SmokeNet Whitepaper

Using the One Time Pad for Zero Trust Networking

Authors: Anthony Scott Thompson
Brig Gen Ian R. Dickinson, USAF (Ret.)
Brian Mielke
Steven Paul Cummings

Abstract

Over the last 20 to 30 years, a lot of effort has been made to discredit using the only known uncrackable encryption in a network - the One Time Pad aka Vernam Cipher. There has been a list of reasons that on the surface have seemed rational. The few that have attempted to use the One Time Pad in the network have been discredited.

This paper will explore a new technology that removes objections through a series of technological advancements. The technology is called the Streaming Transmission One Time Pad Protocol (STOP). The techniques have received multiple patents. The solution is being used daily to great effect by multiple customers. STOP has the potential to altogether remove the Quantum Computing threat. It also has the ability to block nation-state actors from accessing our data as it travels across the Internet.

Each objection will be addressed one at a time. In the end, we will see that STOP and it's IP network implementation SmokeNet is an evolutionary leap, not only for encryption but for network cybersecurity as a whole. It provides a critical piece in the Zero Trust Architecture model being proposed by NIST. In short, STOP and its IP network implementation SmokeNet completely removes the trust in the public Internet.

Disclaimer and Intellectual Property Statement

This report has been created by Introspective Power, Inc. dba Introspective Networks for the purpose of furthering the understanding of Introspective Power, Inc. intellectual property. An additional purpose of the paper is to explore potential future business activities. Accepting this report and/or reading it signifies that the recipient/reader agrees that all ideas and concepts that may constitute intellectual property, whether trade secret, trademark, copyright or patent, remain the property of Introspective Power, Inc. This report and its commissioning in no way grant rights to any other party, including the recipient and/or reader including but not limited to license to or transference of any of the aforementioned property.

Both the recipient of this report and Introspective Power, Inc. agree that this report in no way commits either party to future work or contracts and is no guarantee of future business.

By accepting this report and/or reading it, the recipient/reader agrees to keep all information contained in this report confidential and not to release it to any third party without the consent of Introspective Power, Inc.

Table of Contents

Introduction to STOP

Streaming Transmission One-time-pad Protocol or STOP provides several techniques to allow a One Time Pad aka Vernam Cipher to be utilized to encrypt data traveling across any network including the public Internet. Independent, third party testing has shown that STOP also removes all known network vulnerabilities and protects the edge, hiding the endpoints from network scanning. STOP can be embedded into applications and is a foundational technology for Zero Trust Architecture. STOP simply removes the attack planes found in the public Internet.

STOP is seminal technology that removes all objections to using an OTP for the encryption of network traffic. By using an OTP, we completely remove any threat of cracking encryption. This is the only real protection from the looming Quantum Computing threat.

STOP also utilizes the proven military technique Moving Target Defense. By moving the target of interest in a random way, the adversary will not be able to locate the assets of interest. When used over the Internet or other IP network, these assets of interest would be the data streams STOP uses. In that case, it can also rotate Ports and IP addresses. In the case of a transport network, it can rotate channels or wavelengths. For radio, it can perform deterministic frequency hopping.

This paper explores STOP technology and how it allows an OTP, the only known encryption method that can not be cracked, even in theory, to be utilized for protecting data.

Removing Objections to the One Time Pad

There have been multiple reasons given for why using a One Time Pad in a network is no more secure than using Algorithmic encryption. The main reason given is, if you use Algorithmic encryption to encrypt the entropy that is stored in the virtual “Pad”, it will be no more secure than the encryption itself. While this is an argument, it is not completely true. If you only send the random information using Algorithmic encryption, it will no longer be susceptible to Brute Force attack because the data being sent is random. In order to know you have a solution in a Brute Force attack, you need to know something about the data to know when you have a correct solution. This chink in the reasoning opens the question - are there ways to get around all the other arguments?

With STOP, we set out to address each point one at a time:

1. **Issue:** The entropy is no more secure than the method used for encrypting it.
   Solution: The entropy is encrypted with a One Time Pad so it is perfectly secure.
2. **Issue:** The Pad material has to be as long as the data being encrypted.
   Solution: Expand the entropy in a cryptographically secure way.
3. **Issue:** If the pad material is as long as the data, there will never be enough material to encrypt the pad itself.
   Solution: By expanding the entropy and generating the virtual “Pad Page”, there is more than enough material to encrypt both the data and the One Time Pad.
4. **Issue:** The Pad material will get out of sync and there is no way to recover.
   Solution: This is no longer true and methods of resyncing data are well understood. This

argument likely predates TCP streaming that guarantees packet delivery.

5. **Issue:** The Pad material doubles the amount of data being sent and is inefficient.
   Solution: Today we have a glut of network bandwidth. In nearly all cases, this is simply no longer a valid issue. Also, by expanding the entropy, the Pad material sent is only a fraction of the size of the data.
6. **Issue:** The Pad material can be intercepted as it travels through the network making it easy to intercept and collect.
   Solution: The entropy is: a) encrypted with a One Time Pad, b) utilizing Moving Target Defense so it can not be easily intercepted, and c) the entropy sent is not the Pad material that is actually used for encryption because of expansion.
7. **Issue:** You cannot generate enough True Random to satisfy network encryption.
   Solution: Breakthroughs in Quantum Random Number Generators allow for Gbps of quantum to be generated. This, along with entropy expansion, solves this problem.

With these issues addressed, we can move forward with utilizing OTP encryption for network traffic. Next, we will provide an in-depth look at both the methods and technical reasoning behind STOP technology.

Overview of STOP Technology

STOP has three main components:

1. Moving Target Defense - Removes the ability to easily intercept data.
2. One Time Pad - This is an encryption technique derived from an unsolvable problem.
3. Entropy Expansion - By expanding entropy in a cryptographically sound way, we solve many issues with transporting entropy materials across the network.

This section will explore the technical methods and merits of these primary concepts.

MTD - Moving Target Defense

Recently, some of the techniques used in STOP have been quantified by various groups involved in national cybersecurity and are being referred to as Moving Target Defense[1]. The technique was successfully used in the 2015-2016 DNC hack[2] of email messages but was used as a Moving Target Attack instead of defense[3]. This was used to hide the exfiltration of data allowing collection of emails to continue for 9 months. It was not until another, more easily detected, Russian attack from a different agency that it was discovered. Had that second attack not happened, it is likely the original attack may have never been discovered.

This is clearly a powerful method to hide data. Using MTD (defense), this method is even more effective. For defense, you are not necessarily trying to hide the data but to stop attackers from latching onto a data stream to attack it. This method constantly changes key information. In military terms,

1 "CSD-MTD | Homeland Security." https://www.dhs.gov/science-and-technology/csd-mtd. Accessed 28 Dec. 2016.

2 "Bears in the Midst: Intrusion into the Democratic National ... - Crowdstrike." 15 Jun. 2016, https://www.crowdstrike.com/blog/bears-midst-intrusion-democratic-national-committee/. Accessed 28 Dec. 2016.

3 "Moving target defense vs. moving target attacks: The two faces of ...." 4 Jan. 2016, http://www.networkworld.com/article/3018881/tech-primers/moving-target-defense-vs-moving-target-attacks-the-two-faces-of-deception.html . Accessed 28 Dec. 2016.

MTD constantly changes the attack vector required to attack. The way this is done, finding the new stream before it changes again is currently not possible. Given the processing power required by the network equipment, it is not clear if a data stream utilizing MTD would ever be exposed to attack.

Polymorphic Networking

The concept of Polymorphic Networking, a set of techniques defined by Moving Target Defense[4] (MTD), is a concept of moving around or changing the characteristics of a network connection to make it impossible to latch onto.

Polymorphism is a concept in Computer Science where one thing or object can take on many forms. In polymorphic networking, this is exactly what is happening. In this system, a network connection can be viewed simply as a channel and can move or change over the lifespan of a network transmission. When using STOP, the system implementer only sees “send” and “receive” and is unaware of the changes happening to the network itself. In a SmokeNet private network implementation, it looks like any other Internet Protocol (IP) network. The complexity of the Polymorphism is masked completely from the end user and the system implementer.

The polymorphic changes can be physical and/or logical. This hides the information being transmitted as there is no way of knowing where this information is in the network at any given time. The starting point is quickly changed and will never transmit valuable information across the network itself.

For a simple example in an IP network, let’s start with an SSH session on its standard listen port of 22. A polymorphic approach may immediately hop to initiate a new connection on another port, in the ephemeral range, of let’s say 12,304. The listen socket, which can be detected by a port scan, is never open that long. It is dropped immediately after the single connection is made in sub-second time. What this means is it’s highly improbable, nearly impossible, for a port scan to detect this port leaving the subsequent starting port hidden from detection. Even if one hop was detected, this hopping will continue to happen. To detect the specific communications channel after each switch makes the detection of an entire transmission more and more improbable.

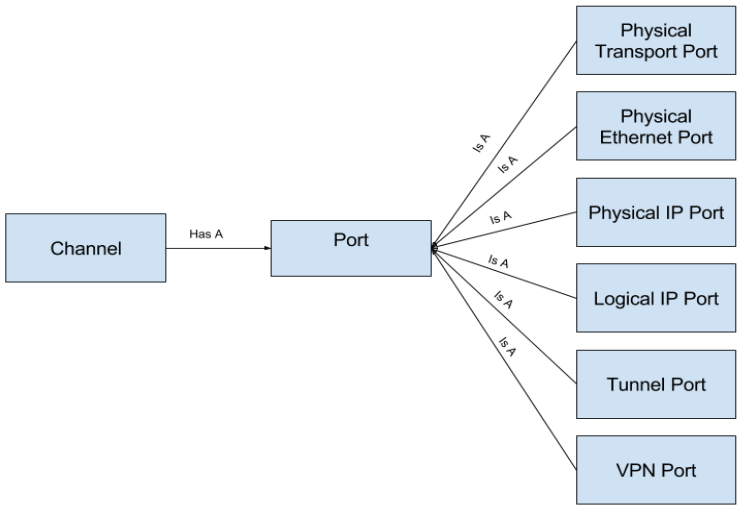

Figure 1: A code Class diagram displaying a few possibilities for port polymorphism

As that was a simple, single IP network example, we can carry these changes even further to include different subnet(s), physically different networks including multiple carriers, and/or different network types such as layer 3 TCP/IP and layer 1 or 2 private line. These can also include path diversity by using private lines, MPLS tunnels, or SD-WAN. When we start to use multiple carriers on different systems with path diversity, we begin to further decrease the

4 "CSD-MTD | Homeland Security." https://www.dhs.gov/science-and-technology/csd-mtd. Accessed 27 Dec. 2016.

probability of being detected. Figure 1 shows an example of a code Class diagram displaying a fraction of the multitude of possibilities for port polymorphism in code.

Ancillary to the high level changes, there must be a process to rotate the port hence changing the underlying properties of the channels' connectivity. To accomplish this, there are really only two options: 1) sequential change or 2) parallel change. Figure 2 shows a high level, simple flow diagram for sequential change. While this will work and likely has some benefits in ensuring one Port/Channel change has been made before making another, we lose some of our unpredictability. To improve on the randomness and make the change less predictable, we can do these in parallel. This would do all or some of the changes at once with random, natural delays occurring due to resource constraints. We could also add random delays to the sequential version or randomly change the sequential order of changes.

The Polymorphic and MTD concepts are core elements of STOP patents which, going back to 2013, predate any literature we have found on the subject with regards to networking. In short, Introspective Networks is the inventor not only of these techniques, but also a working implementation that makes STOP likely the only truly secure Polymorphic Networking system available to the public.

Polymorphic Encryption

Similar to Polymorphic Networking, Polymorphic Encryption is constantly changing some aspect of the encryption in a way that makes it more difficult to decipher. In the classic code definition, an algorithm is used to constantly change the encryption/decryption key so that it can not be easily guessed[5]. To do this, the key and decryption method must also always be changing. This can be handled by outputting the decryption as an ever changing OTP or having an algorithm both sides agree on that constantly rotates. There are also notions of Polymorphic Encryption Algorithm (PEA) for quantum computing[6] and Polymorphic Encryption and Pseudonymisation (PEP) which is used to distribute access to data without giving access to the actual holder of the data.[7] These are both interesting and the latter can easily be incorporated into the STOP stack but, for this conversation, we will deal specifically with Polymorphic with regards to Polymorphism which, at its base, is the same definition provided for Polymorphic Networking: one code object that can take on many different characteristics without the user or system knowing or needing to know the changes to the implementation details. Moreover, what we will describe is very much in line with MTD principles in general as encryption also will become a moving target.

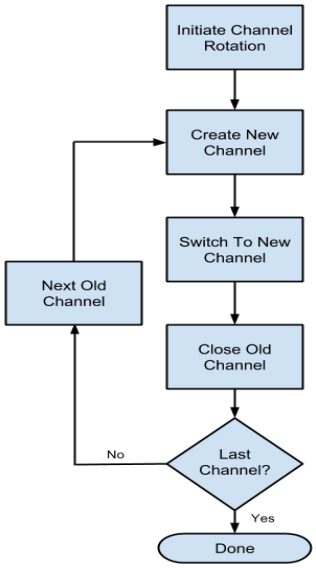

Figure 2: Simple sequential rotation of network Channels. The change would really be to the underlying port but channel is considered new because of the dynamic polymorphism taking place when the port is changed out.

With this fundamental definition, Polymorphic Encryption can be as simple as changing the keys in a symmetric encryption scheme. It could also mean the changing of the underlying encryption cipher algorithm. For example, we may rotate our keys for AES256 periodically

5 "Polymorphic code - Wikipedia." https://en.wikipedia.org/wiki/Polymorphic_code. Accessed 28 Dec. 2016.

6 "PEA: Polymorphic Encryption Algorithm based on quantum computation." http://openaccess.city.ac.uk/2509/.Accessed 28 Dec. 2016.

7 "Polymorphic Encryption and Pseudonymisation for Personalised ...." 30 Sep. 2016, https://eprint.iacr.org/2016/411.pdf. Accessed 28 Dec. 2016.

and then, without letting an eavesdropper know, switch to another strong symmetric algorithm like Blowfish[8]. By constantly changing keys and cipher algorithms, it becomes very difficult to determine how to even start to decrypt discovered data. Even if data is captured, the key or the cipher algorithm are not easily discoverable because this information is encrypted using an OTP. To make this work, transmitting the change in key and/or algorithm to the other side in secrecy is the critical step.

In another incarnation of Polymorphic Encryption, an algorithm randomly changes both the encryption and decryption code. Like the prior examples using existing, predefined algorithms, the new decryption code would need to be transmitted to the other side. For networking, this code would likely be longer than a Pad itself so it might be impractical to transmit. Another option with this technique would be to output a “Pad” that can be decrypted simply on the other side using a single, known arithmetic logic-unit (ALU). Again, the Pad would need to be transmitted securely it therefore encounters the classic key exchange problem and even if the algorithm is rotated, because it is calculated, it can be solved for. For these reasons, these techniques provide no advantage to, and are less efficient than, STOP using natural entropy.

Research into Polymorphic Encryption will reveal many different implementation variants. While each one is Polymorphic in nature, they are all, including simply rotating keys, forms of Polymorphism[9]. In fact, an OTP system itself is Polymorphic by design simply based on the fact the key is always changing for the length of the data being sent. In a stream, it goes on until that stream is discontinued. Used to secure an entire network, it is ever changing and never predictably repeating as long as that network is needed. Because all of these, including the OTP, are changing the underlying encryption in a substantial way, they should all be considered Polymorphic and an encryption based MTD technique. It should be noted that OTP encryption is the simplest solution, requires the fewest resources and has the lowest possible latency.

The One Time Pad and Expanded Entropy

There is only one currently known way to encrypt data with perfect security: the One Time Pad aka Vernam Cipher. In order to use this method of encryption, you need one byte of random for each byte of data. The Pad or Pad book of information is the key in this type of one-way symmetric cipher. In order to have two-way communication, we need to have two sets of pads – one for each direction. Very quickly, we run into an issue of how to generate the amount of random to protect our communications. When we start to contemplate video streaming and the like, it becomes a daunting amount of random required for a single communication stream.

Quantum Random Number Generator (QRNG) technology that is also certified as a True Random Number Generator (TRNG) solves part of this. These are commercially available for speeds up to 1Gbps from a single line card or processor. While this would handle multiple connections at what would be a reasonable 5 Mbps, this would still be inadequate on a massive scale network.

In order to satisfy that need and realistically use an OTP in a network economically and efficiently, we need much more random Pad material. By expanding the entropy in a cryptographically secure manner,

8 "Schneier on Security: The Blowfish Encryption Algorithm." https://www.schneier.com/academic/blowfish/. Accessed 4 Jan. 2017.

9 "Polymorphism (computer science) - Wikipedia." https://en.wikipedia.org/wiki/Polymorphism_(computer_science). Accessed 28 Dec. 2016.

we solve this problem. We could essentially create hundreds, thousands, or even millions of times more entropy than was generated by a TRNG.

This method also solves three other issues of using an OTP in a network:

1. The random Pad information no longer doubles the size of data that needs to be transported across the network.

2. The data that is sent across the network is no longer the exact material that is being used for encryption. Interception and decryption is no longer a sure way to decrypt the data being secured.

3. We can now encrypt the random Pads with a One Time Pad making interception a nonissue as it can not be decrypted.

Testing using the Dieharder test suite on SmokeNet expansion has shown that, if the random material is derived from a TRNG, the expanded material is of the same quality as the original material. This leaves an open question as to whether the expanded material is still True Random or Pseudo Random. Since it passes the same tests and exhibits the same properties, this author argues that expanding random in a way that utilizes the properties of the random, implemented carefully, is still True Random.

Putting the Pieces Together: How STOP Works

One of the most crucial elements to making any MTD solution really work is starting with at least one streaming channel of directly uncrackable, encrypted data. This channel, on its own, provides a method for the two sides to communicate without fear that the messages can be intercepted and cracked as it traverses a network like the Internet without special, controlled equipment. The only method known that is provably uncrackable by itself is an OTP. This channel not only contains the data to be transmitted but all the messages that are required to make the MTD work in a deterministic manner without shared calculations on both ends of a data connection. This also comes into play during the first communication as the system is “primed” with two small sets of Pads. This happens one time only at initiation. These sets of Pads let the system connect for the first time to a known port without fear that the data will be intercepted or cracked. This Pad needs to be large enough to encrypt the first port hop away from the well-known port for Channel 0 and also share the key for the Channel 0 symmetric or asymmetric encryption. Once Channel 0 is established, the process of sending Pad material across the network starts and subsequent channels can be brought up using Pad material to encrypt the data.

To get this first, small Pad for priming the system to each side of the connection, a variety of techniques can be used. The easiest is to simply send it from a third system that brokers this exchange. Another method is to manually prime each side, temporarily storing the pads on removable media like a USB Stick. The two can be combined so that the manual method is used to prime the connection to the central broker system. In this scenario, the clients would only be primed once and then would receive subsequent priming pads from the centralized broker service that is already transmitting messages over a STOP enabled connection. A third method would be to put the priming pads in a message or reasonably secured remote file storage system but set an expiration for use. This method

would send the priming pad directly in an email or store it in a cloud storage location. Again, this technique can be used with a centralized broker to prime that specific connection.

To protect the data further, we want to ensure it is difficult to find. We know for a fact that, without a compromise to the system on either end, there is no way this information can be directly recorded and later decrypted using the current state of the art. As we laid out earlier, the reason for this is the Polymorphic Networking that's been employed. Current "man in the middle" (MITM) recording methods require knowing not just the IP address, but, to understand what the data is related to, also the port to be recorded. Without these two pieces of information, there is no way to understand what the data stream is related to. Encrypted in a way that is, as we have seen, fundamentally secure, the content provides no clear information with regards to what it is being used for. To understand what the data is related to, the connect or listen port must be known. With polymorphic behavior, after the first connection is made, rotation goes to an unknown port. That listen port is closed immediately in sub-second time so, even if there is a port scan, it is highly improbable the listen port would be around long enough to be detected. Lastly, the same entropic randomness used for the Pad is also used for this port rotation.

The question probably being contemplated is, how does one get truly random entropy for the Pad and random number generation? The answer to that is simple and all around you. Taking any measurement from the analog world and measuring it to a high enough fidelity produces random, unpredictable fluctuations in readings. For example, the ambient temperature of any area, enclosed or otherwise, measured to a Pico reading or 10^{-12}, even in an extremely controlled environment, will fluctuate wildly in an unpredictable manner. Many other measurements taken to a similar level of fidelity, such as a voltage measurement or the reading from a gyroscope or accelerometer will produce similar results. The key to using analog sources is to ensure you have a measurement capability to provide the desired level of fidelity where readings start to become entropic. Before moving forward, to ensure the high fidelity of readings, a test should be performed that takes in a large sample set of data from the source of entropy and validates that the readings have good, valid number frequency across all possibilities within the desired numeric range being captured. This is a critical step in validating the measurement source is providing real, accurate high fidelity readings and not providing some kind of approximation with a limited numeric set.

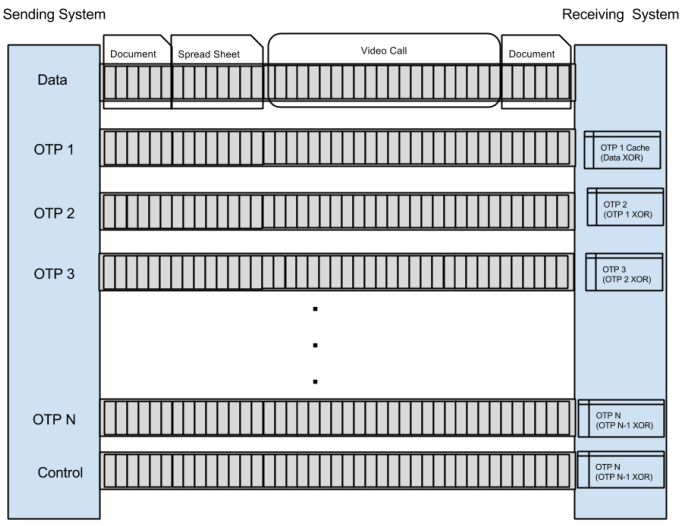

Figure 3: OTP caching example diagram. For channels, the numbers are the inverse of the OTP numbering and zero based. So, the Control with OTP N is actually Channel 0. The real thing to take away from this is the OTP can and must be cached ahead of the data channel. This caching can be done at any time and should have a randomness to it. The key is to have the network speed to stay well ahead of the data itself.s

Taking this concept to the next level, multiple high fidelity, analog entropy readings can be combined using binary operators to further ensure that someone cannot guess one analog source or the other. This ensures the most secure Pad possible making a true OTP system actually live up to its mathematical proof.

Now that we have viable Pad material, we need a way of transmitting it from the sender to the receiver in a reliable manner. This is where Introspective Networks STOP's seminal network Cyber MTD inventions really start to take shape. STOP can use both Polymorphic Networking and Polymorphic Encryption to provide secure key transmission that transcends simple symmetric or asymmetric encryption itself. That said, if we think about how brute force attacks are conducted, if our keys are not compromised and we send nothing but non-repeating entropy across that channel, it would still be impervious to brute force techniques because, as we discussed in the introduction, the underlying data can never be guessed. For brute force to work, you have to know something about the data being sent at a given point in time. In the case of sending entropy derived in the analog world that never has two repeating bytes, brute force techniques, very logically, become impossible. Even with all that, it is better to leave nothing to chance and, using Polymorphic Networking and OTP encryption (even for the Pad material), add MTD to the security provided by STOP.

Another concept we've mentioned that is worth explaining is how this works in Time and Space. Space should be obvious as we are moving the channels in physical or virtual space as it relates to the network(s) used. The time dimension is another story altogether. This is done by sending data at different intervals. In the simplest implementation, only two channels are used: 1) for the OTP stream and 2) containing ciphertext. The time dimension comes into play as the OTP channel will be cached on the far end so, the key stream needed to decrypt the data is never sent at the same time as the data itself. If we were to have more than two channels, subsequent channels used for encryption would have a completely new OTP stream. This stream would be encrypted with the prior OTP stream. What this adds is more complexity to the variables in the time and space dimensions. Caching, if able to stay consistently ahead of data decryption needs, can have random temporal offsets to create further randomness in the timing. We also account for possible decoy streams that have nothing to do with anything but add noise and complexity to both the time and space aspects of STOP. Figure 5 shows how this data is sent across the network. With a little imagination, you can envision how each of these streams could be sent at vastly different timesFigure 6 provides a further example of how this rotation works with multiple ports. This figure emphasizes the rotation which, as we've discussed, can also contain a random time component along with its inherent space movement.

One thing that truly separates STOP from all other forms of communication/data encryption and obfuscation is the deterministic nature of all system MTD changes. There are no calculations on either end to guess or crack. Data is moved, as we've discussed, in both space (across the network) and time (pertaining to the frequency of these changes) in a random manner that, while it may be bound, cannot be guessed with any certainty. The bound set of possibilities is too large and the time to discover too short to make discovery practical. This

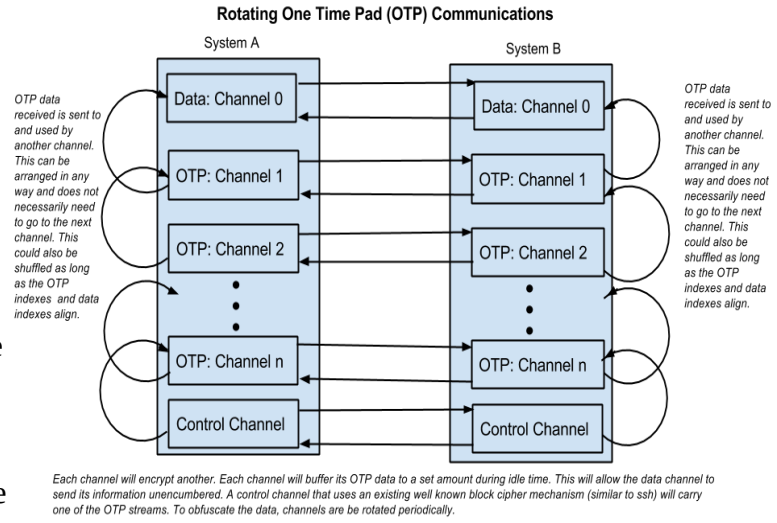

Figure 4: Rotating One Time Pad Communications

determinism can only be performed if the messages are guaranteed to be secure which is made possible by analog sources of entropy and following all the rules of the OTP.

As the last step, we expand the encryption in a way that keeps both sides in sync. It is critical that this not be an Algorithm but a single calculation that has a simple proof that will not aid in cracking the encryption. The nature of mathematical proofs is the reason no symmetric or asymmetric key encryption that relies on math will ever be secure. There will always be a way to derive the key or keys through simple mathematical proof. The myth of intractable problems is a fallacy as you can always solve each possible problem and create a lookup table. So, in our expansion, we create a system that uses a single calculation like the One Time Pad. The same simple proof for the One Time Pad - "x $\oplus$ y = z" - will hold true if you can keep x and y secure. We have already established we are doing that for the pad material so it goes without saying that the expansion is extremely secure.

Summary

STOP technology uses multiple techniques to make it possible to use a One Time Pad in a network. The techniques remove prior objections to using an OTP for network encryption point by point. One of the techniques used is Moving Target Defense. This, in an IP network, has been proven to remove all known network vulnerabilities. Finally, we use Entropy Expansion to allow for cryptographically secure expansion of the entropy. Altogether, we have the ability to send a stream of entropy securely across a network to provide encryption that can not be cracked; even in theory.

The Moving Target Defense, as well as stopping the interception of data, removes all known network vulnerabilities. This also protects the network edge as there are no long-standing open network ports. A server that is only servicing a private network using STOP will look like any other computer not acting as a server.

With these techniques, all threats, current and future, are protected by STOP technology when traversing a SmokeNet network. The OTP encryption used in STOP creates an unsolvable problem. Even a Quantum Computer can not solve the unsolvable. This makes STOP and SmokeNet not just Quantum Resistant but Quantum Proof. This makes SmokeNet a great option for ZTA networking