Secure Data Communication via Lingual Transformation

Jeffrey J. Johnson
Utah State University
Jeffrey.Johnson@usu.edu

Robert F. Houghton
Idaho State University
Hougrobe@isu.edu

Thomas S.E. Hilton
University of Wisconsin – Eau Clare
HILTONTS@uwec.edu

Ivan Cheah
Utah State University
Ivancheahkf@gmail.com

Abstract

This paper proposes a new form of data communication that is similar to slang in human language. Using the context of the conversation instead of an encryption key, nodes in a network develop a unique alternative language to disguise the real meaning of the communication between them. Implementation of such a system, and its potential benefits and challenges are discussed.

1. Introduction

For all practical purposes, modern cryptography can be described as mathematics-based. In contrast this paper proposes a language-based cryptography and discusses how language-based cryptography might be implemented, reasons why it might be advantageous, and some challenges to practical implementation. After a brief background review, the proposed idea will be described, followed by some notes about practical implementation. Some benefits, challenges, and potential future uses and enhancements also will be presented.

2. Background

Recently newspaper and trade press articles have appeared with great frequency and quantity implying to the general public that current security policies and encryption methods may be insufficient to protect data communication. The need to disguise communication, particularly in this digital age, is well established and well documented.

While math-based encryption generally has proven successful in widespread application, at least three eventualities credibly challenge that success. First, traditional threats, such as brute force and replay attacks, are not guaranteed to fail. Indeed, in the case of brute force attacks, they are guaranteed eventually to succeed. The probability of their success in the short term is low, but it is greater than zero, and over time increases to 1. The answer to this threat has been to use larger encryption keys. Also, as computing speed has increased in accordance with Moore's Law; and because our preferred response seems to have been to use essentially the same technology only with larger numbers, we now are in a kind of arms race, hoping ever-larger encryption keys will withstand ever-faster computing power. Is this strategy sustainable? [1]

Second, in all keyed systems the secret is secure only so long as the key is secure. Therefore, even the strongest keyed encryption might be vulnerable to an unsophisticated phishing attack, for example.

Third, quantum computing, once considered to be beyond the horizon, now appears to be forthcoming [2]. Machines that can do all the steps of a complex calculation at once have the potential to render our current math-based encryption schemes obsolete.

2.1. Evolution of Modern Encryption

The Oxford dictionary defines encryption as "the process of converting information or data into a code, especially to prevent unauthorized access." [22] In this paper we more specifically define encryption as the *reversible* algorithmic process of scrambling clear-text into an unreadable stream. One of the early documented uses of encryption is the Caesar cipher, named after the Roman general Julius Caesar who was documented to use this method to communicate with his subordinates. A Caesar cipher is a type of substitution of characters with a shift of letters in a known position [3]. An example of this would be to shift each letter +3 positions in the alphabet so any letters "A" in the original message would become the letter "D" in the encoded version. This is among the oldest known and documented forms of encryption [3].

As communication methods between humans evolved so have the means of encryption. Electronic communication began in the form of the telegraph. Wires were overtaken by radio waves and soon anybody with a quartz crystal could receive broadcast communications. Encryption was needed, especially during wartime, to keep secrets from being directly overheard by the enemy. Perhaps the most famous of

URI: http://hdl.handle.net/10125/50574
ISBN: 978-0-9981331-1-9

wartime encryption was the Germans' Enigma machine used in World War II [4]. This machine used a changing number of rotors that could be set to transpose the alpha characters. Each machine was capable of 17,576 combinations. This machine provided the supposed secure communication that the Axis powers needed to coordinate their methods. However, thanks to Alan Turing and associates, the Allied powers were able to break this code and thereby gain knowledge of the Axis powers' communications. Another famous wartime example is the Navajo code talkers of World War II. [24]

The Data Encryption Standard or DES is a 56-bit encryption standard developed by IBM in 1977 [5] at the behest of the National Bureau of Standards and was used primarily by the United States Government throughout the 1980s and into the 1990s. As computers became more powerful and able to conduct exponentially greater numbers of operations per second, DES was relegated to obsolete status because it was too easily and quickly broken. By 2000, the Advanced Encryption Standard (AES) had supplanted DES as the de facto standard of encryption [6].

As data communication evolved from wired to wireless, encryption again was needed to help keep this process safe from eavesdropping. The 802.11 standard included encryption [7]. This encryption protocol called Wired Equivalent Privacy (WEP) was a method of scrambling the communication using a 40-bit key. As Arbaugh et al [7] pointed out, this method was quickly defeated and new, stronger standards were needed.

One fundamental difficulty of these and other "symmetric" encryption methods is the needed secrecy of the key. Somehow, both sides of the communication need to have possession of the key while keeping it secret from all other parties.

"Asymmetric" or "public private key encryption" has become the de facto solution for most modern encryption schemes. Public key or asymmetric key encryption was in development in the UK as early as 1973 but finally declassified in 1997 [8]. Asymmetric encryption uses two keys: a publicly available one to encrypt and a different, privately held one to decrypt.

New security measures, based on biometric tokens and other non-mathematical substitution schemes are now coming into practice. An example pertinent to this paper is the "Captcha" program, which takes advantage of something humans can do well, but computers do poorly – recognize characters in a distorted picture. This paper proposes a new process that we call Lingual Transformation-based encryption, or simply Lingual Transformation, which would apply a similar concept, protecting data via schemes at which computers are not inherently proficient. It would disguise communication by evolving the language used.

2.2. Human language evolves over time

Even without computers, humans often disguise their communication by introducing changes to their language, as in slang. Also, languages evolve over time. Generally, the changes that occur include spelling and pronunciation, the meaning of words and/or phrases, and additions/deletions in the lexicon. [9] Examples of evolutionary spelling changes can be seen in American versus British spellings of words like color (colour), and catalog (catalogue). Pronunciation differences can be observed between one time and another, as well as in regional differences. For example, the word "leisure" may rhyme with "treasure" or with "seizure", often depending on the age of the person saying it. Lexical changes occur as new words come into widespread use (megabyte, snowboard, google) and old words fall into disuse (deliciate, aerodrome).

Another way in which language changes is by development of slang. The difference in slang is that it often assigns different meanings to existing words. Although slang may have been used much earlier, "What we mean by English slang today didn't really start until the 16th or 17th century in England. It developed out of what was then called the 'thieves cant', or the jargon developed by criminals. It's estimated that perhaps only 10,000 people out of the 4 million English speakers spoke the Thieves Cant, and its purpose originally was the same as all jargon – to be able to speak to each other without others knowing what they were saying" [10].

In modern slang the meanings assigned to slang words often depend on the context in which they are spoken. In reference to a desired drink on a summer day, the word "cool" has a completely different meaning than when used in reference to a person who remains calm under stress. Therefore, given a specific context, the words and phrases employed in a conversation may be understood differently, according to that context. Thus, "Skier slang is different from surfer slang, which is different from any other subgroups' slang, and the only way to know it, is to be a part of that group" [11].

2.3. The Problem

One major problem with modern cryptography is that keyed encryption remains vulnerable to different kinds of traditional attacks. Much of the threat against encrypted communication can be classified into three types [12]:

1. Man in the middle
2. Brute force
3. Replay attack

A man in the middle attack has the communication routed through a computer that is controlled by the attacker. The attacker copies the entire communication and then, based upon the captured key, may pretend to be the person at either end of the communication. This allows the attacker to access or change the data. This attack can be difficult to detect because after the attacker obtains the desired data, the message can be forwarded to the intended recipient, who presumably will have no clue the communication was intercepted.

A brute force attack is when the attacker makes an unending attempt to guess the password or key needed to pretend to be the user. Given enough time and unlimited attempts, brute force will always be successful against math-based cryptography.

A replay attack is when the attacker listens and records the communication stream. They then take this recorded stream and replay one side of it against the other side. This can lead to the attacked system replying with the missing key. This attack also may be difficult to detect because all aspects of the communication are copied from the original source, and therefore appear to be legitimate.

In addition to these known attack vectors, another and ultimately more likely possibility is discovering the files or databases where secret keys are kept [12]. By password cracking or other means, the attacker obtains access to the files containing encryption keys and then has the ability to compromise security for all communication which relies on any of those keys. Further, theoretically random key generation in real implementation often is pseudo-random, which means potentially predictable [13].

In recent history, a persistent challenge in mathbased encryption is represented by a corollary to Moore's Law, which states that computing power (or calculations per unit of time) will double every 18 months or so. This presents a problem to mathematically "unsolvable" decryption equations. As numbers of transistors in computer chips increase the time required to solve these mathematical equations decreases. Solving the equation is brute forcing the decryption of the data. Additionally, there are time-saving factors such as rainbow tables to pre-generate solutions to the mathematical equations to inject into encrypted data tables [14]. Such methods can exponentially speed illicit access to encrypted data streams. In this vein, the apparently imminent advent of quantum computers [15] may pose a serious threat to all math-based encryption.

Much current encryption depends on a key and proper key exchange facilitated by public key verification authorities [16]. Unfortunately, consolidated key verification also constitutes a single point of failure. If a user's private key is decoded or stolen, any entity can claim to be that key holder. In the case of a key verification authority being compromised, the attacker then has access to all the root certificates. An example of this is the DigiNotar hack of 2011 [17].

A question arises then, about whether keyed, math-based encryption really is the best we can do? Perhaps humans' intelligent manipulation of language might serve as a different model for secure data communication. The following section explores ideas about mimicking human slang in data communication.

3. Theoretical Propositions

Slang consists, essentially, of substituting a "wrong" word in place of a right word, while conveying the right meaning. Drawing upon context, speakers and listeners can infer the right meaning in spite of saying or hearing the wrong word. Consider the following short conversation:

Q: How did you like the movie? A:

Oh, it was good.

The answer, "it was good" would be understood by anyone steeped in American culture if the respondent had used any of the following words instead of "good". Note that none of the words' original meaning corresponds to "good".

Swell	Hip
Hot	Bad
Cool	Groovy
Radical	Sick

Although at times slang has been considered vulgar and low, historically it seems to have been widely known and used [19]. With the goal of instilling the ability to create and use slang into computers, one might first attempt to model the way humans create and use slang. Extrapolating from the conversation above, a basic model would depict two participants, A and B, exchanging words via a communication medium (Figure 1).

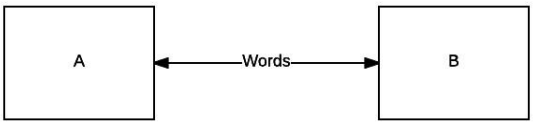

Figure 1.

Such a simplistic model clearly fails to capture the purpose of communication. A better model is shown in Figure 2.

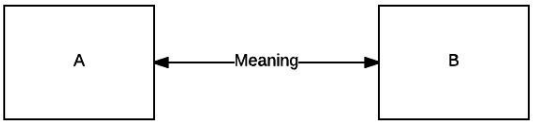

Figure 2.

In Figure 2, A and B exchange meaning. Words constitute the medium by which they do so ("words" in this context are defined loosely to include any representation of meaning, whether by a string of characters or other representation.) Thus, A and B may choose from among many words in their respective vocabularies to convey a particular meaning (each arrow represents a possible word choice; the longer arrows represent the chosen words in Figure 3).

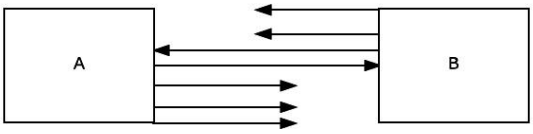

Figure 3.

In the models presented so far, the presence of a third entity, C, must be assumed as an illicit eavesdropper. As long as A and B use standard, accepted vocabulary to convey their meaning, C will have no problem understanding the same meaning.
Under the model in Figure 3, slang can be defined simply as A and B agreeing on a new pathway (word) to convey a specific meaning, particularly if agreement can be reached without C's participation. The only way A and B can reach such agreement is by reference to some context of which A and B are, but C is not, a part.

Because use of slang is both a part of everyday language and a fact of history, arriving at the following is easy:

Axiom 1: Humans, communicating within a specific context, can create and use slang words and/or phrases to convey meaning in a way that is, at least temporarily, unintelligible to others who are not participants in that context.

The slang words may be new creations, invented specifically to convey meaning (e.g., "nerd"), or they may be existing words that are assigned new meaning (e.g., "swell"). In either case, no additional effort is required to disguise the communication – it can be spoken aloud in mixed company. Further, no restrictions exist on which words or phrases may be chosen, as long as A and B can agree on the assigned meaning. Thus, for example, even the word "bad" can be used as a term to express something desirable.

While the models depicted in Figures 1-3 apply to human communication, they may also apply to computer communication. Viewing computer communication as an extension of human communication, logically:

Proposition 1: Like humans, computing machines could achieve the same benefits communicating by slang created from their own context.

Over time, however, the secretive purpose can be frustrated as slang words come into more widespread use and eventually are accepted into common language. The remedy, of course, is to update the slang lexicon frequently, thus staying ahead of eager but uninitiated would-be users. Thus:

Axiom 2: Because slang is not guaranteed to remain arcane for long, human slang continuously evolves to meet the requirements of the users.

Computers mimicking human slang use would face a similar challenge that any message "encrypted" by slang would soon be discovered as the slang words' meaning would eventually become known. Therefore, the vocabulary used by communicating machines similarly must evolve. In human communication, language evolution takes time, and although evolution occurs, sometimes the meaning can still be understood as if the language had not changed. For example, Shakespearean language is still intelligible today despite obvious differences from modern English.

In contrast, however, slang may evolve for expressly clandestine purposes. One requirement for successfully achieving such purposes is changing fast enough to keep ahead of unwanted users who try to

decipher the evolving slang. Of course, speed is one of the great strengths of modern computers. Therefore:

Proposition 2: Computers can "compress" the time required for slang to evolve, thus rapidly and continuously changing their shared slang and consequently keeping outsiders – even fast ones – out.

The value humans derive from creating and using slang lies in the ability to authenticate speakers as members of an exclusive group and/or to maintain a level of secrecy conversing with group members while in the presence of non-members. The goals of modern keyed encryption are almost identical. If computers can create their own slang, and if they can do it quickly enough to keep eavesdroppers flummoxed, then

Proposition 3: Computer-generated slang can meet the same goals as modern encryption techniques.

Of course, proving or disproving theoretical propositions such as these would be more than a simple matter of practical implementation. Extensive testing and real-world experience over time would also be necessary. However, the process must begin somewhere. The following section describes one way it might be accomplished.

4. Implementation

As described above, language evolves over time in several ways. Natural evolution seems to occur along at least three dimensions [18]: the meaning of words, spelling and pronunciation, and additions and deletions in the lexicon. Purposely manipulating language to render it arcane could be done in the same three ways and also in other ways. Word substitutions, based on any number of schemes from rhyming, to homonyms, to synonyms; spelling variations, based on phoneme disassembly/reassembly; and rearrangement of grammar are a few possibilities among a potentially very wide range of schemes [18]. Indeed, the substitutions need not follow any externally discernible method at all, as long as the communicating parties can agree on them. Choosing and combining such schemes in an unpredictable mix would create a unique "slang" that could only be understood by the entities that participated in making it.

One crucial element for development of slang among groups of humans is context. In order to create a type of slang that could be used between two data communication devices, some type of common context must be established. The context must be unique and exclusive such that only the entities involved in the conversation are privy to the context. This makes an interesting challenge because of the necessary assumption that an eavesdropper is always listening.

Some manipulations that could be employed to "evolve" a usable slang might include the same types of methods employed by humans in creating slang, like simply substituting context elements, or rhyming words, synonyms, purposeful misspellings, homonyms, or disassembling/reassembling phonemes. Of course, additional methods beyond those modeled by generations of humans, would be limited only by our ability to imagine and implement them. The evolvable context of a flow of data between two devices could include any protocol-related exchanges, packet history, and/or some agreed-upon and mutually external elements.

4.1. The All-Important Context

The common context is the critical component of the system for several reasons. First, drawing from a common context eliminates the need to exchange an encryption key. The context gives the communicating nodes a common "pool" from which they can draw inferences about the intended meaning of "wrong" words sent and received to convey "right" meaning. This is important, for example, because traditional stream ciphers suffer from a potentially debilitating weakness: the key (called the seed) from which their keystream originates, supposed to be entirely random, in practice is often generated from internal computer states that are pseudorandom, i.e., potentially predictable. If a seed can be identified by an adversary, the adversary can decrypt any encrypted messages resulting from that seed [13]. An example of this problem is the untimely obsolescence of WEP encryption in early Wi-Fi implementations [21]. Drawing on a unique, common context eliminates the need for an encryption key.

The context facilitates another purpose beyond the being the basis for "encryption". Because it is created exclusively by two communicating nodes, and is unique to just those two nodes, their conversation, and specific elements thereof (time, place, etc.), no other entity can correctly apply it. Therefore, proper application of the unique context serves as a means of authentication of the nodes involved.

4.2. The Process

A process to create and use an exclusive slang between two nodes in a network would require the following steps:

1. Given a need to communicate securely a message between two nodes in a communication network, begin by exchanging non-secure, inconsequential flows between the two nodes (ICMP commands, for example), keeping a log as the exchanges continue. The log may include any and all of the flow(s) exchanged between the two nodes, including protocol details (packet headings, time stamps, etc.) as well as message content.
2. Using the log as a reference, agree on methods, possibly similar to the natural evolution of human slang, to effect substitutions for the individual words (or other subdivisions of the secure message).
3. Apply the agreed-upon methods, transforming the original message into a series of words (or other subdivisions of the message) which, despite being transmitted in clear text, appear to constitute nonsense, or content in which an eavesdropper is not interested.
4. Upon receipt, reverse the agreed-upon method to obtain the original message.

Now apply these four steps to the process between every pair of communicating nodes in a network. As message content traverses a network, every node-node pair along the way would repeat the process, each time creating a new and unique context and using a different transformation method. While the "encryption" achieved at this point may appear to be a simple substitution cypher, it is important to note that no encryption key has been generated or exchanged. Further, every node-node pair along the communication pathway is using a different and unique "language." Still, the meaning of the original message would be understood by each node.
Therefore, the two terminal nodes (the original sender and the ultimate receiver), going through the chain of node-node pairs, must create their own unique context and choose their own transformation methods.

Viewed from another perspective, Lingual transformation based encryption can be understood in the following way:

Given a network consisting of several nodes, A, B, C, … Z, and assuming that when node A has a message for node Z, the message must be communicated through the intermediate nodes, B, C, etc., node A must communicate first with node B. Node B then communicates with node C in order to pass the message along on its path toward node Z. Node C then communicates with node D, and so on until the message arrives at node Z.

With lingual transformation, nodes A and B first create their own unique language or slang, using the context of their conversation that is known only to A and B. We can refer to this language as AB language. Then, node B similarly establishes another, different new language with node C: BC language. Node C does similarly with node D, and so on until all of the node-node pairs from A to Z have their own unique language. Finally, when the full chain of unique-language-speaking node-node pairs has been established, node A communicates with node Z via the chain and creates yet another unique language, established between node A and node Z (AZ language) in the same manner as described for all the node-node pairs (see figure 4). After all these languages have been established, nodes A and Z can begin exchanging secret information. At this point, node A translates its secret message from human language to AZ language, then translates the result into AB language and sends it to node B. Lacking context (AZ context), Node B will not understand the message, but will translate it to BC language and send it to node C. Node C similarly will not understand the AZ-language message, but will translate it to CD language, and so forth.

Throughout the duration of the conversation between node A and node Z, every node-node pair continuously modifies its language by reference to its continuously-changing common context. Nodes A and Z also continuously evolve their own unique language in the same way.

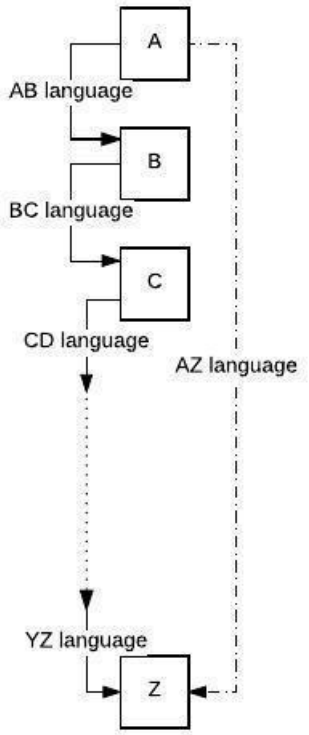

Figure 4.

5. Benefits of Lingual Transformation Based Encryption

This new method of computer communication functions as an encryption scheme as well as a potential one-time pad. This will reduce the risk of the three vectors of attack mentioned earlier: Man-in-the middle, brute force, and replay attack. It also eliminates the danger of stolen keys (because there are no keys to steal).

A man-in-the-middle attack is only effective if the attacker can copy the data and extract the cleartext. The only conversation that the attacker can capture is between only two nodes in the chain. These two nodes (pick any communicating pair) are only speaking their version of slang. The actual conversation between nodes A and Z appears as nonsense to C and D, for example, because it is conducted in a language (AZ language) they don't know. However, C and D translate and pass on the "nonsense" words into their language (CD language) which when "decrypted" still would be in a language unknown to anyone except A and Z.

The use of Brute Force would be irrelevant as there is no password or key to brute force out of the conversation. Agreeing on a transformation method is not equivalent to sharing a key, just as agreeing to speak French is not equivalent to giving a specific password. Even if one element of the conversation could be deciphered, the other elements would not be compromised because each is "encrypted" differently. A replay attack would be foiled by the fact that all of the languages employed are continuously evolved, effectively creating a one-time pad. An attacker would gain nothing by replaying a recorded conversation as the conversation is either dropped or evolved beyond the limit of the replayed recording.

Of course, other threats also exist that are not solvable by encryption because they involve attacking static data, human users, or operating systems. Since social engineering involves attacks on the human user and not on the data, encryption of any kind is irrelevant against it. A port scan or vulnerability attack is a server side attack and not a conversation attack therefore this proposed encryption method will have little effect on reducing any threat via port scan. Denial of service attack also does not have anything to do with encryption and therefore any method of encryption will not reduce this attack vector. Any attack on the physical computer is also not defensible by communication-based encryption [12].

The scheme described in this paper successfully would improve the defense against the methods of attack which target weaknesses in encryption. These results would help any type of computer communication be more effective in keeping the secrets secure.

6. Challenges

Two major challenges immediately become apparent with lingual transformation-based encryption. The first is the time necessary to establish context in order to morph computer language into an unrecognizable quasi-slang. In addition to the original establishment of the context, the need for continual updates will also prove time-consuming. Along with the context, the actual lexicon of the unique language also must evolve. Of course, the speed at which computers operate will greatly compress the time needed to evolve language compared to humans; nonetheless there will be a time lag. Bandwidth is a related concern due to the amount of overhead data exchange required to establish unique slangs before secrets can be shared.

The second challenge lies in the actual methods for morphing language. In addition to contributing to the time problem, the potential complexity of creating a new language on the fly will likely be challenging. While historically, language evolution takes time, slang develops more quickly by substituting new words or changing the meaning of existing words. The easiest substitutions would probably come directly from the context, but other substitution might be more secure. A system might instead use context elements as "menu choices" for other methods of substitution. Some possible methods are listed:

- Rhyming words:
- Synonyms
- Antonyms
- Homonyms
- Cockney Rhyming Slang
- Phoneme deconstruction/reconstruction
- Language translation
- Quasi-random replacement

The greater number of methods used, the more robust lingual transformation based encryption will become. Slang is apparently a universal human language concept. [23] We envision a constant and ongoing quest for new methods. The methods above focus only on changing the vocabulary and no other characteristic of language. Other possible areas to search for manipulation methods might include: spoken language (including voice pitch, speed, and pronunciation), purposeful and meaningful misspellings, and grammar (including sentence structure).

7. Future Uses

As this technology emerges and evolves, ideas about future uses include, but certainly are not limited by, the following.

- Morphing data within a single computer. Using the internal I/O of the computer, this method could be built into the controllers themselves to encrypt the data stored on the hard drive.
- Creating an ad-hoc network in a company's DMZ to force an enhanced language morphed network communication, thus ensuring that language morphing is occurring. This technique would help keep the technology future-proof. By adding nodes to the system the morphed language gets more complicated and morphed. Adding system after system would force subsequent attacks to keep up with similar resources.
- Keeping records of previously created morphed languages to use as digital fingerprints. Each node to node communication could be logged and stored. Once logged computers could identify themselves based upon that previous, and unique communication. This could speed up future communications with same devices.
- As quantum computers are created and brought to market, this method, due to its lack of math, can still be used to protect secret messages. Quantum computers create serious problems for math-based encryption. Lingual transformation-based encryption uses the communication as the basis for morphing rather than an exchanged key or known algorithm.

8. Summary

In this paper we have put forward the theoretical proposition that the benefits of human slang use could be effectively realized in data communication by applying similar principles. Humans routinely manipulate standard language to achieve purposes like authentication and secure communication. Computers might just as well emulate that behavior. Some important features of a practical implementation of this idea are:

- Keyless encryption, eliminating the need for safe storage and exchange of encryption keys.
- Continuous evolution, essentially creating a one-time pad.
- Non-math logic, allaying concern about the threat of quantum key cracking.

Some additional benefits include:

- Greatly increased effort required for attackers to compromise large numbers of records (because the language continually morphs, a successful breach would yield only a small amount of data).
- Automatic authentication and compromise detection (because communication is based only on the exact common context of the two nodes involved – the very presence of a third party alters the context, and therefore is detectable).

- A new method that diverges from the "arms race" (bigger and bigger key sizes) way of protecting data communication.

Of course, a healthy skepticism is appropriate when an unproven idea is advanced. Naturally many questions about proposed benefits remain until a practical implementation can provide answers. The following list is representative of some objections that might be raised:

- This is no better than a simple substitution cypher.
- A trained and/or experienced linguist could easily crack the code.
- An eavesdropper at the terminal node would know everything the terminal node knows.

Addressing such concerns is more than a matter of debate. These and other concerns will require thorough proving through rigorous testing (which is underway, but without reportable results as of this writing). However, with careful consideration of this list we observe:

- Ultimately all encryption is substitution. However, this idea is different from other methods because
    - there is no key exchange; ○ while the terminal nodes participate in lingual transformations with their immediate neighbors, they remain unaware of the other transformations between them;
    - the transformations are performed continually, so the encryption evolves as the conversation continues. If an eavesdropper could decipher the encryption, he would almost immediately have to do it again;
    - the communication between nodes, for example in BC language, CD language, DE language, etc. consists of translations from AZ language. An eavesdropper who deciphers one of the languages would still have to decipher AZ language (then return to the 3rd point, above)
- The linguist would have to be quick, and his/her work continuous. In any case, if the language were deciphered, the breach would consist of one communication unit (word, packet, record), not tens of millions as in contemporary breaches. Additionally, as the data entering any given node has already been substituted with a different language, the original data can only be translated at the terminal nodes.
- The presence of an eavesdropper at a terminal node indicates a lapse in all types of security. Any security/encryption method will eventually be breached if an attacker has access to the terminal data node.

9. Conclusion

There will always be a need for secret communication to be kept from prying eyes and ears. Harking back to Julius Caesar, the need to keep secrets has always been at the forefront of military, government, corporate and other private operations. Because opposing entities will always attempt to discover important secrets, we always will have a need to create new and better ways of keeping our communication safe. Language-morphing techniques will contribute to the evolution of keeping secrets.

10. References

[1] J. J. Johnson, R. F. Houghton, and T.S.E. Hilton, "A Completely Different Concept: Non-Mathematical Encryption?" 56th Annual Conference of International Association for Computer Information Systems (IACIS). Nashville, TN October 2016 pp 19-20.

[2] IBM Newsroom https://www03.ibm.com/press/us/en/pressrelease/51740.wss 6 Mar. 2017.

[3] Dey, S. "SD-AREE: A New Modified Caesar Cipher Cryptographic Method Along with Bit-Manipulation to Exclude Repetition from a Message to be Encrypted", (2012), arXiv preprint arXiv:1205.4279

[4] Gold, J. I., & Shadlen, M. N., "Banburismus and the brain: decoding the relationship between sensory stimuli, decisions, and reward". Neuron, 2002, 36(2), 299-308. [5] R. Davis, "The Data Encryption Standard in Perspective", Communications Society Magazine, IEEE, Nov. 1978, pp. 5-9.

[6] J. Daemen, V. Rijmen, "The design of Rijndael: AESthe advanced encryption standard", Springer, Berlin 2002.

[7] W. A. Arbaugh, N. Shankar, Y.J. Wan, and K. Zhang, "Your 80211 Wireless Network Has No Clothes", Wireless Communications, IEEE, 9(6), 2002, pp. 44-51.

[8] T. Espiner, "GCHQ Pioneers on Birth of Public Key Crypto" ZDNet.com October 26, 2010 http://www.zdnet.com/article/gchq-pioneers-on-birth-ofpublic-key-crypto/

[9] M. Liberman, "Linguistics 001: Introduction to Linguistics (fall 2016 Lecture 9 Morphology" http://www.ling.upenn.edu/courses/ling001/morphology.html

[10] K. Pearson, "Word History: Where Does Slang Come From?", http://www.primarysources.com/blog/2010/12/01/word-history-where-doesslang-come-from/, 2010.

[11] A. Bushong, "Slang: How invented Words Become Part of Our Language", http://www.bridgeenglish.com/slang-how-invented-wordsbecome-part-of-our-language/, 2013.

[12] Verizon "2016 Data Breach Investigations Report" http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf

[13] P. Ekdahl & T. Johansson. "A new version of the stream cipher SNOW. In Selected Areas in Cryptography" (pp. 47-61). Springer Berlin Heidelberg 2002.

[14] P. Oechslin, Making a faster cryptanalytic timememory trade-off. In Annual International Cryptology Conference (pp. 617-630). Springer, Berlin, Heidelberg, 2003.

[15] Bernstein,D.J., J. Buchmann & E. Dahmen (Eds.). Post-quantum cryptography. Springer Science & Business Media 2009.

[16] Adams, C. and S. Lloyd, Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations, New Riders Publishing, Indianapolis IN, 1999.

[17] Keizer, G. "Hackers may have stolen over 200 SSL certificates" Computerworld, 2011. http://www.computerworlduk.com/it-vendors/hackers-mayhave-stolen-over-200-ssl-certificates-3300651/

[18] M. Liberman "Linguistics 001 Lecture 22 Language Change" https://www.ling.upenn.edu/courses/Fall_2003/ling001/language_change.html

[19] J. C. Hotten, The Slang Dictionary; Or, The Vulgar Words, Street Phrases and 'Fast' Expressions of High and Low Society, Hotten, London, 1870.

[20] P. Ekdahl, T. Johansson , "A new version of the stream cipher SNOW" In Selected Areas in Cryptography (pp. 47-61). Springer Berlin Heidelberg. August 2002.

[21] A. H. Lashkari, M. Mansoor, A. S. Danesh, (2009, "Wired equivalent privacy (wep) versus wi-fi protected access (wpa)" 2009 International Conference on Signal Processing Systems (pp. 445-449) IEEE May 2009.

[22] "encryption, n.1." *OED Online*. Oxford University Press, June 2017. Web. 22 August 2017.

[23] Internaţională, S. Ş. Universitatea „1 Decembrie 1918" Din Alba Iulia Facultatea De Istorie Şi Filologie Departamentul De Filologie.

[24] Aaseng, N. (2009). *Navajo code talkers*. Bloomsbury Publishing USA.