Rating The Non-Vanishing Entropy of Randomness-Rich Ciphers

_The Remarkable Efficacy of Sub-Vernam keys_

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

Abstract: We rate our cryptographic products by one theoretical secrecy status, dismissed as academic, and one computational secrecy where we show that barring new mathematical insight, cryptanalysis will require a prohibitive computational burden. Alas, how do we regard a situation where there exist just two plausible messages co-encrypted into a single ciphertext, decrypted to either message, depending on which of the two keys the intended reader holds? Cryptanalysis practiced on that ciphertext will be as unfruitful as a cryptanalytic attack on a Vernam cipher. In general, sub-Vernam ciphers generate sub-Vernam terminal entropy (equivocation), and hence offer a remarkable efficacy which is not properly expressed in our established formalism. We propose a metric to reflect the benefit from ciphers with keys much larger than commonly used, but not prohibitively large.

Introduction

Randomness may be regarded as "cyber oil". Much as we gas-up our automobiles regardless of how big or how fast they are, so we random-fuel our cryptographic products regardless of their nature, or purpose. In fact cryptography may regarded as a discrimination science: marking a difference between holders of a piece of randomness and non-holders thereto. De-facto we have two categories of cryptographic products: the
'academic variety' which produces perfect secrecy but is a randomness-guzzler, and all the rest, which can be named 'short key' products where the randomness, the cryptographic key, is very short compared to the size of the plaintext that the key is designed to handle.

So, if we think of rating, and evaluating a cipher over a course of a lifetime, or say, a course of a year of service, and we expect to need to handle a quantity P of plaintext load, then using Vernam -- the mathematical secrecy cipher -- we will need to deploy a key (randomness) of size $|K| = |P|$. Which for ordinary service is an infeasible amount. Albeit, using any of the prevailing ciphers the required randomness $|K| \ll |P|$ is much smaller than the amount of processed plaintext.

What we have not had, until now, are cryptographic ciphers, which require randomness in between these two categories. The difference between the Vernam cipher and the short keys ciphers may be described through the concept of equivocation. Short keys ciphers commit to the plaintext that generated them. This generating plaintext may be hard to extract, but with sufficient computing power and persistence, it will be pulled out, and result in a successful cryptanalysis. By contrast, a Vernam ciphertext of size $n$ bits may be decrypted to $2^n$ plaintexts of same length. We may regard this as 'total equivocation'. It is this attribute, equivocation, that renders the Vernam ciphertext unbreakable.

So between the fully committed ciphertext -- zero equivocation, and the total equivocation we today have an "application desert". Alas, this desert begins to bloom, we see a crop of cipher products requiring more randomness than the typical short key ciphers, and less randomness than the Vernam option. Correspondingly, these ciphers -- sub-Vernam ciphers -- offer a varying degree of equivocation, more than zero, and less than total. This degree of terminal equivocation reflects a degree of cryptanalytic boundary, or conversely a degree of cryptographic security that needs a metric -- which is what we offer herein.
The Remarkable Efficacy of Sub-Vernam Ciphers

Key Size

Common Short Keys Ciphers

Vernam Cipher

Trans-Vernam Ciphers

Residual Equivocation Defense. Non-Vanishing Entropy

Vulnerability to Advanced Math Attack

Key Size

Key Size

Messages: Possible v. Plausible

Many textbooks analyze cryptography in a mathematical bubble while in fact it is a technology applied in practical adversarial situations. Mathematically Alice can send Bob any message she choses out of what is commonly regarded as a 'message space' -- a set of all sizes of all distinct bits strings. But a real life Alice wishing to send a real life Bob a message is looking at a small number $P_l$ of plausible messages to choose from. where $P_l < P_o$; $P_o$ being the number of possible messages.

Illustration: Alice moves towards a fork in the road, and she wishes to inform Bob that she will choose the left turn, and not the right turn. Alice will face two possible messages: "I choose the right way" or "I choose the left way". Alice and Bob are both aware of the circumstances and aware that Alice's binary choice is at stake. They are further aware that Alice message to Bob is designed to resolve this ambiguity. Eve, the adversarial eavesdropper, will likely read the map the same way, and is equally eager to
crack Alice's message to Bob to find out whether Alice will go right or go left. Now, if she applied powerful cryptanalytic effort onto Alice's message -- encrypted into C -- and finds that C contains both messages: \( M_r \) and \( M_l \) with one key, \( K_r \), decrypting C to \( M_r \), and another key, \( K_l \), decrypting C to \( M_l \) then without knowledge which key Bob is using, Eve faces a terminal equivocation that leaves her not a scintilla wiser than she was before she captured Alice's ciphertext. Come to think about it, this is the situation with Vernam. Had Alice used the Vernam cipher to send her message to Bob, Eve would have been exactly at the same state of ignorance.

Mathematically Vernam appears much superior to the two-message equivocation state because under Vernam Alice could have sent any and all messages of same size as C, not just two. Alas, the circumstances were such that only two messages were in play, and not knowing which one of these two, is as frustrating for Eve as having captured a Vernam cipher.

We argue, hence, that plausible messages are of greater interest then possible messages, and we set forth to handle them mathematically.

**Cryptanalytic Entropy**

Suppose that Alice, immersed in some specific adversarial situation, wishes to send Bob message \( M=m_0 \) [\( M \) is the message space that contains \( m_0 \)]. Her cryptographer advises her that she may choose another message \( m_1 \neq m_0 \) such that both \( m_0 \) and \( m_1 \) will be seamlessly packed into a combined ciphertext C, which when Bob will decrypt it he will see only \( m_0 \) and not \( m_1 \). His question to her is: what is the message \( (m_1) \) that will create maximum confusion with adversary Eve in the worst-case event when she cracked C to yield both \( m_0 \) and \( m_1 \).
Identifying $m_1$ is a matter of insight into the particular adversarial circumstances. It is a subjective judgment by Alice ($m_1 \left\Rightarrow m_1^a$). Albeit, in virtually all circumstances $m_1$ will be the negation of $m_0$. If $m_0 =$ "let's have a meeting" then Alice will choose $m_1 =$ "let's not have a meeting". If Alice approaches a fork in the road, and $m_0 =$ "taking the left turn", then Alice will choose $m_1 =$ "taking the right turn". Of course, should Alice choose $m_1 =$ "I will wear a green shirt" then the confusion imposed on adversarial Eve by the combined $[m_0 + m_1]$ will be negligible. Arguably in the former case the combined $[m_0 + m_1]$ will be equivalent to a total (Vernam grade) equivocation.

Alice's cryptographer could extend the cryptographic procedure to co-encrypt Alice's message ($m_0$) with some $d > 1$ so called decoy messages: $m_1, m_2, ..., m_d$, and allow Alice to choose such messages that will maximize the confusion (equivocation) faced by Eve when she chews on $C = \text{CoEnc}(m_0, m_1, ..., m_d)$, the co-encrypted ciphertext of the $(d+1)$ messages -- even if Eve has no computational limitations.

Unlike the binary case ($d=1$) where a negation of $m_0$ is the natural choice, for $d>1$ the choices are more subjective, and require insight into the particular adversarial circumstances. In the case of approaching the fork in the road, Alice could choose $m_2 =$ "I will stop at the road's split", as well as $m_3 =$ "Not advancing, waiting for your advice", and also: $m_4 =$ "Turning back", $m_5 =$"I think this ruse is working"... and so on, each decoy message that gets packed into the co-encrypted ciphertext increases Eve's equivocation. Again, assuming Eve is omnipotent and will unearth any and all messages packed into the co-encrypted ciphertext $C$.

Let's analyze the situation from Eve's side. Eve is informed that Alice communicated an encrypted message $C$ to Bob. Her cryptographer then asks her to list $d'$ plausible messages: $m_1^e, m_2^e, ..., m_d^e$ she suspects Alice could have sent Bob, given the particular adversarial circumstances at hand. Eve's list of Alice's plausible messages will help her cryptanalysis.
We first review the case where Alice's list and Eve's list are the same: \( \{m^a\}_d = \{m^e\}_d' \). In that case when omnipotent Eve finished cryptanalyzing \( C \) she ended up with a list of message candidates which is congruent with her list before she cryptanalyzed \( C \). Hence, this is a situation of perfect security since the awareness of \( C \) had no impact on the a-priori probabilities of the plausible messages. Say:

\[
\text{for } i=1,2,...,d' \quad \Pr_i | C = \Pr_i
\]

In the more general case we have \( \{m^a\}_d \neq \{m^e\}_d' \). Let's mark all messages that appear in both lists as \( m^{ae}\) for \( i \) taken from: 1,2,...,\( \min(d,d') = \{m\}^{ae} \), and mark \( m^{ea}\) for \( I \) taken from 1,2,...\( d' = \{m\}^{ea} \) all the messages that appear in Eve's list and not in Alice's list. Similarly \( \{m\}^{ae} \) denote all the messages that appear in Alice's list and not in Eve's list. When Eve exhausts her cryptanalysis of \( C \) she finds out that \( C \) contain some messages listed in her a-priori list, \( \{m\}^{ea} \) and some message that were not on her a-priori list \( \{m\}^{ae} \). The latter kind Eve will dismiss. If Eve associated some possible messages with zero probability to be the message Alice sent, then Eve will conclude that these messages are decoy messages in \( C \).

In this case the cryptanalysis of \( C \), while imposing a terminal equivocation did in fact give Eve an entropic advantage because the sum probability of 100\% is now distributed over a smaller list of plausible message \( \{m\}^{ea} \) then Eve's original list \( \{m\}^e \). Clearly in this case the security projected by the co-encrypted ciphertext \( C \) is less than perfect. How much less -- for that we need a metric.

Before cryptanalyzing \( C \), and only exploiting the fact that Alice sent Bob a message, Eve analyzed the situation and came up with her list of plausible messages: \( \{m\}^e = m^{e_1}, m^{e_2},..., m^{e_d'}, \) each of which was associated with a probability \( \Pr_1, \Pr_2=,...,\Pr_d' \). Clearly;

\[
\sum \Pr_i = 1 \quad \text{for } i=1,2,...,d'
\]
Using Shannon’s entropy, H, concept we can compute the entropy facing Eve versus her attempt to cryptanalyse C:

\[
H = -\Sigma \Pr_i \log (\Pr_i) \text{ for } i=1,2,...,d'
\]

After cryptanalyzing C, Eve will face a lower entropy, H':

\[
H' = -\Sigma \Pr[m^{ea}_i] \log (\Pr[m^{ea}_i]) \text{ for } l \text{ running over } m^{ea}_i \in \{m^{ea}\}
\]

The entropic reduction H-H' may be normalized to VSI = H'/H, where VSI stands for Vernam Security Index. VSI ranges from VSI=0, the case where the cryptanalysis is successful, given an omnipotent cryptanalyst, to VSI=1, the case where the co-encryption achieves a Vernam perfect security. The in-between values of the VSI measure the equivocation based security achieved in various circumstances. It covers the 'security desert' between the two extremes VSI=0 and VSI=1.00

Illustration: in the case where Alice approaches a fork in the road, Alice might consider the following three most effective decoy messages to protect the identity of the message she wishes to send to Bob ("taking the left turn" - L). They are \(m^a_1 = R = "\text{taking the right turn}"\); \(m^a_2=S=\text{"will stop at the road's divide"}\); \(m^a_3 = D = \text{"will divide me team between the right turn and the left turn"}\). Accordingly Alice co-encrypts:

\[C=\text{CoEnc}(m^a_0, m^a_1, m^a_2, m^a_3) = \text{CoEnc}(L,R,S,D)\]

Eve, on her part listed as plausible messages: \(m^e_1 = R, m^e_2 = L, m^e_3 = S \) and \(m^e_4 = W = "\text{I will withdraw}"\).

Applying her insight into the situation Eve assigned the following probability ratings to her plausible candidates:

\[Pr[R]=40\%, \ Pr[L]=40\%, \ Pr[S]=8\%, \ Pr[W]=12\%\]
Accordingly Eve's faces an entropy \( H = H(R, L, S, W) = 1.72 \) bits. After an exhaustive cryptanalysis of C, Eve finds that C is comprised of messages R, L, W, and D. Since Eve evaluated \( \text{Pr}[D]=0 \) before cryptanalyzing C, she will conclude that D is a decoy message. From the absence of message W in C, Eve will conclude that \( \text{Pr}[W]=0 \) and adjust her plausibility assessment to \( \text{Pr}[R]=44\%, \text{Pr}[L]=44\%, \text{Pr}[S]=12\% \), which computes to entropy \( H' = H(R, L, S) = 1.41 \) bits. Accordingly the Vernam security index of Alice's encryption is:

\[
VSI = \frac{H'}{H} = \frac{1.41}{1.72} = 0.82
\]

Interpreted as a message encrypted with 82% efficacy relative to having used Vernam cipher to protect Alice's message to Bob.

**Co-Encryption, Fusion**

Some m message \( P_1, P_2, \ldots P_m = \{P\}_m \) may each be encrypted using corresponding m keys: \( K_1, K_2, \ldots K_m = \{K\}_m \) over the same, or different ciphers. The resultant m ciphertexts: \( C_1, C_2, \ldots C_m = \{C\}_m \) may be combined in some way to a collective ciphertext \( C^* = f(\{C\}_m) \). \( C^* \) is a co-encrypted ciphertext of the m messages, \( \{P\}_m \). Alice could then send \( C^* \) to m correspondents where each correspondent has a different key from the \( \{K\}_m \) set. If the \( C^* \) combination function (f) is so structured that each correspondent i will be able to extract her respective ciphertext, \( C_i \), then Alice would accomplish her goal of sending m messages to m correspondents combined into one co-encrypted ciphertext, \( C^* \).

An omnipotent cryptanalyst capturing \( C^* \) will be able to separate \( C^* \) to its ingredients: \( \{C\}_m \), and crack each ciphertext to its corresponding plaintext. The next cryptanalytic step will be to find out which correspondent received which message. The cryptanalyst will be able to do it by positioning himself as the "man-in-the-middle", unraveling \( C^* \) to its \( \{C\}_m \) ingredients, then sending each individual ciphertext to each
correspondent. Every correspondent will respond with "I don't understand" message to all the ciphertexts, except the one meant to her.

For example C* could be a concatenation of individual ciphertexts with markers between them to allow the correspondents to identify their ciphertext:

\[ C^* = header_1-C_1-trailer_1-header_2-C_2-trailer_2-...-header_m-C_m-trailer_m \]

where the header_i and trailer_i are part of the shared secret between Alice and correspondent i. This technique is in line with the various Chaffing and Winnowing procedures in the literature [Rivest 1998].

In the above illustration the headers and trailers will be readily exposed as soon as Alice sends a second message with this protocol. And this raises the question: can we identify a cipher to be used in this combination protocol where the combination function f will be such that each correspondent i will readily identify C_i in C*, while a cryptanalyst aware only of C*, and not of \{K\}_m will not be able to winnow any C_i from C*. Or to be exact, is there a cipher and an f function such that a cryptanalyst acting as "Man-in-the-Middle" will have to check with each correspondent i all the possible bit sequences in C*.

A cipher like this, if found, will be regarded as a 'fusion cipher' and the corresponding f will be regarded as "fusing function".

By definition a fusion cipher can be used to distribute m messages to m recipients by passing along a single file, C*. C* can be broadcast in m copies to the m recipients, or in one copy to be passed along sequentially among the recipients, or any combination thereto. Whichever the distribution pattern, each recipient will extract from the fused ciphertext exactly the message intended for him or her.

Likewise, Alice could use any fusion cipher to send Bob message m_0, and load into the ciphertext the encrypted versions of her choice of d decoy messages: m_1, m_2,..., m_d,
and thereby effect functional secrecy by equivocation. Bob will readily retrieve $m_0$, and adversarial Eve will --at best-- be terminally confounded by the list $m_0, m_1, m_2, \ldots, m_d$.

**Security Discussion**

We consider the case where Alice uses a fusion cipher to send Bob message $m_0$ by packing it with $d$ decoy message $m_1, m_2, \ldots, m_d$ into a co-encrypted ciphertext $C$. What is Alice's security standing?

We first analyze this question by taking all assumptions in favor of adversarial Eve: we consider her omnipotent, and of complete visibility as to the actions and movements of Alice and Bob. In this case Eve will crack $C$ and list $m_0$ plus all the $d$ decoys. In the best case for Eve, she would cryptanalyse $C$ as "man-in-the-middle", and then send Bob the extracted $d+1$ ciphertexts one by one, to find out which one is aimed at Bob. Alice and Bob can counter this by pre-agreeing on decoy messages. That is, from time to time each will send each other a meaningless message to which no response is expected. This procedure will deny Eve the chance to associate $m_0$ with Bob. At best she will enjoy an entropic advantage as described above, and as measured by the Vernam Security Index (VSI) based on any difference between Eve's list of plausible messages, and Alice's list.

Looking beyond this instant to a state where Eve watches Bob's behavior, and learns from it which of the $(d+1)$ messages was intended for him, and hence will be able to identify the next message Alice sends to Bob using the same parameters. Aware of this risk Alice and Bob may exchange the other $d$ keys corresponding to the $d$ decoy messages. Bob will then be able to read the decoy messages and then behave in a way that will not expose the true message Alice sent him. Alternatively, or, in addition, Alice and Bob could agree on some rotation so that they switch the key that decrypts the true message Alice sends to Bob.
A particular risky scenario for Alice and Bob comes up if Eve is not aware that C is a fusion cipher, and by chance she extracts $c_0$ from C, and cryptanalyses $m_0$ from it. The more decoy messages there are, the lower the chance for that scenario. But in a binary case, Alice and Bob face a 50% chance of such risk. Alice and Bob may counter this risk by advertising the fact that they use a fusion cipher. Since a fusion cipher also hides the number of messages fused into it, Eve might extract all or some of the (d+1) messages, and worry that it is not all, may be the 'real message' is still hiding in C.

**Trans Vernam Ciphers**

The reason for the above mentioned "application desert" between short key cryptography and Vernam cryptography is: (i) Vernam's requirement for a key as large as the plaintext, is very rigid. Re-using any non-trivial segment of the key will collapse the equivocation to a vanishing entropy; (ii) short key cryptography is based on extensive mixing of key bits with plaintext bits -- the computational burden rises exponentially with the size of the key. So it turns out that Vernam does not allow for any smaller key, and the prevailing ciphers do not allow for a much larger key. This leaves the in-between zone unoccupied.

This ‘desert’ has recently begun to bloom: ciphers which allow the key (the randomness) to grow as desired without a prohibitive computational hurdle. Powered by Moore's law for memory, which enabled larger and larger memories everywhere, these randomness-rich ciphers bridge the gap between the prevailing short key cryptography and the "academic" Vernam cipher.

We may define a Trans-Vernam cipher, TVC, as a cipher, which operates effectively on large as desired key (randomness), projecting increased security for increased randomness, up to equivalence with a corresponding Vernam cipher. Unlike short keys
cryptography where in general the amount of processed plaintext is not a critical security factor, the security of a TVC is critically affected by the total amount of processed plaintext. As long as the size of the key, \(|K|\) is equal or larger than the size of processed plaintext, \(|P|\), the TVC projects Vernam grade security: \(|P| \leq |K|\) \(\Rightarrow\) Vernam Security. When the amount of processed plaintext starts to exceed the key size, Vernam security is lost, but for a while equivocation security is maintained. The degree of such equivocation security is captured by the Vernam Security Index defined herein. As more and more plaintext is processed via this TVC, the residual equivocation security diminishes, until at some point the TVC, like ordinary short key cryptography, commits to its generating plaintext. At this state, it is mathematical intractability that projects the security of the cipher. Since the measure of randomness used is part of the secret, a cryptanalyst faces the question of potential futility, for the possibility that the cipher still projects Vernam security, or close to it, as measured by the Vernam Security Index.

The co-encrypted ciphertext described herein is a case where randomness measured as the summation of the \((d+1)\) keys (the message plus its decoys) is creating strong Vernam Security Index (high level of equivocation). In general, by using a sufficiently large key, the cipher writer does not have to a-priori specify decoy messages since the richness of the key will statistically bring them out. This is most clearly visible in the Ultimate Transposition Cipher below.

Read more in “Randomness Rising” [Samid 2016R]

"Walk in the Park" cipher

We offer here a brief overview of the cipher described in [Samid, 2002, 2004, 2016B, 2016C]. In the Walk-in-the-Park cipher (WaPa) one uses a \((t-1)\) letters alphabet, and then introduced a \(t\)-th letter to break any repetition in any given plaintext. For example for \(t=4\), the \((t-1)\) alphabet will be comprised of \(X, Y,\) and \(Z\), used to express any message. One arbitrary message may then look like \(m'=XYYZXZZZYX\). The 4-th letter,
W will be wedged between any two like letters to construct m=XYWYZXZWZWXY which is a repetition-free string. Accordingly m' can be expanded at will by duplicating any letter to create m". There will be no ambiguity in contracting m" back to m. In the illustration above m may become m" = XYYYYYYYYYYYYWYZXZWZZZZZWWWWZYXXXXXXXX and will readily be reconverted back to m by eliminating all duplications.

The WaPa cipher views the plaintext as a sequence of vertices in a graph: v1, v2,..., and the corresponding ciphertext is a series of edges e1, e2,... that marks the same pathway on the shared graph. The intended reader of the message, in possession of the graph, will readily convert the ciphertext (the series of edges) to the plaintext (the series of vertices).

Let Alice's message to Bob be: m0 = XYZXWY (no repetitions). She now wishes to co-encrypt it with a decoy message m1 = Y'Z'W'Z'X' (where the single quotes only signifies that the letters belongs to m1). To do that Alice will expand her message to Bob:

\[ m''_0 = XYYYYXWWY \]

as well as expand the decoy message:

\[ m''_1 = YYYYYWWWZWWWZXX' \]

Both messages are comprised of 11 letters. Alice will now construct a WaPa graph on which the two messages will mark a pathway defined by the sequence of their letters. Alice will allow for two versions of the same graph, one marked by the letters in m''0 and the other by the letters in m''1. Alas, the shape of the pathway will be the same in both versions, and both versions will have the same designations of the graph edges, and therefore both pathways will be expressed via these edges:

\[ C = CoEnc(m_{0}, m_{1}) = e_1 \cdot e_2 \cdot \ldots \cdot e_{11} \]
Bob will decrypt C to m\"_0, which he will readily shrink to m_0, and read Alice's message. But Eve might also discover the second version of the graph, and use it to interpret the same ciphertext to m\"_1, and shrink it to m_1. So in summary, omnipotent Eve will face a terminal equivocation as to which message was sent to Bob m_0 or m_1.

**The BitFlip Cipher**

The BitFlip cipher is a polyalphabet cipher where each letter may be represented by any member of a very large set of n bits strings. Using a t letters alphabet this cipher may accommodate d decoy messages by arranging for $2^n/(d+1)t$ strings to represent a given letter to interpret a given message. When a reader encounters a string that does not evaluate to any of the t letters in her key, she ignores it. This will allow Alice to put together a series of n-bits strings such that each of the (d+1) messages will be the one read by a reader who holds the key for that message. Eve, unaware which key Bob holds, will at best, for her, decrypt the co-encrypted ciphertext to Bob's message plus the d decoy messages, alas, will not be able to resolve this terminal ambiguity. Read more [Samid 2016].

**The Ultimate Transposition Cipher**

The ultimate transposition cipher (UTC) offers a key space of n! to randomly transpose any n items list to any other. Alice could arrange her message to Bob (m_0) and her d decoy message, m_1, m_2,....m_d as follows:

$$P = m_0 \[*\] m_1 m_2 ...... m_d$$

where '[*]' is the divider between the message for Bob (left of the marker), and the decoy messages (right of the marker). Alice would then transpose P into a T(P) permutation. Bob, in possession of the transposition key will readily reconstruct P and identify his message m_0. Adversarial Eve will identify at least (d+1) transposition keys that would each reverse transpose T(P) to:
\[ P = m_i \lbrack^* \rbrack m_0 m_1 \cdots m_{i-1} m_{i+1} \cdots m_d \]

And will face terminal equivocation. The price: using a much larger key space: \((n_0 + n_1 + n_2 + \ldots + n_d)!\) rather than the smaller \(n_0!\) key space, where \(n_i\) is the number of transposed entities in message \(i\). Note: there is a strong chance that the decoy messages share transposed entities, which will reduce the size of the key needed for equivocation.

**Summary**

We highlighted here a rising class of cryptographic ciphers which build security not on algorithmic complexity but rather on terminal equivocation. These ciphers feel the gap between zero equivocation (small unicity distance) typical of all the prevailing ciphers, and the total equivocation offered by Vernam ciphers. We propose a metric to capture the level of security offered by various degrees of equivocation. The resultant metrics (the Vernam Security Index) will be a practical tool to assess the growing number of ciphers of this kind.

**Reference:**


Nies 2008: “Computability and randomness” Niels A. The University of Auckland, Clarendon, Oxford, UK

Samid, 2001 “Re-Dividing Complexity between Algorithms and Keys:
http://link.springer.com/chapter/10.1007/3-540-45311-3_31#page-1
Samid 2002: " At-Will Intractability Up to Plaintext Equivocation Achieved via a Cryptographic Key Made As Small, or As Large As Desired - Without Computational Penalty " G. Samid, 2002 International Workshop on CRYPTOLOGY AND NETWORK SECURITY San Francisco, California, USA September 26 -- 28, 2002
Samid 2004: "Denial Cryptography based on Graph Theory", US Patent #6,823,068
Samid 2003 “Essential Shannon Security with Keys Smaller than the Encrypted Message”
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.1585&rep=rep1&type=pdf
Samid 2016A: “Shannon’s Proof of Vernam Unbreakability” G. Samid
https://www.youtube.com/watch?v=cVslW1WdVI
a1c46380a815#.9qq9w8lb8
Smart 2016: “Cryptography Made Simple” Nigel Smart, Springer.