Binomial Sieve Series

A Prospective Cryptographic Tool

Gideon Samid
Department of Electrical Engineering and Computer Science
Case Western Reserve University
gideon.samid@case.edu

Abstract: a Binomial Sieve Series (BSS) is an infinite monotonic set of natural numbers, $b_1, b_2, ...b_n$ ($b_i < b_{i+1}$) generated, ('naturally') from any two natural numbers ($x, y \leq x$) . If one repeatedly counts b_i elements over the set X= 1,2,….x (recycled counting) and eliminates each time the element of X that stops each round of counting, then the surviving element of X is y. Every natural number, per any x, is associated with a certain survivor. We prove that per any x all BSS are infinite and approach an equal size, regardless of the identity of the survivor element y. These infinite series (in count and length) have no simple pattern, their disorder is reminiscent of primes. We suggest some intriguing cryptographic applications based on the poor predictability of the next element in each series, combined with good predictability of the computational load to develop the series (by the users and by the cryptanalyst). Using x as a shared secret, and a random, per-session, y, Alice and Bob may mark successive messages between them with the next element of the respective BSS, thereby mutually authenticating themselves throughout their conversation. Other cryptographic possibilities are outlined.

1.0 Introduction:

Mathematical insight is often acquired by regarding a known construct as a member of a set, which then attracts investigation. Accordingly, we may define a set for which the natural numbers is a member. We are seeking an abstraction which is different from the familiar sequence: integers, rationals, irrationals, and complex numbers.

We shall define a procedure that would associate any two natural numbers, x and $y \leq x$ with an infinite, rising monotonic series comprised of natural numbers: $b_1, b_2, ...b_n$ (where $b_i < b_{i+1}$ for all i=1,...(n-1), writing as:

$$[x:y] = \mathbf{b_1, b_2, ...b_n}$$

To do so we shall apply a simple sieve operation. The concept of procedural elimination of ordered element according to some rule has been made famous by Eratosthenes (Bokhari 1987). It has been applied sporadically in recent years (Chen-98, Heyde-76, Shen-99 and Telgarski-88). This abstract notion is hereby reapplied: Let x (range) and p (period) be any two natural numbers. We shall define the "initial x set" as the ordered set containing: 1,2,3,... x. We shall use a cyclical counting operation over the initial x set, namely, counting from 1,2,3,... to x, and continuing with 1,2,... -- as many times as desired. (We shall regard this counting method as 'clockwise' and the opposite direction as 'counter clockwise'). That way we shall count p numbers over 1,2… x and identify the member of the initial x set where the counting stopped. That member ("the hit") will be excluded from the initial x set, defining 'the first round x set' containing x-1 elements, and being a proper subset of the initial x set. We shall now resume counting with the next member of the first-round x set, and end up with some other member of the same set as we conclude our counting ("the second hit"). That member will also be removed, thereby defining 'the second round x set' containing x-2 members and being a proper subset of the 'first round x set'. We can repeat the process for i=1,2...(x-1), defining each round 'the i-round x set', containing x-i members. For i=x-1 the resulting '(x-1) x set' will contain one member. We shall designate this surviving member, as 'the survivor' or s, and define the above procedure as the sieve operation of order zero, writing:

$$s = S_0(x,p)$$

Clearly: $1 \leq s \leq x$. Examples: for x=50, and p=35, we get $S_0(50,35)=3$, similarly $S_0(60,9)=30$. For x=4 and p=3 we have the initial x set as 1,2,3,4; the first-x-set will be: 1,2,4; the second x set will be: 1,4, and the third x set will be: 1, so we may write: $S_0(4,3)=1$

If we consider periods, 1,2,3,…p, such that $p >> x$ then necessarily for some survivors s we shall have established a rising monotonic series: $b_1 < b_2 < .... < b_n$ such that:

$$S_0(x,b_1) = S_0(x,b_2) = \ldots\ldots S_0(x,b_n)$$

We refer to such series as binomial sieve series, or BSS. We may now introduce the following symbolism: [x:y] will mark the BSS generated over the range, x, and yielding a survivor y; [x:y](i), will designate the i-th member of the series (b_i). So we can write:

$$[x:y] = [x:y](1), [x:y](2), ...$$

We may designate for convenience: [x:y](0)=0.

Examples:

[35 : 12] = 15, 17, 215, 262, 357, 427, 459, 492...
[24 : 8] = 7, 25, 88, 115, 125, 155, 160, 178, ...
[50 : 10] = 11, 39, 139, 143, 149, 183, 239, 281...
[40 : 20] = 157, 164, 211, 250, 264, 350, 351, 458, ...

These infinite series may be defined per section. The expression [x:y]{u,v} will define the elements [x:y]{u,v}(i), [x:y]{u,v}(i+1),.....[x:y]{u,v}(j) of the series [x:y] such that:

$$[x:y]\{u,v\}(i-1) < u \leq [x:y]\{u,v\}(i)$$

and:

$$[x:y]\{u,v\}(j) \leq v < [x:y]\{u,v\}(j+1)$$

for u, v two natural numbers. So we may write: [40 : 20]{200,400} = 211, 250, 264

We shall also introduce the nomenclature of [x:y]{u//n} to indicate the section of [x:y] beginning with element i where $[x:y]\{u,v\}(i-1) < u \leq [x:y]\{u,v\}(i)$ and ending with element (i+n-1). Similarly [x:y]{u/n} will designate element (i+n-1), where i is defined as above. Enclosing a BSS section within vertical lines will imply a count of section elements. Accordingly, since:

[24:8]{80,160}=88, 115, 125, 155, 160

We shall write: 5 = |[24:8]{80,160}|

The basic theorem of binomial sieve series: For any value of x there is at least one BSS which is infinite because every natural number leaves one survivor in the finite range $1 \leq s \leq x$.

We shall now prove that every survivor defines an infinite BSS, and further prove that for a given range, x, the sizes of the various BSS (per the x survivors) approach equality when more natural numbers find their place in the range of x BSS. The basic theorem of binomial sieve series may be written as:

$$\lim_{n\to\infty} |[x{:}\, y]\{1//n\}| = \frac{n}{x}$$

for all values of x and y.

Proof: all the natural numbers of the form k_1x+1 where k_1 is any natural number don't end up with "1" as their survivor because they "sieve" it out in the first hit. Among them there are numbers of the form: $k_2(x-1)+1$, and they don't end up with "2" as their survivor. Among them there are those of the form $k_3(x-2)+1$, which don't end up with "3" as their survivor. And so on, we may point to smaller and smaller sets that don't end up with 1,2,…i as their survivor. For i=x-1 we thereby identified numbers that satisfy the following (x-2) equations:

$$k_1x+1 = k_2(x-1)+1 = \ldots k_i(x-i+1)+1 \ldots = k_{x-1}(x-x+2)+1$$

There are infinite solutions for the $k_1,k_2,\ldots k_{x-1}$ values that satisfy these x-2 equations, and hence there are infinite numbers that end up with x as their survivor.

If instead for element x, we focus on some other element s $(1 \le s \le x)$ we can repeat the above analysis with the set of numbers of the form $s + k_1x+1$, $s + k_2(x-1) +1$, … and similarly prove that there are infinite number of natural numbers that end up with s as their survivor.

The form of the (x-2) equations that must be solved in order to find the numbers that end up with some arbitrary s as their survivor are the same for all *s* values, and hence the number of natural numbers that point to *s* as their survivor is the same for all *s* values. In other words, large enough sets of natural numbers are equally distributed among the x possible survivors. This proves the basic binomial sieve series theorem.

Some obvious relationships:

[1:1] = 1,2,3,4,.... (the natural numbers)
[2:1] = 2,4,6,8,..... (even natural numbers)
[2:2] = 1,3,5,7,.... (odd natural numbers)

These natural binomial sieve series behave in a peculiar way. We may investigate them by tracking the numerical difference between successive members. Let's define the corresponding interval series:

$$[x:y]_{int} = [x:y]_{int}(1), [x:y]_{int}(2),....$$

where: $[x:y]_{int}(i) = [x:y](i) - [x:y](i-1)$

One readily lists:

$[1:1]_{int}$= 1,1,1,.... repeating sequence: {1} of size 1
$[2:y]_{int}$= 2,2,2... repeating sequence: {2}of size 1 for y=1,2
$[3:y]_{int}$ has a repeating sequence: {1,5} of size 2 for y=1,2,3
$[4:1]_{int}$ and $[4,3]_{int}$ have a repeating sequence: {2,1,9} of size 3
$[4:2]_{int}$ and $[4,4]_{int}$ have a repeating sequence: {1,2,9} of size 3

It becomes stranger yet: $[5:y]_{int}$ has a repeating sequence: {3, 1, 3, 7, 4, 4, 10, 3, 10, 4, 4, 7} of size 10, for y=1,2,3,4,5 And for [6:y] there are six distinct patterns, all of size 10:

$[6:1]_{int}$ has a repeating sequence: {26, 3, 2, 9, 2, 2, 3, 2, 9, 2}
$[6:2]_{int}$ has a repeating sequence: {17, 1, 2, 2, 11, 2, 3, 2, 11, 9}
$[6;3]_{int}$ has a repeating sequence: {13, 4, 1, 4, 11, 2, 5, 11, 4, 5}
$[6:4]_{int}$ has a repeating sequence: {13, 5, 4, 11, 5, 2, 11, 4, 1, 4}
$[6:5]_{int}$ has a repeating sequence: {17, 9, 11, 2, 3, 2, 11, 2, 2, 1}
$[6:6]_{int}$ has a repeating sequence: {26, 2, 9, 2, 3, 2, 2, 9, 2, 3}

The average gap between successive members of a binomial sieve series [x:y] is x, but the variance around this average seems to defy a clear sense of order. The sum of the repeating elements above is uniformly 60.

These binomial sieve series stand apart from the common mathematical structure built as an extension of the operation of addition, and as such they are of some non-applicative interest. In this discussion we shall focus on cryptographic aspects of these series.

2.0 Pattern-Recognition/Cryptanalysis/Compression

Given any random looking or arbitrary series expressed, say, as a decimal or similar sequence of natural numbers: r_1, r_2,..., one could construct a matching monotonic rising series, M: m_1, m_2,.... by setting:

$$m_i = m_{i-1} + r_i$$

This constructed M series may be fitted, section by section if necessary with sections of the infinite number of infinite BSS. Say:

$$M = [x_1:y_1]\{u_1,v_1\}, [x_2:y_2]\{u_2,v_2\},....$$

For example: the monotonic series M =11, 16, 20, 27, 30, 31, 64, 67, 82, 95, 109, 126 may be faithfully represented as follows:
[23:4]{10,20},[5:3]{20,31}.[45:7]{60,70},[33:11]{80,130}

The driving idea is to use as few binomial sieve series as possible. The fewer series the more pattern is extracted from the 'random looking series' and prospectively more insight, and cryptanalysis capability, is gained. The prospect of representing a random looking series as a short list of binomial sections also provides an opening to unbound compression.

The decomposition of a random looking monotonic series to binomial sections may also proceed through a linear combination of two or more sections of BSS each with the same number of elements.

3.0 Computing-Load Based Cryptography

The nominal sieve construction of the binomial series is (i) very predictable as to the effort needed, and (ii) is very controllable as to same effort. By choosing proper values for x, u

and v in the section series [x:y]{u,v} one could dictate the computing load necessary to specify the series. The computing load is proportional to the size of x, and to the span (v-u). The risk for a computational shortcut can be mitigated by introducing "switch conditions" as follows:

The above described nominal sieve operation may be modified by introducing some condition C(z), where z is the value where a counting round stopped (a 'hit'). If the conditions are satisfied, then the counting switches direction. If it were counting upward, (clockwise) it now goes downward, (counterclockwise), and vice versa. Such switch conditions will require the algorithm to execute the search for the survivor round by round (negating the possibility for mathematical shortcut – extracting the survivor without performing the many rounds). The condition itself may be controlled as to the computing load for compliance evaluation.

For example, the nominal value of [5:2]{3,9} is: 5,9. If we introduce the condition that the search direction will switch if a counting round stops at a prime number, then the result will be: $[5:2]\{3,9\}_{\text{prime switch}}$= 3,6,9. And since there is no known (or published rather) method for a quick determination of primality, it would be possible to use this method on large range, x values where the numbers marked by the counting cycle will have to be evaluated tediously as to whether they are prime or not.

Sieve computing works efficiently in dedicated hardware and firmware, and the switch conditions may be made volatile and part of the ciphersecret.

Alice and Bob will exchange the values of x,y,u, and v and use these value (along with any applicable switch conditions) to generate a shared key in the form of the at-will size of the respective binomial series. This shared key will be usable for any purpose, including a one time pad configuration. The exchanged values could be such that if exchanged at time point T_1, then Alice and Bob will need until time point T_2 to compute the full series (the shared key). They could then use the shared key to exchange short lived secrets until time point T_3. By properly choosing the values of T_1, T_2 and T_3, Alice and Bob will insure that even if their adversary will right away guess the values of x, y, u and v -- he will not have time to compute the shared key (provided the adversary uses a computing machine with known, or properly estimated performance).

Alternatively, Alice and Bob will use such values for x,y,u and v that both Alice and Bob may need to spend a well measured short time to compute the shared key. Albeit such measured

time will pose an acceptable delay. Alas, for a cryptanalyst checking a range of possibilities, this computing burden, spread over the range of options, will be prohibitive.

3.1 Cryptanalysis of Computing Load Based Cryptography of Binomial Series

The shared key may be used by Alice and Bob as pseudo one time pad key, or as a frequently changed block cipher key, etc. But for the purpose of cryptanalysis we shall assume that the shared key is exposed to the adversary, who now tries to expect the following bits of the key string.

Assuming a decimal representation of the key, and a full concatenation, it should be relatively easy for the adversary who knows that this cipher is used, to identify the rising monotonic series. So, if the exposed key looks like: 78126172177265276, then we assume that the adversary would readily list the following monotnic rising series: 78, 126, 172, 177, 265, 276, and the adversarial task ahead is to recognize that this series is generated by: [49 : 19]{50,300}.

On the face of it the adversary may start to test sieve operations beginning with the period value of 78, and checking for x=1,2,... and y=1,2,....x, ending up with a list of candidates: $[x_1:y_1]$, $[x_2:y_2]$,...$[x_k:y_k]$ that comply with the period of 78. For every x=1,2,3,.... there is some value of y such that [x:y]{78,78}=78, hence the value of k is as large as the examined x range. For all those k possibilities the cryptanalyst will compute: $[x_i:y_i]${126,126}, and discard all the empty lists. If all the lists are empty, the cryptanalyst will increase the examined x range and repeat the above until at least one list is not empty. The non empty lists will then be examined for $[x_i:y_i]${172,172}. If all the lists are empty, the cryptanalyst will have to examine higher x values, otherwise the next number in the list (177) will be checked, and so on, until the cryptanalyst finds x, y, u, and v values that are consistent with the given list. Once these values are at hand, the rest of the series can be computed and the cryptanalysis is complete -- unless there are more than one sets of x,y,u, and v that produce the same list, and the one used is different.

In this nominal case, the size of x is the main factor for the cryptanalysis burden, since x can be determined by Alice and Bob they are in a good position to anticipate the lead time they may have before their secret is compromised. Albeit, the cryptanalyst may very well estimate the

size of x. The average gap between successive numbers on the binomial list is x (dictated by the basic theorem). By computing the average gap, the size of x may be well estimated.

3.2 Countermeasures for Binomial Series Cryptanalysis:

Alice and Bob may deny the cryptanalyst the ready estimate of the size of x, and a ready sieve testing of the apparent monotonic series by using a secret linear combination of several binomial series.

Alice and Bob agree on the following secret values: x_i, y_i, u_i, f_i and n, for i=1,2,3...k. then each constructs the following linear combination: $L = l_1, l_2,...l_n$. where:

$$l_j = \sum_{i=1}^{i=k} f_i[x_i : y_i]\{u_i / j\}$$

for j=1,2,..n

So, for example: Alice and Bob agree on: $x_1 = 49$, $y_1 = 19$, $u_1 = 126$, $f_1 = 2$, $x_2=14$, $y_2=5$, $u_2 = 200$, $f_2 = 4$, n=3. They first compute the two constituent binomial series: (i) [49 : 19]{126//3} = 126, 172, 177 and (ii): [14:5]{200//3} = 210, 211, 214 then they compute the members of the linear combination series:

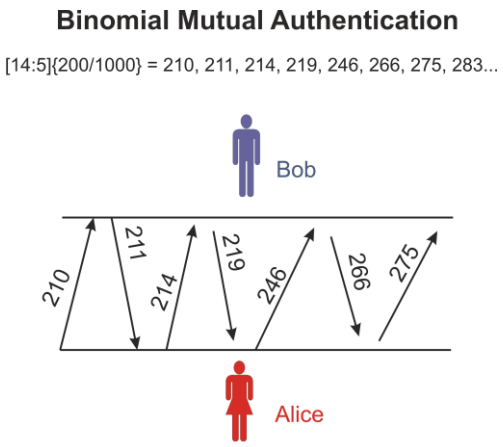
l_1 = 1092 = 2*126 + 4*210 l_2= 1188 = 2*172 + 4*211 l_3=1210 = 2*177 + 4*214

It is easy to see that the composite series will always be rising monotonic. The cryptanalyst might not suspect that the series is a linear combination and try in vain to look for a generating pair for it. Alternatively, the cryptanalyst might not know whether the series is a composite of two binomial series, or more and he will have to check a prohibitive variety of linear combinations of proper binomial series.

The linear combination option offers a means to reduce computation, without allowing brute force cryptanalysis to ease up proportionally. If Alice and Bob use a shared secret in the form of the range, x, and its size is 8 digits (to inhibit brute force attack), then the parties computation efffort will be proportional to 10^8. If the same number is interpreted as two four

digits numbers that are lined in a linear combination then the computation effort will only be proportional to 10^4 . The brute force approach will still have to contend with eight digits.

4.0 Authentication Solutions



Alice and Bob mark their bilateral messages with the next entry on their secret binomial series.

The binomial series may serve Alice and Bob to mutually authenticate one to the other by prefacing each message with the next item of a secret shared BSS. An adversary without knowledge of the [x:y] generating pair will be unable to foretell the next number on the binomial sieve series and will be unable to pretend to either Alice or Bob that he is the other party. Since the series is infinite it can be used for long term on going communication, between two centers or organizations or, say, between a merchant and a customer. Both parties will need to remember the last number they have used, and send the next number in the series for each message they send over to the opposite party. This procedure will also guard against loss, or error.

The above described bilateral authentication may be extended to multilateral mutual authentication. Alice, Bob, Carla, and David may be chatting or exchanging messages with mutual visibility. Each party will mark every one of its messages with the next entry on the binomial list. There are standard means to prevent a collision that can be used here too. The marked messages will serve as an organizer and as a protection from a hacker who might otherwise pose as a proper member of the group and participate with harmful me**ssages.**

4.1 Initial Authentication

The binomial series can also be used for initial authentication. Alice initiates a conversation with Bob. Bob then needs to identify Alice. Alice and Bob may share the range, x, value for BSS. Bob will challenge Alice with a period of choice, p. Alice will invoke the sieve operation with x and p and compute the survivor s, such that [x:s]{p,p}=p. She will then compute the next item on the list [x:s]{p/2}, and convey it to Bob as her proof. If Bob gets the right

answer (which he computes much the same) then he is confident that he is communicating with Alice, and in reply he sends the next item: [x:s]{p/3}. Alice computes the same, and if Bob sent her the right number she becomes confident that she indeed communicates with Bob. Neither one could have responded properly without knowledge of the value of x -- the shared secret that is not exposed in the mutual authentication process.

The next time around when Alice initiates a conversation, (or Bob initiates one with Alice), Bob will randomly choose another period p as a challenge, and same for the third conversation and on.

4.2 Authentication Cryptanalysis

We envision an adversary privy to the challenges and the responses, trying to deduce the value of the range, x, so he can pose as Alice to Bob the next time around.

Alice initiates the conversation. Let Bob's challenge be designated as b_1. Alice proof be designated as a_1, and Bob's proof to Alice as b_2. The adversary knows: b_1, a_1, and b_2. He now searches for a value x, so that in conjunction with $1 \leq y \leq x$ will satisfy: $[x:y]\{b_1/1\}=b_1$; $[x:y]\{b_1/2\}=a_1$; $[x:y]\{b_1/3\}=b_2$

Since every x is associated with some value y that satisfies the b_1 equation, the adversary will have to list the full range of possibilities for x (from x=1 to some indefinite high value for x). Next, the adversary will have to examine all the identified pairs (x,y) to sort out those that satisfy the a_1 equation, and among them to find the one (or perhaps more) that also satisfy the b_2 equation. If Alice and Bob conduct more conversations based on the same shared secret x, then the adversary will surely have enough equations to pin point the value of x.

It would seem that cryptanalysis is straight forward, alas the choice of the size of x may be such that the above cryptanalysis will last too long, and by the time it is complete, Alice and Bob switch to another value of x. Suppose x is chosen such that Alice and Bob will need to use 10 seconds of computing time to develop their response. Ten seconds delay seems a reasonable burden for the sake of security. Now if x is limited to be up to a size of 999,999, then the adversary using comparable computing power will need to dedicate more than 115 days (24 hours days) to exercise the above strategy. For a seven digit x, the cryptanalysis time is 1157 days. Alice and Bob will surely change their shared secret by then.

4.3 Authentication cryptanalysis countermeasures

There are quite a few countermeasures against the above described cryptanalysis:

- **linear combination**
- **clockwise-counter clockwise switching conditions**
- **skipping elements in the series**
- **random initial pattern**
- **variant end point**

Linear combinations, and switching conditions were discussed earlier. The other measures are discussed below. The implementation of these countermeasures may be done through method-key diffusion mode, explained ahead.

Skipping Elements in the series: The initial authentication procedure (as described) calls for Alice and Bob, each in his or her turn, providing the other with the next item on the respective binomial sieve list. This could be changed to any, (k-th), element on the list. Namely, given the shared secret, x, Bob will challenge Alice with p, leadidng Alice to compute the survivor $S_0(x,p)$, and then instead of proving herself to Bob by sending $[x:S_0(x,p)]\{p/1\}$, she will send $[x:S_0(x,p)]\{p/k\}$ where k is also a shared secret along with x. Bob will then prove his identity to Alice by sending her $[x:S_0(x,p)]\{p/(k+j)\}$, where j is also part of the shared secret. This add-on will increase the brute force analysis effort facing the adversary.

Random Initial Pattern: for a range, x, one could attach an x bits long "mask" bit string such that any number from 1 to x corresponds to a single bit on the mask such that any number (1,2,...x) corresponding to a binary digit of value zero will be eliminated a-priori, before the sieve operation begins. This mask will be part of the shared secret designed to further complicate the task of the cryptanalyst.

Example: Nominally we have: $S_0(4,3)=1$, but if we mask it with binary "11" (1011), then the sieved range changes from 1-2-3-4 to 1-3-4 and the result $S_{masked}(4,3)=2$.

Variant End Point: The nominal sieve operation concludes with the last survivor of the initial range. We can in turn stop the sieve operation earlier. We can stop it in the next to last surviving number, or even earlier. We can run the sieve operation with a condition to stop it when the counting of the period hits a prime number, or a number of any property, or

altarnatively to stop it after a certain number of cyclical rounds. Such variability will be part of the shared secret, also designed to further complicate the task of the cryptanalyst.

4.4 Method/Key Diffusion Mode

Traditionally cryptography has developed with a clear distinction between the method, (procedure, algorithm), and the cryptographic key. The former was in the open, subject to review and analysis and the latter comprised the complete and full secrecy of the operation. Recent thinking challenged this tradition (e.g. Samid 2001, YouDeny.com, US Patent 6,823,068), and introduced a fuzzy line between the two. While he strives to identify the key in order to crack a cryptogram, it is equally necessary to identify the method used. The key itself may be uncovered using 'brute force' but an ingenious method defies exposure by a 'dumb computer' and makes the cat-and-mouse 'game' between cryptographer and cryptanalyst into a pure race of imagination and creativity. All the countermeasures discussed here against an aggressive cryptanalysis may be activated or de-activated by a special data element that is part of the 'secret key'. It is true about the bit mask, the end point variant, the clockwise/counterclockwise shifting, etc.

4.5 Hierarchy Oriented Authentication:

The binomial series can be used to manage communication security in a hierarchical organization where upper echelon people need to communicate without exposure to lower echelon individuals. We shall use the linear combination method. We shall describe an organizational configuration where the top echelon is level 0, the one below is level 1, then further down level 2, until level t -- the bottom one. We shall associate a shared echelon secret in the form of the BSS generating x and y. This will define x_1, x_2, ... x_t.and y_1, y_2, ... y_t. Members of each echelon will be privy to all the shared secrets in all the echelon below them. When members of echelon i communicate with each other they will use a linear combination formula that includes x_i, x_{i+1}, ...x_t, and y_i, y_{i+1}, ...y_t and where the k-th element of the combined series is computed as:

$$l_k = \sum_{j=1}^{j=t-i+1} f_j[x_j : y_j]\{1/k\}$$

Any communicating party will use the next item in line in the combined list, and each will check that the communication they receive from fellow echelon members carries the appropriate marker from the combined series. Members of echelon i are unaware of x_{i-1} and y_{i-1} and hence cannot participate in conversations among the higher i-1 echelon. But they can participate in any conversation running among lower echelon agents. To address a higher echelon person, the lower echelon agent will use his known secrets, which are known upwardly, and a conversation may be struck using the same or different message counter.

5.0 Closing Notes:

Binomial Sieve Series are infinite in the third power: each series is infinite, for each range (x) there are x series, and x is any natural number. They are generated in a 'natural' way (very little arbitrariness), and their apparent lack of order is reminiscent of the disorderly appearance of primes in the scale of natural numbers (is there a BSS that generates a long enough section of the primes?). Intuitively BSS represent an enticing subject matter for research. But even now, when the sequence shown by BSS is a puzzle, they appear to offer an effective cryptographic tool, which may find its place in the growing tool box of mathematical means to face the challenge of handling a mutually mistrustful situation.

Reference:

Bokhari S. 1987 " Multiprocessing the Sieve of Eratosthenes" Computer 1987 pp50 58

Chen, X. 1998 "Sieve Extremum Estimates for Weakly Dependent Data" Econometrica, Vol. 66, No. 2 pp. 289-314

Heyde C. 1976 "On Asymptotic Behavior for the Hawkins Random Sieve" Proceedings of the American Mathematical Society, Vol. 56, No. 1 pp. 277-280

Samid, G. 2001 "Re-Dividing Complexity Between Algorithms and Keys (Key Scripts)" The Second International Conference on Cryptology in India, Indian Institute of Technology, Madras, Chennai, India. December 2001.

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

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.

Shen X. 1999 "Random Sieve Likelihood and General Regression Models" Journal of the American Statistical Association, Vol. 94, No. 447 pp. 835-846

TelgÃ¡rsky, R. 1988 "Complete Exhaustive Sieves and Games" Proceedings of the American Mathematical Society, Vol. 102, No. 3 pp. 737-744 1988

US Patent 6,823,068