CRYPTOGRAPHY

Which Computational Universe Do We Live In?

By ERICA KLARREICH

April 18, 2022

Cryptographers want to know which of five possible worlds we inhabit, which will reveal whether truly secure cryptography is even possible.

2

Olena Shmahalo for Quanta Magazine

Many computer scientists focus on overcoming hard computational problems. But there's one area of computer science in which hardness is an asset: cryptography, where y hard obstacles between your adversaries and your secrets.

Unfortunately, we don't know whether secure cryptography truly exists. Over millennia, created ciphers that seemed unbreakable right until they were broken. Today, our interne and state secrets are guarded by encryption methods that seem secure but could conceivab moment.

To create a truly secure (and permanent) encryption method, we need a computational problem that's hard enough to create a provably insurmountable barrier for adversaries. We know of many computational problems that seem hard, but maybe we just haven't been clever enough to solve them. Or maybe some of them are hard, but their hardness isn't of a kind that lends itself to secure encryption. Fundamentally, cryptographers wonder: Is there enough hardness in the universe to make cryptography possible?

In 1995, Russell Impagliazzo of the University of California, San Diego broke down the question of hardness into a set of sub-questions that computer scientists could tackle one piece at a time. To summarize the state of knowledge in this area, he described five possible worlds — fancifully named Algorithmica, Heuristica, Pessiland, Minicrypt and Cryptomania — with ascending levels of hardness and cryptographic possibility. Any of these could be the world we live in.

Algorithmica

In this world, the most natural computational questions are all easy, which makes cryptography impossible. Here, the set of problems with efficient solutions — a set called P — doesn't just contain the problems we've already figured out how to solve. It also includes all the problems in another set called NP, which consists of the problems for which it's easy to check a proposed solution if someone hands it to you.

On the face of it, P and NP feel like different categories. For example, take the problem of deciding whether your suitcases will hold all the items you want to pack for a trip. If a friend packs for you, it's easy to verify whether they've fit everything in — just check for any items they've missed. So the suitcase-packing problem is in NP. But packing the suitcases yourself is much harder — you might have to try many different arrangements. It's not clear whether there's an efficient algorithm that solves this problem for all possible combinations of items and suitcases. That is, we don't know whether this problem is in P.

The problem of decrypting an encryption scheme is also in NP. After all, if you have an encrypted message and a friend claims to have decrypted it, you can check by feeding their decrypted message into the encryption machine and seeing whether the output matches your original encrypted message. (Of course, you must possess one of the encryption machines to do this, but cryptographers don't consider a scheme secure unless it can withstand attacks from an enemy who gets hold of one of the machines.)

In Algorithmica, P and NP are the same set of problems. A proof of this would be an algorithmic bonanza, since it would mean there are fast algorithms for things like suitcase packing and all the other seemingly hard problems in NP. But it would be a disaster for cryptographers, since one of the problems we'd be able to solve efficiently is decryption.

Most computer scientists believe that P is different from NP, for the simple reason that there are so many problems in NP that we've been unable to solve efficiently. But no one has ever been able to prove (or disprove) this, even though the "P versus NP" question has been considered the most famous problem in theoretical computer science for five decades. Then again, said Yuval Ishai of the Technion in Haifa, Israel, "apart from the constant failure of the smartest people, we have no evidence that it's hard to show that P is not equal to NP."

Heuristica

In this world, there are problems in NP that aren't easy to solve, but every problem in NP is easy "on average," meaning it can be solved efficiently in most cases. For example, if we're in Heuristica, then there exists an efficient suitcase-packing algorithm that nearly always succeeds, but that might fail for a few rare combinations of suitcases and items to pack. (These fast and usually successful algorithms are commonly called "heuristics.")

From the point of view of cryptography, there's not a big difference between Heuristica and Algorithmica. If we come up with an encryption scheme in Heuristica, there will be some inefficient decryption method that can handle nearly every message, making the scheme useless for practical purposes.

Pessiland

This is the worst of all possible worlds. In Pessiland, some problems in NP are hard even on average. For these problems, any efficient algorithm will fail not just occasionally but often. Yet these hard problems are not of a kind that is useful for hiding secret information.

"We definitely don't want to live in Pessiland," said Eric Allender of Rutgers University. "Here we get all the bad aspects of [computational] complexity, but without any of the advantages like cryptography."

Minicrypt

In this world, some problems in NP are hard on average, and this hardness is enough to build t
fundamental building block of cryptography: a "one-way function," which is a function th
carried out efficiently but can't be reversed efficiently. Cryptographers have shown that s
cryptography requires one-way functions. And if we have them, we get an array of crypto
goodies, such as secret key encryption, digital signatures and pseudorandom number gen

"Whether one-way functions exist is, without question, the most important problem in cryptography," said Rafael Pass of Cornell University and Cornell Tech. "If we don't have them, all these things can be broken."

Cryptomania

In this world, we have enough hardness to create everything in Minicrypt plus even more advanced cryptographic protocols such as public key encryption (in which people can send encrypted messages without knowing the secret key).

Eliminating Worlds

Most cryptographers, Ishai said, believe that at least some cryptography does exist, so we likely live in Cryptomania or Minicrypt. But they don't expect a proof of this anytime soon. Such a proof would require ruling out the other three worlds — and ruling out Algorithmica alone already requires solving the "P versus NP" problem, which computer scientists have struggled with for decades.

Recently, though, Pass and his doctoral student Yanyi Liu found a new approach to sifting through these worlds. For the first time, they identified a natural problem — called time-bounded Kolmogorov complexity, or K^t for short — whose difficulty level creates a bright dividing line between the worlds that include cryptography and the worlds that don't. If K^t is easy on average, Liu and Pass showed, then secure cryptography cannot exist, so we're in Algorithmica, Heuristica or Pessiland. But if K^t is hard on average, then we can make one-way functions, so we're at least in Minicrypt, and possibly Cryptomania.

This new result means computer scientists can eliminate Pessiland — the worst world — if they can prove one more statement: If K^t is easy on average, then so is every problem in NP. In that case, we'll have narrowed things down to the worlds where K^t is hard on average (Minicrypt and Cryptomania) and the ones where K^t — and every other problem in NP — is easy on average (Algorithmica and Heuristica).

Researchers have been chipping away at Pessiland for some time, said Ryan Williams of the Massachusetts Institute of Technology. "I think the general consensus is that Pessiland can be ruled out, but whether we're going to do that sooner or later, I don't know."

Cryptographers would really like to eliminate Heuristica as well, which would involve proving that if K^t is easy on average then every problem in NP is easy in all cases (not just on average). Ruling out these two worlds would mean that either we live in Algorithmica, where everything is easy all the time, or we have enough hardness for basic cryptography.

Cryptographers widely refer to this goal as the field's holy grail. Ishai doesn't expect to see it p
his lifetime, but even that is uncertain. "When hard problems are cracked, sometimes we se
but sometimes we don't," he said. "Definitely our best shot is this new line of work."