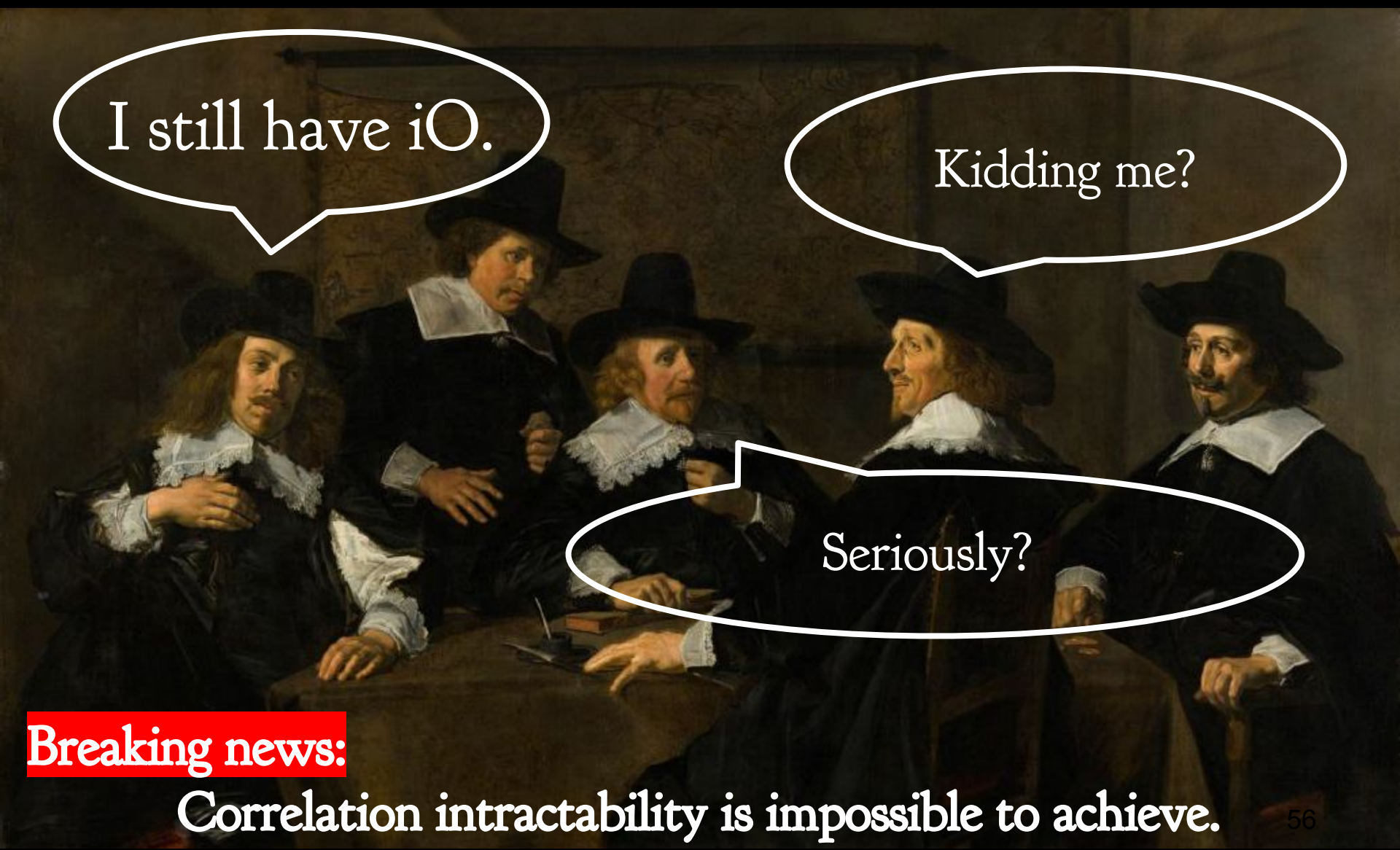


Breaking news:

Correlation intractability is impossible to achieve.



In a field of wealth



Breaking news:

Correlation intractability is impossible to achieve.

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

Fiat-Shamir for proofs: R(a, b) = 1 if $x \notin L$ and $\exists$ c s.t. Verifier(x, a, b, c)=1

[CGH 98]

Correlation intractability is impossible to achieve

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

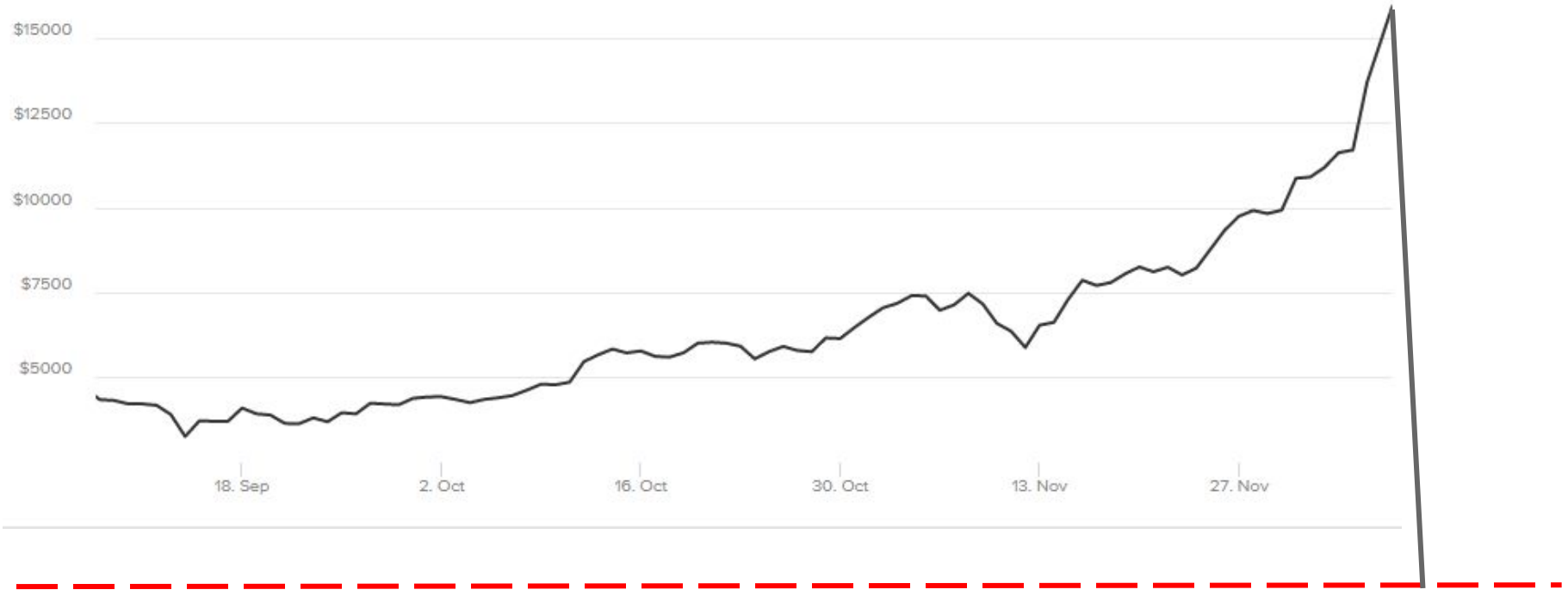
Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

Fiat-Shamir for proofs: R(a, b) = 1 if $x \notin L$ and $\exists$ c s.t. Verifier(x, a, b, c)=1

[CGH 98]

Correlation intractability is impossible to achieve … in some cases.

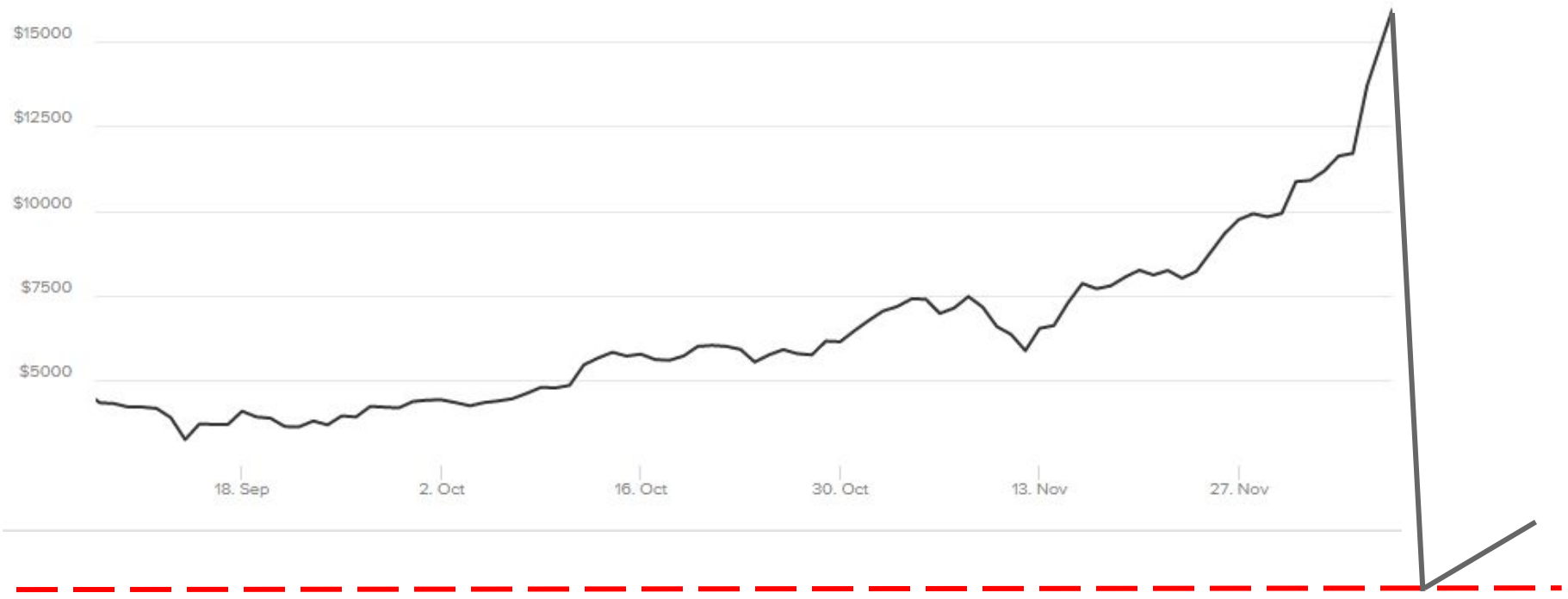
Bitcoin price
live



Breaking news:

Correlation intractability is impossible to achieve.

Bitcoin price
live



Breaking news:

Correlation intractability is impossible to achieve in some cases

Canetti, Goldreich, Halevi:

H cannot be correlation intractable if **the key is short** !!!

Canetti, Goldreich, Halevi:
H cannot be correlation intractable if **the key is short** !!!

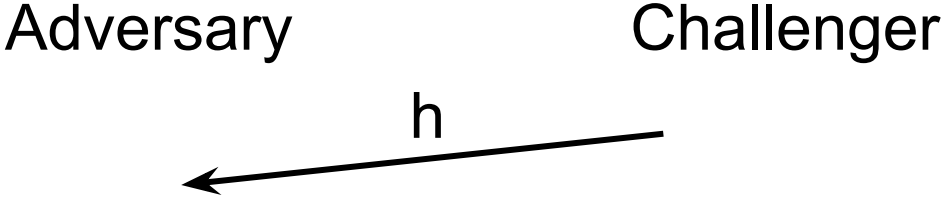
Consider the “Diagonal” relation:

$$R^H(x, y)=1 \text{ iff } y=x(x)$$

Canetti, Goldreich, Halevi:
H cannot be correlation intractable if **the key is short** !!!

Consider the “Diagonal” relation:

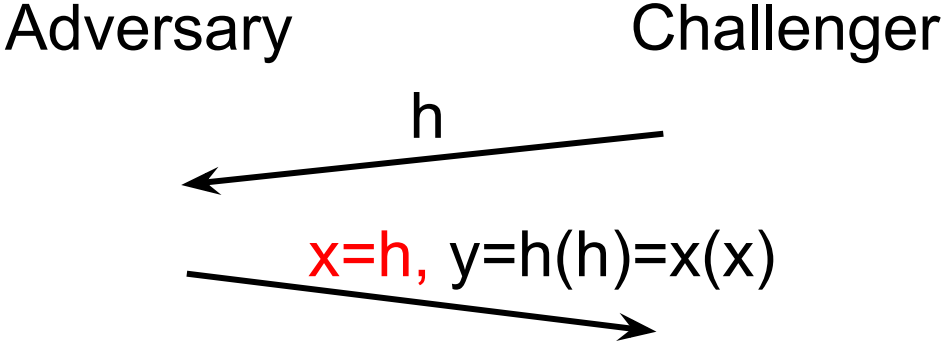
$$R^H(x, y)=1 \text{ iff } y=x(x)$$



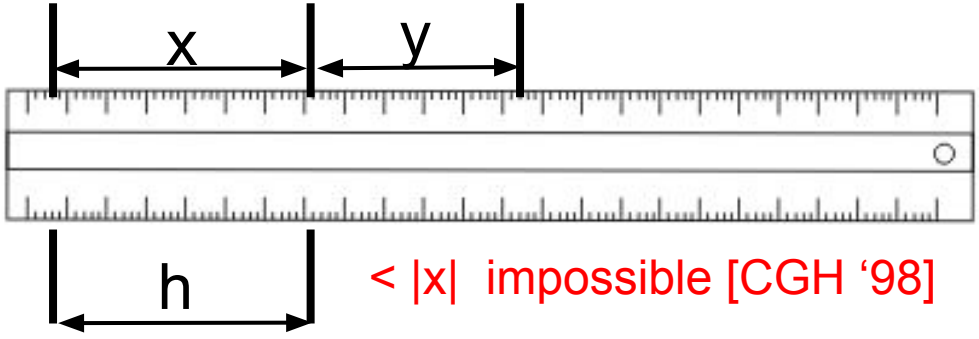
Canetti, Goldreich, Halevi:
H cannot be correlation intractable if **the key is short** !!!

Consider the "Diagonal" relation:

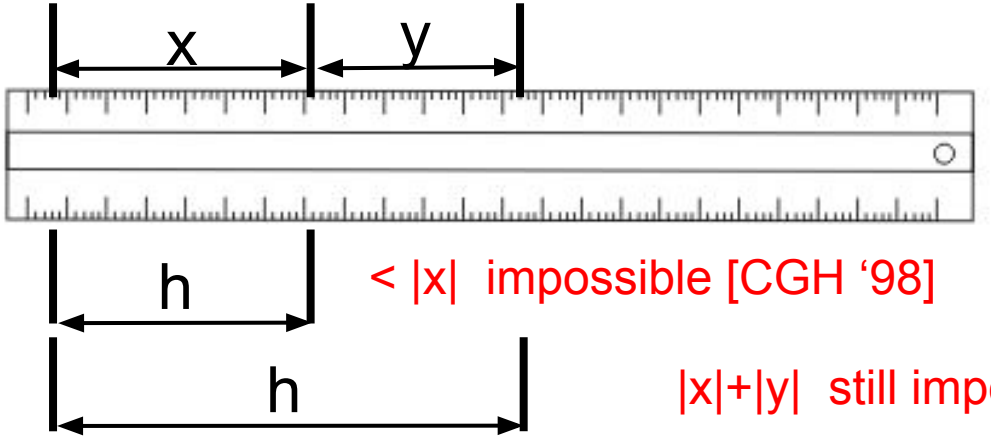
$$R^H(x, y)=1 \text{ iff } y=x(x)$$



Adversary wins if $R^H(x, x(x))=1$

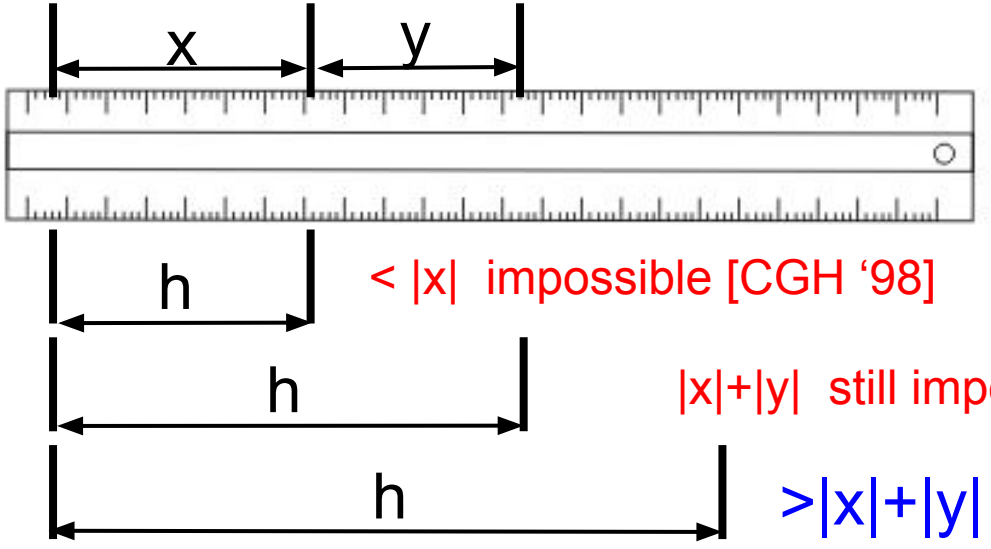


< |x| impossible [CGH ‘98]



< |x| impossible [CGH ‘98]

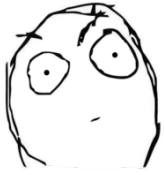
|x|+|y| still impossible [CGH ‘98]

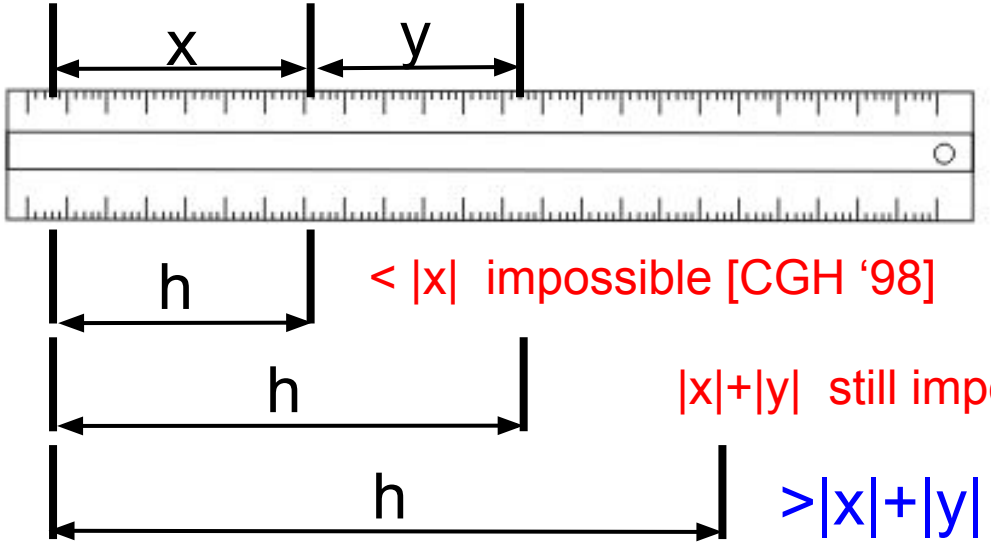


< |x| impossible [CGH ‘98]

|x|+|y| still impossible [CGH ‘98]

>|x|+|y| may be possible!!





< |x| impossible [CGH '98]

|x|+|y| still impossible [CGH '98]

>|x|+|y| may be possible!!

Possible for hash functions with even just 'slightly' longer keys… not too bad.

Functions from $\{0,1\}^* \rightarrow \{0,1\}^m$ can never be correlation intractable.

Examples: Interesting sparse relations

Constant relation: R(x, y) = 1, if y=c

Partial constant relation: R(x, y) = 1, if the first half of y=c'

Bitcoin **need quantitative hardness**

Fixed point relation: R(x, y) = 1 if y=x (or prefix of x if |y|<|x|)

Fiat-Shamir: Long story ...

"Diagonal relation": $R^H(x, y) = 1$ if $y=x(x)$ [CGH 98]

"Small family relation": $R^H(x, y) = 1$ if $\exists\ h \in H$ s.t. $y=h(x)$ [this work]

Examples: Interesting sparse relations

Constant relation: $R(x, y) = 1$, if $y=c$

Partial constant relation: $R(x, y) = 1$, if the first half of $y=c'$

Bitcoin: Require moderately hardness

Fiat-Shamir for proofs: $R(a, b) = 1$ if $x \notin L$ and $\exists\ c$ s.t. $Verifier(x, a, b, c)=1$

Fixed point relation: $R(x, y) = 1$ if $y=x$ (or prefix of x if $|y|<|x|$)

"Diagonal relation": $R^H(x, y) = 1$ if $y=x(x)$

"Small family relation": $R^H(x, y) = 1$ if $\exists\ h \in H$ s.t. $y=h(x)$

Summary:

Goals: 1-input-output; capture as much relations as possible; Including Fiat-Shamir.

Examples: Interesting sparse relations

Constant relation: $R(x, y) = 1$, if $y=c$

Partial constant relation: $R(x, y) = 1$, if the first half of $y=c'$

Bitcoin: Require moderately hardness

Fiat-Shamir for proofs: $R(a, b) = 1$ if $x \notin L$ and $\exists$ c s.t. Verifier$(x, a, b, c)=1$

Fixed point relation: $R(x, y) = 1$ if $y=x$ (or prefix of x if $|y|<|x|$)

"Diagonal relation": $R^H(x, y) = 1$ if $y=x(x)$

"Small family relation": $R^H(x, y) = 1$ if $\exists$ $h \in H$ s.t. $y=h(x)$

Summary:

Goals: 1-input-output; capture as much relations as possible; Including Fiat-Shamir.
Constraints: the key must be longer than $|x|+|y|$, the seed must be longer than $|x|$.

Examples: Interesting sparse relations

Constant relation: $R(x, y) = 1$, if $y=c$

Partial constant relation: $R(x, y) = 1$, if the first half of $y=c'$

Bitcoin: Require moderately hardness

Fiat-Shamir for proofs: $R(a, b) = 1$ if $x \notin L$ and $\exists$ c s.t. $Verifier(x, a, b, c)=1$

Fixed point relation: $R(x, y) = 1$ if $y=x$ (or prefix of x if $|y|<|x|$)

"Diagonal relation": $R^H(x, y) = 1$ if $y=x(x)$

"Small family relation": $R^H(x, y) = 1$ if $\exists$ $h \in H$ s.t. $y=h(x)$

Summary:

Goals: 1-input-output; capture as much relations as possible; Including Fiat-Shamir.
Constraints: the key must be longer than $|x|+|y|$, the seed must be longer than $|x|$.
Obvious difficulty: prove "weird" relations.

Other concrete properties for RO-like hash functions:

Perfect one-wayness [Canetti 97, Canetti, Micciancio, Reingold 98]

Non-malleability [Boldyreva, Cash, Fischlin, Warinschi 09]

Magic Functions [Dwork, Naor, Reingold, Stockmeyer 03]

Entropy preservation [Barak, Lindell, Vadhan 04]

Seed-incompressible CI [Halevi, Myers, Rackoff 08]

Correlated-Input security [Goyal, O'Neill, Rao 11]

Universal Computational Extractor [Bellare, Hoang, Keelveedhi 13]

ELF [Zhandry 16]

Canetti, Chen, Reyzin (TCC 2016-A)

IO(Puncturable.PRF) is correlation intractable for bounded relations.

Assuming subexponential iO, subexponential owf, input-hiding obfuscation for evasive functions.

Canetti, Chen, Reyzin (TCC 2016-A)

IO(Puncturable.PRF) is correlation intractable for bounded relations.

Assuming subexponential iO, subexponential owf, input-hiding obfuscation for evasive functions.

Kalai, Rothblum, Rothblum (Crypto 2017)

IO(Puncturable.PRF) is correlation intractable for all sparse relations, therefore implies the soundness of Fiat-Shamir for proofs.

Assuming subexponential iO, subexponential owf, exponentially secure input-hiding obfuscation for multi-bit point functions.

Canetti, Chen, Reyzin (TCC 2016-A)

IO(Puncturable.PRF) is correlation intractable for bounded relations.

Assuming subexponential iO, subexponential owf, input-hiding obfuscation for evasive functions.

Kalai, Rothblum, Rothblum (Crypto 2017)

IO(Puncturable.PRF) is correlation intractable for all sparse relations, therefore implies the soundness of Fiat-Shamir for proofs.

Assuming subexponential iO, subexponential owf, exponentially secure input-hiding obfuscation for multi-bit point functions.

the existence of a sub-exponentially secure one-way function.

While the hash function we construct is far from practical, we believe that this is a first step towards instantiations that are both more efficient and provably secure. In addition, we show that this result resolves a long-lasting open problem in the study of zero-knowledge

Canetti, Chen, Reyzin (TCC 2016-A)

IO(Puncturable.PRF) is correlation intractable for bounded relations.

Assuming subexponential iO, subexponential owf, input-hiding c
functions.

Kalai, Rothblum, Rothblum (Crypto 2017)

IO(Puncturable.PRF) is correlation intractable for all sp
implies the soundness of Fiat-Shamir for proofs.

Assuming subexponential iO, subexponential owf, exponentially
obfuscation for multi-bit point functions.

the existence of a sub-exponentially secure one-way function.

While the hash function we construct is far from practical, we believe that this is a first step towards instantiations that are both more efficient and provably secure. In addition, we show that this result resolves a long-lasting open problem in the study of zero-knowledge

This work:

[Kalai, Rothblum, Rothblum ‘17] is right!

the existence of a sub-exponentially secure one-way function.

While the hash function we construct is far from practical, we believe that this is a first step towards instantiations that are both more efficient and provably secure. In addition, we show that this result resolves a long-lasting open problem in the study of zero-knowledge

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with ...

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties:

- For all key k*, a random ciphertext CT decrypts to a random message m.
- For all key k* and message m*, the following distributions are the same:

CT s.t. Dec(k*, CT) = m* and CT <- Enc(k*, m*)

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties:

- For all key k*, a random ciphertext CT decrypts to a random message m.
- For all key k* and message m*, the following distributions are the same:

$$\{CT \text{ s.t. } Dec(k^*, CT) = m^*\} \quad \text{and} \quad \{CT \leftarrow Enc(k^*, m^*)\}$$

(2) Exponentially hard KDM security for all key-dependency function f:

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties:

- For all key k*, a random ciphertext CT decrypts to a random message m.
- For all key k* and message m*, the following distributions are the same:

$$\text{CT s.t. } \text{Dec}(k^*, \text{CT}) = m^* \quad \text{and} \quad \text{CT} \leftarrow \text{Enc}(k^*, m^*)$$

(2) Exponentially hard KDM security for all key-dependency function f:

For all f (possibly inefficient), any polytime Adv, for all superpoly function s

$$\Pr_k[\, \text{Adv}(\, \text{Enc}(k, f(k))\,) \rightarrow k\,] < \frac{s(n)}{2^n} \text{ , where } n = \text{secp} = |k|$$

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties:

- For all key k*, a random ciphertext CT decrypts to a random message m.
- For all key k* and message m*, the following distributions are the same:

CT s.t. Dec(k*, CT) = m* and CT <- Enc(k*, m*)

(2) Exponentially hard KDM security for all key-dependency function f:

For all f (possibly inefficient), any polytime Adv, for all superpoly function s

$$\Pr_k[\ \mathrm{Adv}(\ \mathrm{Enc}(k, f(k))\ ) \rightarrow k\ ] < \frac{s(n)}{2^n}\ , \text{ where } n = \mathrm{secp} = |k|$$

We provide parameters where ElGamal and Regev encryptions plausibly satisfy that level of security.

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f:

For all f (possibly inefficient), any polytime Adv, for all superpoly function s

$$\Pr_k[\ \mathrm{Adv}(\ \mathrm{Enc}(k, f(k))\ ) \to k\ ] < \frac{s(n)}{2^n}\ , \text{ where } n = \mathrm{secp} = |k|$$

For the instantiations from ElGamal or Regev, the assumptions are morally:

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f:

For all f (possibly inefficient), any polytime Adv, for all superpoly function s

$$\Pr_k[\ \mathrm{Adv}(\ \mathrm{Enc}(k, f(k))\ ) \to k\ ] < \frac{s(n)}{2^n}\ , \text{ where } n = \mathrm{secp} = |k|$$

For the instantiations from ElGamal or Regev, the assumptions are morally:
Discrete-log where polynomial time adv succeeds with probability
no-better-than-guessing + KDM for any f.
LWE where polynomial time adv succeeds in key-recovery with probability
no-better-than-guessing + KDM for any f.

H(???...?)=000000….XYZ3d83h

The instantiations from ElGamal or Regev do not suffice for the quantitative correlation intractability required for Bitcoin (will explain later).

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties.

(2) Exponentially hard KDM security for all key-dependency function f:

For all f (possibly inefficient), any polytime Adv, for all superpoly function s

$$\Pr_k[\ \mathrm{Adv}(\ \mathrm{Enc}(k, f(k))\ ) \to k\ ] < \frac{s(n)}{2^n}\ , \text{ where } n = \mathrm{secp} = |k|$$

How do we get around the black-box lower bound of [Bitansky et al. 13]?

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f.

Corollary: under the same assumptions, we get
Fiat-Shamir for proof.

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f.

Corollary: under the same assumptions, we get
Fiat-Shamir for proof.
NIZK for NP. (soundness is from FS; zero-knowledge need to prove)

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f.

Corollary: under the same assumptions, we get
Fiat-Shamir for proof.
NIZK for NP.
[Reingold, Rothblum, Rothblum ‘16]: Constant round doubly efficient IP for any language computable in polytime and fixed polynomial space.
=> a non-interactive one
(for non-interactive delegation with public verifiability, previous results assume RO, or iO, or “mmaps-looking FHE” by [Paneth-Rothblum ‘17])

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f.

Construction of the hash function.

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with

(1) Natural statistical properties.

(2) Exponentially hard KDM security for all key-dependency function f.

Construction of the hash function.

Keygen: $CT \leftarrow_r CTspace$.

$H: \{0,1\}^n \rightarrow \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

This work:

Correlation intractability for all sparse relations from symmetric encryption schemes with
(1) Natural statistical properties.
(2) Exponentially hard KDM security for all key-dependency function f.

Construction of the hash function.
Keygen: $CT \leftarrow_r CTspace$.
$H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

[Fiat-Shamir, '86]

3.2 Security

The formal proof of security in this extended abstract assumes that n is sufficiently large and that f is a truly random function. Consequently, there can be no generic attack which breaks the scheme for any n and f unless factoring is easy. Practical implementations which use particular moduli n_0 and psuedo-random functions f_0 may still be vulnerable to specialized attacks, but they mearly show that n_0 is too small or that f_0 is demonstrably non-random. When n_0 is at least 512 bits long and f_0 is sufficiently strong (*e.g.*, multiple DES with a fixed cleartext and variable key), such attacks are quite unlikely.

Explain the assumption.
Explain the proof idea.
Possible relaxations & generalizations.

Instantiation via Regev's scheme.

Parameters: n = security parameter, B = O(n), $q=O(n^3)$, n' s.t. $(2B+1)^{n'} = 2^n$

sk: $s \in [-B, B]^{n'}$;

Enc(sk, b): sample $a \in Z_q^n$; sample $e \in [0, q/2)$. CT: a, $z = \langle a,s \rangle + b \cdot q/2 + e$

Dec(sk, CT): $z - \langle a,s \rangle$, then round by 2.

Instantiation via Regev's scheme.

Parameters: n = security parameter, B = O(n), q=O(n^3), n' s.t. $(2B+1)^{n'} = 2^n$
sk: $s \in [-B, B]^{n'}$;
Enc(sk, b): sample $a \in Z_q^n$; sample $e \in [0, q/2)$. CT: a, $z = \langle a,s \rangle + b \cdot q/2 + e$
Dec(sk, CT): $z - \langle a,s \rangle$, then round by 2.

Hash function (to l bit output): $H_{A,Z}(x)$: $Z - Ax \in Z_q^l$, then round by 2.
(it is not a RO-like function due to approx. linearity.)

Instantiation via Regev's scheme.

Parameters: n = security parameter, B = O(n), q=O(n^3), n' s.t. $(2B+1)^{n'} = 2^n$
sk: $s \in [-B, B]^{n'}$;
Enc(sk, b): sample $a \in Z_q^n$; sample $e \in [0, q/2)$. CT: a, $z = \langle a,s \rangle + b \cdot q/2 + e$
Dec(sk, CT): $z - \langle a,s \rangle$, then round by 2.

Hash function (to l bit output): $H_{A,Z}(x)$: $Z - Ax \in Z_q^l$, then round by 2.
(it is not a RO-like function due to approx. linearity.)

Statistical properties are easy to verify.

Instantiation via Regev's scheme.

Parameters: n = security parameter, B = O(n), q=O(n^3), n' s.t. $(2B+1)^{n'} = 2^n$
sk: $s \in [-B, B]^{n'}$;
Enc(sk, b): sample $a \in Z_q^n$; sample $e \in [0, q/2)$. CT: a, $z = \langle a,s \rangle + b \cdot q/2 + e$
Dec(sk, CT): $z - \langle a,s \rangle$, then round by 2.

Hash function (to l bit output): $H_{A,Z}(x)$: $Z - Ax \in Z_q^l$, then round by 2.
(it is not a RO-like function due to approx. linearity.)

Statistical properties are easy to verify.
Exponential KDM assumption: hard for polynomial time algorithms to find s with better-than-guessing probability given a, $y = \langle a,s \rangle + b \cdot q/2 + e$.

Instantiation via Regev's scheme.

Parameters: n = security parameter, B = O(n), q=O(n^3), n' s.t. $(2B+1)^{n'} = 2^n$
sk: $s \in [-B, B]^{n'}$;
Enc(sk, b): sample $a \in Z_q^n$; sample $e \in [0, q/2)$. CT: a, $z = \langle a,s \rangle + b \cdot q/2 + e$
Dec(sk, CT): $z - \langle a,s \rangle$, then round by 2.

Hash function (to l bit output): $H_{A,Z}(x)$: $Z - Ax \in Z_q^l$, then round by 2.
(it is not a RO-like function due to approx. linearity.)

Statistical properties are easy to verify.
Exponential KDM assumption: hard for polynomial time algorithms to find s with better-than-guessing probability given a, $y = \langle a,s \rangle + b \cdot q/2 + e$.

Lattice open problem: find a polynomial time algorithm for LWE with polynomial modulus that achieves better-than-guessing success probability.

Instantiation via ElGamal

If we don’t worry about KDM for a moment, then it is discrete log.

Instantiation via ElGamal

If we don't worry about KDM for a moment, then it is discrete log.

Any polynomial time algorithm success with probability $poly(n)/2^n$

Is there a group where this assumption is plausible?

Instantiation via ElGamal

If we don't worry about KDM for a moment, then it is discrete log.

Any polynomial time algorithm success with probability $\text{poly}(n)/2^n$

Is there a group where this assumption is plausible?

Discrete log over finite field (of size roughly 2^n)	T(n) time, ~ 1 success probability	Polytime, P success probability
Pollard's rho algorithm	$T(n) = \exp(n/2)$	$P = \text{poly}(n) \cdot 2^{-n}$
Index calculus algorithm	$T(n) = \exp(n^{1/3} (\log n)^{2/3})$	???

The success prob. of index calculus in polynomial time:

We don't know how to achieve
better-than-guessing success probability.

Exercise: find a polytime algorithm for discrete-log over finite field that achieves better-than-guessing success probability.

In the online-offline model:

The offline phase gets g, F_q, runs infinite time, keep polysize advice.

The online phase gets $h = g^x$, and the advice, runs in polytime.

Claim: index-calculus achieves $2^{-n/C}$ success probability for any constant C.

In the online-offline model:
The offline phase gets g, F_q, runs infinite time, keep polysize advice.
The online phase gets $h = g^x$, and the advice, runs in polytime.
Claim: index-calculus achieves $2^{-n/C}$ success probability for any constant C.

How:
The offline phase gets g, F_q, picks a polynomial bound $B=n^C$, computes all the
$\log_g(p)$ for p in {2, 3, 5, 7, …, B}

The online phase gets $h = g^x$, picks a random r, compute $w = h \cdot g^r = g^{x+r} \bmod q$.
Then see if all the factors of w are below B. [Rankin 1938: $2^{-n/C}$]
If so, $w = 2^{x2} \cdot 3^{x3} \cdot 5^{x5} \cdot \ldots \cdot B^{xB}$
Then $x = \log_g(2) \cdot x2 + \log_g(3) \cdot x3 + \ldots + \log_g(B) \cdot xB - r \bmod \varphi(q)$.

Used also in the Logjam attack [Adrian et al., CCS15]

Discrete log over elliptic curve groups:

("bad" means subexponential time algorithms are known)
MOV 93: supersingular is bad.
ADH 94: hyperelliptic is bad.
GHS 02: Composite order extension over F_2 is bad.
Semeav 04: Summation polynomial is useful but I don't know how.
Gaudry 09 and Diem 11: Yes it is useful to attack some fields.
…
Still no algorithm achieves non-trivial running time or success probability for $E(F_q)$ where q is a prime, $\#E(F_q)$ has a large prime factor.

Discrete log over elliptic curve groups:

("bad" means subexponential time algorithms are known)
MOV 93: supersingular is bad.
ADH 94: hyperelliptic is bad.
GHS 02: Composite order extension over F_2 is bad.
Semeav 04: Summation polynomial is useful but I don't know how.
Gaudry 09 and Diem 11: Yes it is useful to attack some fields.
…
Still no algorithm achieves non-trivial running time or success probability for $E(F_q)$ where q is a prime, $\#E(F_q)$ has a large prime factor.

For discrete-log problem, Shoup 97 showed generic algorithm that runs in time T can only achieve success prob $T^2/2^n$

Discrete log over elliptic curve groups:

("bad" means subexponential time algorithms are known)
MOV 93: supersingular is bad.
ADH 94: hyperelliptic is bad.
GHS 02: Composite order extension over F_2 is bad.
Semeav 04: Summation polynomial is useful but I don't know how.
Gaudry 09 and Diem 11: Yes it is useful to attack some fields.
…
Still no algorithm achieves non-trivial running time or success probability for $E(F_q)$ where q is a prime, $\#E(F_q)$ has a large prime factor.

For discrete-log problem, Shoup 97 showed generic algorithm that runs in time T can only achieve success prob $T^2/2^n$

For Bitcoin, T time ~ T^2 success probability doesn't suffice for "decentralize"

Explain the assumption. ✓

Explain the proof idea.

Possible relaxations & generalizations.

Proof idea:

Again, [Kalai, Rothblum, Rothblum ‘17] is right!

Proof idea:

Again, [Kalai, Rothblum, Rothblum ‘17] is right!

Translate correlation intractability where many possible x can be the right answer, to some property where only one x* is the right answer.

Proof idea:

Again, [Kalai, Rothblum, Rothblum ‘17] is right!

Translate correlation intractability where many possible x can be the right answer, to some property where only one x* is the right answer. Cost an exponential loss in the success probability in the underlying assumption.

Proof idea:

Again, [Kalai, Rothblum, Rothblum ‘17] is right!

Translate correlation intractability where many possible x can be the right answer, to some property where only one x* is the right answer. Cost an exponential loss in the success probability in the underlying assumption.

KRR17: some property = finding the input in the Input-hiding obfuscation. Transition is done by iO + puncturing (computational).

Proof idea:

Again, [Kalai, Rothblum, Rothblum ‘17] is right!

Translate correlation intractability where many possible x can be the right answer, to some property where only one x* is the right answer. Cost an exponential loss in the success probability in the underlying assumption.

KRR17: some property = finding the input in the Input-hiding obfuscation. Transition is done by iO + puncturing (computational).

This work: some property = key recovery in a KDM ciphertext. Transition is done via statistical properties.

Keygen: $CT \leftarrow_r CTspace$. $H: \{0,1\}^n \rightarrow \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

From any R, pad it to R' so that
(1) R' is still sparse,
(2) every x has almost equal number of y.

Keygen: $CT \leftarrow_r CTspace$. $H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

Fix a sparse relation R with density d

0. Suppose by contradiction:
$Pr_{CT}[\ Adv(CT) \to k \text{ and } (k, Dec(k, CT)) \in R\ ] > v.$

Keygen: $CT \leftarrow_r$ CTspace. $H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

Fix a sparse relation R with density d

0. Suppose by contradiction:
$Pr_{CT}[\ Adv(CT) \to k \text{ and } (k, Dec(k, CT)) \in R\ ] > v.$

1. Averaging over one of the 2^n possible inputs k*:
$Pr_{CT, k^*}[\ Adv(CT) \to k \text{ and } k=k^* \text{ and } (k^*, Dec(k^*, CT)) \in R\ ] > v \cdot 2^{-n}.$

Keygen: $CT <-_r$ CTspace. $H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

Fix a sparse relation R with density d

0. Suppose by contradiction:
$Pr_{CT}[\ Adv(CT) \to k \text{ and } (k, Dec(k, CT)) \in R\ ] > v.$

1. Averaging over one of the 2^n possible inputs k*:
$Pr_{CT,\, k^*}[\ Adv(CT) \to k \text{ and } k=k^* \text{ and } (k^*, Dec(k^*, CT)) \in R\ ] > v \cdot 2^{-n}.$

2. k*, m* randomly, CT' s.t. Dec(k*, CT')=m*

-- For all k*, a random CT decrypts to a random msg.

$Pr_{k^*, m^*, CT'}[\ Adv(CT') \to k \text{ and } k=k^* \text{ and } (k^*, m^*) \in R\ ] > v \cdot 2^{-n}.$

Keygen: $CT <-_r$ CTspace. $H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

Fix a sparse relation R with density d

0. Suppose by contradiction:
$Pr_{CT}[\ Adv(CT) \to k \text{ and } (k, Dec(k, CT)) \in R\ ] > v.$

1. Averaging over one of the 2^n possible inputs k*:
$Pr_{CT, k^*}[\ Adv(CT) \to k \text{ and } k=k^* \text{ and } (k^*, Dec(k^*, CT)) \in R\ ] > v \cdot 2^{-n}.$

2. k*, m* randomly, CT' s.t. Dec(k*, CT')=m*

-- For all k*, a random CT decrypts to a random msg.

$Pr_{k^*, m^*, CT'}[\ Adv(CT') \to k \text{ and } k=k^* \text{ and } (k^*, m^*) \in R\ ] > v \cdot 2^{-n}.$

-- conditional probability

3. k* randomly, m* s.t. $(k^*, m^*) \in R$; CT' s.t. Dec(k*, CT')=m*
$Pr_{(k^*, m^*) \in_r R, CT'}[\ Adv(CT'=Enc(k^*, m^*)) \to k^*\ ] > (v/d) \cdot 2^{-n}.$

Keygen: $CT <-_r$ CTspace. $H: \{0,1\}^n \to \{0,1\}^l$, $H_{CT}(k) = Dec(k, CT)$.

Fix a sparse relation R with density d

0. Suppose by contradiction:
$Pr_{CT}[\ Adv(CT) \to k \text{ and } (k, Dec(k, CT)) \in R\ ] > v.$

1. Averaging over one of the 2^n possible inputs k*:
$Pr_{CT, k^*}[\ Adv(CT) \to k \text{ and } k=k^* \text{ and } (k^*, Dec(k^*, CT)) \in R\ ] > v \cdot 2^{-n}.$

2. k*, m* randomly, CT' s.t. Dec(k*, CT')=m*

-- For all k*, a random CT decrypts to a random msg.

$Pr_{k^*, m^*, CT'}[\ Adv(CT') \to k \text{ and } k=k^* \text{ and } (k^*, m^*) \in R\ ] > v \cdot 2^{-n}.$

-- conditional probability

3. k* randomly, m* s.t. $(k^*, m^*) \in R$; CT' s.t. Dec(k*, CT')=m*
$Pr_{(k^*, m^*) \in_r R, CT'}[\ Adv(CT'=Enc(k^*, m^*)) \to k^*\ ] > (v/d) \cdot 2^{-n}.$

-- Exponential KDM

-- For all k*,m*, CT'=Enc(k*,m*) is stat. close to CT' s.t. Dec(k*, CT')=m*

Possible relaxations & generalizations.

Bigger success probability?
Weaker KDM assumptions?

What if someone achieves success prob. $2^{-n/10}$ for DLOG over $E(F_q)$?

What if someone achieves success prob. $2^{-n/10}$ for DLOG over $E(F_q)$?

Impact on DLOG over elliptic curve groups:

must explore additional structure, due to [Shoup 97]

What if someone achieves success prob. $2^{-n/10}$ for DLOG over $E(F_q)$?

Impact on DLOG over elliptic curve groups:

must explore additional structure, due to [Shoup 97]

Impact on our result.

Fun fact: $2^{-n/10} >> \text{poly}(n) \cdot 2^{-n}$, ☹

Condition to form a contradiction:

$2^{-n/10} < \text{non-negl}/(d \cdot 2^n) \Rightarrow d > 2^{-9n/10}$

=> the density of a non-trivial relation has to be larger than $2^{-9n/10}$

=> |output| > 9n/10, where n = |input|

=> Implies e.g. Fiat-Shamir for protocols where

|first msg| = n, |second msg|=9n/10, soundness = $2^{-9n/10}$

To imply general Fiat-Shamir, we need KDM for unbounded key-dependency functions.

To imply general Fiat-Shamir, we need KDM for unbounded key-dependency functions.

Schemes that supports KDM for affine functions
([Boneh, Halevi, Hamburg, Ostrovsky 08], etc.)
Schemes that bootstrap to bounded functions
([Barak, Haitner, Hofheinz, Ishai 10], etc.)

To imply general Fiat-Shamir, we need KDM for unbounded key-dependency functions.

Schemes that supports KDM for affine functions
([Boneh, Halevi, Hamburg, Ostrovsky 08], etc.)
Schemes that bootstrap to bounded functions
([Barak, Haitner, Hofheinz, Ishai 10], etc.)

We don’t know any method that implies unbounded KDM and at the same time guarantees exponentially hard key-recovery.

To imply general Fiat-Shamir, we need KDM for unbounded key-dependency functions.

Schemes that supports KDM for affine functions
([Boneh, Halevi, Hamburg, Ostrovsky 08], etc.)
Schemes that bootstrap to bounded functions
([Barak, Haitner, Hofheinz, Ishai 10], etc.)

We don't know any method that implies unbounded KDM and at the same time guarantees exponentially hard key-recovery.

Assuming KDM security for
Enc(k, m) = Ext(owf(k)) + m?

Summary:

Correlation intractability for all sparse relations (implies Fiat-Shamir) from symmetric encryption schemes with
(1) Natural statistical properties;
(2) Exponentially hard KDM security for all key-dependency function f.

Multiple-input-output relations

Quantitatively correlation intractable.

Future directions

Correlation intractability for multiple-input-output relations.

It will be more useful.

For example, lots of Fiat-Shamir application in practices starts from an argument.

As another example, Gennaro-Halevi-Rabin signature.

Of course, need to bypass the impossibility results by CGH and Goldwasser-Kalai.

Maybe start from “simple” relations.

Quantitative correlation intractability.

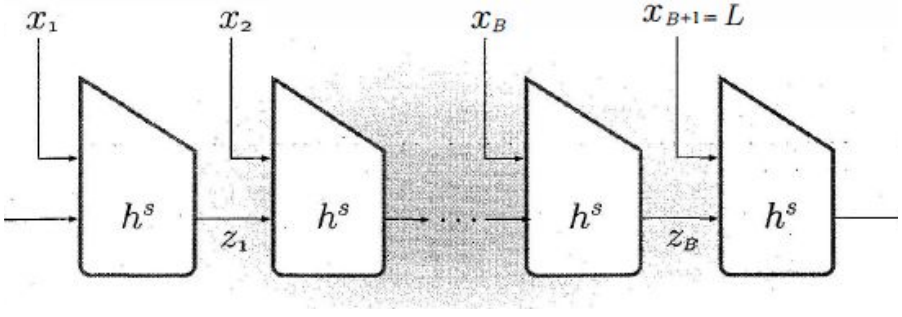
[Ball, Rosen, Sabin, Vasudevan 17] “proof of useful work”, can we do it for hash functions?

The CGH impossibility results for f: $\{0,1\}^* \rightarrow \{0,1\}^m$ also holds for relatively sparse CI.

Corollary: Domain extension techniques (like Merkle-Damgard) might preserve collision resistance, but do not preserve correlation intractability.

In fact, AsicBoost takes advantage of Merkle-Damgard to speed up bitcoin mining.
*(but that’s not because of CGH impossibility)

Achieve f: $\{0,1\}^* \rightarrow \{0,1\}^m$ that is quantitatively CI for “bitcoin relation”?




Thanks for
your time!