Cryptography of Things

Cryptography Designed for Low Power, Low Maintenance Nodes in the Internet of Things

Gideon Samid
Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, Ohio
BitMint, LLC
Gideon@BitMint.com

Abstract Proposing a cryptographic premise where intensive computation is avoided, and security is achieved via non-complex processing of at-will size keys. Memory is cheap, power is expensive throughout the Internet of Things; maintenance may be an issue, and the risk is not less than for the Internet of People. Cryptography must adjust itself: first in its approach and philosophy, then in practical ciphers. The proposed philosophy is to increase the role of randomness, and to build ciphers that can handle any size key without choking on computation. Orthodox cryptography seeks to create a thorough mix between key bits and message bits, resulting in heavy-duty computation. We propose simple, fast ciphers that allow their user to adjust the security of the ciphertext by determining how much randomness to use. We present “Walk in the Park” cipher where the “walk” may be described through the series of visited spots (the plaintext), or, equivalently through a list of the traversed walkways (ciphertext), the “park” being the key, the size of which determines security, but does not affect nominal computation load. We describe a use scenario for the proposed cipher: a drone taking videos of variable sensitivity and hence variable required security – handled by the size of the “park”.

Keywords—low-power encryption, randomness, Trans-Vernam Cipher, User-Controlled Security.

I. Introduction

The Internet of Things is comprised of a large number of nodes where mainstay cryptography is overly stressed. Computational power may be limited, data speeds are too demanding, maintenance of remote nodes is a major constraint. We propose a conceptual solution based on the idea that algorithmic complexity may be replaced by large size keys.

We discuss (i) the application environment, and (ii) the principles of the proposed solutions.

A. Application Environment

Our human environment is about to be inundated with sensory nodes, transducers, controllers, invading our living space as they monitor, report, guide and control our modern living space. Swarms of camera fitted drones, and “wall climbers” will watch over us, and help us get out of trouble, as well as exploit opportunities. The communication framework for all this new crop of communicating devices is none other than the Internet of Things. And hence, much as the Internet of People exposed human participants to malware, fraud and abuse, so it will surely happen for the IOT. And much as encryption is the only principled and effective defense for human security, so is the case for security of things.

The IOT connects remote network nodes, which are not easy to access in a physical way. Some may be exposed to the sun, and may generate modest amount of energy on their own, others are not so exposed, and need to be fitted with batteries, which are hard to replace. At either case, there is a strong limit on processing power. Some IOT nodes are tiny flying devices, they cannot 'carry' heavy batteries.

The IOT poses the challenge of saving on computational energy. Alas, modern cryptography is based on algorithmic complexity, which is effected through computational complexity, which in turn drains the device battery. So we need to rethink cryptography with an eye towards computational economy.

Speed: some IOT applications may involve high-speed communication. High speed, high-resolution cameras fitted on flying drones may be required to transmit to an operational center, to serve an important rescue operation, or other social task. Similarly, an isolated device somewhere may be activated with a large stream of commands, most of them should be further transferred to devices down the road. All in all, the IOT may need to accommodate high volume, high speed information exchange. The existing popular ciphers slow down that flow rate, and are not friendly to this requirement. Admittedly, stream ciphers, which are very popular with military applications, are much faster than the standard block ciphers, but they suffer from the vulnerability of synchronization. Losing a couple of bits will create havoc in the entire stream in the downstream.

Maintenance: Quite a few IOT nodes will be placed in hard to access locations, and no physical maintenance will be feasible. Hence the use of any specific cipher, which at any moment may be mathematically breached, is a risky practice. This applies to all algorithmic complexity ciphers. As Prof. Nigel Smith articulates in his book “Cryptography (an Introduction)”: “At some point in the future we should expect our system to become broken, either through an improvement in computing power or an algorithmic breakthrough.” Normally the IOT gravitates towards very few ciphers considered 'secure'. If one of them is suddenly breached (e.g. GSM communication cipher), then all the IOT nodes which rely on it, have lost their security, and physical attention is not practical.

Magnetic Vulnerability: Many IOT nodes are placed in very harsh environment, and are subject to lightening violence, as well as man made electromagnetic impacts. Software based cipher may be at greater risk.

In summary, IOT nodes are vulnerable both to malicious attack, and to environmental punishment. These vulnerabilities may be remedied to a large extent if we come up with a new cryptographic approach: Cryptography Of Things (COT).

B. Principles of the Proposed Solution

Modern cryptography erects security around data using two parameters: (i) algorithmic complexity, and (ii) randomness. It's generally believed that the more complex an algorithm the more secure the ciphertext, and also the more randomness that is being used (the larger the key), the more secure the ciphertext. Randomness is in a way dull, and of no much interest mathematically (except of course with respect to its definition and to metrics of quality). By contrast, algorithmic complexity is an exciting math dilemma. Academic cryptographers are attracted to this challenge and develop new and newer complex algorithms. Unfortunately in today's state of affairs, we only manage to compare complexities one to the other, not to ascertain their level in an objective mathematical way. And even if it turns out that $P \neq NP$ as most complexity researchers believe, in cryptography complexity is used in combination with randomness, hence one is using a random key selected from a large key space. What is hard to know is how many specific keys when applied with specific plaintexts, offer some mathematical vulnerability, leading to effective extraction of the message. In other words, the de facto complexity, or security of algorithms cannot be ascertained. Worried about this, we come up with increasingly complex algorithms, which require more and more computational effort. They in turn require more and more power -- which many IOT nodes simply don't have.

Randomness, on the other hand is passive memory, and even the smallest and most unsophisticated devices can be fitted with gigabytes of memory, serving as key. These realities lead one to aim to develop cryptography where the role of reliable, passive, manageable, secure randomness is enhanced, while the role of doubtful complex algorithms that are power hogs, is decreased.

This thinking brings to mind the famous Vernam cipher: the algorithm could not have been simpler, and the key could easily be as large as hundreds of gigabytes -- memory is both cheap and light. It may be stored without requiring power. Too bad that Vernam is so impractical to use. Yet, can we re-analyze Vernam as a source of inspiration for security through more randomness and less algorithmic complexity?

Let's envision a Vernam Inspired Cipher (VIC) where at any stage the user can 'throw in a few more key bits' and by that achieve a large increase of cryptanalytic burden, together with a modest increase of nominal processing burden (encryption, and decryption). Let us further demand from the VIC the Vernam property of achieving mathematical secrecy at the minimum key size required by Shannon's proof of perfect secrecy.

To better analyze this vision let's regard any cryptographic key, k, as the natural number represented by binary interpretation of its bit sequence. Accordingly, the Vernam key space associated with n-bits long messages, will be: $1,2,....2^{n+1}-1$ corresponding to $\{00....0\}_n$ to $\{11....1\}_n$. We may further agree that any natural number $N > 2^{n+1}-1$ will be hashed to an n-bits size string. Once we agree on the hashing procedure we have managed to recast Vernam cipher as a cipher that accepts any positive integer as a key, with which to encrypt any message m comprised of n bits to a corresponding ciphertext. We regard this as natural number key representation (NNKR).

We can similarly recast any cipher according to NNKR. We consider a cipher for which the series $n_1, n_2,.....n_{max}$ represents the allowable bit counts for the keys. E.g for DES the series has one member $n_1=n_{max}=56$; for AES the series contains three members: $n_1=128$, $n_2=196$, $n_3=n_{max}=256$. For a cipher where the key is a prime number then the series is the series of primes. For ciphers defined over every bit string of length n_{max} all the natural numbers from 0 to $2^{n+1}-1$ qualify as a n_{max} key. Larger keys will be hashed to a n_{max} bits long hash. For ciphers where the series $n_1, n_2, ... n_{max}$ represents discrete possible keys, we may agree to hash any natural number to highest member of the list $n_1, n_2,....$ which is lower than that natural number. For all natural numbers smaller than n_1, we will "hash" them to the null key ($|k|=0$), and we may formally agree that the case of k=NULL is the case of no encryption (the ciphertext is simply the plaintext).

With the above definition we have recast all ciphers as accepting every natural number as a key.

The basic idea of security is that the larger the key, the better the security. In other words, we "buy" security, and "pay" for it with a choice of a random number -- the larger the number, the higher the price. Let $s_i(k)$ be the security achieved by a user of cipher i, "investing" key k. The metric s, will reflect the average computational effort required of the cryptanalyst for extracting the message m from a captured ciphertext c, computed over the distribution of $m \in M$, where M is the message space from which m is selected. Let $p_i(k)$ be the average combined processing effort (encryption plus decryption) required of a user of cipher i, while using key, k, over the distribution of message $m \in M$.

For any cipher i, using a natural number k as key, we may define the utility of the cipher at this point as the ratio between the cryptanalytic effort and the nominal processing effort:

$$(1)....... U_i(k) = s_i(k)/p_i(k)$$

We can now define a Vernam Inspired Cipher as one where over some range of natural number k (k_1.....k_2) as key, the utility of the cipher will be somewhat stable:

$$(2)\ldots\ldots\ U_{K1}, U_{K1+1}, \ldots\ldots\ U_{K2} \sim U$$

In that case a user encrypting with k_1 will be able to increase the security he builds around the data, while still using the same cipher, by simply ratcheting up the key from k_1 to k_2. She will then -- again, using the same cipher -- increase its associated security from $s(k_1)$ to the higher value of $s(k_2)$

$$(3)\ldots\ldots\ldots\ S(K_2) = S(K_1) + \Sigma\, (U(K+1) * P(K+1) - U(K) * P(K)) \text{ FOR } K=K_1 \text{ TO } K=K_2$$

which is reduced to:

$$(4)\ldots\ldots\ S(K_2) = S(K_1) + U\, \Sigma\, (P(K+1)-P(K)) \text{ FOR } K=K_1 \text{ TO } K=K_2$$

We may now add the Vernam limit for a large enough key. Recasting cryptographic keys as natural number leads to redefinition of the key space, K, as a subset of the natural numbers from 1 (or formally from zero) to the highest natural number to be considered as a key, k_{max}:

$$(5)\ldots\ldots\ |K| \leq K_{MAX}$$

And hence, for messages comprised of n bits, a key max of value 2^n ($k_{max} = 2^n$) will allow for a cipher where the user could simply ratchet up the integer value used as key, $k' < 2^n$, to the point of achieving mathematical security. We can define a special case of a Vernam Inspired Cipher, as a **Trans Vernam Cipher (TVC),** being a cipher where increase in the integer value used as key will eventually reach "Vernam Security Levels", or say, Shannon's security, for n-bits long messages:

$$(6)\ldots\ldots\ S_{MAX} = S(K_{MAX} = 2^N) = S(K') + \Sigma\, U(K) * (P(K+1)-P(K)) \text{ FOR } K=K' \text{ TO } K=K_{MAX}$$

Existence: It's readily clear that DES, AES and their like will not qualify as Vernam Inspired Ciphers. For DES:

$$(7)\ldots\ldots\ S(K < 2^{56}) = 0$$
$$S(K > 2^{56}) = S(K=2^{56})$$

For AES:

$$(8)\ldots\ldots\ S(K < 2^{128}) = 0$$
$$S(2^{128} < K < 2^{192}) = S(K=2^{128})$$
$$S(2^{192} < K < 2^{256}) = S(K=2^{192})$$
$$S(K > 2^{256}) = S(K=2^{256})$$

A positive example for existence can be shown on the basis of any cipher where the ciphertext, c, is larger than the message, m: $|c| > |m|$. We shall designate such a cipher as EXP (expansion cipher). Let $|k_{exp}|$ be the bit size of the key used by the EXP cipher. Let k be any positive integer larger than $2^{|k}$exp$^{|+1}$-1. We may hash k è k_{exp} then use it to iteratively encrypt k to a larger k'> k, k" > k'... until some $k^{(t)}$ is as large as the required Vernam key. ($k^{(t)} \geq 2^n$) for n-bits long messages. . The larger the key k (closer to 2^n) the closer it is to a perfect Vernam, and the greater its security measured. Even this simple procedure highlights the advantage of 'any size key': the cryptanalyst cannot bound his task, as she does with fixed size keys. This procedure may be used with keys that are a tiny fraction of the Vernam size, or very close to Vernam. At the latter case there will remain non-reducible equivocation regardless of the computational ability of the cryptanalyst.

The security of such a procedure depends of course on the nature of the EXP cipher. But it illustrates the underlying notion of a cipher framework where a user can "buy" more security, by simply investing in more randomness, and thereby buying whichever security one desires, up to perfect mathematical (Shannon) security.

II. "WALK-IN-THE-PARK" CIPHER

We present here a Trans-Vernam Cipher (TVC), that runs by the name *Walk-in-the-Park* because both encryption and decryption is taking place by "walking" – charting a path determined by the message, and then describing it through various entities in the "park" where the walk happens. It is based on the idea that a 'walk' can be described either via the places visited, or via the roads taken. One needs the "park" to convert one description to the other.

The cipher is defined as follows:

We employ a four-letter alphabet: X, Y, Z, and W, expressed via 01,10,11,00 respectively. The key is a table (or matrix) of size *m * 2n* bits, which houses some arrangement of the four alphabet letters (m*n letters in total). We regard every letter as a node of a graph, and regard any two horizontally or vertically contiguous letters as connected with an edge. So every letter marked on the graph has between 2 to 4 edges connecting it to other letters on the graph.

We define a path on the graph as a sequence of marked letters such that any two contiguous letters on the graph are connected via an edge.

Informally, the cipher works by mapping the plaintext into a sequence of X,Y,Z, and W; then using this sequence to mark a pathway on the graph. Given the starting point, it is possible to describe the very same graph via denoting the edges traversed by the pathway. Each node, or vertex on the graph has up to four edges; let's mark them Up, Down, Right, Left: U,D,R,L, and assign the bit combinations 01,10,00,11 respectively to them. The translation of the pathway from a sequence of vertices to a sequence of edges amounts to encrypting the plaintext to the ciphertext. And similarly for the reverse (decryption).

Why is this a Trans Vernam Cipher? Because the graph may be large or small. The larger it is the more security it provides. It may be so large that it will be Vernam equivalent, and it may be so small that brute force will extract it relatively easily. The processing effort is not affected by the size of the graph, only by the length of the pathway, which is the size of the encrypted message. By analogy given a fixed walking speed, it takes the same time to walk, say, 10 miles on a straight stretch of a road, or zig-zagging in a small backyard.

Detailed Procedure:

1. Alphabet Conversion: Map a list of symbols to a three letters alphabet: X, Y, Z. By mapping every symbol to a string of 5 letters from the {X,Y,Z} alphabet, it is possible to map 3^5=243 distinct symbols (a few less than the ASCII list of 256 symbols).

2. **Message conversion**: let m=m_0 be the message to be encrypted, written in the symbols listed in the 243 symbols list. Using the alphabet conversion in (1) map m_0 to m_3 - a sequence of the 3 letters alphabet: X, Y, Z.

3. **DeRepeat the Message:**: enter the letter W between every letter repletion in m_3, and so convert it to m_4. m_4 is a no-repeat sequence of the letters {X,Y,Z,W}. Add the letter W as the starting letter.

4. **Construct a key**: construct a n*m matrix with the letters {X,Y,Z,W}. The matrix will include at least one element for each of the four letters. The letters marking will abide by the *'any sequence condition'* defined as follows: Let i, and j represent two different letters of the four {X,Y,Z,W}. At any given state let one of the n*m elements of the matrix be "*in focus*". Focus can be shifted by moving one element horizontally (right or left), or one element vertically (up or down) – reminiscent of the Turing Machine. Such a focus shift from element to an adjacent element is called "*a step*". The '*any sequence condition*' mandates that for any element of the matrix marked by letter i, it will be possible to shift the focus from it to another element marked by the letter j, by taking steps that pass only through elements marked by the letter i. The 'any sequence condition' applies to any element of the matrix, for any pair of letter (i,j).

5. **Select a starting point:** Mark any matrix element designated as "W" as the starting point (focus element).

6. **Build a pathway on the matrix reflecting the message (m_4):** Use the {X,Y,Z,W} sequence defined by the m_4 version of the message, to mark a pathway (a succession of focus elements) through the matrix. The "any sequence condition" guarantees that whatever the sequence of m_4, it would be possible to mark a pathway, if one allows for as much expansion as necessary, when an 'expansion' is defined as repeating a letter any number of times.

7. **Encrypt the pathway** : Describe the identified pathway as a sequence of edges, starting from the starting point. This will be listed as a sequence of up, down, right, left {U,D,R,L} to be referred to as the ciphertext, c.

The so generated ciphertext (expressed as 2 bits per edge) is released through unsafe channel to the intended recipient. That recipient is assumed to have in her possession the following: (i) the alphabet conversion tables, (ii) the matrix, (iii) the identity of the starting point, and (iv) the ciphertext c. The intended recipient will carry out the following actions:

8. **Reconstruct the Pathway:** Beginning with the starting element, one would use the sequence of edges identified in the ciphertext, as a guide to chart the pathway that the writer identified on the same matrix.

9. Convert the pathway to a sequence of vertices: Once the pathway is marked, it is to be read as a sequence of vertices (the matrix elements identified by the letters {X,Y,Z,W}), resulting in an expanded version of the message, m_{4exp}. The expansion is expressed through any number of repetitions of the same letter in the sequence.

10. Reduce the Expanded Message: replace any repetition of any letter in m_{4exp} with a single same letter.

11. **Reduce m_4 to m_3:** eliminate all the W letters from m_4.

12. **Convert m_3 to m_0:** use the alphabet conversion table to convert m_3 to the original message m_0.

Illustration: . Let the message to be encrypted be: m=m_0 ="love". Let the alphabet conversion table indicate the following:

l –- XYZ
o -- ZYX
v -- XYZ
e -- ZYY

Accordingly we map m_0 to m_3 = XYZ ZYX XYZ ZYY.

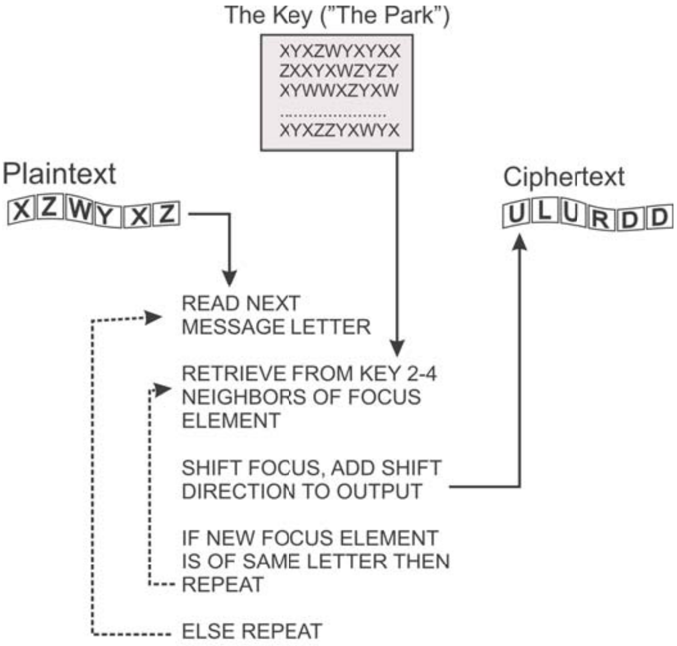
We now convert m_3 to m_4 = WXYZWZYXWXYZWZYWY.

We build a matrix that satisfies the 'any sequence condition':

$$\begin{matrix}1&2&3\\4&5&6\\7&8&9\end{matrix} = \begin{matrix}X&X&Y\\X&W&Y\\Z&Z&Z\end{matrix}$$

Using m_4 as a guide we mark a pathway on the matrix: 5,2,3,6,9,6,5,8,9,6,3,2,5,2,3,6,9,8,5,8,9,6,5,6

The pathway is now defined through the traversed edges, and construct the ciphertext c = URDDULDRUULDULDDLUDLULR. In order to decrypt c, its recipient will have to use the matrix (the graph, the key, or say, "the walking park"), and interpret the sequence of edges in c to the visited vertices: 5,2,3,6,9,6,5,8,9,6,3,2,5,2,3,6,9,8,5,8,9,6,5,6. This is the same pathway marked by the plaintext. Once it is marked on the matrix it can be read as a sequence of the visited vertices: m_{4exp} = WXYYZYWZZZYYXWXYYZZWZZYWY.Which is reduced $m_{4exp} > m_4$: WXYZWZYXWXYZWZYWY; Which, in turn, is reduced to the three letters alphabet: $m_4 > m_3$ = XYZ ZYX XYZ ZYY, which is converted to m = "love"



"Walk in the Park" Encryption Algorithm

"Turing Machine" algorithmic simplicity.
Security achieved via at-will randomness

Walk-in-the-Park as a TVC: There are various procedures, which would translate the matrix (the key) into a natural number and vice versa. Here is a very simple one. Let k be a square matrix (key) as described above, comprised of n^2 letters. Each letter is marked with two bits, so one can list the matrix row by row and construct a bit sequence comprised of $2n^2$ bits. That sequence corresponds to a non-negative integer, k. k will be unambiguously interpreted as the matrix that generated it. To transform a generic positive integer to a matrix one would do the following: let N be any positive integer. Find n such that $2(n-1)^2 < N \leq 2n^2$. Write N in binary and pad with zeros to the left such that the total number of bits is $2n^2$. Map the $2n^2$ bits onto a n^2 matrix, comprised ot 2 bits element, which can readily be interpreted as n^2 letters {X,Y,Z,W}. If the resultant matrix complies with the 'any sequence' condition, this matrix is the one corresponding to N. If not, then increment the $2n^2$ bit long string, and check again. Keep incrementing and checking until a compliant matrix is found, this is the corresponding matrix (key) to N.

It is clear by construction that Walk-in-the-Park is a TVC: the key (the map) gets larger with larger integer keys, and for some given natural number k_{Vernam} a message m will result in a pathway free of any revisiting of any vertex. The resultant ciphertext can then be decrypted to any message of choice simply by constructing a matrix with the traversed vertices fitting that message.

Cryptanalysis: A 9-letters key as in the illustration above will be sufficient to encrypt any size of message m. Alas, a cryptanalyst who is aware of the size of the key will readily apply a successful brute force analysis. Clearly, the larger the size of the key the more daunting the cryptanalysis. As long as the pathway revisits a vertex twice, the resultant cipher is not offering mathematical security, but for a sufficiently large map (key) the pathway may be drawn without revisitation of same vertices -- exhibiting Vernam, (or say, perfect) security. The critical feature of this cipher is the fact that the size of the map (the key) is integral part of its secrecy, so a cryptanalyst has no clear end point within which to conduct brute force cryptanalysis. For further depth and background see Samid 2002, Samid 2004.

III. USAGE SCENARIOS

We describe here a use case that is taken from a project under evaluation. The IOT node in this case is a micro drone equipped with a versatile video camera. The drone is extremely light, it has a small battery, and a solar cell. It is designed to land on large objects like trees and roofs. The camera streams to its operators a life video of the viewable vista. The drone requires encryption both for interpretation of commands, and for transmitting videos. The high-powered multi mega pixel camera may be taping non sensitive areas like public roads; it may stream medium sensitive areas, like private back yards, and it may also stream down highly sensitive areas, like industrial and military zones. The micro drone is dropped in the vicinity of operation, with no plans of retrieval, It should operate indefinitely.

Using Walk-in-the-Park the drone will be equipped with three keys (matrices, graphs): 1. a small hardware key comprised of square flash memory of 500x500 {X,Y,Z,W} letters. This will amount to a key comprised of 500,000 bits. 2. A flash memory holding 1000x1000 {X,Y,Z,W} letters, comprising 2,000,000 bits. 3. A flash memory holding 2500x2500 {X,Y,Z,W} letters comprising 12,500,000 bits. The latter key should provide perfect secrecy for about 6 megabytes of data, and will be used to communicate to the drone transposition keys to be used on another TVC transposition based cipher, not described here.

The determination of the security sensitivity of the photographed area (and the corresponding security level used) may be determined onboard the drone, or communicated from the reception center based on the transmitted pictures.

IV. Summary Notes

We presented here a philosophy and a practice for 'Cryptography of Things' (CoT) -- means to facilitate data security associated with things-nodes in the IP protocol. The CoT is mindful of processing parsimony, maintenance issues, and security versatility. The basic idea is to shift the burden of security away from power-hungry complex algorithms to variable levels of randomness matching the security needs per transmission. This paper presents the notion of Trans-Vernam Ciphers, and one may expect a wave of ciphers compliant with the TVC paradigm. It's expected that the IoT will become an indispensable entity in our collective well being, and at the same time that it should attrack the same level of malice and harmful activity experienced by the Internet of People, and so, despite its enumerated limitations, the IoT will require new horizons of robust encryption to remain a positive factor in modern civil life.

References

1. Auguste Kerckhoffs, « La cryptographie militaire », Journal des sciences militaires, vol. IX, pp. 5–38, Janvier 1883, pp. 161–191, Février 1883.
2. M. Hellman. 1977 “An extension of the Shannon theory approach to cryptography”. IEEE Transactions on Information Theory, V. 23 , 3 1977 , pp. 289 - 294
3. Ma´t´e Horva´th, 2015 “Survey on Cryptographic Obfuscation” 9 Oct 2015 International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/412
4. Masanobu Katagi and Shiho Moriai "Lightweight Cryptography for the Internet of Things" Sony Corporation 2011 https://www.iab.org/wp-content/IAB-uploads/2011/03/Kaftan.pdf
5. Rein Canetti, Cynthia Dwork, Moni Naor, Rafail Ostrovsky “Deniable Encryption” CRYPTO '97Volume 1294 of the series Lecture Notes in Computer Science pp 90-104Date: 17 May 2006
6. S. Zhou (ZTE Corporation) Z. Xie (ZTE Corporation) 2011 "On Cryptographic Approaches to Internet-Of-Things Security" http://www.lix.polytechnique.fr/hipercom/SmartObjectSecurity/papers/ZhouSujing.pdf
7. Samid, (A) 2015 “Equivoe-T: Transposition Equivocation Cryptography” 27 May 2015 International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/510
8. Menezes, A. J., P. van Oorschot and S.A. Vanstone. The Handbook of Applied Cryptography. CRC Press, 1997.
9. Samid, 2004 "Denial Cryptography based on Graph Theory", US Patent #6,823,068
10. Samid, 2015 “The Ultimate Transposition Cipher (UTC)” 23 Oct 2015 International Association of Cryptology Research, ePrint Archive https://eprint.iacr.org/2015/1033
11. Samid, 2016 “Shannon’s Proof of Vernam Unbreakability” https://www.youtube.com/watch?v=cVsLW1WddVI
12. Samid, G. "Re-dividing Complexity between Algorithms and Keys" Progress in Cryptology — INDOCRYPT 2001 Volume 2247 of the series Lecture Notes in Computer Science pp 330-338
13. Samid, G. 2001 "Anonymity Management: A Blue Print For Newfound Privacy" The Second International Workshop on Information Security Applications (WISA 2001), Seoul, Korea, September 13-14, 2001 (Best Paper Award).
14. Samid, G. 2001 "Encryption Sticks (Randomats)" ICICS 2001 Third International Conference on Information and Communications Security Xian, China 13-16 November, 2001
15. Samid, G. 2002 " At-Will Intractability Up to Plaintext Equivocation Achieved via a Cryptographic Key Made As Small, or As Large As Desired - Without Computational Penalty " 2002 International Workshop on CRYPTOLOGY AND NETWORK SECURITY San Francisco, California, USA September 26 -- 28, 2002
16. Samid, G. 2003 "Intractability Erosion: The Everpresent Threat for Secure Communication" The 7th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2003), July 2003.
17. Samid, G. 2003 "Non-Zero Entropy Ciphertexts (Stochastic Decryption): On The Possibility of One-Time-Pad Class Security With Shorter Keys" 2003 International Workshop on CRYPTOLOGY AND NETWORK SECURITY (CANS03) Miami, Florida, USA September 24 - - 26, 2003
18. Shannon, Claude, 1949 “Communicaiton Theory of Secrecy Systems” http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf
19. Smart, Nigel “Cryptography (an Introduction)” 3rd Edition http://www.cs.umd.edu/~waa/414-F11/IntroToCrypto.pdf
20. Stallings Williams, 2002 “Introduction to Cryptography” http://williamstallings.com/Extras/Security-Notes/lectures/classical.html
21. Vernam 1918; Gilbert S. Vernam, US Patent 1310719 Filed 13 September 1918.

.