inside NOVA

Cryptography: Encryptions Past

By Sarah Lewin on October 18, 2012 9:07 AM |

NOVA scienceNOW's "Can Science Stop Crime" introduced you to Tadayoshi Kohno, a University of Washington computer scientist who's made it his mission to take over networked machines, identifying surprising security holes that the bad guys can make use of. But hacking isn't just a question of guessing a password--systems use a process of encryption to mask information, making it a meaningless jumble of code unless you possess a secret key.

Ever since writing has existed, people have wanted to send secret messages to one another--and others have wanted to intercept and read them. These days, the "messages" are bank information, government files, or control over surveillance systems. The eavesdroppers have evolved into hackers and government agencies like the NSA. But the constant power struggle between secret keepers and secret stealers continues apace, spurring new invention and broadening the horizons of science along the way. Over the next week, this four-part blog series will take you through the history of cryptography, its present, and future possibilities of unbreakable codes.

Encryptions Past

LHIEL XEU SRWP R SLNVAB VX TNO VQ XDMZ SRV DLLH XMABADIU QVWVXDIE

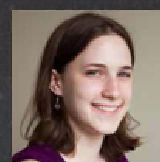
Ever send a secret message to a friend to avoid the snooping eyes of a younger sibling? If so, you're probably already familiar with the sort of encoding techniques I'll describe here, the earliest examples of substitution ciphers: schemes for replacing letters with others to render a message unreadable.

The first recorded examples of ciphers--specific procedures for encrypting information--replaced each letter in a message with a different letter. ("A" would always be replaced by "S," for example.) Simple versions of these ciphers were supposedly first used by Julius Caesar to transmit battle commands, and they're even sometimes called Caesar ciphers.

This sort of cipher is *monoalphabetic* because it assigns just one specific code letter to each letter of the alphabet in the message. It comes with a key that specifies how to change each letter to disguise the message and how to reverse the change to read it. The key could look something like:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
L A Q E F J S W B H I M P G U T X R Z O K N C Y V D

Using this key, the code LOOK BEHIND YOU would translate to MUUI



Sarah Lewin

Sarah Lewin (NOVA Research Intern, Summer 2012) enjoys explaining things. She will graduate from Brown University in the Spring of 2013 with a bachelor's degree in mathematics and as much classwork in science writing as she can get. After this, she will most likely either disappear in a poof of smoke or continue her habit of writing about the awesome bits of math and science. She newly explains things at sarahexplains.wordpress.com.

Other posts by this Contributor

AFWBGE VUK.

Of course, as these ciphers entered wide use, people quickly began to discover ways to read the message without being given the key. Look at our example--the repeated UU in the first word could only be something like EE or OO, and trying different substitutions lets you work out the original message. The longer the message, the better letter frequency analysis will work.

Secret keepers were forced to up the complexity of their codes. The result was the *polyalphabetic* ciphers--a method which switched the encoding process of each letter throughout the message, so "A" might be replaced with "S" at one time and then with "R" later on in the message, all according to a set pattern. The more complicated the key, the harder the message would be to figure out. However, tools like frequency analysis could still crack secret messages when the text was long enough, because the key would have to repeat--meaning some parts of the message would be encoded with the same substitutions as others.

Armies and civilian secret-keepers alike quickly took up polyalphabetic ciphers as a much more secure way to communicate. To code them quickly, senders used tools like the cipher disk in the image below. This disk was standard issue for Confederate officers during the Civil War.

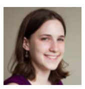
A reproduction of a Confederate cipher disk. Via Wikimedia.

This Confederate cipher wheel can implement a Vigenère Cipher, which uses a key phrase or sentence to encode a message. For each letter of the raw message, you'd turn the dial so "A" on the outer circle lined up with the letter in the code phrase on the inner circle. You'd then find the letter from your message on the inner circle and substitute the matching letter on the outer circle! Rinse and repeat for each letter of the code phrase and message.

This was the state of substitution cryptography before the advent of complex calculating machines. Soon, mechanical developments would make this sort of code look like child's play.

Our blog series on cryptography will continue later this week.

inside NOVA



Cryptography: The First Unbreakable Code

By Sarah Lewin on October 23, 2012 8:22 AM |

Ever since writing has existed, people have wanted to send secret messages to one another--and others have wanted to intercept and read them. This is the second installment of a blog series taking you through the history of cryptography, its present, and future possibilities of unbreakable codes. Click here to read Part 1: Encryptions Past.

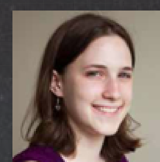
Some of the very first secret codes were substitution ciphers--schemes for transforming the letters in a message to render them unreadable to anybody who didn't know the secret to decoding them. The reader of the message would use a "key," information that revealed how to translate the message back into normal text, that could come in the form of an exact list of letters or numbers, a code word, or another variable. In theory, only people with knowledge of the key could read the encoded messages. In practice, though, the earliest ciphers were simple enough to break by analyzing the frequency of letters or simple trial and error.

The good news for secret-keepers: In World War II, an unbreakable encryption scheme was invented. The bad news: The One-Time Pad, as it was called, never really caught on. And for good reason.

Here's how it worked. Users would need two identical copies of a long book of random numbers--the "One-Time Pad" itself. The first message sent would use the first page of the One-Time Pad, and each subsequent message would use a new page, so that by the 999th message both communicators would have gone through 999 matching pages of random numbers.

Because the numbers in the key didn't repeat, there were no patterns to analyze: hence the unbreakableness. Sure, a spy could guess the exact string of random numbers used and decode the message. But how would he know it was the right message? With a slightly different string of random numbers, the message could decode to say something completely different, and there was no way to verify a correct decoding.

Unbreakable it may have been, but the One-Time Pad was also rather inconvenient. Bulky books of random numbers were impractical to carry and use on the battlefield. There was always the risk that they would be stolen by the enemy. Even the process of generating truly random numbers was much more difficult than you'd expect, and any patterns in the numbers could be exploited to crack the code. The unbreakable One-Time Pad points to the problem with all ciphers invented up to this point--in order to send a secret message, the sender and recipient had to already share a secret: the key, whether it was a list of letters and their counterparts or a book of random numbers. And the system would only be as secure as the



Sarah Lewin

Sarah Lewin (NOVA Research Intern, Summer 2012) enjoys explaining things. She will graduate from Brown University in the Spring of 2013 with a bachelor's degree in mathematics and as much classwork in science writing as she can get. After this, she will most likely either disappear in a poof of smoke or continue her habit of writing about the awesome bits of math and science. She newly explains things at sarahexplains.wordpress.com.

Follow NOVA's Twitter Feed

Other posts by this Contributor

Cryptography: Encryptions Present
Cryptography: The First Unbreakable Code
Cryptography: Encryptions Past
Adventures in Swarm Robotics

method used to share the key. It would be quite some time before anybody overcame that particular hurdle.

German Enigma machine. Via Wikimedia.

So far I've only talked about pen-and-ink ciphers: the kind that are easy to encrypt and transmit by hand. But the push for more complexity meant, that during World War II, armies were always on the lookout for a fast, convenient way to send out orders and information. Pen and paper gave way to simple mechanical devices, which soon blossomed into complex machines whose codes required even more complicated machines to crack. A famous example: Germany's use of the mechanical encryption system called ENIGMA to conceal its plans. ENIGMA was powerful because it was flexible: the machine's settings allowed users to access a huge number of encryption schemes based on keys that were shared among all the operators and changed by the day.

Here is how the process worked: ENIGMA sent each typed letter through three of its many scrambling rotors. At each rotor, ENIGMA switched the letter with another letter. Then, the letter went through a plugboard that could swap several letters with each other--the swap list was changed every day. After every key type, the rotors would increment forward, ensuring that the encryption of the next letter would be different. The message could only be decoded if the machine was set up with the identical rotors in the same position and the same plugboard settings--leaving 159 million million million possible "keys" or settings for a given message's beginning. (See How the Enigma Works for more about ENIGMA's inner workings.)

Because the plugboard and rotor settings were changed by all ENIGMA users on a daily basis (each ENIGMA operator had a thick book of settings to use each month), British scientists were forced to rush and break the code each day to read transmissions before the information became obsolete.

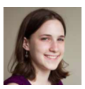
The race to break ENIGMA is a famously dramatic story. Ultimately, the scientists at Britain's Bletchley Park invented a mechanical device the size of several rooms to crack the code. Their machine was built of several pieces called bombes that recreated ENIGMA's internal machinery. These bombes automatically cycled through trying all the possible rotor combinations to break the day's transmissions. The bombes were precursors to the computers we know today; ENIGMA motivated scientific development and showed the world the possibilities of using machines to encode and transmit information.

Today, you can use a computer to create a polyalphabetic substitution code complicated enough that it would take impossibly long for someone to decode without the key. And indeed, many encryption systems available commercially rely on that basic format.

But isn't there a way to get rid of this reliance on secret keys?

Well, yes--as we'll explore next.

previous post next post



Cryptography: Encryptions Present

By Sarah Lewin on October 24, 2012 10:23 AM |

Ever since writing has existed, people have wanted to send secret messages to one another--and others have wanted to intercept and read them. This is the third installment of the blog series taking you through the history of cryptography, its present, and future possibilities of unbreakable codes. Follow the links to read the first and second parts of the series.

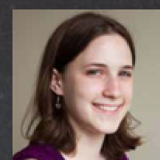
Last time I talked about complex polyalphabetic ciphers--techniques of encoding information that make it nearly impossible to reveal the message without access to a secret key. Because the encryption processes themselves are so intricate, the main security challenge becomes keeping that key private. In order to tell a secret, you already have to share a secret key, and your encoded message will be useless if an eavesdropper finds a way to get ahold of that key.

But in 1976, Ron Rivest, Adi Shamir, and Leonard Adleman invented a method that eliminated the need to give away the key at all. Their method, public key cryptography, is still used today to make secure transmissions of sensitive data and to prove the identities of people online. Public key cryptography turns traditional cryptography on its head: instead of keeping the key a secret, every receiver creates and broadcasts his own individualized key for everyone to see, and anybody who wants to send him a message will use that key to encode it. Because of the way the key was created, he will be the only one who can decode it.

The trick to creating this kind of public key is to use what mathematicians call a one-way function: a type of math operation that's easy to do in one direction but nearly impossible to undo without additional information. One example of a one-way function is multiplying two super-big prime numbers. The multiplication part is easy, but to *undo* that multiplication, you need to know what the original prime numbers were. And as mathematicians can attest, factoring a number into primes is a ton of work--pick a number big enough, and it'll take all the computers in the world longer than the age of the universe to find the factors.

The RSA public-key cryptosystem that they invented (named after Rivest, Shamir, and Adleman themselves, of course) is still in use today, and it works along exactly these lines. Each person's public key is a version of a large number built from two primes, and only someone with the knowledge of the number's factors--the private key--can decode something encoded using their public key.

The other popular public key cryptosystems today work similarly. They each use a mathematically "hard problem" to create keys so that anyone can encode messages to specific people but only the intended recipients will



Sarah Lewin

Sarah Lewin (NOVA Research Intern, Summer 2012) enjoys explaining things. She will graduate from Brown University in the Spring of 2013 with a bachelor's degree in mathematics and as much classwork in science writing as she can get. After this, she will most likely either disappear in a poof of smoke or continue her habit of writing about the awesome bits of math and science. She newly explains things at sarahexplains.wordpress.com.

Follow NOVA's Twitter Feed

Other posts by this Contributor

Encryptions Future: Quantum Cryptography
Cryptography: Encryptions Present
Cryptography: The First Unbreakable Code
Cryptography: Encryptions Past
Adventures in Swarm Robotics

have the extra information needed to reverse it.

Now, if people wanted to stop at this level of security it would be perfectly understandable. With the computers we have now, public key cryptography is certainly secure enough--so secure, in fact, that it's prompted governments of several countries to put limits on key size, and even to try and ban the exportation of big prime numbers. After all, governments want to be able to read everybody's mail--it wouldn't do for foreign states to have better encryption systems. Public key cryptography is the system that makes e-commerce possible, and it is a standard for high-importance confidential messages. But there is always a chance that someone will find a way to beat the system and find the extra information from the public key.

Enter the next big step, at least in theory--the quantum computer.

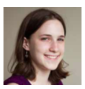
More on that next time.

PBS HOME | PROGRAMS A-Z | TV SCHEDULES | WATCH VIDEO | DONATE | SHOP PBS | SEARCH PBS

inside NOVA

Archive Most Commented NOVA Home

Search this blog



Encryptions Future: Quantum Cryptography

By Sarah Lewin on November 20, 2012 11:40 AM |

Ever since writing has existed, people have wanted to send secret messages to one another--and others have wanted to intercept and read them. This is the fourth installment of a blog series taking you through the history of cryptography, its present, and future possibilities of unbreakable codes. Follow the links to read the first, second, and third parts of the series.

In the last post I talked about Public Key Cryptography, a system that derives its security from "hard" math problems like finding the factors of large numbers. But increases in computing power have made once-impenetrable codes solvable in just a few months. Keeping ahead of technological development by increasing the length of the secret number used to encode information--the key--is always an option, but there is still a chance that someone will find a tricky way to calculate your private key, letting them read all your messages.

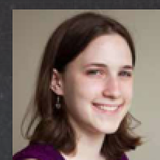
One of those tricks could be quantum computing. Scientists have shown that a quantum computer could use some clever properties of quantum mechanics to help codebreakers solve many of the math problems at the heart of public key cryptography.

How? A quantum computer encodes "bits" of information in the properties of subatomic particles. And here is where things get strange: Because unobserved particles--according to quantum mechanics--exist in a combination of all possible states, a quantum computer is able to store and operate on multiple numbers at once. Suddenly, "impossible" tasks like factoring large numbers can be done in the same time it would take to multiply them.

Although quantum computers are still in their infancy--nothing with enough complexity to run that sort of program can be built yet--codemakers can see the demise of public key cryptography on the horizon. It's not like everyone would have a quantum computer, but chances are that someday in the future, if the government or a big business wants to read your encrypted mail, they'll have access to a computer that can do so.

But in 1984, Charles Bennett and Gilles Brassard found a way to transmit a key that is provably secure, even in the face of quantum computers. Their method, which is still used today, uses properties of quantum mechanics to transmit a random key--and any eavesdropping party would not only get useless information, but would even *alert the sender and receiver to an attack*. Pretty cool stuff.

Here's how it works: The One-Time Pad, which I discussed in a previous



Sarah Lewin

Sarah Lewin (NOVA Research Intern, Summer 2012) enjoys explaining things. She will graduate from Brown University in the Spring of 2013 with a bachelor's degree in mathematics and as much classwork in science writing as she can get. After this, she will most likely either disappear in a poof of smoke or continue her habit of writing about the awesome bits of math and science. She newly explains things at sarahexplains.wordpress.com.

Follow NOVA's Twitter Feed

Other posts by this Contributor

Encryptions Future: Quantum Cryptography
Cryptography: Encryptions Present
Cryptography: The First Unbreakable Code
Cryptography: Encryptions Past
Adventures in Swarm Robotics

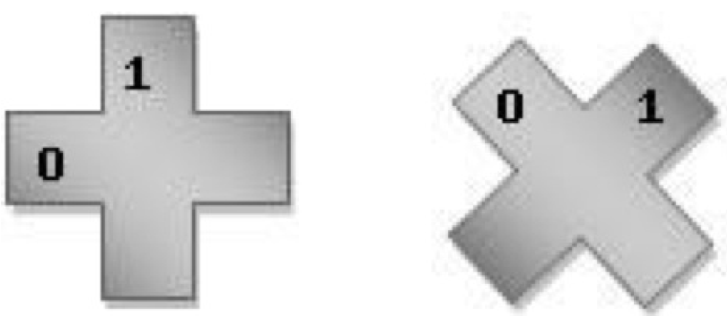
article, basically encodes data by using a randomly-generated key as long as the intended message. It's impossible to break the code and read the message without the key.

Bennett and Brassard's quantum key distribution protocol, called BB84, acts as a sort of high-tech One-Time Pad: A random key is generated using quantum mechanics and shared securely between two people, who can then use it to encode and send unbreakable messages however they want. It eliminates the problem of transferring a key securely. The process of generating the key takes advantage of the quantum mechanical property that measuring something can change it.

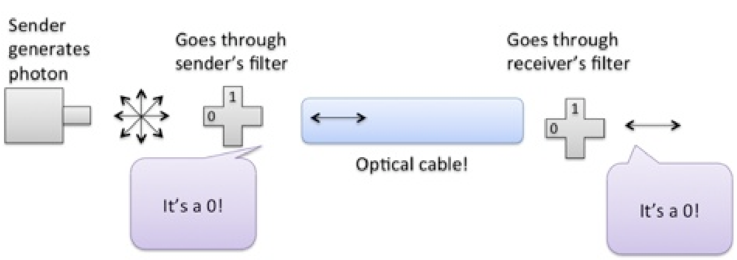
Here are the basics: photons are sent from one person to another, measured at both ends. The measurements that match up will be used as the basis for a randomized key.

The sender and receiver each have two kinds of polarized filters: one that only lets in horizontal or vertically oriented photons, and one that only lets in the diagonals. They agree that photons that are vertically or diagonal-forward polarized will represent binary 1, and photons that are horizontal or diagonal-backward will be 0.

The two possible filters



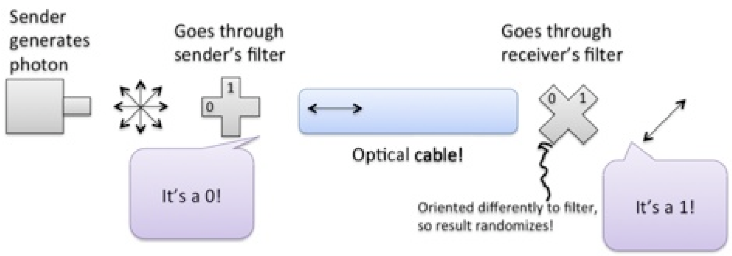
The sender generates a photon and prepares it with one of the two random filters before sending it along an optical cable to the receiver. Once it's there, the receiver measures the photon with his own randomly chosen filter.



Sender and receiver both choose a horizontal/vertical filter, both measure 0.

Here's the tricky part. If the receiver measures the photon with the same filter as the sender, he'll get the same result. But if he uses the wrong filter, there's no such guarantee: If he's been sent a vertically polarized photon and he measures it with a diagonal filter, he has a 50% chance of getting each of the diagonals as his result.

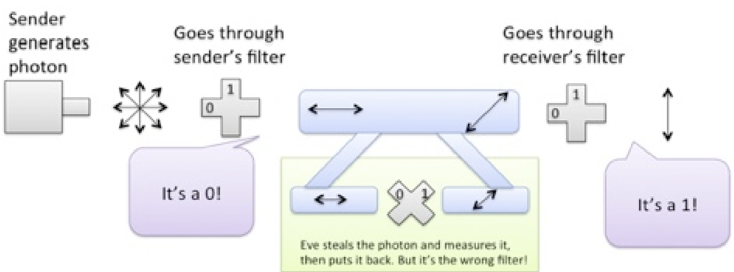
Sender randomly chooses horizontal/vertical, but receiver randomly chooses diagonal. Chaos ensues!



The sender and receiver go through a long string of photons in this way, recording the bit values and which filter they used for each. Afterwards the receiver uses another line of communication--it doesn't have to be secret--to tell the sender which filters he used for each (without giving away the results of each measurement). The sender reveals which filters she used, and they agree to only count the photons where they used the same filters. That way, they know that they've measured each photon the same way, so they'll have the same values.

In this way they build up a secret, random string of numbers using the photons that they both measured in the same way. This string of numbers will be their key.

Now imagine that you are an eavesdropper. If you are able to intercept a photon between the users, you won't know how the sender prepared it, so you have a 50% chance of using the wrong filter. That means that, not only might you get the wrong answer, you might also mess up the value for the receiver because of that quantum mechanical property I mentioned where measuring the photon can change its value. In fact, if the sender and receiver compare the values of a few of their photons and find any disagreements, they can tell that somebody's been trying to read their photons and discard the suspect values from the key.



Eavesdropper tries to measure photon but uses different filter than the sender and receiver, messing up measurements.

Oh, and by the way: Once the sender and receiver start revealing the filters they used, it's far too late for you to use those filters yourself. The photons are long gone.

There are a few commercial systems that implement quantum key distribution today, including ID Quantique, a spinoff from a University of Geneva experimental physics group. The technology has already been used in highly secure transmissions, from Swiss ballots to World Bank transactions.

But quantum key distribution hasn't become mainstream quite yet, mostly due to a few basic issues. For one, the machines are all handmade by physicists, so they are expensive and inconvenient to commission. Another issue is that the system requires dedicated optical cables to send the photons, whereas almost all currently existing fiber-optic infrastructure, although fairly widespread, relies on sending multiple signals on the same cables. And finally, there's the issue of scalability. Right now quantum key communications must be cabled directly from one user to another--like from a bank to a single high-powered client--but a vast new infrastructure would be needed to connect a large network of users.

Richard Hughes, a quantum researcher at Los Alamos National Laboratory, is working on answers to these design problems. In the future, he expects quantum cryptography to be used in smart grid applications and to eventually extend to everything from smartphone and tablet security to securing data in the virtual cloud. He says that quantum cryptography is much further along than we realize: The technology that exists today can already be used reliably in optical fiber networks for systems of medium scale, and experiments suggest that through line-of-sight delivery of secure keys--that is, sending the photons through open air--it could be possible to generate keys with satellites in orbit.

Although the key creation is perfectly secure, there still may be ways to outwit the system. For instance, in April 2010 researchers at the Norwegian University of Science and Technology found a way to trick a commercial system into revealing its secrets by shining a laser into the receiver's filter, blinding it while they read the photons themselves. The team was kind enough to warn companies using quantum technology before publishing their results so the security hole could be fixed.

Certainly this won't be the only flaw that researchers--and hackers--discover. After all, you can have the strongest, most well-secured door in the world, but the room's only safe for the time it takes to blast through the wall. As time goes on, security flaws will be found and repaired, approaching the perfectly secure system promised by quantum physics, and maybe even revealing more about how our universe works.

The fight between codemakers and codebreakers has driven technological and mathematical advances through history--from frequency analysis to mechanical bombes during World War II to computers to quantum programs. Quantum key distribution promises an unbreakable one-time cipher that companies, governments, and even individuals will be able to use to send information with perfect security and store private data with an unbeatable cipher. So--at least for now--the secret keepers have won. What will we do with that power?

previous post **next post**