

Is One-time Pad History?

DIRK RIJMENANTS

Abstract Are one-time pads a thing of the past? There has been quite a bit discussion and some security and cryptography experts argue that one-time pad is no longer a system for today's needs, that it is impractical and creates enormous key distribution problems. They say that current computer algorithms provide enough security and public key schemes solve the problem of key distribution. That's all true. But there are some things they don't tell you.

Keywords Cryptography, One-time Pad, Manual Encryption, Cryptographic Algorithm, Public Key Encryption, Mathematical Security, NSA, GCHQ, FAPSI,

One-time Pads and One-time Encryption

Let us first explain one-time encryption, and what this paper is about. One-time pad encryption is a most basic encryption algorithm where the readable data is combined with a truly random key of the same length as the data. The key should never be reused and always destroyed after use. The system was invented in 1917 and it is mathematically unbreakable. There is no way to crack it with current or future computer power, simply because it is mathematically impossible. The downside is that the rules of one-time use create a cumbersome key distribution with associated problems.

I must point out here that this paper is about modern one-time encryption applications, not the pencil-and-paper spy trade craft (although it is just as secure). Neither is this paper about small one-time passwords or one-time keys, which are only valid for a single encryption session by some crypto-algorithm under control of that key, and the paper is certainly not about the many snake-oil applications that pretend to be unbreakable because they claim to be using one-time encryption, while they actually are not. Remember: key as long as the data, truly random and used only once. There is no way around these three conditions without messing up the unbreakable part!

Many cryptologists believe that one-time encryption is something from the past. They claim that modern encryption algorithms offer secure communications and privacy, that the current key exchange schemes solve the complex key distribution and that there is no longer a need for one-time encryption. This paper explains why they are wrong (and why they don't admit that).

Insecure Systems

For a start, there is the problem of implementing secure systems. A strong encryption algorithm is useless on a computer that contains viruses or spy ware that captures your keystrokes or retrieves your data before it is encrypted. Today, virtually all computers are vulnerable to attacks, and most computers are actually infected, especially those connected to an external network like the security nightmare called 'the Internet'.

The modern Personal Computer is a true security disaster, everything leaks out, and anyone can get in. In fact, today, all our means to communicate are completely digitalized and automated, but at the same time, we no longer have any control over these systems. We have no idea of what our own computer is doing, which processes are running in the background, or what plug-ins, add-ons and other unidentified software is downloaded automatically to "stay compatible". A most dangerous evolution which has gone way too far already.

There are very strong algorithms available, but we use completely insecure computers. Even firewalls of government agencies have proven to be vulnerable to attacks. In 99 percent of cases, Intelligence agencies don't have to break any encryption, they simply retrieve the information before it is encrypted. That is why the only truly secure encryption is performed by dedicated crypto devices or computers, well separated from the outside world. All OSI model based network connections are to be considered insecure. Cryptologists or software designers who claim that their software provides security and privacy on your personal computer really do not know what they are talking about, not because of incompetence, but simply because they have no idea of all the processes that are running on your computer. Actually, we all don't have any idea.

Mathematical Security

What about mathematical security? There are two main types of encryption: symmetric and asymmetric. The traditional symmetric encryption uses the same key for both encrypting and decrypting (one-time pads are a type of symmetric encryption). This creates the problem of secure key exchange. Asymmetric public key encryption uses key pairs. Each pair consists of a public key to encrypt and a private key to decrypt. You can distribute your public key openly, and everyone who wants to send you something can encrypt data with your public key, but only you can decrypt it with your private key. This is great. We no longer have to securely exchange secret keys.

Unfortunately, a public key algorithm can't encrypt your data. Although quite simple and straightforward, its process is far too slow and computationally too heavy to do that. Instead, we use a random key to encrypt the actual data with a traditional symmetric algorithm. Next, that random key is encrypted with the public key of an asymmetric algorithm, provided by the receiver. Finally, the data and encrypted key are sent, nicely packed together. The receiver takes his private key to decrypt your random key and uses the retrieved random key to decrypt the actual data. Great! This solved the key distribution problem, a major disadvantage of one-time pads. Today, asymmetric public key schemes are used worldwide to protect communications, e-mail, websites and e-commerce. There would simply be no public cryptography without public key cryptography, as this would leave us with enormous and expensive key distribution problems.

Now, it's pretty impressive to say that factoring the product of two large primes, used in public key cryptography, is mathematically almost impossible. It sounds great to say that current technology will not break it in a trillion years. But we don't need to break that fantastic asymmetric algorithm! Instead, breaking the traditional symmetric algorithm that is used to encrypt the actual data will do just fine. So, who cares your key exchange is perfectly secure? It is the traditional symmetric algorithm that must be strong, and that is not a question of so-called insurmountable mathematical problems to crack asymmetric encryption, but a case of cryptanalysis of man-made algorithms. Design problems, weak key schedules or bad implementation can be exploited, not to

mention backdoors or other mathematical shortcuts to crack them. Today's public key schemes only provide – unproven - key exchange security. They do not add to the message security.

Trusting Algorithms

So, can we trust those symmetric algorithms? All currently used algorithms are openly examined and tested in the public. But there are other players than the public. Government agencies have far more resources, technology and knowledge than the public. It is known that agencies such as the NSA, GCHQ, FAPSI or the FSB Academy have an unmatched collection of brain power, money and computational power.

Are the current algorithms really unbreakable, or at least rather strong? In both the US and Russia, two key players in the world of cryptography, the use and export of cryptography is approved and controlled by the government. In the US, cryptography export is legally regarded as weapons export, and Russia forbids using non-approved cryptography.

On what basis do they approve or forbid the use or export of a new algorithm? Do they only approve weakened versions of an algorithm? Why do they lower key sizes of algorithms? A simple question gives you the answer: would they cut in their own throats, and approve and spread unbreakable encryption and deprive themselves of their primary job, collecting information?

Now you will ask, why don't we prove that those algorithms are breakable? Well, we all know what we need for that: lots of money, and the government has it. You also need very capable people, and guess who has most of these capable guys, and the money to recruit them? Indeed! That's not paranoia from our side, that's efficient pro-active management and safeguarding sources from their side. Be honest, you and I would do just the same.

Modern crypto algorithms are doing a great job of protecting the little man against their snooping neighbours or low tech criminals, but it doesn't provide absolute security or privacy. Some experts argue that modern algorithms are practically unbreakable. Are they? Where is the mathematical proof? Ask any cryptologist to provide the mathematical proof his algorithm is unbreakable. He simply cannot (one-time pad is the only one). If they do not find a way to break something, it's not unbreakable. They have only proved they cannot break it themselves (or claim they cannot break it). They could claim their encryption is strong, at best. Nothing more or less.

A good old quote of NSA's David Boak says it all: *"the 'approved' systems have simply been shown to adequately resist whatever kinds of crypto-mathematical attacks we, with our finite resources and brains, have been able to think up. We are by no means certain that the [opponent] equivalent can do no better"*.

So, we'll just have to trust them and hope no one is smarter or has better computers. Can we trust them? Do you prefer trust and hope, or do you prefer proof? Use your common sense!

Long Term Security

If an algorithm is safe today, will it be safe tomorrow? No single computer developer could have imagined 40 years ago that we would have computers with a speed of 1.105 Peta Flops, that's $1.105 \cdot 10^{15}$ or 1.105 quadrillion floating point operations per second (this record is probably already broken when you read this). Nothing is unbreakable, except for one-time pad. Therefore, we make algorithms that take so much time to break, that the information will be useless by the time we decrypted it. Actually, we are talking here about the time, required for a brute-force attack on the key, because we can't predict if or when some smart guy finds an efficient cryptanalytic attack to shortcut the job. That is how all current cryptography works.

However, Intelligence agencies collect and store huge amounts of possibly interesting data traffic for the future. If new technology or cryptanalytic techniques enable the decryption of these archives, this can have devastating consequences. Imagine critical information about important people, operations or political decisions, decrypted after 20 or 30 years. Many of the involved people would still be alive or even in office.

On the other hand, a one-time pad encrypted message will never be broken if the keys have been destroyed. Just take a look at the past. Messages that were encrypted in the 1950's with 'state of the art' cipher machines and were kept archived by the adversary (which actually happened) are now generally broken within a few seconds, minutes or some hours at the most. Let's hope for them there wasn't anything crucial to hide. The messages that were sent 50 years ago with a one-time tape device such as the ETCRRM or ROCKEX will stay unbreakable for ever.

Home Banking

Now let's have a quick off-topic detour. If my computer isn't safe and I can't trust current cryptography, does this mean I have to stop home banking? Absolutely not! Hacking someone's bank account or home banking involves more than a smart hacker. Technically, it is rather easy to hack into someone's home banking (although some one-time key or identification systems are more secure than others). But you also need transfer accounts, false identities and credit cards to redraw money. All these things leave traces and to earn a living by emptying bank accounts will get you the law on your tail pretty fast. The same goes for other on-line fraud. It's just not an efficient way to get money (although it apparently keeps increasing because of unawareness). But again, that's another ball game. The goal of online fraud is to leave a trace, namely the hole in your wallet. The interception and unauthorized use of poorly protected or encrypted information however often doesn't leave traces, can be automated efficiently and can be very damaging.

One-time Pads, Obsolete or Not

Modern crypto algorithms provide practical reasonable security and privacy, essential to our economy and everyday life. However, sometimes you need ever lasting absolute security and privacy, and that's only possible with one-time encryption. Some experts argue that the distribution of large quantities of one-time pads or keys is impractical. This is nonsense. Today's electronics are capable of generating huge numbers of truly random keys, and current one-time encryption software can process large quantities of data at higher speeds than any other complex algorithm. Current data storage technology such as USB sticks, memory cards, DVD's, external HDD or SSD drives enable the physical transport of enormous quantities of truly random keys.

Actual sensitive communications are often limited to a small number of users. In such cases, one-on-one communications with the associated key distribution, possibly in configuration with a star topology, is no longer a practical problem, especially considering the security benefits (this quote will not be popular with cryptologists, but it is true). By using a co-called sneakernet (transferring data on removable media by physically couriering), you can reach a throughput of one-time keys that is greater than what a network can process on data. In other words, it could take a few hours to drive a terabyte of key material, stored on an external drive, by car to someone, but it will take days or even weeks to consume that amount of keys on a broadband network. A terabyte sized key can easily encrypt you e-mail traffic for a year, including attachments (many internet providers won't even allow this amount of traffic). If security is preferred above practical key distribution, and exchange of keys is possible on beforehand, then one-time pad is the right choice. Some commercial firms offer such one-time encryption solutions, mostly to government and defence agencies, and for good reasons. Recent revelations prove what any serious security expert already knows for decades: commercial encryption software and public computer algorithms on personal computers are not suitable and not capable of protecting private communications.

One-time Pad Encryption in Practice

But how could we possibly send, receive and process unbreakable one-time pad encryption on a personal computer? We all know that such computers are absolutely not suitable for cryptographic tools. There are a few practical yet absolutely secure solutions.

You can use a dedicated notebook or laptop computer exclusively for one-time pad encryption. This laptop is stripped from the usual insecure network connections (both physical and wireless) that work via the insecure full path of OSI layers. This secure laptop's only connection to the outside world is a basic IO port (optically or electrically) from which the crypto software sends or receives raw message bytes to or from any – insecure – personal computer.

Incoming bytes are processed by the crypto laptop exclusively as message bytes and the laptop will therefore be absolutely resistant against any external attack or intrusion because any non-conform or malicious data would simply be discarded as unusable/unreadable message data.

Header bytes could be agreed upon to facilitate the recognition and processing of the message data. Checksums or hash values can be added, prior to encryption, to assure authenticity and detect corruption, and addresses can be added in plain to tell the connected insecure PC to whom the data should be sent (automatically).

We now have an absolutely secure method to process, encrypt and transmit one-time pad encrypted data from a dedicated secure laptop or notebook through any insecure personal computer. The insecure computer carries some basic software to (automatically) send or receive the encrypted raw message bytes (data, files...) from or to the secure laptop and to relay encrypted data over any insecure network. Any one-time pad encrypted data that is sent, processed or stored (for infinite time) on any insecure computer or server remains unbreakable. Note that the dedicated laptop obviously should be stored on a physically secure location when not in use.

Another very cheap solution for text messages is a small device with LCD display and small keyboard. Its microprocessor is programmed only to encrypt and decrypt messages. The securely encrypted message are sent through a basic IO port or USB port to any insecure personal computer which, given some basic software, relays the secure message through any insecure network.

The required keys can be stored in its memory and automatically erased after one-time use. A generator for truly random keys can be added to generate and store own keys, and a memory slot enables copying generated keys onto a memory card, carried safe-hand to the person you want to communicate with. Using a simple ATtiny or ATmega processor, the complete device easily fits in a cigarette box sized housing.

This small device is an easy and practical way to send absolutely secure and unbreakable text messages through any insecure channel. By programming an audio modem into the device, and adding a small speaker and microphone, you can even send and receive messages by telephone without requiring any physical connection.

However, even pencil-and-paper one-time pads still have a future as a low-tech way to encrypt small text messages. It's easy to learn, anyone can create truly random one-time pads and you don't need any compromising equipment or insecure computers to perform the encryption. Manual one-time pads are therefore ideal for secure (covert) communications, useful in many situations. This 'poor man's encryption' brings absolute security and privacy in everyone's reach, and that's a basic right.

Conclusion

So, is one-time encryption still useful? Yes! It's the only system, proven mathematically unbreakable, regardless any future mathematical or technical developments like quantum computing. It's also the only system that provides real long-term protection. Only one-time encryption is so basic and transparent that anyone can trust and use it. That's why it has been used for so long in the past (although there were other cryptographic solutions) and that it will stay the preferred solution when unconditional security is required.

Public key algorithms and traditional symmetric algorithms are of course useful, and they have earned their place in the commercial market for reasonably secure large-scale communications. However, in some specific circumstances, absolute security is preferred above practical considerations. We need both practical applications and secure applications, and we can favour one of them for a specific situation. We cannot compare them, nor can we pretend that we don't need the one because we have the other.

Unfortunately, many cryptologists do make that mistake, and one can ask why. Is it because they forgot what it is all about in the first place (security, not comfort) or because absolute security is a logistical and technical problem (key distribution, physical security) that doesn't need a cryptologist? Maybe the answer is even simpler: some just seek commercial profit and others just don't want you to communicate securely.

Is one-time pad encryption a thing of the past? Absolutely not!