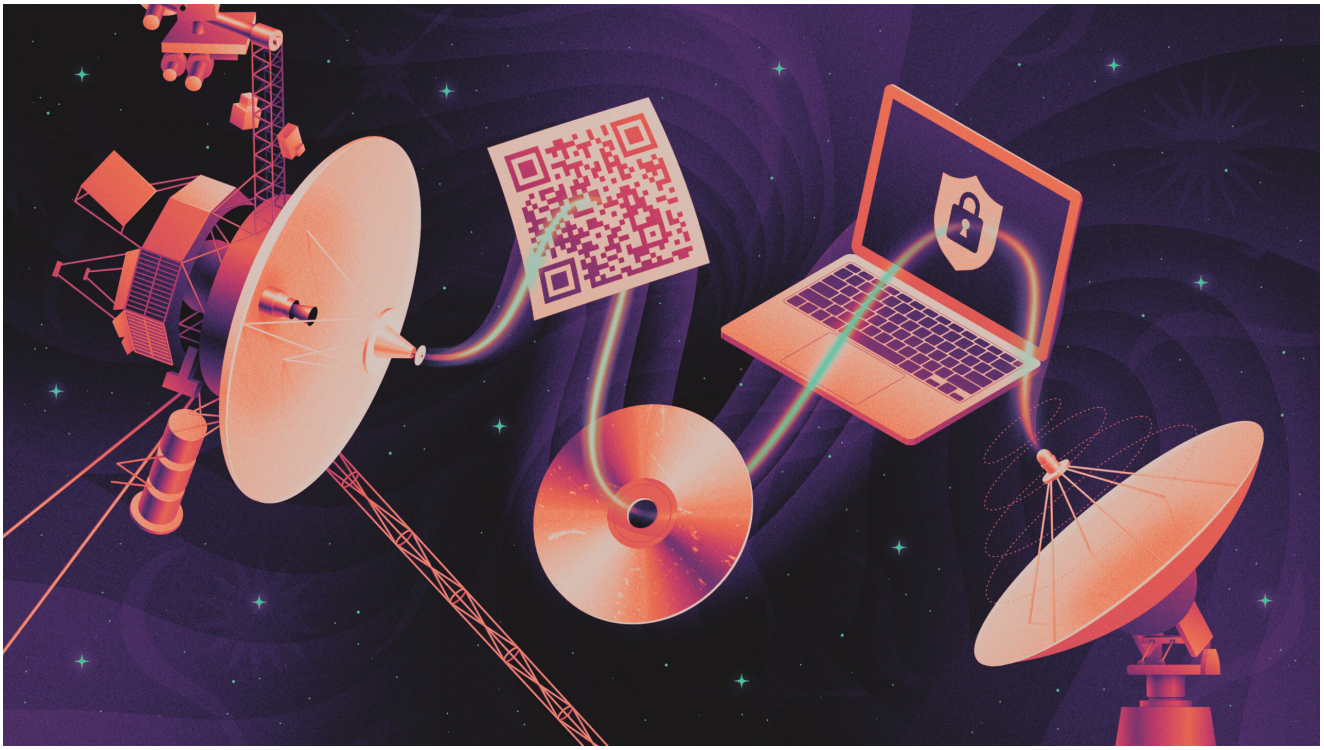
How Mathematical Curves Power Cryptography

quantamagazine.org/how-mathematical-curves-power-cryptography-20220919

By Jonathan O'Callaghan September 19, 2022



Reed-Solomon codes help power an enormous number of information technologies.

Kristina Armitage/Quanta Magazine

Introduction

Given a collection of points in space, can you find a certain type of curve that passes through all of them? This question — a version of what's called the interpolation problem — has interested mathematicians since antiquity. Earlier this year, the mathematicians Eric Larson and Isabel Vogt solved it completely.

But while the work has generated a lot of excitement among pure mathematicians, interpolation has practical consequences that extend far beyond the realm of geometry. Interpolation is central to storing and communicating electronic data, constructing cryptographic schemes, and more. It's why you can scratch a CD and still hear music, or get a QR code dirty and still scan it. It's why space missions like the Voyager program could send clear digital images back to Earth. It's why a cluster of computers can perform a complex computation even if one of those computers malfunctions.

These applications all rely on a strikingly beautiful and conceptually straightforward use of interpolation: so-called Reed-Solomon codes, and the codes that build on them.

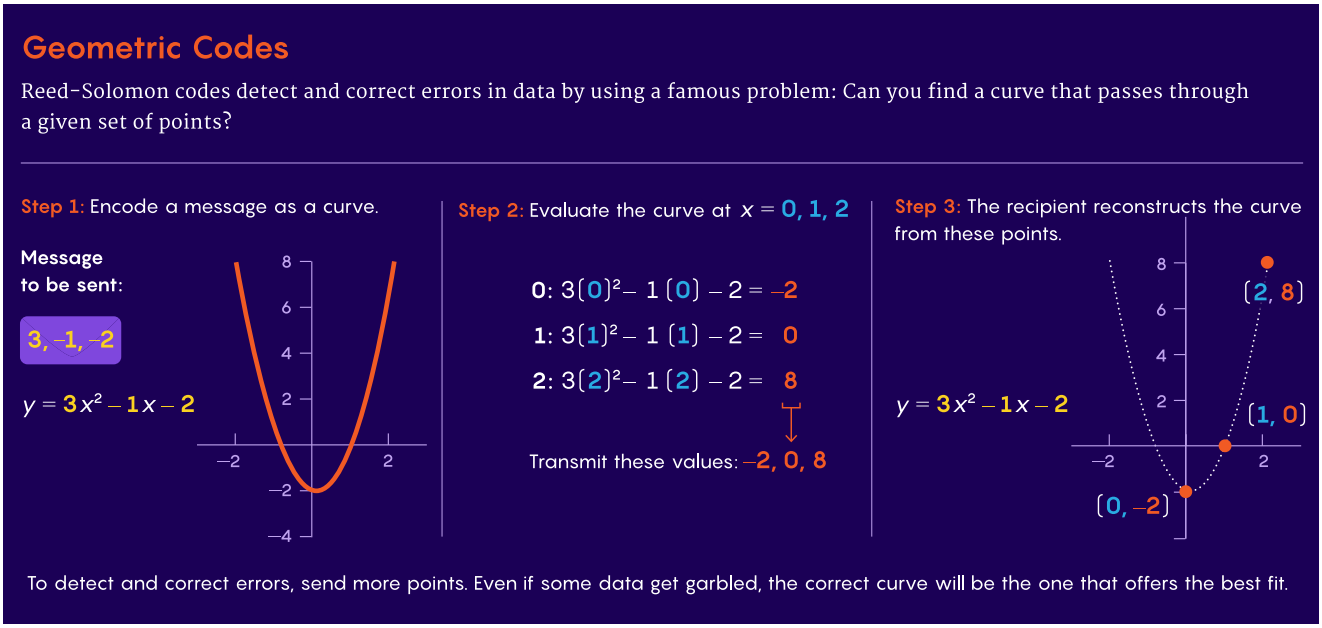
Point by Point

Say you want to send a message consisting of two numbers: 2 and 7. It's possible that some of the data you're transmitting will get lost or corrupted — the 2 might flip to a −2, for instance. So instead of simply sending the data, you can add extra information to help the recipient identify and fix errors that might arise. This is what's called an error-correcting code.

The simplest example of such a code involves transmitting the same message multiple times. To allow the recipient to identify whether an error occurred, send the same message twice: 2, 7, 2, 7. If the numbers in corresponding positions don't match (say, if the transmission instead reads 2, 7, −2, 7), the recipient will know one of them is wrong — but not which one. To let them figure that out and correct the error, send the same message three times: 2, 7, 2, 7, 2, 7. The recipient simply needs to take the majority vote to figure out your intended message.

But this means of correcting errors is wildly inefficient. Here's a smarter approach: Encode the message as a curve, and send just enough information to allow the recipient to reconstruct that curve.

In our simple case of transmitting 2 and 7, the curve would be the line $y = 2x + 7$. Evaluate this curve at two predetermined values of x, and transmit the resulting y-values. The recipient now has two points, and because the interpolation problem tells us that two points determine a unique line, the recipient simply has to find the line that passes through the points they received. The coefficients of the line reveal the intended message.

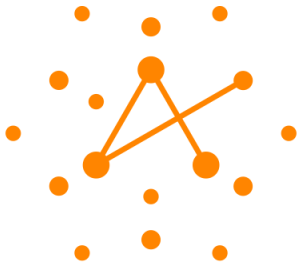


Merrill Sherman/Quanta Magazine

Introduction

To avoid errors, you once again add extra information. Here, you send the *y*-value that corresponds to another predetermined *x*-coordinate. If the three points do not fall on the same line, there's an error. And to figure out where the error is, you just send one more value — meaning you've sent four numbers total, rather than the six required by the previous method.

The advantage grows with the size of the message. Let's say you want to send a longer message — 1,000 numbers. The less efficient code would require sending 2,000 numbers to identify an error, and 3,000 to correct it. But if you use the code that involves interpolating a polynomial through given points, you only need 1,001 numbers to find the error, and 1,002 to correct it. (You can add more points to identify and correct more potential errors.) As the length of your message increases, the difference in efficiency between the two codes grows starker.



Abstractions

Exploring the world of scientific ideas

Abstractions navigates promising ideas in science and mathematics. Journey with us and join the conversation.

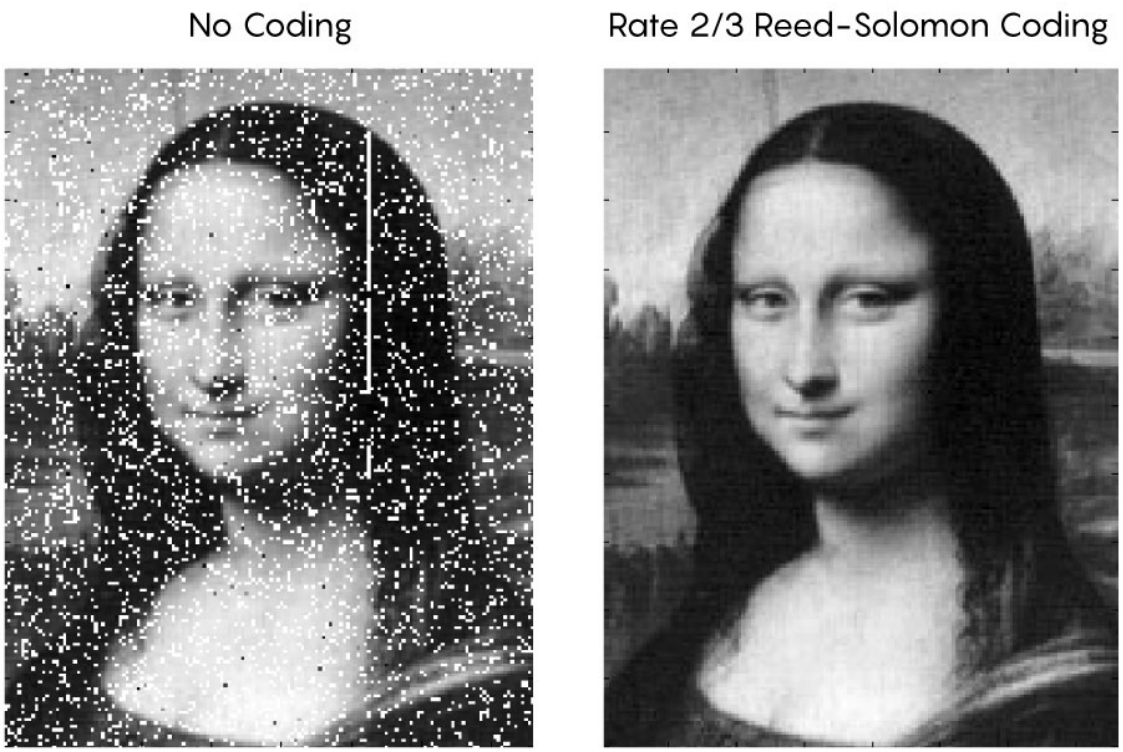
See all Abstractions blog

The more efficient code is called a Reed-Solomon code. Since its introduction in 1960, mathematicians have made further breakthroughs, developing algorithms that can correct more errors with greater efficiency. "It's very elegant, clean, concrete," said Swastik Kopparty, a mathematician and computer scientist at the University of Toronto. "It can be taught to a second-year undergraduate in half an hour."

Reed-Solomon codes have been particularly useful for storing and transmitting information electronically. But the same concept has also been essential in cryptography and distributed computing.

Take secret sharing: Let's say you want to distribute a secret among several parties such that no one person can access the entire secret, but together they can. (Imagine an encryption key, for instance, or a missile launch code.) You encode the numbers in a polynomial, evaluate that polynomial at a predetermined set of points, and distribute each of the results to a different person.

Most recently, Reed-Solomon codes have been employed in areas like cloud computing and blockchain technology. Say you need to run a computation that's too complicated for your laptop, so you have a large computational cluster run it — but now you need to verify that the computation you get back is correct. Reed-Solomon codes let you ask for additional information that the cluster likely won't be able to produce if it hasn't done the computation correctly. "This works magically," said Jade Nardi, a research fellow at the Mathematics Institute of Rennes in France. "This process is really wonderful, and the way it relies on [these codes] blows my mind."



In a proof-of-concept test, NASA scientists encoded the Mona Lisa onto a laser beam and sent it from Earth's surface to a lunar spacecraft. Reed-Solomon codes were used to correct transmission errors introduced by Earth's atmosphere.

Courtesy of Xiaoli Sun, NASA Goddard

But Reed-Solomon codes also have an important constraint. They're constructed in such a way that you can only evaluate your polynomial at a fixed (and usually relatively small) set of values. That is, you're limited to using a certain set of numbers to encode your message.

The size of that set, or alphabet, in turn restricts the length of the messages you can send — and the bigger you try to make your alphabet, the more computational power you'll need to decode those messages.

And so mathematicians sought an even more optimal code.

Future Codes

A more general, more powerful code would allow you to store or send longer messages without needing to increase the size of your alphabet. To do this, mathematicians devised codes that involve interpolating a function — which lives in a special space associated to a more complicated curve — through given points on that curve. These so-called algebraic geometry codes "came out of nowhere, and they're better than any other code we know how to make [with a smaller alphabet]," Kopparty said. "This beats everything. It was a real shock."

Related:

1. **Old Problem About Mathematical Curves Falls to Young Couple**
2. **How Shannon Entropy Imposes Fundamental Limits on Communication**
3. **'Post-Quantum' Cryptography Scheme Is Cracked on a Laptop**

There's just one problem. In practice, implementing a Reed-Solomon code is much, much easier than implementing an algebraic geometry code. "This is state-of-the-art, but it's still under investigation to really turn into something practical," said the cryptologist Simon Abelard. "It involves quite abstract mathematics, and it's hard to handle these codes on a computer."

For now, that's not worrisome: In real-world applications, Reed-Solomon codes and related forms of error correction are sufficient. But that might not always be the case. For instance, if powerful quantum computers become available in the future, they'll be able to break today's cryptography protocols. As a result, researchers have been searching for schemes that can resist quantum attacks. One top contender for such schemes would require something stronger than Reed-Solomon codes. Certain versions of algebraic geometry codes might just work. Other researchers are hopeful about the role algebraic geometry codes might play in cloud computing.

But even in the absence of such potential uses, “in the history of mathematics, sometimes you discover new things that really don’t have applications nowadays,” said Elena Berardini, a researcher at Eindhoven University of Technology in the Netherlands who works on algebraic geometry codes. “But then after 50 years, you find that it might be useful for something completely unexpected” — just like the ancient problem of interpolation itself.

Next article

Chaos Researchers Can Now Predict Perilous Points of No Return