Multiple Use Pads

Publication date: March 19, 2017
Revision date: April 6, 2017

Joseph Latouf
CORA Cyber Security Inc.
519-254-4703
Windsor, Ontario

Abstract

There is a problem when current spending on cyber security valued in excess of $70 **B**/year, fails to prevent cyber criminals from costing the global community more than $500 **B**/year. The speed and sophistication of cyber-attacks have been steadily increasing in frequency and effectiveness. Brilliant minds continue to devise new attacks on encryption, such as: man-in-the-middle attacks, statistical attacks, frequency analysis and pattern recognition, prime number generators, and side-channel attacks.

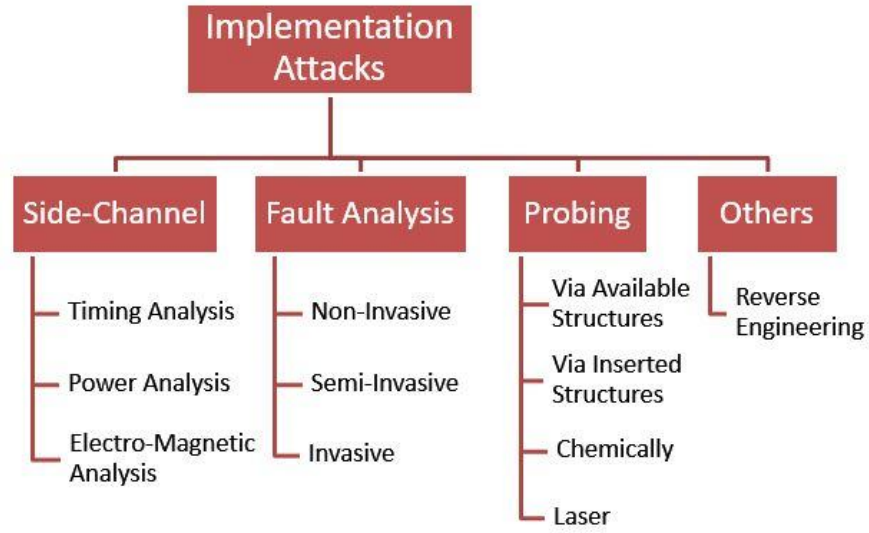
Information theory and cryptography recognize 'One Time Pads' as '**perfect secrecy**' in theory, however, until today, they have remained impractical to use. CORAcsi has invented MUP as a **practical and reusable** form of a OTP. **MUPs** are 'probability' based, eliminating the effectiveness of these attack vectors. The only possible attack is a brute force attack, which due to the probabilities involved, require more time than is available to our universe.

The Problem

One-Time-Pads (Vernam's Cipher [1]) have been widely, and theoretically accepted as a perfect cipher. **They are in principle, impossible to crack. In practice, OTPs are not practical**: 1) the pad must be randomly generated, and, 2) OTPs can only be used once.

At the advent of binary systems for computational analysis (computers), memory and processing power was expensive and hard to obtain. This led to brilliant mathematical implementations of encryption that protected data, including communication. Due to the impracticality of OTPs, modern encryption is based upon limited, finite size keys – which produces creative attacks other than a 'brute force' attack.

Figure 1 - Sample of attacks on 'finite key size' Encryption



Today, memory and processing power are inexpensive and abundant. Capable mathematicians and technologists are highly motivated in their attempts to break encryption; they are succeeding. They have devised many attacks such as, man-in-the-middle, statistical, side channel attacks, and many more.

It should be noted that there is a correlation between 'Cyber hygiene' and the 'brute processing' involved with 'finite key size' encryption.

The Solution

'Multiple Use Pads' (MUP *Patent pending***)** are an abstraction and procedural implementation of One Time Pads in a digital environment that **allow for the repeat, secure, and practical use of a given, limitless pad**. MUP is a probabilistic approach to cyber security;

one that combines the size of an OTP with a randomized form of steganography that results in a practical process for which the only attack is a brute force attack, and such an attack cannot reliably succeed.

MUP represents several practical techniques that, when used properly, remove the 'one time only' limitation that has previously been applied to these conveyors of 'perfect secrecy'. Moreover, these techniques reduce the dependency on maximizing the entropy of a specific pad.

Our 'secret sauce' uses multiple methods of producing the desired result of Multiple Use Pads, which may be categorized broadly into one of two categories:

1. Pad morphing
2. Message diffusion

Both result in the ability to use 'perfect secrecy' in a randomize manner to produce an implementation in which Pads can be securely reused, while concealing any potential, statistical connections (redundancy) between multiple uses with different messages.

Message diffusion removes this ‘redundancy’, effectively removing a 2-pad attack. That said, combining message diffusion with morphing pads is the ultimate in perfect secrecy.

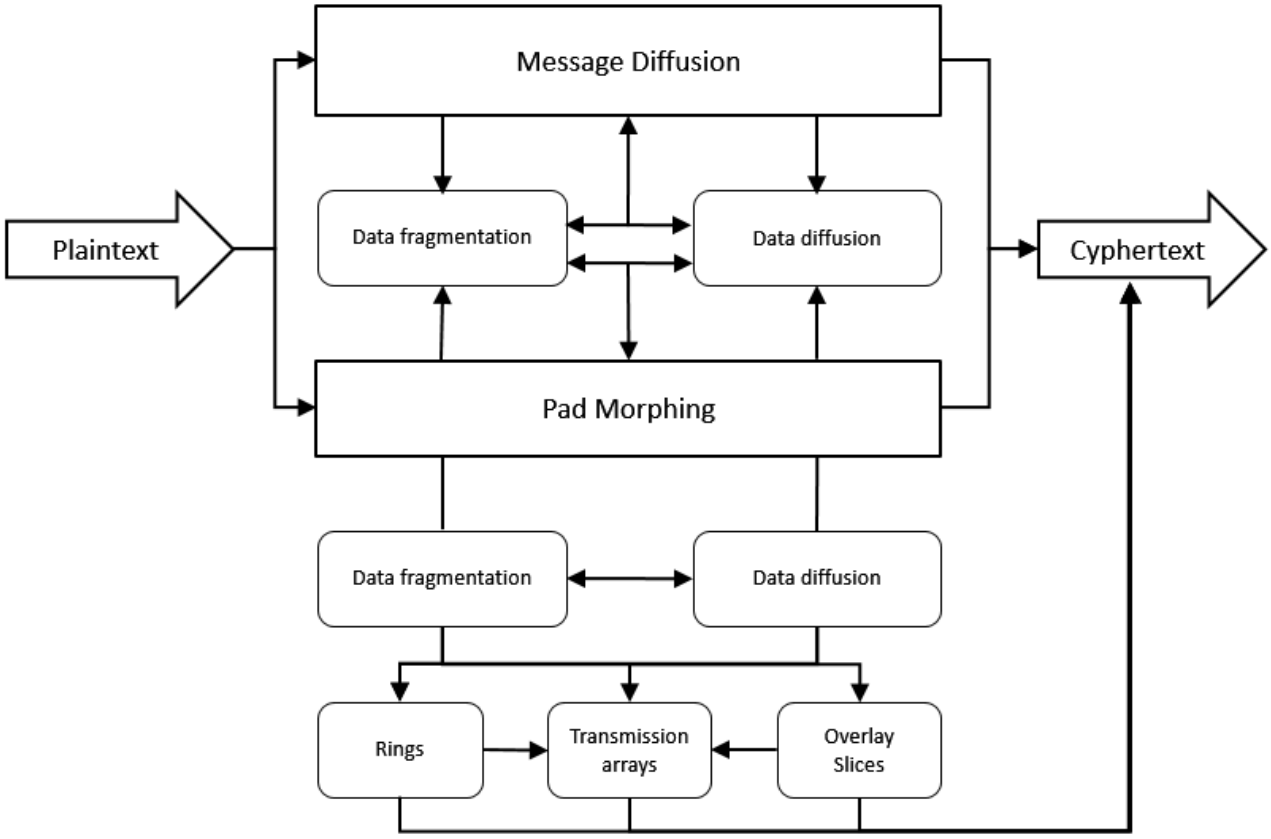
Techniques in which the pad inherently morphs (adapts) to varying implementations and boundary conditions produces probabilistic implementations that are too numerous to anticipate or identify. The combination of 2 or more of these techniques secures data while reducing the dependency on pad entropy and empowering the reuse of existing pads.

Recall that two-pad attacks are notoriously successful (Verona Project) since the capture of two ciphers results in the XOR of the 2, original plain texts messages.

$$(m_0 \oplus \text{pad}) \oplus (m_1 \oplus \text{pad})$$
$$= m_0 \oplus m_1$$

As cited in many resources, the redundancy of the English (and other) languages, along with ASCII itself, lends itself to statistical tools that will allow for the realization of these messages (and the pad).

Figure 2 - Overview of MUP (decryption works in the reverse direction)



Message Diffusion

As is prevalent in the literature, Vernam's Cipher, also known as OTPs, or perfect secrecy, involves the "exclusive or" (which is also represented as XOR and has the symbol ⊕) operation performed on a binary level (bitwise).

$M = D \oplus OTP$

Where M ≡ Mapped data
D ≡ original data
OTP ≡One Time Pad

Most forms of encryption fundamentally use some form of 'pad morphing', ranging from poorly designed protocols such as 802.11b WEP to more robust methods such as Salsa 20 [2] (eStream [3]). The concept of stream ciphers; using a limited size key with a Pseudo Random Generator (PRG) could be viewed under a lens called "pad morphing".

Unlike these methods, MUP does not impose the use of a finite size key, or seed, to generate the larger streaming pad. This provides advantages to MUP that more closely aligns MUP with perfect secrecy [4]; probabilities for breaking the resulting cipher are thousands of orders of magnitude larger than all other forms of encryption.

Message diffusion is implemented in two manners: a) data diffusion and b) data fragmentation.

a) Data diffusion

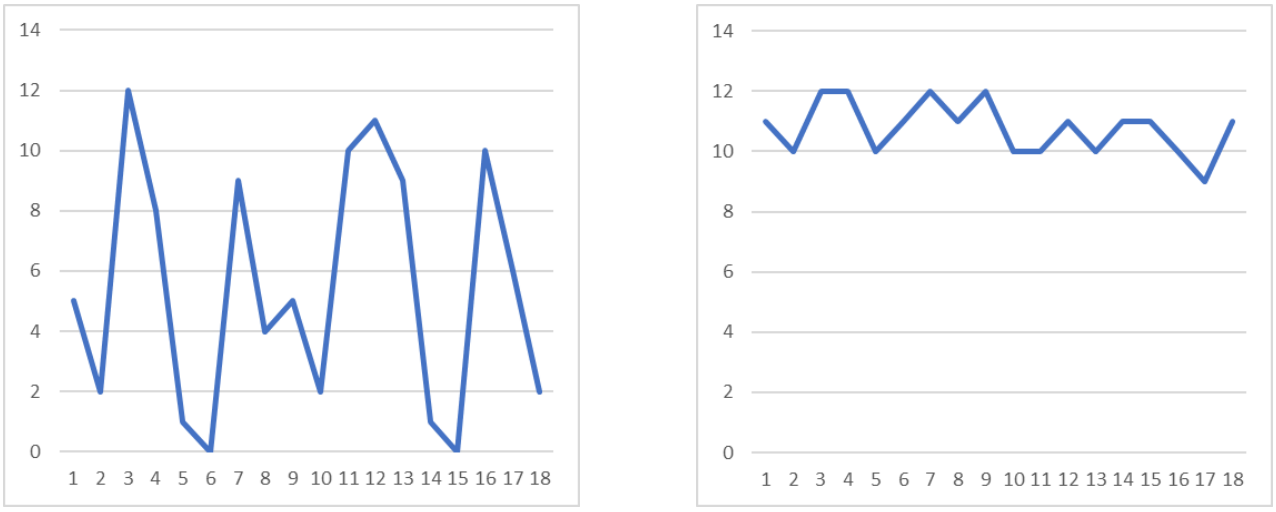
Data diffusion has rudimentary implementations in related techniques, such as with Huffman encoding [5] and related compression techniques [6].

In addition to methods used for compression, dispersion may be applied to the data, making the statistical methods of determining the pad, given two messages, incompatible with MUP.

Data diffusion (dispersion) is not dissimilar to steganography, except that:

1. The ratio of "hidden data" to "random data" is significantly lower.
2. The message body is composed of "random data" rather than "meaningful data (pattern specific)".

Figure 3 – Sample frequency distributions by byte – before and after dispersion

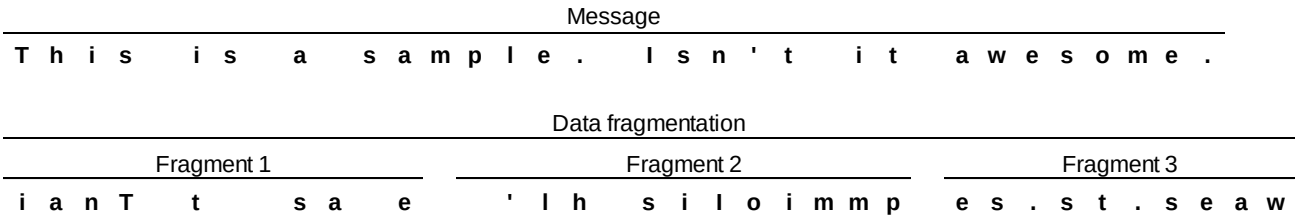


b) Data fragmentation

To further eliminate statistical analysis on the messages that may be XORed together, bytes, may be channeled into multiple fragments thereby removing any 'connectedness' that may naturally occur due to language nuances (human or computer based). These fragments may then be distributed to different locations (files), or may be recombined into a stream for real time, secure communication.

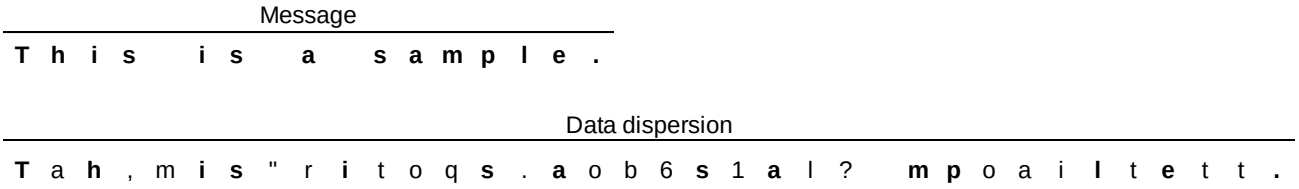
Consider the following examples:

Figure 4 - Data fragmentation



Notice this illustrates the redundancy cited earlier: there are 5 's', 3 'e', 3 'a', 3 'i', 2 'l', 2 'm', etc. With dispersion, these number might be normalized to all have the same # of occurrences (all characters appearing "5" times), or they could be reversed (10 'm', 10 'l', 8 'i', 7 'a', 7 'e' and 5 's' characters), or they could be mapped to a different distribution all together.

Figure 5 - Data dispersion



Notice in this example, the dispersion has removed the connection between 'most' connected letters, and, it has changed the distribution of characters. Previously 's' was the most frequently occurring letter, however after dispersion, both 'a' and 't' occur more frequently than 's'; the entire distribution has changed. It should be noted that these examples are simple examples for illustrative purposes.

Combining data dispersion with data fragmentation results in the best implementation of Message diffusion.

Pad Morphing

The literature often cites that 'perfect secrecy' is obtained if the key (pad) is the same size as the plain text. Herein lies the first divergence from this base line introduction to OTPs:

A. Variability.

Use additional boundary conditions in which:

1) The **length** of the MUP varies.
2) The 'CM **offset'** used to map the data varies.
3) The 'data offset' used to map the data varies.
4) The number of XORs performed on the data varies (**overwrapping**).

Consider the following examples; while the examples refer to the 'bit position', it could also refer to the 'byte position'.

Figure 6 - Sample mappings with variation alone

Len of Data: 8 Data offset: 0
Len of MUP: 8 MUP offset: **0**

		Bit Positions							
Plaintext:		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MUP:	⊕	**0**	1	2	3	4	5	6	7

This implementation in which both the data and MUP offsets are 0, and both have the same length, illustrates the prior art with respect to OTP. This OTP should never be reused with data that has 'redundancy' associated with it.

Figure 7 - Example 2 of Variability

Len of Data: 8 Data offset: 0
Len of MUP: 8 MUP offset: **1**

		Bit Positions							
Plaintext:		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MUP:	⊕	**1**	2	3	4	5	6	7	0

In this example, there is a minor change using the MUP's offset; the resulting MUP is different from the original implementation.

Figure 8 - Example 3 of Variability

Len of Data: 8 Data offset: 3
Len of MUP: 10 MUP offset: **5**

		Bit Positions							
Plaintext:		**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**
MUP:	⊕	**5**	6	7	8	9	0	1	2
	⊕	3	4						

This is the most interesting example of Variation, in which the length of the map is varied (number of XORs performed), producing a second XOR operation on the first two mapped bytes (the 3rd and 4th bit positions within the data).

This varied implementation potentially impacts the entropy of the resulting map. This lends itself to a “chaos” perspective; in some cases, the entropy will be decreased slightly, and in some cases, the entropy may be increased slightly. As the length of data and MUP increase, the corresponding reduction, or increase, in entropy diminishes. In all cases, the strength of the MUP is maintained beyond a single use.

Dispersion

Similar to data dispersion described earlier, dispersion may also be applied to the MUP itself; inserting bytes into the stream to disperse the original MUP. There are multiple implementations for dispersion. For MUP, the following implementations have met with success:

1) A byte at index n is inserted into the same MUP at index m. Alternatively this byte could be truncated or otherwise manipulated before insertion. This may be done with 1 or more bytes.
2) One or more bytes could be generated, then inserted into the pad to disperse the pre-existing bytes.

Block implementations

Block implementations and the subsequent topic of ‘Morphing’ are described in greater detail in a subsequent paper that deals more exhaustively with MUP. This white paper is an introduction to MUP, and so, only a cursory introduction to these concepts is included.

Rings

The concept of cyclic implementations dominates the methods used for encryption. When the various forms of encryption are distilled to the fundamental concepts, this author identifies a few key elements:

- Logical operation(s); example - an XOR operation.
- Patterns of applying these logical operations; example Feistel Structures (or rounds).

MUP can be creatively applied to blocks of data, cycling over this "ring" such that they repeatedly cycle around this ring of data.

Rings further eliminate the concern for 'reusing the pad' and further minimizes the dependency on entropy. Rings are best applied to select 'pieces' of a data stream; enough to minimize the potential complications of entropy and pad reuse.

Rings are useful for validation (inquire about our detailed paper on MUP).

Figure 9 - MUP and Rings

data; len = 10

pad (fragment); len = 9

		Bit Position									
		1	2	3	4	5	6	7	8	9	10
⊕	1	1	2	3	4	5	6	7	8	9	1
⊕	2	2	3	4	5	6	7	8	9	1	2
⊕	3	3	4	5	6	7	8	9	1	2	3
⊕	4	4	5	6	7	8	9	1	2	3	4
⊕	5	5	6	7	8	9	1	2	3	4	5
⊕	6	6	7	8	9	1	2	3	4	5	6
⊕	7	7	8	9	1	2	3	4	5	6	7
⊕	8	8	9	1	2	3	4	5	6	7	8
⊕	9	9	1	2	3	4	5	6	7	8	9

max number of iterations = 9

Transmission arrays

A transmission array incorporates a random carrier with a specified pad and plain text. The random carrier's boundary length must differ from that of the pad. This is an effective means of transferring data back and forth. Each transmission involves another random carrier. The wrapping of the pad and key onto this carrier eliminates the statistical possibility that multiple messages may be used to determine the key.

Figure 10 - Example - Transmission array

		Byte position																							
Random Array:		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Data:	⊕			0			1	2			3				4				5						6	
MUP:	⊕		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15								

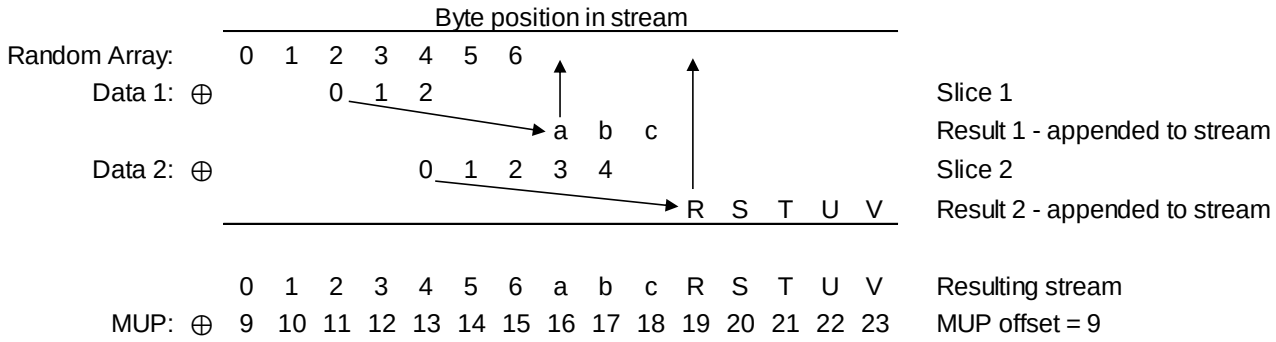
Transmission arrays are used to form Transmission Chains described below.

Overlay slices

Overlay slices are groups of data, either blocks of comparable size, or varying size blocks of data. Each slice is XORed with a slice from a randomly set stream of bytes, and the resulting XORed bytes are appended to this stream. The position within the original stream, and the relative order of being appended is governed by a control array. Each slice becomes part of the stream that is used for subsequent slices.

This entire stream may then be safely mapped with a MUP since there isn't a logical or statistical relationship between bytes contained within the original message.

Figure 11 - Example - Overlay slices



Notice the FILO (First In Last Out) process.

Morphing

The dispersion described above is an elementary form of morphing. More elaborate methods may be incorporated by implementing:

1) Control Arrays (dynamic)
2) Adaptation
3) Challenge – response communication
4) Transmission chains

Control Arrays

Control arrays are used to hold the parameters used to specify a unique MUP, and related techniques that are in use during a particular encryption of data. Control arrays also contain parameters for signaling adaptations and communication (response requests) with the various end points.

Control arrays may be organized into segments (blocks) for the parameters used for adaptations and responses. These segments (parameters) may be morphed during the lifetime of a MUP to provide many benefits including:

1. Unique implementations that are specific to a particular set of ‘shareholders’.
2. Temporary implementations that are unique to a particular time; these are similar to the proposed use of rotating keys for V2V communication, however, they are much lighter and simpler, while producing significantly greater numbers of permutations.

CAs may incorporate any number of segments, depending on the needs in a given architecture and implementation. For example, V2V communications would involve different sections as compared to an exclusive connection between an OEM and an individual vehicle (updates, repairs, etc.) and within the vehicle's multiple BUSs.

Adaptation

Adaptation allows the Control array to:

1. change its parameters.
2. change the order of its parameters.
3. initiate controlled resets on parameters, and or pads.

The adaptation and response blocks may work together in the CA shown above to modify the values contained therein, however, they may also modify the 'spacers' and the order of the blocks.

Communication

Communication allows for end points (stakeholders) to initiate modifications and authenticate connections in the event of a ‘misbehavior’.

Adaptations between multiple end-points and stakeholders requires a method by which these adaptations may be coordinated. This is one of the primary roles for the Communication structures included in MUP.

Transmission Chains

Transmission Chains (TC) may be used to build, reset, or adapt existing pads. In a distributed environment, TC allows for the joint enactment and synchronization of pads.

Given a limited size pad, TC allows for repeated, secure exchanges that may build a larger pad; each of which accelerates the process of producing a significantly sized pad at multiple endpoints, with common or varying stakeholders. For example, a smaller MUP (say 1 kB) may be initially issued to 2 or more endpoints. TCs can be used to 'grow' the MUP to any desired length (there are no limits), say 1 MB; this can be done efficiently and quickly, with validation during each TC.

Conclusion

All students in information theory and cryptography will study the theory of OTPs and 'perfect secrecy'. It has been delegated to the annals of theory and history; found in good spy movies where the operator has his/her miniature pad. Modern computers haven't utilized this unlimited, 'perfect secrecy' – having found ingenious mathematical algorithms that use a 'finite size key'; that is, until now!

Ingenious technologists have found numerous attacks to break these mathematically based algorithms. MUP implements a 'probabilistic' approach by creatively using pads that have no imposed limit on their size. The use of pad morphing and message diffusion makes MUP practical and reusable, while eliminating many of the attacks that focus on 'brute processing' (side channel attacks) versus brute force (running the numbers).

Join with CORAcsi in securing the global community!

- ✓ Industry and commercial partnerships
- ✓ academic analysis and applications
- ✓ Vehicle 2 vehicle and Vehicle 2 Infrastructure protocols and proposals
- ✓ Internet of Things
- ✓ The Cloud
- ✓ Smart cities

Collaboration, pilots and partnerships are the fastest and most efficient way to properly secure our global marketplace. Care to learn more about partnerships, CORA and MUP? **Just ask** (*use the Contact Form available at* CORAcsi.com*, in the CORAcsi menu*):

Joe Latouf
President

Gerry Simpson
VP of Business Development

References

[1] G.S. Vernam. Cipher Printing Telegraph Systems. American Institute of Electrical Engineers, 1926

[2] D.J. Bernstein. The Salsa20 family of stream ciphers. University of Illinois at Chicago, 2007

[3] S.Babbage, C De Canniere, A Canteaut, C Cid, H Gilbert, T. Johansson, B. Parket, B. Preneel, V. Rijmen, M. Robshaw, The eSTREAM Portfolio. 2008

[4] C.E. Shannon and W. Weaver. The Mathematical Theory of Communications. University of Illinois Press, 1949

[5] D.A. Huffman, A Method for the Construction of Minimum-Redundancy Codes, Proceedings of the I.R.E., 1952

[6] J. Duda, Asymmetric Numeral Systems as Accurate Replacement for Huffman Coding, Warszawa, 2016