The Nino Cipher

The Foundation to Next-Generation Security

Abstract. In this paper, we introduce the concept of Equivocation Augmentation and then propose a new information-theoretic multi-purpose encryption system which uses the concept in a practical manner, namely the NINO-Cipher. The NINO-Cipher is a next-generation encryption algorithm which is impossible to compromise using brute-force. In essence it is composed of a chain of One-Time Pads, bound together using a random entropy stream. By actively engineering Shannon's equivocation characteristic, we demonstrate that the key entropy to a cryptosystem can be augmented at a faster rate than it can be consumed in the encryption process, allowing for the augmentation of the equivocation of key $H_E(K)$ and message $H_E(M)$, such that they never attain unity, thereby guaranteeing absolute security using a finite length key, irrespective of the length of the message M.
The NINO-Cipher is unique in that it is the first cipher to break Shannon's "equivocation-barrier".

A Simple Solution to a Complex Problem

Our digital world has evolved at a faster rate than the security solutions which we use to secure it, such that what was assumed "secure" yesterday is "insecure" today, making it more dangerous and less secure with every passing day. The problem is basically one where we have built our security future on flawed foundations. With the advent of quantum computing and artificial intelligence security threats, our current digital security paradigm is rapidly reaching the point where it is no longer fit for purpose. We are now at the point where a return to basic cryptographic principles – Shannon's principles – is inevitable.

In his 1949 paper "Communication Theory of Secrecy Systems", Claude E Shannon mathematically defined what it means to be cryptographically secure and proposed three secrecy systems, namely "ideal secrecy" and "perfect secrecy" which are information-theoretically secure, and "practical secrecy" which is by definition information-theoretically insecure. Any info-theoretically secure system, specifically an "ideal secrecy" system, is secure in that a brute-force attack on the key to an intercepted ciphertext is guaranteed to always result in 2 or more viable messages being decrypted. The objective being that the assailant will have to solve an "unsolvable" cryptographic problem. We adopt a very pragmatic view that only information-theoretically secure systems may be considered "secure", as opposed to being "safe". We therefore reject the concept of "reduced notions of security", since they are in effect only "notions of safety" – unfortunately, it is illogical to deduce "security" by redefining the requirements, and scientifically flawed to use an affirmation of the consequent as a logical basis for such a deduction. If a cipher is secure, it is an intractable problem. But just because a cryptographic problem is intractable does not make the cipher secure.

"Perfect secrecy" is a specific instance of "ideal secrecy", in that a brute-force attack on the key will result in all possible messages appearing as valid decipherments. In contrast, information-theoretically insecure systems or "practical secrecy" systems are guaranteed to result in a single viable message, and therefore have an inherent insecurity guarantee. Since they cannot prevent the inevitable compromise under upon a brute-force attack on the key, many such systems rely on the use of mathematical complexity as a means of increasing the workload required to breach such systems.

Interestingly, the common analogy of defining "security" as a search for a needle in a haystack is fundamentally flawed. Security is only present if two or more indistinguishable needles are found in the haystack. Unlike info-theoretical secrecy systems, "practical secrecy" systems do not require the whole stack to be searched, and may be compromised almost instantly.

In his paper, Shannon proved mathematically using equivocation (and scientifically, since equivocation can be verified by deductive logical experimentation) that the simple "one-time pad" (OTP) is a "perfect secrecy" system. An OTP is a simple encryption cipher which entails the modular addition of the one-time pad values (the key) and the message values (note that all information can be represented as numbers) to produce a ciphertext. Decryption entails a reversal process using modular subtraction of the ciphertext using the same OTP key to produce the initial message.

An OTP is information theoretically secure in that all possible messages appear as valid decryptions, but requires a truly random encryption key (the one-time pad) which can only be used once, is as long as the message, and which must be securely distributed before the encryption process takes place. Unfortunately, until now, whilst practically simple to implement, an OTP secrecy system has proven almost "impossible" to solve in a digital context, since it requires another secure communication channel for distribution and storage of an OTP sufficiently long to secure all future messages, and requires the OTP to be securely stored. This can only be attained using another OTP, and so we have an infinite security problem.

This practical difficulty remained unsolved until 2015, although developments in the 60's and 70's resulted in the development of mathematically-based asymmetric public/private key encryption systems, which whilst being info-theoretically insecure "practical secrecy" systems, provided a temporary key distribution solution where previously there was none. Objections were made by the "purist" cryptographic community that such systems were flawed in that Shannon specifically stated that "practical secrecy" systems cannot be relied upon as security systems, unless it can be proven that the required high workload was assured, namely that there was no possible means by which a brute-force attack could be accelerated. In any event, it is not possible to detect when such systems have been compromised, allowing an attacker to choose the optimum moment in which to compromise a system.

This new cryptographic field of mathematical complexity had massive appeal amongst the mathematical community, such that today, all commonly used secrecy systems are unproven "practical secrecy" systems, with no scientific equivocation basis for their security. We term this "safety" in contrast t "security". Note that mathematicians are still wrestling with the mathematical complexity problem of whether P=NP, so in effect, we are currently resigned to using encryption systems which are unproven "information theoretically insecure" systems. This unfortunate trend is in direct contrast to Shannon's basic cryptographic principles such as equivocation.

The security requirements of today's digital users have changed considerably. They want simplistic "information-theoretically secure" encryption systems which are based on scientific method, not mathematically complicated "information-theoretically insecure" systems which are based on assumptions of complexity or the opinions of "experts". In order to satisfy this requirement, we will have to return to Shannon's basic cryptographic principles.

Specifically, this paper describes a technique used by the NINO cipher called equivocation augmentation, which finally solves the OTP's distribution and storage problems, thus enabling its practical use in a digital context. The NINO cipher is simply an OTP-chain info-theoretically secure secrecy system.

One Time Pad – Going Back to the Future

The simple solution to the OTP distribution and storage problem, is that the separate secure distribution and storage of the entire OTP key prior to encryption can be eliminated completely. Instead of relying on a single long OTP operation using an OTP key as long as the message to be encrypted, the OTP encryption sequence may be composed of any number of "minor" OTP encryptions in a structure called an OTP-chain. In a similar manner to blockchains which use hashes to link transactions, an OTP-chain uses random entropy to link the "minor" OTP encryptions. Random entropy (not keys) is transmitted with every "minor" OTP encryption, and then combined with the current OTP key to reconstruct the next OTP key.

This technique, known as "equivocation augmentation", allows for an OTP-chain information-theoretically secure encryption sequence to be initiated with a key much smaller than would ordinarily be required, irrespective of the message length.

OTP Key distribution therefore occurs "just in time" using a separate independent OTP encrypted random entropy sequence, transmitted simultaneously with encryption, eliminating the need to store or manage large OTP keys locally. In effect the NINO-cipher is composed of two independent OTP cryptosystems, one encapsulated inside the other. Whilst it is generally impossible to maintain info-theoretic security for an infinite length of time by encrypting a joint message and distributing the next OTP key simultaneously in the same encryption operation, one need only distribute enough random entropy to replenish the key equivocation lost in the current OTP key, prior to each subsequent encryption, like refueling a car before it runs out of fuel.

By applying Shannon's principles of equivocation, it is possible to preserve the OTP's info-theoretic security characteristics by breaking up the OTP encryption into a number of smaller OTP encryption steps. A small OTP key can be used to start the first OTP encryption step, encrypting a part of the message and a random string which is generated on demand. The random string is then used by both sender and receiver, not as a key, but as entropy to "top up" the lost key entropy / equivocation of the current OTP for the next encryption step. This process can be repeated in an infinite manner to encrypt a message of any length.

As we will shortly demonstrate, since key and message equivocation can never attain the value of 0, the NINO-cipher is guaranteed to be information theoretically secure (it has a security guarantee!) irrespective of the key size used (although a minimum of 10 bits is required). It is therefore also quantum-resistant. Since it does not depend on workload for security, it is extremely simple in composition, and has superior performance characteristics. Most importantly, it is a cipher which is purely based on Shannon's principles of equivocation, and can be validated using scientific experimentation. It has no backdoors and doesn't use mathematical complexity, assumptions of security, or rely on the opinions of experts. Strictly speaking, it is not a cipher with a static design but a cryptographic design framework, in that it may be used to create an almost infinite number of variants, and can dynamically change its structure whilst in use. It does not use keys of a fixed length, and can be adapted to include any existing secrecy system, making it info-theoretically secure.

Equivocation = Security

At this point, one may ask "what is equivocation"?

Equivocation is in the simplest of terms, information theoretic security. In his 1949 paper, "*Communication Theory of Secrecy Systems*", Shannon introduced "equivocation" as a secrecy index which could be used to analyze and graphically visualize the security of any cryptosystem. As we shall see, any encipherment with key or message equivocation equal to zero (log 0 = 1) is in effect compromised and insecure. Shannon defined key and message equivocation as the conditional entropy of key and message with respect to a specific piece of intercepted ciphertext. For a complete mathematical treatment of equivocation, we refer you to Shannon's paper.

In layman's terms, equivocation is the technique used by cryptanalysts to count valid keys and messages, much like a card player counts cards at a blackjack table. In cryptography one does not count the actual value of keys and messages, but the logarithm of the actual values, so 64 (2^6) keys have a key equivocation value of 6, and 256 (2^8) keys would have a key equivocation value of 8. The same principle is applicable to message equivocation.

With symmetric encryption, a key ***K*** is first selected at random from the set of all possible keys ***H(K)*** (the entropy/information of the key). Key ***K*** is then used to encrypt a message ***M*** selected from a set of possible messages ***H(M)*** (the entropy/information of the message), to produce a unique ciphertext ***C***. Decryption entails the reverse process of applying the identical key ***K*** to the ciphertext ***C*** to produce the initial message ***M***. With asymmetric encryption, the key ***K*** is split into 2 inverse keys such as **K/2** is used for encrypt and **2K** used for decryption.

From a cryptanalytic perspective, we take the role of the assailant that has captured a ciphertext ***C***. The assailant first determines the encryption/decryption cipher used, and then determines the entropy ***H(K)*** of the key ***K*** (the total number of keys to try). At first, all the keys ***K*** to be tested are treated as valid and constitute the key equivocation $H_E(K)$ value, such that a 256-bit key with 2^{256} possible keys has a key equivocation $H_E(K)$ of 256. The assailant then tries every possible key ***K*** with the ciphertext ***C***, and checks to see if a valid ***M*** is produced. If a valid message ***M*** is produced with a key ***K***, the message equivocation count $H_E(M)$ increases. If an invalid message ***M*** is found, denoting an invalid key ***K***, the $H_E(K)$ count is reduced.

When the ciphertext ***C*** is as long as the key ***K***, as is the case with an OTP, the assailant will be left with a pile of all possible valid messages ***M*** and their associated keys ***K***. In other words, $H_E(K)$ and $H_E(M)$ will be of equal value. This is the security condition known as "perfect secrecy", since all possible messages will be present. Note that the assailant's efforts are pointless, even if he was able to complete the search with unlimited computing resources, for he cannot distinguish which message is the correct one from the residual message set (the message equivocation).

However, when the ciphertext ***C*** is longer than the key ***K***, then the assailant has the benefit that the residual set of keys ***K*** can be retested, resulting in a further elimination of the residual set of valid messages ***M*** until they are reduced to a single valid message ***M***, such that $H_E(M)$ and $H_E(K)$

eventually equal zero, or log 0 which is equal to 1 – the eventual compromise of the system for a unique message has been identified.

Understanding Equivocation

The beauty and simplicity of equivocation, is that one can graphically depict the security characteristics of any secrecy system. It is probably for this reason that it does not feature prominently in the cryptanalysis of "practical secrecy" systems, for it depicts the insecurity of such systems quite readily.

If we take the characters of a language alphabet and compose messages using all possible variations of the language, we will have two sets of messages, one set which is valid for that language (information) and another set composed of invalid sequences of characters (redundancy). With normal English messages over 30 characters in length, using an 8-bit medium (so 256 characters), 1.3 bits of every 8 bits is information and 6.7 bits is redundancy. The proportion is therefore 16% info and 84% redundancy. Shannon demonstrated that the amount of ciphertext required to solve any encipherment (its unicity distance) could be calculated by dividing the entropy of key by its redundancy. So a 256 bit AES encipherment pf an English message is guaranteed to be solved if the ciphertext is longer than 39 characters. Note that 256/0.8 = 304.76 bits = 38.09 characters.

In the following diagram, we denote the equivocation graphs for 4 separate messages of 8 characters in length, each character being of 8 bits, encrypted using a one-time pad with a key of 32 bits. Each message differs only from the others in terms of the amount of information that they contain. By information, we mean any meaningful and valid message in the specific language used. These different message types include the following, which are each displayed in a separate colour:

- M0 – A known plaintext message (has 0% information, 100% redundancy) – [red]
- M1 – A normal English message (16% info, 84% redundancy) – [black]
- M2 – A balanced language message (50% info, 50% redundancy) [green]
- M3 – A random string message (100% info, 0% redundancy) [blue]

Note that their respective key equivocations $\boldsymbol{H_E(K)}$ are plotted in their respective colours, using an unbroken line, whilst their $\boldsymbol{H_E(M)}$ message equivocations are denoted with a dotted line.

Whilst this paper has drafted with laypersons in mind, academic readers are referred to Shannon's paper "Communication Theory of Secrecy Systems" for a more robust mathematical treatment of equivocation.

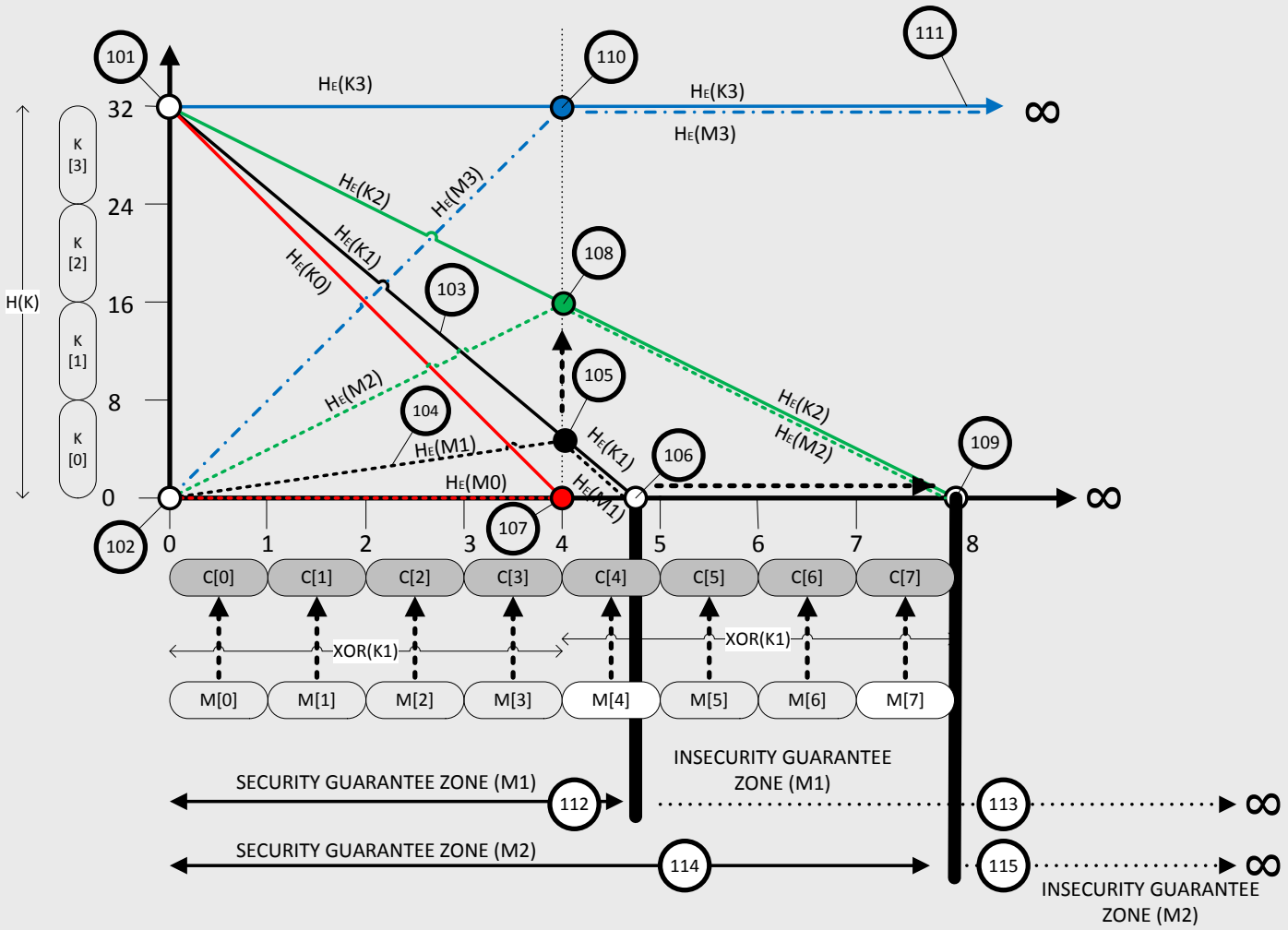

FIGURE 1: Equivocation Graphs of 4 message information types.

From the equivocation graphs of the 4 message types above, the following observations can be made:

- The key entropy **H(K)** is denoted on the Y-Axis, whilst the number of characters are denoted on the X-axis. Both the Y and X axis are logarithmic, allowing projections for keys and messages of any size and length.
- All message equivocations ***HE(M)*** begin with a value of 0 at **102**.
- All key equivocations ***HE(K)*** begin with a value of 32 at point **101**, since they all use the same 32-bit key **K**.
- The ***HE(M)*** and ***HE(K)*** of all messages meet at their respective *perfect secrecy* points – M0 (**107**), M1 (**105**), M2 (**108**) and M3 (**110**).
- The height of each *perfect secrecy* point is determined solely by the amount of information in the message type, with the lowest being M0 (the known plaintext) and the highest being M3 (the random string).
- The slopes of ***HE(M)*** and ***HE(K)*** for each message type are solely determined by the amount of information in the language message type, not the key. Ordinarily it is the language used which solely determines the slope of the key and message equivocations.

- $H_E(K)$ ordinarily has a downward trend towards its respective *secrecy point* where after it continues along its current projection until it reaches unity, or 0 as shown by points **106** and **109**.
- When $H_E(M)$ meets $H_E(K)$ at the *perfect secrecy* point, $H_E(M)$ follows $H_E(K)$ to unity.
- When any key or message equivocations hit 0 (the *unicity point*) the ciphertext is effectively solved, since there will be a unique message. Log 0 = 1.
- The *unicity point*, determines the length of ciphertext up to which the encipherment is info-theoretically secure. This length is known as the *unicity distance*. As mentioned earlier, the unicity distance can be calculated by dividing the key entropy $H(K)$ by the redundancy in the message. So, for **M1**, and English message, 32/.84 = 38.09 = 4.76 characters as denoted by point **106**.
- Ciphertexts of each message type shorter than the *unicity distance* have "ideal secrecy" and are info-theoretically secure (they have a security guarantee). Note that **M0** (known plaintext) has no *unicity distance* since it's $H_E(M)$ never rises above 0.
- Ciphertexts with any amount of information and which are exactly and only exactly as long as the key **K** have "perfect secrecy", which is a special instance of "ideal secrecy".
- Ciphertexts of each message type longer than the *unicity distance* have "practical secrecy" and are info-theoretically insecure (they have an inherent insecurity guarantee). Such encipherments cannot be considered secure to begin with, much like the security of a car journey cannot be securely guaranteed if there is insufficient fuel to complete the journey to start with.
- Any encipherment will continue to remain info-theoretically secure, as long as neither the key nor the message equivocations never equal to 0, or as long as the message is shorter than the *unicity distance*.
- Increasing the key entropy $H(K)$ (or key size), will increase the *unicity distance* proportionately.
- Random strings **M3** are the most secure of encipherments in that they are always info-theoretically secure, and have an infinite length *unicity distance*.
- It is not the secrecy system but the length of each specific encipherment which determines information theoretic security. All secrecy systems are capable of producing encipherments which are either "ideally secret", "perfectly secret" or "practically secret", differing only in the length of the message that was encrypted.
- The equivocation principles mentioned above are generally applicable to all secrecy systems, such that an AES-256 encipherment longer than 39 characters (32 bytes divided by 0.84 = 38.09) is info-theoretically insecure, and guaranteed to be compromised using a brute-force attack on the key (or the message).
- As mentioned, it is the specific encipherments and not the secrecy systems which are info-theoretically secure. It is exceedingly difficult for an encryption system to be deemed info-theoretically secure, for it must be secure under all possible message lengths. In other words, it's key and message equivocation must NEVER reach zero, irrespective of the message length.

Note that whilst we have considered a brute-force attack on the key in the example above, it is in general more efficient (but not always possible) for an assailant to perform a brute-force attack on

the message (an infinitely smaller search space), by trying every possible message to get the key, and validating the key against the rest of the ciphertext. In an "ideal secrecy" system, such a technique has no benefit. But in an insecure "practical secrecy" system, this is usually the fastest way to breach the cipher if an "inverse function" is available since viable keys can be eliminated and verified if encipherments are longer than the key. Any encryption system which cannot prove that it is impossible to engineer an "inverse function" of the system (allows the key to be derived by trying the message) should not be trusted. Likewise, any cipher that cannot be verified in a scientific context using deductive logic, should not be trusted. In the same manner that we get "pseudo-random" number generators (random number generators which appear to be random but are not), information-theoretically insecure systems should be considered "pseudo-secure" secrecy systems (for such "secrecy systems" merely appear to be secure but aren't).

Note that it is in general, extremely irresponsible bad practice to make any security assumptions – since the assumption of insecurity is a mandatory requirement.

Engineering Equivocation

Most importantly, from the equivocation graph of **M1** above, it should be clear that in the case of a one-time pad, the provision of a key **K** as long as the message **M** will result in an information-theoretically secure encipherment where all message **M** will be produced as valid decryptions upon a brute force attack.

The fundamental problems to be solved, is that the OTP key must be (a) truly random, (b) securely distributed before the encryption, (c) sufficiently long to accommodate a message of any size, (d) securely stored.

All 4 problems are addressed simultaneously using the NINO-cipher. First, we break up the OTP operation into smaller OTP encryption blocks. In each OTP encryption block, we XOR encrypt some message characters and the independent XOR encryption of some randomly generated characters. This is then subjected to a transposition of fixed period (a shuffle). Following each encryption block, the random characters are used to augment the lost OTP key equivocation making all keys possible for the next OTP encryption block. Note that we do not pass OTP keys, but the random material used to create new OTP keys. In this manner, we only need an OTP key large enough to complete the first OTP encryption block, in order to encrypt a message of any length.

The only real technical cryptographic question to be answered, is whether it is possible to augment the key equivocation such that it is as high as the key entropy. Whilst Shannon stated that the uncertainty of a cryptosystem is limited to the key, we will demonstrate that this is only the case when one uses a single secrecy system. The NINO-cipher uses at least two independent OTP encryptions, an internal OTP XOR for entropy exchange, encapsulated by an external OTP XOR and transposition for the main encryption. In this manner, the initial ***H(K)*** can be exceeded. Whilst ordinarily, message expansion is an unwanted trait, in this case it allows for the "impossible" practical limitations of the OTP to be easily solved.

In contrast to most secrecy systems which merely stand by and watch as the very life of the cipher (the key and message equivocation) ebbs away, we are going to actively manipulate the key and message equivocation of an OTP encryption to our advantage. Now that we have a firm grasp on the basic fundamentals of equivocation, we are going to put this knowledge to use by engineering the equivocation of a one-time pad (OTP) in a beneficial manner. This technique is applied to all OTPs in the NINO-cipher, which is effectively an OTP-chain bound by entropy.

We are first going to engineer the key equivocation of each OTP encryption block with the following 3 basic operations:

- We reduce the rate of equivocation decay by transmitting random strings.
- We block the assailant's ability to equivocate the entire key, using super-encapsulation (inserting an encryption ciphertext within another encryption ciphertext).
- We reset the key equivocation after every micro-OTP encryption.

We will then focus on the message equivocation of the OTP using the following basic operation:

- We increase the message equivocation using a transposition.

Following these four operations, we will have the basic composition of a NINO-cipher, a simple and secure cryptosystem, capable of ensuring that any encryption is information-theoretically secure using a finite length key, irrespective of the message length, or the amount of information contained in the message. Note that any existing encryption system may be incorporated by using it to encrypt the message, but this is ordinarily unnecessary and inefficient.

Engineering Key Equivocation

Note that it is a basic requirement for cryptographers and cryptanalysts to have a thorough understanding of equivocation at a mathematically complex level. Any cryptosystem designer who does not know what equivocation is does not belong in the field, much like an accountant that does not know how to count.

Since the downward key equivocation slope is a terminal condition indicating the inevitable "death" of the security of the cipher, we need to slow this rapid rate of decay down. The slope of the key equivocation of an encipherment depends solely on the amount of information in a message being encrypted (remember that the height of the *perfect secrecy* point increases with an increase in message information, and an increase in key size), a characteristic determined solely by the language of the message, we are therefore first going to increase the information in a message by adding random strings to messages.

The resultant impact on the key and message equivocation of an encipherment using random strings is visible in the following diagram, where two message characters are joined to two random characters in every minor OTP encryption block of the OTP-chain:

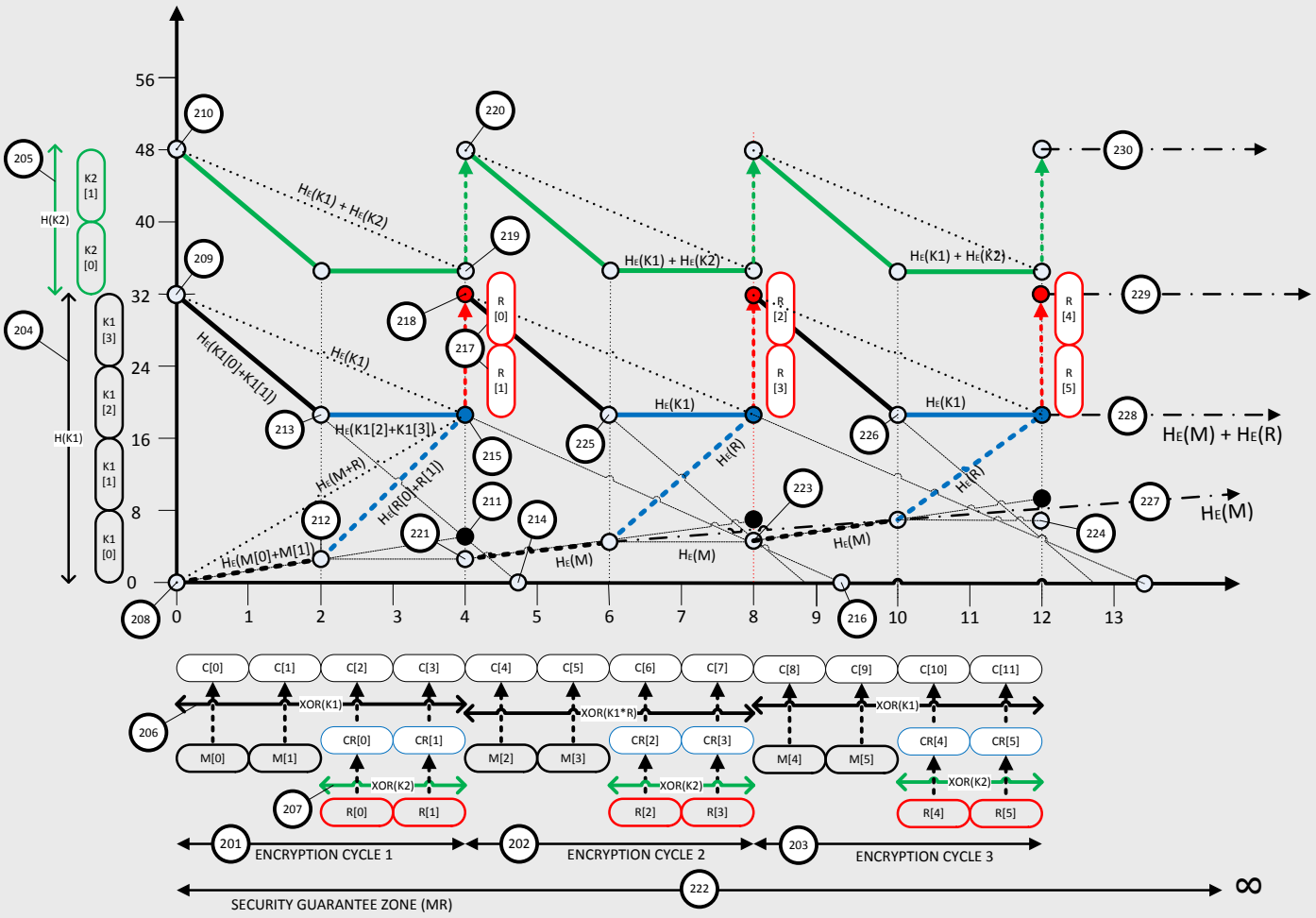

FIGURE 2: Equivocation Graph demonstrating effects of Key Equivocation engineering

From the equivocation graph above, the following observations can be made with regards to changes in the equivocation slopes of the encipherment:

- The message equivocation ***$H_E(M)$*** and key equivocation ***$H_E(K)$*** of a NINO-cipher OTP chain is demonstrated using 3 OTP encryption blocks to encrypt a 6 letter English message ***M*** with a 32-bit key ***K1***, and a 16-bit key ***K2***.
- The 32-bit key ***K1*** is used to perform the main XOR encryption of each individual 4-byte/character OTP encryption block composed of 2 ***M*** message characters and 2 ***R*** random characters as shown in step **206**. The random characters are generated on demand and need not be stored.
- For the first two ***M*** characters, the equivocation slopes (in black) are identical to those found in English messages. For the random R characters, the key and message equivocation slopes (in blue) are those found with random strings. The overall effect is that the secrecy point for the overall 4 characters is raised from point **211** to **215**.
- Thus, adding random entropy increases the average information in the joint message, extending the *unicity distance* from point **214** to **216**. In actual fact, the increase is larger,

- By increasing the information, we reduce the overall $H_E(K)$ slope and increase the $H_E(M)$ slope. The benefit of the increase in equivocation, is that it means a greater number of residual messages and keys upon a brute force attack.

The following observations can be made with regards to the increase in ***H(K)*** after every encryption step (denoted in red):

- With the *perfect secrecy* point raised at **215**, it is now possible to break through Shannon's "entropy" barrier (going higher than the initial ***H(K)***). We now use the unencrypted transmitted entropy characters **R[0]** and **R[1]** to increase the $H_E(K)$ of the encipherment, completing the chain. The simplest manner in which this can be performed is to use multiplication and discarding the least and most significant sections, but this requires solving an efficiency problem.
- The efficiency problem may be solved by mixing the new entropy with the contents of an entropy pool, from which the next OTP key is derived.
- In any event, since our total residual entropy after augmentation is now 5.2 + 16 +16 = 37.2 bits, the total number of key possibilities which can result from augmentation is $2^{32} * 2^{5.2}$. Although there is some inevitable duplication, all key possibilities should be covered in the next OTP, restoring its H(K). If we had used an OTP encryption block with 8 characters, the total would have been 10.2 + 32 +32 key possibilities. Note that this figure is increased even further after the transposition of fixed period.
- This process can be repeated ad nauseum, with every link in the OTP chain effectively resulting in the evolution of the key.

In order to successfully break through the "entropy" barrier, the new random entropy characters **R** must first be encrypted with an independent OTP secrecy system using key **K2** (as denoted in green above). This operation has the following effects:

- It raises the effective overall key entropy which must be solved, since **K2** must now be solved in addition to **K1**. The overall increase in ***H(K)*** is denoted in green.
- XOR encrypting any message twice with an OTP using two separate keys **K1** and **K2** limits the maximum key equivocation lost to one of the keys only - this is so, even if the message is known. As an example, let's use the simple OTP algebraic notation of modular addition **K1 + K2 + M = C** (Key1 + Key2 + Message = Ciphertext).
    - **K1** and **K2** are a number between 0 and 9 and M is known to be a 4.
    - If **K1**=6, **K2**=5 and **M**=4, then **C**=15 (actually 5 since 15 mod 10 = 5).
    - The attacker sees **C**=5.
    - Knowing that **M**=4, he can only deduce that **K1 + K2** = 1 or 11mod 10. He cannot deduce exactly what **K1** and **K2** are, since there are 10 possible value combinations.
- In the graph, **K2** will continue to have key equivocation, even if **K1** hits unity and is compromised. The opposite is also true. (Two independent info-theoretic security systems)

Note that every OTP in the OTP chain raises the message equivocation $H_E(M)$ higher and higher until it meets $H_E(K1)$, where after it will enter a state of equilibrium, never dropping below 18.2, even in the case of a known plaintext, rising as the key equivocation $H_E(K1)$ is augmented, and dropping when it meets the downward trending $H_E(K1)$.

Engineering Message Equivocation

A number of potential risks are still evident in the previous graph after "equivocation augmentation" has occurred.

The first risk relates to the fact that in the case where an MMRR OTP encryption has been used, it may be possible for an assailant to derive **K1** eventually if given access to the message and the random entropy. He will never gain access to **K2** or **K3**. Although this has yet to be determined scientifically, we will keep to our primary design directive that an assumption of insecurity must prevail at all times. We therefore need to hide the relative positions of the MMRR composition characters.

The second risk relates to the need to ensure that the "equivocation augmentation" of the *perfect secrecy* point is as high as possible in order to ensure the integrity of the augmentation operation.

Both risks are addressed by a single simple solution - applying a transposition over each OTP XOR encryption block. The impact on the equivocation of the NINO-cipher is demonstrated below.

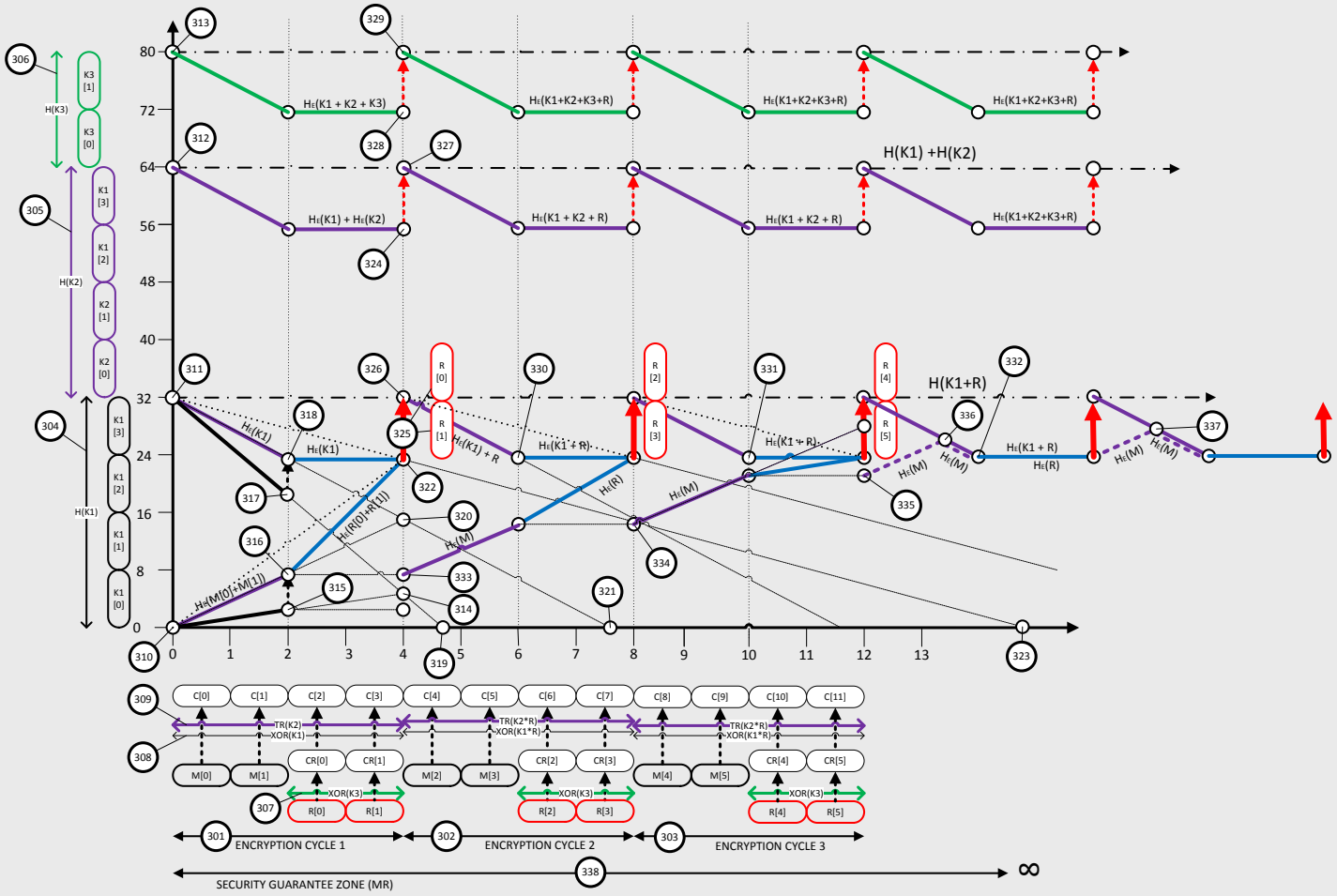

FIGURE 3: Equivocation Graph demonstrating effects of Message Equivocation engineering using a transposition of fixed period

From the equivocation graph above, the following observations can be made with regards to the impact of a transposition of fixed period on the message equivocation of the cipher depicted in the previous graph:

- A new key **K2** is introduced into the mix. This adds another level of key equivocation segregation, since **K2** and **K3** cannot be derived. It effectively "seals" the cipher.
- **K2** has the additional property of being an entropy pool of sorts, in that with 4 characters in the OTP encryption block, there are only 4! = 1x2x3x4 = 24 possible ways to "shuffle" the ciphertext using 2^{32} keys. Even if the shuffle order was known (which cannot be derived from the ciphertext alone), it would not result in a complete compromise of **K2**.
- Importantly, a shuffle introduces an important dilemma to our brute-force assailant, in that he will now have 24 possible viable avenues of pursuit, as opposed to the initial 1, only one of which is correct. This effectively increases the $H_E(M)$ and $H_E(K)$ by 4.58, the logarithm of 24. In the case where the OTP block contains 8 characters, the total avenues of pursuit increases to 40320, increasing $H_E(M)$ by 15.29. This figure prevents $H_E(K)$ from ever being reduced to 0, even where the message and the random strings are known, thus ensuring that the key **K1** cannot be fully compromised, since it is protected by **K2** due to compound or super encryption using two independent encryption algorithms.
- The additional entropy also allows for greater efficiency with regards to the "equivocation augmentation" process.
- At point **337** we note that the equivocations of the cipher have entered a state of equilibrium, such that they never attain unity. The security guarantee is therefore guaranteed.

Thus, from the equivocation graph above, it is clear that "equivocation augmentation" is at least viable from a theoretical perspective. Provided that a second independent cryptosystem is used to continually update lost entropy, the equivocation of the first cryptosystem may be prevented from ever attaining zero. To use a real world analogy – the range of a vehicle is usually limited by its fuel consumption and size of fuel tank. The ability to refuel the vehicle in motion allows for an infinite range, if infinite fuel is available.

Putting the theory into practice, forces us to deal with the practical manifestation of the concept of "independence". The very act of applying the random entropy directly to the existing key equivocation creates a dependency, since the new key **K** will be derived wholly from the new entropy and the previous key **K**. What is required, is a larger intermediary "pool of entropy". It is the augmentation of the equivocation of this entropy pool that is required. Thus, the use of a variable length entropy pool or array populated by a variable length initial key adds to the uncertainty of this entropy pool, and thereby, the equivocation of key. However, in order to defeat a super-machine, (our assumed opponent), one must create an unsolvable dilemma – this is achieved by placing a limit on the duration of any equivocation-counting process that may be used against a specific entropy pool or array. This is achieved by using two or more entropy pool arrays, one of which is "active" the others "passive". An "active" entropy pool is used to derive keys and perform "equivocation augmentation" (essentially multiplication of the active array pool contents with the new entropy). After each "equivocation augmentation", surplus bits from the active array are transferred to the "passive" array. After a while the passive array will become full, be designated as

the active pool, and the contents of the previous "active" pool are evacuated (deleted or transferred). This action then introduces an ever "expanding set" of equivocation possibilities which do not exist, making the determination of equivocation an impossible problem for a machine to solve.

At this point, it should be evident that considering the absolute security characteristics of the NINO-cipher, our strict pragmatic perspective of "absolute" security is justified, and that the NINO-cipher is unparalleled as a secrecy system, in that it is simple in design, yet post-quantum secure, easy to implement, and verifiable through scientific method. It is the first cipher to exceed the boundaries of Shannon's equivocation principle. It is verifiably immune to all forms of ciphertext/key/message cryptographic attack, and the concept of equivocation augmentation can be applied to other cryptographic primitives such as pseudo-random number generators, making them information theoretically secure. The "equivocation augmentation" principle used by the NINO-cipher represents an entirely new field of information-theoretically secure cryptography.

With regards to the question of key distribution, or the "authentication problem", as it is so often referred to. The NINO-cipher was never designed to support authentication-based "session key" distribution, since the concept is entirely unnecessary in an info-theoretically secure context. Once an initial key is distributed between parties, not further authentication/session key distribution is required. "Authentication" mechanisms may be used to affirm specific actions, such as denoting consent with signatures.

The NINO-cipher represents the solid foundation to a three-tier security system, and "digital asset management" (authentication, identify, entity, assets, groups etc.) and related matters are supported in the next tier. To this extent, there is nothing stopping existing "authentication systems" from being protected, or being integrated into the mechanics of the cipher.

A NINO-Cipher Example

In contrast to most cryptographic ciphers, the OTP-Chaining NINO-Cipher does not have a fixed length key. A key of any length may be used. It also does not have fixed operations or dimensions and other cryptographic algorithms may be used. In operation, it is possible to introduce control information into the encryption stream, which may be used to alter the operational dynamics of the cipher whilst it is in use. It may for example increase or decrease the key, the amount of entropy injected into the system, or alter the dimensions of the OTP encryption blocks used. Indeed, the cipher may be as varied as the key used. Its simple design allows for its use as a network transmission or local storage system, allowing for information-theoretically secure storage.

In the following diagram we demonstrate a simple 4-byte implementation of the NINO-cipher:

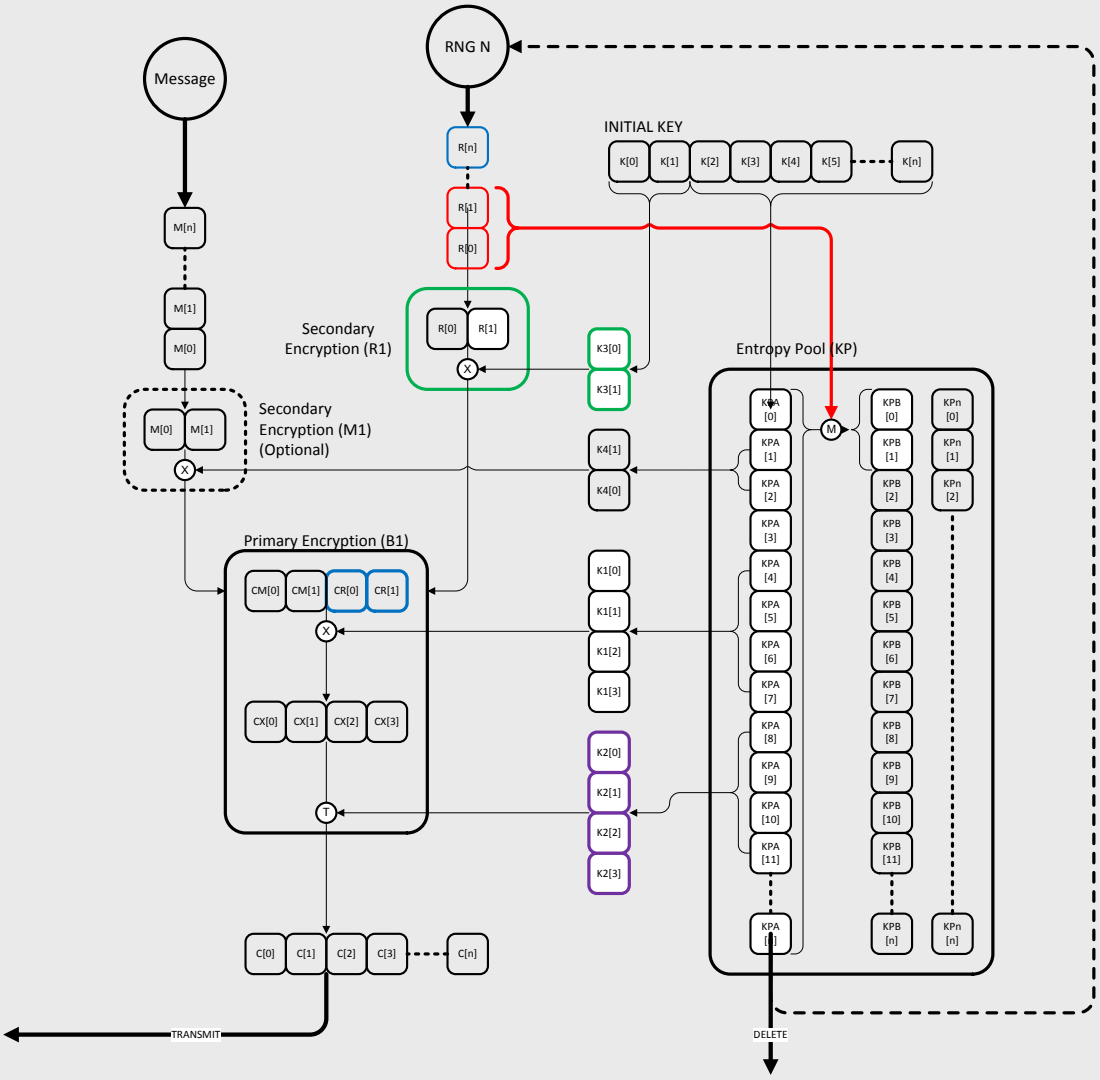

FIGURE 4: A simple 4-byte NINO-cipher implementation

From the diagram above, the following observations can be made with regards to the implementation of a NINO-cipher in encryption, the decryption operation differs in that the entropy is extracted from the decrypted message sequence:

- An entropy pool composed of two or more arrays, one designated active, is used
- An initial key **K** of unknown length is used to populate an independent key **K3** and the active entry pool array, repeating the key if necessary. The cipher key and entropy pool dimensions may be randomly determined before encryption.
- Keys **K1**, **K2** and **K4** (if required) are derived from the active array. The exact array locations may be altered dynamically.
- A message stream provides message characters and an independent RNG provides random characters.
- Random characters are encrypted with **K3** in a secondary encryption. **K3** may also be altered during operation.
- Message characters and random encrypted characters are XOR'd in a primary encryption with **K1**, **K2** being used to transpose or shuffle the resulting ciphertext.

- After each encryption block, which may be of variable size, the random entropy **R** is applied to the active entropy pool through multiplication (like a PRNG). This will increase the active array bit contents, and the least and most significant bits will be used to populate the passive array. When the passive array is full, it becomes active and the previous active array contents are evacuated out (may be used to provide entropy for an initial RNG.

The NINO-cipher differs from "standard" cipher design, in that no specific sizing is required, indeed it is actively rejected. Any form of "standardization" merely aids the assailant. All that is required, is that any NINO-cipher variant must allow for "equivocation augmentation". So variation in key size, key length, order of encryptions, size of encryption blocks, array sizes, entropy pool arrays, etc. is actively promoted. Indeed, a standard sizing may be initially used to pass configuration parameters which determine the specific configuration during encryption. In addition, the cipher may be altered whilst in motion and need not be static in design.

Practical Considerations

The NINO-cipher, whilst being a simple and practical "ideal secrecy" system composed of a chain of "perfect secrecy" OTPs, and being superior in almost all security respects when compared to current cryptosystems has two issues which require special attention:

- All messages will be extended by at most 100% of the original message length. This is the inevitable price to pay for "absolute" security (a price which must be paid with OTPs).
- The cipherstream of the cipher is especially susceptible to propagation of errors, such that a single bit error may render the ciphertext illegible. This issue may be addressed externally, by hashing the ciphertext, internally, by hashing the message, or the issue of error detection and correction may be addressed by the transmission medium, such as with TCP/IP. However, since this is the only real viable attack open to an assailant (prevention of communication), the NINO-cipher has an automatic error recovery mechanism which allows for the cipher to continue from a previous key stage. This mechanism is handled in the next security level of the overall layered security model.

Conclusion

The NINO-Cipher represents a "quantum-leap" in practical cryptographic capability, with unprecedented security characteristics which make it ideal for "quantum-secure" security environments. Numerous advantages which flowed from the research conducted during the development of the NINO-cipher include:

- Practically viable "absolute" security in a digital context.

- Simple cipher design, with unlimited adaptability – highly configurable, any number base, any key length, any message or message streams, any number of entropy pools, with unlimited potential for introducing variables allowing for morphism.
- Underlying principle of “equivocation augmentation” can be verified through scientific experimentation.
- Equivocation augmentation principle applicable to almost all cryptographic primitives which use entropy, such as RNGs, hashes etc.
- High adaptable and variable cipher design, which can be made dynamic in configuration.
- Fast due to simple mechanics, and no workload requirements.
- Can be integrated with and by current cryptosystems, securing their communication absolutely.