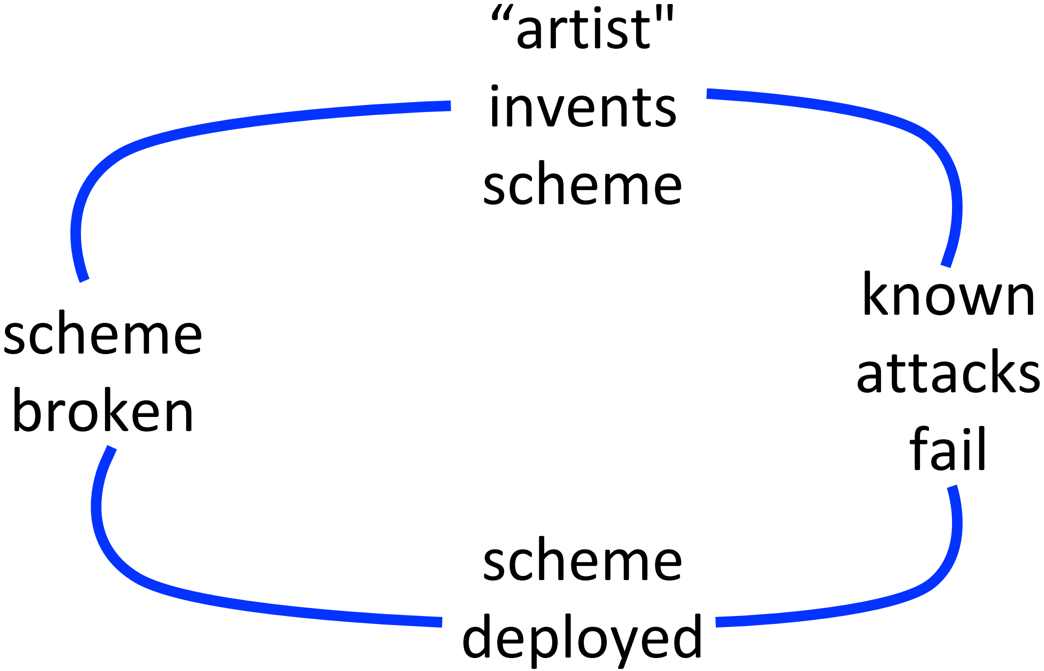
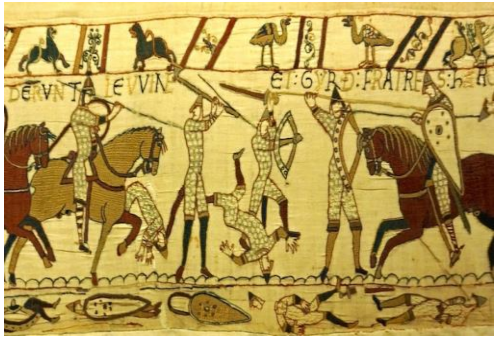
On One-way Functions and Kolmogorov Complexity

Yanyi Liu and Rafael Pass
Cornell Tech and Cornell University

The “Dark Ages” Crypto Cycle

(the last 2000 years)

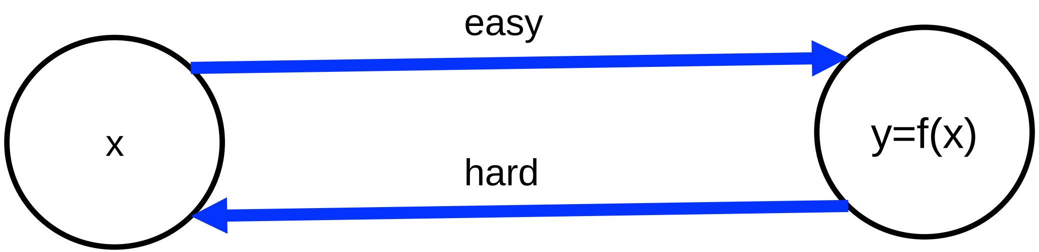




One-way Functions (OWF) [Diffie-Hellman'76]

A function **f** that is

- **Easy to compute**: can be computed in poly time
- **Hard to invert**: no PPT can invert it, even with "small" probability



Ex [Factoring]: use x to pick 2 random "large" primes p,q, and output y = p* q

One-way Functions (OWF) [Diffie-Hellman’76]

A function f that is

- **Easy to compute**: can be computed in poly time
- **Hard to invert**: no PPT can invert it

OWF both necessary [IL’89] and sufficient for:

- Private-key encryption [GM84,HILL99]
- Pseudorandom generators [HILL99]
- Digital signatures [Rompel90]
- **Authentication schemes** [FS90]
- Pseudorandom functions [GGM84]
- Commitment schemes [Naor90]
- Coin-tossing [Blum’84]
- ZK proofs [GMW89]
- …

Not included:
public-key encryption, OT, obfuscation

Whether OWF exists is the most important problem in Cryptography

OWF v.s NP Hardness

Observation: OWF => NP $\notin$ BPP

"Holy grail" [DH'76]

Prove: NP $\notin$ BPP => OWF

In the absence of the holy-grail…

~~Discrete Logarithm Problem [DH'76]~~

~~Factoring [RSA'83]~~

Lattice Problems [Ajtai'96]

DES,
SHA,
AES…

QUANTUM COMPUTERS

So far, not broken…but for how long?
"Cryptographers seldom sleep well" - Micali'88

Have we really escaped from the "crypto cycle"?

In the absence of the holy-grail…

Discrete Logarithm Problem [DH’76]

Factoring [RSA’83]

Lattice Problems [Ajtai’96]

DES,
SHA,
AES…

Central question: Does there exist some **natural average-case hard problem** (a “master problem”) that **characterizes existence of OWF?**

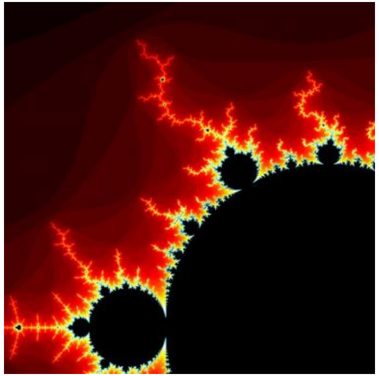
Main Theorem

For every polynomial t(n)>1.1n:

OWFs exist iff **t-bounded Kolmogorov-complexity** is mildly hard-on-average

Deep Connection between Cryptography and Kolmogorov Complexity; the **central problems in these fields are connected!**

Kolmogorov Complexity [Sol'64,Kol'68,Cha'69]



Which of the following strings is more "random":

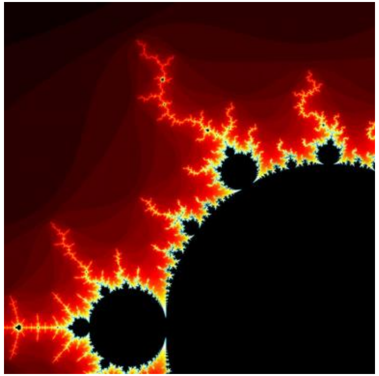
- 123123123123123123
- 173054445934739403

K(x) = length of the shortest program that outputs **x**

Formally, we fix a universal TM U, and are looking for the length of the shortest program Π = (M,w) s.t. U(M,w) = x

Lots of amazing applications (e.g., Godel's incompleteness theorem)
But **uncomputable**.

Time-Bounded Kolmogorov Complexity



Which of the following strings is more "random":

- 123123123123123123
- 1730544459347394037

$K(x)$ = length of the shortest program that outputs **x**

$K^t(x)$ = length of the shortest program that outputs **x** within time **t(|x|)**

Can K^t be **efficiently computed** when **t** is a polynomial?

- Studied in the Soviet Union since 60s [Kol'68,T'84]
- Independently by Hartmanis [83], Sipser [83], Ko [86]
- Closely related to **MCSP** (Minimum Circuit Size Problem) [T'84,KC'00]

Average-case Hardness of K^t

Frequential version [60's, T'84]

Does ∃ algorithm that computes **$K^t(x)$ for a "large" fraction of x's**?

Observation [60's, T'84]: **K^t** can be approximated within d log n w.p $1-1/n^d$

Proof: simply output n.

Average-case Hardness of K^t

Frequential version [60's, T'84]
Does ∃ algorithm that computes **$K^t(x)$ for a "large" fraction of x's**?

Observation [60's, T'84]: **K^t** can be approximated within d log n w.p $1-1/n^d$
Proof: simply output n.

Def: **K^t** is **mildly-HOA** if there exists a polynomial p, such that no PPT heuristic H can compute **K^t** w.p 1-1/p(n) over random strings x for inf many n.

Def: **K^t** is **mildly-HOA to c-approximate** if there exists a polynomial p, such that no PPT heuristic H can c-approximate **K^t** w.p 1-1/p(n) over random strings x for inf many n.

Main Theorem

The following are equivalent:

1. **OWFs** exist
2. ∃ poly $t(n)>0$, s.t. **K^t is mildly-HOA**.
3. ∀ $c>0$, $\varepsilon>0$, poly $t(n)>(1+\varepsilon)\,n$,
   K^t is mildly-HOA to $(c\log n)$-approx.

Main Theorem

The following are equivalent:

1. **OWFs** exist
2. ∃ poly $t(n)>0$, s.t. **K^t is mildly-HOA**.
3. ∀ $c>0$, $\varepsilon>0$, poly $t(n)>(1+\varepsilon)n$,
   K^t is mildly-HOA to $(c\log n)$-approx.

Corr [Crypto v.s. K-complexity]: For all poly $t(n)>(1+\varepsilon)n$,
OWFs exist iff K^t is mildly hard-on-average

Corr [New insight into K-complexity]: For all $c>0$, $\varepsilon>0$, poly $t(n)>(1+\varepsilon)n$,
K^t is mildly hard-on-average to $(c\log n)$-approx iff K^t is mildly hard-on-average.

Main Theorem

The following are equivalent:

1. **OWFs** exist
2. ∃ poly $t(n)>0$, s.t. **K^t is mildly-HOA**.
3. ∀ $c>0$, $\varepsilon>0$, poly $t(n)>(1+\varepsilon)\, n$,
   K^t is mildly-HOA to (clog n)-approx.

Proof: (2) => (1) => (3)

Today: just sketch idea behing (2) => (1)
(1) => (3) is the harder direction (in the paper)

Theorem 1

Assume there exists some poly $t(n)>0$, s.t. **K^t is mildly-HOA**.
Then OWFs exist.

Weak OWF: "mild-HOA version" of a OWF:
efficient function f s.t. no PPT can invert f w.p. **1-1/p(n)**
for inf many n, for some poly $p(n)>0$.

Lemma [Yao'82]. If a Weak OWF exists, then a OWF exists.

So, we just need to construct a weak OWF.

The OWF Construction:

Let **t** be a (polynomial) time-bound (the time-bound from the K-complexity problem)
Let **c** be a constant so that **$K^T(x) < |x|+c$ for all x**

Define **$f(\Pi',i)$** where $|\Pi'| = n+c$, $|i| = \log(n+c)$ as follows:

- Let $\Pi = [\Pi']_{1\to i}$ = first i bits of Π'.
- Run Π for at most **t(n)** steps;
  let **y** denote its output
- Output i||y.

Reduction idea: if an PPT attacker A inverts f w.h.p, then we can compute the K^T-complexity of random strings y, by feeding **(1,y), (2,y), .. (n+c,y)** to A and see which work.

Proving this works is a bit non-trivial since we feed A the wrong distribution!

In OWF experiment
(where A works):

i $\leftarrow$ $\mathbf{U_{log(n+c)}}$
y $\leftarrow$ output of a random program
of length i

In the emulation by H in $\mathbf{K^t}$ experiment
(where we need to ***prove*** that A works):

i $\leftarrow$ $\mathbf{K^t(y)}$
y $\leftarrow$ $\mathbf{U_n}$

No reason to believe that the output of a random program will be close to uniform!

But: using a counting argument, we can show that they are not too far in **relative distance (details in the paper)**

Main Theorem

For all ε>0, all poly t(n)>(1+ε)n
OWFs exist iff **K^t is mildly-HOA**.

First natural avg-case problem characterizing the feasibility of the basic tasks in Crypto
(i.e., private-key encryption, digital sigs, PRGs, PRFs, commitments, authentication, ZK…)

Identified a natural "**master-problem**" for Cryptography:

Non-trivial crypto is possible iff Kt is hard.

Golden time for Crypto and K-complexity

- **Sublinear time** average-case hardness of K-complexity problems suffice to characterize **subexponential/qpoly OWF** [LP’21]
- Characterize **OWF in logspace, NC0** [RS’21,LP’21]
- Characterize OWF [LP’21], resp. NC0-OWFs [Allender et al’ 21], though **NP-complete problems**
- **Unbounded K-complexity** sometimes suffices [Ilango-Ren-Santhanam’21], and even just **sparse languages** [LP’21]
- [LP’21] argued a potential approach of basing **OWF** on **EXP** $\neq$ **BPP**

Thank You