Welcome!

Attached, you will find the research paper first published at http://ePrint.IACR.org/2014/894 that introduces the mathematics behind the UberCrypt Framework (UCF), a new approach to cryptosystems.

This completely transparent mathematical text details the underlying math of the system. It asks some questions about the security and strength of the system, and introduces some interesting new math assertions. We request your thoughtful feedback.

We often hear questions like: “Why should I bother reviewing this paper?” or “What are the compelling advantages of this system over what exists today?” or “What is it about existing systems that is broken (while not admitting that anything is broken), that this is fixing?” So, while a few paragraphs cannot fully answer these questions, we thought we’d try to touch upon them briefly as a way to start that dialog.

It is vital to note that existing cryptosystems and ciphers like ECC, RSA, AES, RC4 and others are robust, elegant and currently unbroken solutions. There is no intention to in any way criticize these legendary systems.

Rather, the UCF represents an extension of the crypto-paradigm. It is a framework with all the characteristics typically associated with that term. It is a super-structure that is malleable while still maintaining integrity. It is an extensible framework that incorporates many inputs and can be used for many processes and outputs. It is a cryptosystem, but also more. This framework is enabled by a tight coupling of a dynamic three dimensional geometric form with an amorphous two dimensional “entropy field”. These are the abstract descriptors, what of the characteristics and/or behaviors?

Here are a few characteristics or behaviors of the system:

1) It falls into the class of cryptosystems called symmetric key stream ciphers (though an experimental block cipher mode does function).
2) It is a variable strength system with a granularity of 1 bit. But this understates that point. The strength of the system is a minimum of 366 bits and a theoretical maximum of 381,554 bits (though the reference software maxes out at 2930 bits). These numbers address the quantity of strength, but not the quality of it. The UCF allows users to “couple” any encryption to any number of logical, physical or virtual “objects.” This adds quantitative and qualitative strength and security for each object that is “coupled.” This creates new opportunities for users and system administrators.
3) Though the UCF is a variable strength system – the performance of its cipher is constant. At 366 bits or 3666 bits, the cipher speed is the same.
4) Because the UCF is built on a dynamic 3D geometric form, it supports both a standard form that allows all users to interoperate, but also easily supports custom and proprietary forms.
5) The cipher process itself is capable of sharding bytes while ciphering.

Perhaps one way to see these characteristics is within a comparative context with other algorithms.

Characteristic	UCF	RC4	AES	Blowfish	RSA	ECC
Stream Cipher	✓	✓				
Fixed Block Cipher			✓	✓		
Customizable/Dynamic Block Cipher	✓					
Asymmetric Key	?				✓	✓
Symmetric Key	✓	✓	✓	✓		
Stratified Strength		✓ 1684	✓ 128 192 256	✓ 128 to 448 (every 64)	✓ 512, 1024, 2048, 4096	✓ 192, 224, 256, 384, 521
Dynamic Granular Strength	✓ 366 to 381554 (1)					
Cipher Speed Relative to Strength	**Constant** 8 cpu cycles/ byte	**Constant** 7 cpu cycles/ byte	Inverse	Inverse	Inverse	Inverse
User Selectable Strength Couplers	✓					
Byte Sharding	✓		?			
Non-Inflationary Ciphertext	✓	✓				
Customizable Bit Generator (PRGA/DRBG)	✓ (unique 2D & 3D geometries & ‘entropy fields’.)					✓ (unique curves, limits, etc.)
Max Key Stream Period (in bytes, power of 10)	1054	9 (after which patterns appear)	41 (in CTR mode)	136 (in CTR mode)		
Authentication	✓		✓		✓	✓

As this table illustrates, the UCF represents the most dynamic and flexible algorithm, with the widest range of strength (quantitatively and qualitatively), the longest key stream periodicity and (near) fastest cipher speed. However, these high level statements do not adequately address the more subtle nuances of the system that ultimately give rise to new use-cases and ensuing benefits. For these, an in-person briefing is more useful.

With that said, as RC4 is the only other stream cipher on this table, it may be useful to briefly mention a few advantages in favor of the UCF. RC4 has some known weaknesses and is slowly being deprecated by major vendors in web browsers and other tools. This is primarily because of weaknesses in its key schedule (and non-integrated nonce-ing), but also patterns in the resulting cipher key stream. The UCF has a much stronger and more flexible “key schedule” algorithm that results in a larger and more dynamic “entropy field” – but while maintaining the same class of constant high speed performance.

Contact Joe Chiarella at +1 717.610.1119 or JChiarella@ColloidLLC.com to arrange a briefing on the UCF.

THE UBERCRYPT FRAMEWORK: A NEW APPROACH IN CRYPTOSYSTEMS

JOE CHIARELLA

Colloid LLC, Harrisburg, PA

GREG MOSHER

Colloid LLC, Atlanta, GA

J. ROBERT BUCHANAN

Department of Mathematics, Millersville University, P.O. Box 1002, Millersville, PA 17551

Abstract. This article describes a novel and unique cryptosystem making use of a small set of private and public initialization values to produce an infinite, pseudorandom byte stream which can be used as a one-time stream cipher for securing communication between parties and for data archival. The cryptosystem makes use of geometry and number theory to generate the byte stream and is extensible in that additional private authentication factors can supplement the initialization values. The article discusses the design and operation of the system and lists many potential questions of interest to the community of mathematical and cryptological researchers. Foremost among these questions are determining the most appropriate method for assessing the cryptographic strength of the algorithm and determining any weaknesses in the security of the algorithm.

E-mail addresses: JChiarella@ColloidLLC.com.

1. Introduction

The need for secure communication between parties and for securing data from unauthorized access is increasingly important in areas such as email, banking, data archival, entertainment, and others. In the past cryptosystems such as the Data Encryption Standard (DES, [14]) and Content Scramble System (CSS [2]) have served with varying degrees of success. At present the RSA [10] algorithm and the Advanced Encryption Standard (AES [7]) are well received among practitioners of cryptology, government, private industry, and the general public. RSA is computationally intensive for the communication or protection of large volumes of data and has a number of known vulnerabilities including timing attacks [3]. AES has been successfully implemented in modern computer CPUs. This article describes a novel cryptosystem which includes aspects of geometry and number theory in order to create a key stream which can be used as a one-time stream cipher for cryptographic purposes.

Due to the novel nature of this cryptosystem, the authors have included several lists of questions and conjectures about the mathematical and cryptographic properties of the system. It is hoped that by familiarizing the cryptological community with this innovative algorithm, experts will provide answers or insight into these questions and conjectures. The remainder of this article includes a high-level description of the function of the cryptosystem. The development of a key stream which serves as a one-time stream cipher can be broken into two basic activities. The first is the creation of a list of large prime integers from authentication factors and a three-dimensional geometric figure which is constructed from a set of private and public initialization values (Sec. 2). The second is the generation of a pair of matrices whose elements can be thought of as bytes and from which the bitwise exclusive OR operation will produce a key stream (Sec. 3). Each of these sections ends with a list of relevant questions and conjectures for the cryptological community to consider. Sec. 4 outlines some attempts at measures of the strength of this cryptosystem and summarizes some of the challenges faced by an eavesdropper on communications secured by this cryptosystem. The discourse in Sec. 5 concludes the article with some discussion of ways in which the cryptosystem could be generalized further so as to defeat any successful attacks on its current form.

The cryptosystem to be described has been successfully implemented in computer code across a variety of computer operating systems. Readers interested in testing the computer program or in seeing its output should visit the URL at the end of this article.

2. Generation of Prime Numbers

In this section the process used to generate a list of prime numbers from a set of initialization values will be described. The set of initialization values is small; at its simplest consisting of an ordered triple of three private initialization values and a single public initialization value. From the private initialization values a two-dimensional geometric figure is derived. The public initialization value derives a three-dimensional figure from the two-dimensional figure. From the geometry of the three-dimensional figure, a list of prime numbers will be generated. The list of primes thus generated will be used in Section 3 to create a key stream which can be used for encryption or decryption. The remainder of this section describes the process of prime integer generation in more detail and ends with a subsection enumerating open questions about the mapping of the three-dimensional geometric figure to the primes and properties of the primes generated.

2.1. **Private Initialization Values and the 2D Geometrical Form.**

Consider two parties, referred to here as Alice and Bob, who wish to communicate privately with one another. They have securely exchanged an ordered triple of numbers denoted (c, α, m). The first component is an integer for which $2^{256} \leq c \leq 2^{1160}$. The second and third components are real numbers where $\pi/12 \leq \alpha \leq 5\pi/12$ and $4 \leq m \leq 65535$. The parameter c can be interpreted as the length of an edge of a triangle while α is the included angle between the edge of length c and an edge of length $b = c\,m$. From these three initialization values the two parties will each create the same obtuse triangle in the Euclidean plane. In the following explanation line segments joining distinct points P_1 and P_2 will be denoted $\overline{P_1P_2}$. The length of line segment $\overline{P_1P_2}$ will be denoted as $\|\overline{P_1P_2}\|$ and will be calculated as the usual Euclidean distance between the points. Without loss of generality the edge of length c will be parallel to the x-axis in the plane and its endpoints will be denoted A and B. The adjacent side of length b will have one endpoint at A, make an angle of α with $\overline{AB}$, and will have its other endpoint at C. For the sake of simplicity define $a = \|\overline{BC}\|$. Figure 1 illustrates $\triangle ABC$. Several other points in the plane of $\triangle ABC$ will become important to the derivation. Various triangle centers will

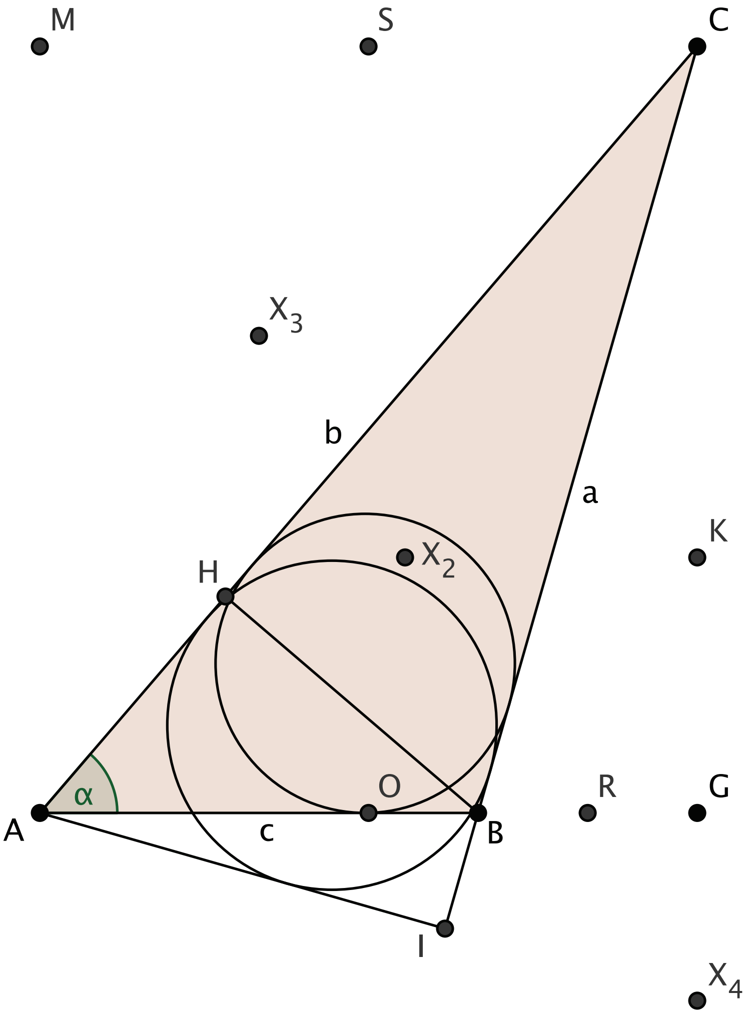

FIGURE 1. The two-dimensional geometric form constructed from the private initialization values.

be mentioned and will be denoted using the nomenclature of Kimberling [11]. Let point X_2 be the centroid (intersection of the three triangle medians, [12]) of $\triangle ABC$. Let point X_3 be the center of the circumcircle (unique circle passing through the vertices, [9]) of $\triangle ABC$. Finally let X_4 be the orthocenter (intersection of the three triangle altitudes, [5]) of $\triangle ABC$. The height of $\triangle ABC$ considering $\overline{AC}$ as its base is represented in Fig. 1 as $\overline{BH}$. Using Heron's formula [4] to calculate the area Δ_{ABC} of $\triangle ABC$, then

$$\varnothing_2 = \|\overline{BH}\| = \frac{2\Delta_{ABC}}{\|\overline{AC}\|}. \tag{1}$$

As a property of the orthocenter, $\overline{AX_4}$ is perpendicular to $\overline{BC}$. Define point I to be the intersection of the line through points B and C with line segment $\overline{AX_4}$. As a consequence $\triangle ACI$ is a right triangle. Letting s_{ACI} represent the semiperimeter of $\triangle ACI$ and Δ_{ACI} its area, the diameter of the incircle (circle tangent to each of the sides, [17]) of $\triangle ACI$ is

$$\varnothing_3 = \frac{2\Delta_{ACI}}{s_{ACI}}. \tag{2}$$

Figure 1 also illustrates the incircle of $\triangle ABC$. If s_{ABC} represents the semiperimeter [6, pp. 113–132] of $\triangle ABC$ then the diameter of the the incircle is

$$\varnothing_4 = \frac{2\Delta_{ABC}}{s_{ABC}}. \tag{3}$$

Let line segment $\overline{AG}$ be the projection of $\overline{AC}$ along $\overline{AB}$. The point labeled M is located at the intersection of the line through A perpendicular to $\overline{AB}$ with the line through C parallel to $\overline{AB}$. Point S is the midpoint of the line segment joining C and M. Let points O and R be the midpoints of $\overline{AG}$ and $\overline{BG}$ respectively. Point K is the intersection of the line through X_2 parallel to $\overline{AB}$ with the line segment $\overline{CG}$. Consequently Alice and Bob having the same ordered triple (c, α, m), construct the same geometric figure.

2.2. **Public Initialization Values and the 3D Geometrical Form.** The public initialization value is an integer n with $10^8 \leq n < 10^9$. When operated upon to produce several values, it is combined with the two-dimensional geometrical object described in Sec. 2.2 to create a three-dimensional geometrical object. In this section the operations performed on n and the construction of the three-dimensional object are described.

The sequence of digits of an irrational number will frequently be calculated and the following functions are defined to describe how these digits are determined. For any prime integer q, define

$$f(q) = \sqrt{q} - \lfloor \sqrt{q} \rfloor. \tag{4}$$

Note that $f(q) \in (0, 1)$ and since the square root of a prime integer is an irrational number, $f(q)$ is always irrational. From such an irrational number an integer with a prescribed length (number of hexadecimal digits) must be extracted. If $f(q)$ is expressed in a mantissa-exponent format of the form

$$f(q) = 0.d_1 d_2 d_3 \cdots \times 10^{-b_0}$$

where $d_1 \neq 0$ and $b_0 \geq 0$, then define function $F(q; N)$ for q prime and $N \in \mathbb{N}$ as

$$F(q; N) = \lfloor f(q) \times 10^{b_0 + \lfloor N \log 16 \rfloor} \rfloor = d_1 d_2 d_3 \cdots d_{\lfloor N \log 16 \rfloor}. \tag{5}$$

For the sake of notation let $\lceil x \rceil_{\mathbb{P}}$ denote the smallest prime number greater than real number x. Referring to $\triangle ABC$, let $q_1 = \lceil b \rceil_{\mathbb{P}}$ and then let $p = \lceil q_1/n \rceil_{\mathbb{P}}$. As constructed, q_1 is a prime in excess of $2^{258} \approx 4.6 \times 10^{77}$. Using Eq. (5) define integer $d = F(p; 4050)$. Since the hexadecimal representation of d will be used so frequently, let the notation $(d)_{16}$ denote d in base-16, where

$$(d)_{16} = (F(p; 4050))_{16} = h_1 h_2, h_3 h_4, h_5 h_6, \cdots, h_{4049} h_{4050} \tag{6}$$

where in Eq. (6) the hexadecimal digits have been grouped into pairs separated by commas for convenience. The paired digits will be used to populate a 45×45 matrix, H (which should not be confused with point H located on $\overline{AC}$). Matrix H is filled starting at its center (the (23,23) entry) in clockwise fashion (the transpose of the Ulam spiral [18]).

$$H = \begin{bmatrix} h_{3873}h_{3874} & & & \cdots & & & h_{3961}h_{3962} \\ & \ddots & & & & \iddots & \\ & h_{33}h_{34} & h_{35}h_{36} & h_{37}h_{38} & h_{39}h_{40} & h_{41}h_{42} & \\ & h_{31}h_{32} & h_{9}h_{10} & h_{11}h_{12} & h_{13}h_{14} & h_{43}h_{44} & \\ \vdots & h_{29}h_{30} & h_{7}h_{8} & h_{1}h_{2} & h_{15}h_{16} & h_{45}h_{46} & \vdots \\ & h_{27}h_{28} & h_{5}h_{6} & h_{3}h_{4} & h_{17}h_{18} & h_{47}h_{48} & \\ & h_{25}h_{26} & h_{23}h_{24} & h_{21}h_{22} & h_{19}h_{20} & h_{49}h_{50} & \\ & \iddots & & & & \ddots & \\ h_{3785}h_{3786} & & & \cdots & & & h_{4049}h_{4050} \end{bmatrix} \tag{7}$$

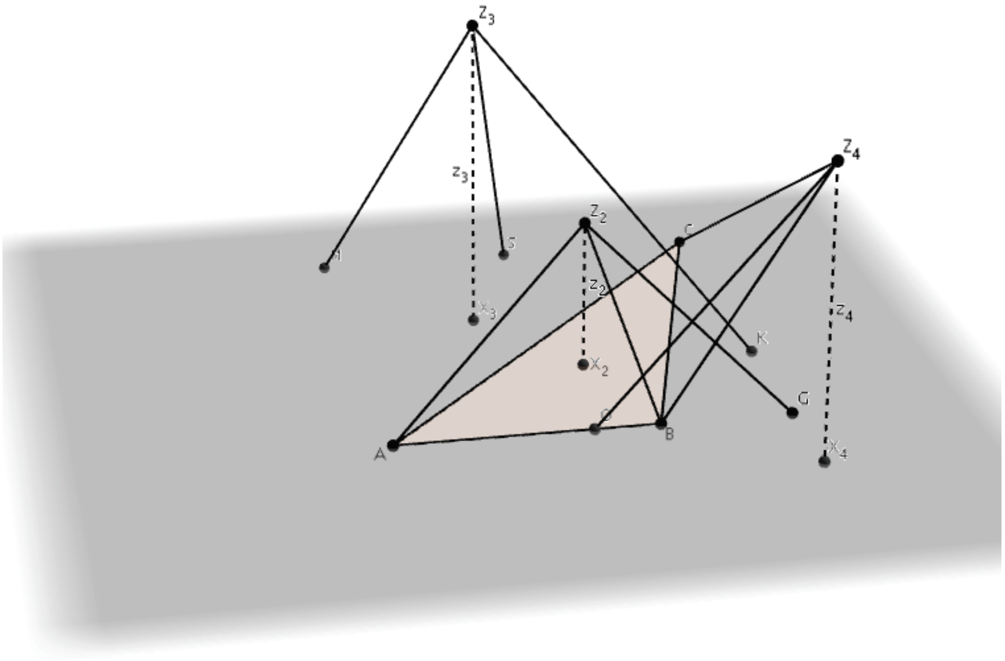

FIGURE 2. The three-dimensional geometric form constructed from the private initialization values. The altitudes and hypotenuses of several right triangles perpendicular to the plane of $\triangle ABC$ are illustrated, though not all of the hypotenuses mentioned in Sec. 2.3 are shown.

When convenient, each entry of matrix H can be thought of as a nonnegative integer less than 256. Let $s = 1 + [d_1d_2d_3d_4d_5 \pmod{45}]$ and $t = 1 + [d_6d_7d_8d_9d_{10} \pmod{45}]$ where d_j refers to the jth digit of $d = F(p; 4050)$. $H_{s,t}$ is the entry in the sth row and tth column of matrix H. If $H_{s,t}$ is even then let k be the integer represented by the concatenation of the five entries on row s of matrix H beginning in column t (wrapping around the matrix if necessary). Otherwise if $H_{s,t}$ is odd let k be the integer represented by the concatenation of the five entries in column t of matrix H beginning in row s (again, wrapping around if necessary). For example if $s = 22$ and $t = 21$ and if $H_{22,21}$ is even then

$$(k)_{16} = h_{31}h_{32}, h_9h_{10}, h_{11}h_{12}, h_{13}h_{14}, h_{43}h_{44}$$

while if $H_{22,21}$ is odd then

$$(k)_{16} = h_{31}h_{32}, h_{29}h_{30}, h_{27}h_{28}, h_{25}h_{26}, h_{59}h_{60}.$$

Define the real number m_2 as

$$m_2 = \left(\left[15 \times 10^6\right] + \left[k \pmod{70 \times 10^6}\right]\right) \times 10^{-8}. \tag{8}$$

Now letting $s = 1 + [d_{11}d_{12}d_{13}d_{14}d_{15} \pmod{45}]$ and $t = 1 + [d_{16}d_{17}d_{18}d_{19}d_{20} \pmod{45}]$ and following the same procedure as just described determine real number m_3. Finally letting $s = 1 + [d_{21}d_{22}d_{23}d_{24}d_{25} \pmod{45}]$ and $t = 1 + [d_{26}d_{27}d_{28}d_{29}d_{30} \pmod{45}]$ determine m_4. The three real numbers m_2, m_3, and m_4 will lie in the unit interval.

With the three real number multipliers just generated from the public initialization value, a three-dimensional geometrical object can be constructed. See Fig. 2. Assuming $\triangle ABC$ and its associated points described in Sec. 2.1 lies in the $z = 0$ plane of $\mathbb{R}^3$, let Z_2 be the point directly above X_2 at altitude $z_2 = m_2\varnothing_2$. Likewise Z_3 will be the point directly above X_3 at height $z_3 = m_3\varnothing_3$ and Z_4 will be directly above X_4 at altitude $z_4 = m_4\varnothing_4$. Just as in the case of the two-dimensional geometric form, if Alice and Bob use the same private and public initialization values, they construct the same three-dimensional geometric figure.

2.3. **Mapping the 3D Geometrical Form to Primes.** Once the three-dimensional geometric form is constructed, the procedure described in this section is used to generate a set of fifteen prime integers.

The smallest primes greater than the distance from point Z_2 to points A, B, C, G, and X_3 form the first set of five primes. The smallest primes greater than the distance from point Z_3 to points C, K, M, S, and X_4 form the next five primes. The final set of five primes are the smallest primes greater than the distance from point Z_4 to points B, C, M, O, and X_2. Using the notation developed earlier,

$$p_{A,Z_2} = \lceil \|\overline{AZ_2}\| \rceil_{\mathbb{P}} \tag{9}$$

would be the smallest prime greater than the Euclidean distance from point A to point Z_2. Thus from the private and public initialization values, ultimately Alice and Bob derive the same fifteen primes summarized below.

$$\left\{ \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \\ p_6 & p_7 & p_8 & p_9 & p_{10} \\ p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \end{array} \right\} = \left\{ \begin{array}{ccccc} \lceil \|\overline{AZ_2}\| \rceil_{\mathbb{P}} & \lceil \|\overline{BZ_2}\| \rceil_{\mathbb{P}} & \lceil \|\overline{CZ_2}\| \rceil_{\mathbb{P}} & \lceil \|\overline{GZ_2}\| \rceil_{\mathbb{P}} & \lceil \|\overline{X_3Z_2}\| \rceil_{\mathbb{P}} \\ \lceil \|\overline{CZ_3}\| \rceil_{\mathbb{P}} & \lceil \|\overline{KZ_3}\| \rceil_{\mathbb{P}} & \lceil \|\overline{MZ_3}\| \rceil_{\mathbb{P}} & \lceil \|\overline{SZ_3}\| \rceil_{\mathbb{P}} & \lceil \|\overline{X_4Z_3}\| \rceil_{\mathbb{P}} \\ \lceil \|\overline{BZ_4}\| \rceil_{\mathbb{P}} & \lceil \|\overline{CZ_4}\| \rceil_{\mathbb{P}} & \lceil \|\overline{MZ_4}\| \rceil_{\mathbb{P}} & \lceil \|\overline{OZ_4}\| \rceil_{\mathbb{P}} & \lceil \|\overline{X_2Z_4}\| \rceil_{\mathbb{P}} \end{array} \right\}$$

The choice of fifteen primes generated in this way is arbitrary. The process could generate more prime integers by calculating additional line segment lengths, or fewer primes by omitting some lengths listed above, or different planar points and circle centers could be used.

2.4. **Generalizations and Extensions.** In this section three generalizations of the derivation of the set of prime integers will be described. The first generalization, which will be called **iteration**, enables the generation of additional sets of fifteen prime numbers (or any other desired increment) in a manner similar to that described in Sec. 2.2. The second method of generalization, named **translation**, involves a modification of the location of points X_j and Z_j for $j = 2, 3, 4$. The two forms of generalization in the construction of the three-dimensional geometric figure can be used individually or in concert. The third generalization involves the use of **additional authentication factors** to generate prime integers. Additional authentication factors are any source of information which can be mapped to a number.

The process called iteration is carried out by constructing one or more triangles similar to $\triangle ABC$ (in the same plane as that triangle). In Fig. 3 the additional triangle is $\triangle LEX_3$ and is constructed as follows. Points D and E are the midpoints of $\overline{AC}$ and $\overline{BC}$ respectively. Point R is the intersection of the line through points A and B with the perpendicular line through point E. Finally point L is the intersection of the line through the circumcenter X_3 and point D with the line through points E and R.

Claim. *$\triangle LEX_3 \sim \triangle ABC$.*

Proof. The perpendicular bisectors of a triangle intersect at the triangle's circumcenter. Thus $\overline{LX_3} \perp \overline{AC}$ and $\overline{EX_3} \perp \overline{BC}$. Consequently $\angle X_3LE$ is congruent to $\angle CAB$. Likewise, since $\overline{AB} \perp \overline{EL}$ then $\angle LEX_3$ is congruent to $\angle ABC$. Therefore $\triangle LEX_3$ is similar to $\triangle ABC$. □

$\triangle LEX_3$ possesses its own centroid, circumcenter, orthocenter, and other points corresponding to those associated with $\triangle ABC$ and described in Sec. 2.1. Following the procedure for constructing the three-dimensional geometric figure described in Sec. 2.2 an additional set of fifteen primes can be generated using $\triangle LEX_3$. To continue generating more prime integers, this process can be repeated by constructing a similar triangle based on $\triangle LEX_3$. Therefore if the set of private initialization values is thought of as $((c, \alpha, m), i)$ where $i \in \mathbb{Z}^*$ specifies the number of repetitions of prime integer generation procedure, then $15(i+1)$ primes can be produced.

The method called translation adjusts the locations of the points X_j and Z_j for $j = 2, 3, 4$. To avoid confusion, the adjusted points will be denoted X'_j and Z'_j and the original (un-primed) symbols will refer to the points described in Sections 2.1 and 2.2.

Two additional private initialization values, denoted (r, x_s), are used to locate the points X'_j and Z'_j where $j = 2, 3, 4$. These values are real numbers satisfying the inequality $0 < x_s < r < \sqrt{2}/2$. Define

$$n_{j,x} = \left\lfloor \frac{\varnothing_j}{x_s \times 10^7} \right\rfloor \tag{10}$$

$$n_{j,y} = \left\lfloor 1 + \frac{100 m_j}{r} \right\rfloor \tag{11}$$

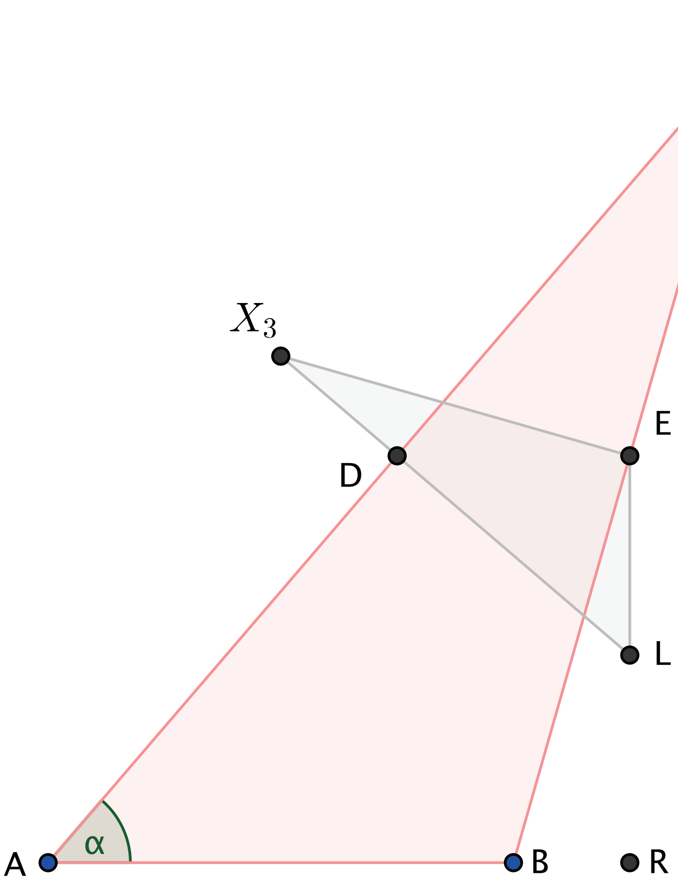

FIGURE 3. $\triangle LEX_3$ is similar to $\triangle ABC$ constructed from the private initialization values.

Let X_j' be the point in the plane of $\triangle ABC$ which is displaced from point X_j by the vector

$$\mathbf{u}_j = \left\langle (-1)^{n_{j,x}} x_s z_j, (-1)^{n_{j,y}} z_j (r^2 - x_s^2)^{1/2} \right\rangle . \tag{12}$$

Since $0 < x_s < r$ the components of vector $\mathbf{u}_j$ are real. Point Z_j' lies above the point X_j' at an altitude of $z_j(1 - r^2)^{1/2}$. Once positioned, points X_j' and Z_j' are used in place of X_j and Z_j to calculate a set of fifteen prime numbers as described in Sec. 2.3.

Incorporating both of these generalizations requires a private initialization value of the form $((c, \alpha, m), i, (r, x_s))$. The original ordered triple of private initialization values (c, α, m) can be thought of as the special case $((c, \alpha, m), 0, (0, 0))$. Iteration without adjusting the points X_j and Z_j is specified by a private initialization value of the form $((c, \alpha, m), i, (0, 0))$, while repositioning the points X_j and Z_j without iteration is specified by $((c, \alpha, m), 0, (r, x_s))$.

The third extension of the process for generating prime numbers involves additional authentication factors. A simple example of an additional authentication factor is a passphrase or password. Other authentication factors may include a file resident on a computer, the media access control (MAC, [1]) address of a computer, global positioning system (GPS) coordinates, and others. For each of these authentication factors the process by which a prime integer is generated is the same and will be outlined for the case in which the private initialization value i (the iteration counter) is zero. Generalization to the cases in which $i > 0$ are straightforward. Referring to the three-dimensional geometric form described in Sec. 2.2 the following fifteen integers (not necessarily prime) can be found.

$$\left\{\begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \\ j_6 & j_7 & j_8 & j_9 & j_{10} \\ j_{11} & j_{12} & j_{13} & j_{14} & j_{15} \end{array}\right\} = \left\{\begin{array}{ccccc} \lfloor \|\overline{AZ_2}\| \rfloor & \lfloor \|\overline{BZ_2}\| \rfloor & \lfloor \|\overline{CZ_2}\| \rfloor & \lfloor \|\overline{GZ_2}\| \rfloor & \lfloor \|\overline{X_3Z_2}\| \rfloor \\ \lfloor \|\overline{CZ_3}\| \rfloor & \lfloor \|\overline{KZ_3}\| \rfloor & \lfloor \|\overline{MZ_3}\| \rfloor & \lfloor \|\overline{SZ_3}\| \rfloor & \lfloor \|\overline{X_4Z_3}\| \rfloor \\ \lfloor \|\overline{BZ_4}\| \rfloor & \lfloor \|\overline{CZ_4}\| \rfloor & \lfloor \|\overline{MZ_4}\| \rfloor & \lfloor \|\overline{OZ_4}\| \rfloor & \lfloor \|\overline{X_2Z_4}\| \rfloor \end{array}\right\}$$

If x represents SHA-256 digest of the value or contents of the authentication factor, then an additional prime associated with the authentication factor is

$$\hat{p} = \left\lceil F\left(\lceil x \rceil_{\mathbb{P}}, 900\right) \quad \left(\text{mod } \left\lfloor \frac{\sum_{s=1}^{15} j_s}{10} \right\rfloor\right) \right\rceil_{\mathbb{P}}. \tag{13}$$

The prime $\hat{p}$ generated is of the same order of magnitude as the geometrically generated primes $\{p_1, p_2, \ldots, p_{15}\}$. Use of the SHA-256 digest of the authentication factor and calculating modulo $\lfloor \frac{1}{10} \sum_{s=1}^{15} j_s \rfloor$ makes inverting the function producing $\hat{p}$ difficult and thus the process may be considered a one-way function. Consequently trying to recover the value or contents of the additional authentication factor from knowledge of $\hat{p}$ will be difficult.

The set of geometrically determined prime integers is augmented by one or more prime integers derived from the additional authentication factors and the process described in Sec. 3 is used to generate a pseudorandom byte stream which now depends on the additional authentication factors. An eavesdropper on the communication between Alice and Bob would require not only the private initialization values $((c, \alpha, m), i, (r, x_s))$, but also the additional authentication factors in use.

2.5. **Number Theoretic Issues and Questions.** The method of selecting prime integers related to the dimensions of a three-dimensional geometric form is novel. Its use raises several questions about the primes selected and about the possibility of reconstructing the three-dimensional figure from knowledge of the primes only. Some of the questions identified as important are outlined below.

(1) Is function F from Eq. (5) a one-way function? Given the value of N is $F(q, N)$ invertible? Is function f from Eq. (4) invertible?
(2) Are the prime numbers generated by this procedure uniformly distributed among the set of prime integers? Is uniformity of distribution important to the security of the system?
(3) Thinking of the process of generating prime numbers as a function whose domain is the Cartesian product of the set of all permissible private initialization values with the set of all possible public initialization values and whose codomain is a set of all possible sets of fifteen prime integers, is this function an injection, a many-to-one mapping, a surjection, or a bijection?
(4) Given private initialization values $((c, \alpha, m), i, (r, x_s))$ and a public initialization value n, is there an open set (in some non-trivial topology) containing these values such that the image of the open set consists only of a single set of prime integers?
(5) Are the fifteen prime integers generated by the procedure described above, necessarily pairwise distinct?
(6) Is the procedure which generates the fifteen primes integers a one-way function? In other words, given only the set of fifteen prime integers, can the private initialization values be deduced?

3. Generation of the Key Stream

Earlier in Sec. 2.2 a geometrically based method for determining $15(i + 1)$ prime integers was described (recall that i is the private initialization value specifying the number of iterations of the two-dimensional geometric figure to construct). In Sec. 2.4 a procedure by which additional authentication factors were used to produce additional prime integers was described. From this point forward the total number of constructed primes will be assumed to be $k \geq 15$ and the primes will be treated the same regardless of the manner in which they were found. In this section those primes will be used to construct a matrix with k rows from which a key stream will be generated.

Given the set of primes $\{p_1, p_2, \ldots, p_k\}$, the members of this set can be sorted by their SHA-256 hashes [16]. Without loss of generality we may assume their sorted arrangement is still $\{p_1, p_2, \ldots, p_k\}$.

3.1. **Generation of Pseudorandom Sequences.** Calculation of the geometrically derived $15(i+1)$ prime integers has used the left-most $30(i+1)$ digits of $d = F(p; 4050)$. To generate a pseudorandom sequence from the first SHA-256-ordered prime, let $s = 1 + [d_{30i+31} \cdots d_{30i+35} \pmod{45}]$ and $t = 1 + [d_{30i+36} \cdots d_{30i+40} \pmod{45}]$. If $H_{s,t}$ of matrix H given in Eq. (7) is even then let $(k)_{16}$ be the integer represented by the concatenation of the four entries on row s of matrix H beginning in column t. Otherwise if $H_{s,t}$ is odd let $(k)_{16}$ be the integer represented by the concatenation of the four entries in column t of matrix H beginning in row s. Treating k as a base-10 integer, calculate

$$l_1 = 900 + [k \pmod{1500}]. \tag{14}$$

Using function F described in Eq. (5), determine the integer $v_1 = F(p_1; l_1)$ which will possess l_1 hexadecimal digits. Since the mantissa from which v_1 is derived is an irrational number, then the $(v_1)_2$ can be thought of as a pseudorandom sequence of 0's and 1's. By concatenating v_1 with itself repeatedly, an infinite l_1-periodic pseudorandom sequence denoted $\mathbf{v}_1$ is created. When convenient $\mathbf{v}_1$ can be thought of as an infinite sequence of two-digit hexadecimal integers.

$$\mathbf{v}_1 = x_1x_2, x_3x_4, \ldots, x_{l_1-1}x_{l_1}, x_1x_2, x_3x_4, \ldots, x_{l_1-1}x_{l_1}, \ldots \tag{15}$$

The example format illustrated in Eq. (15) assumes l_1 is even. A similar example can be created when l_1 is odd.

The remainder of the set of prime integers is used in the same manner with the requirement that the hexadecimal integer lengths $\{l_1, l_2, \ldots, l_k\}$ are all distinct. Suppose pseudorandom sequences $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_\nu$ have been determined. To this point, the left-most $30(i + \nu + 1)$ digits of $d = F(p; 4050)$ have been used. To generate pseudorandom sequence $\mathbf{v}_{\nu+1}$, let $s = 1 + [d_{30(i+\nu)+31} \cdots d_{30(i+\nu)+35} \pmod{45}]$ and $t = 1 + [d_{30(i+\nu)+36} \cdots d_{30(i+\nu)+40} \pmod{45}]$. Determine integer k as described above from matrix H and calculate $l_{\nu+1} = 900 + [k \pmod{1500}]$. If $l_{\nu+1} \in \{l_1, l_2, \ldots, l_\nu\}$, it is re-calculated using the next ten digits of $F(p; 4050)$ until a unique value is found. The $l_{\nu+1}$-periodic pseudorandom sequence $\mathbf{v}_{\nu+1}$ is found by concatenating $F(p_{\nu+1}, l_{\nu+1})$ with itself repeatedly. In this way periodic pseudorandom sequences $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ each having a unique fundamental period are determined from the prime integers found in Sec. 2.2.

3.2. **Generation of a Pseudorandom Matrix.** From the pseudorandom sequences $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ the $k \times \infty$ matrix, M (not to be confused with point M in Fig. 1) is formed where the jth row of M is $\mathbf{v}_j$. Since row j of M is l_j-periodic, matrix M is column-periodic with fundamental period $\operatorname{lcm}(l_1, l_2, \ldots, l_k)$. The fundamental column period can be large. For example if $i = 0$ the column period can be as small as approximately 5.354×10^{11} and as large as approximately 4.399×10^{50}, while when $i = 1$ the lower bound for the period is 2.214×10^{22} and the upper bound for the period is approximately 1.18×10^{101}. As was done for the pseudorandom sequence $\mathbf{v}_j$, when convenient, columns of M will be merged pairwise to form a matrix whose entries will be thought of as two-digit hexadecimal integers.

Another matrix will be used in concert with matrix M to produce a pseudorandom byte stream. This matrix is constructed in a similar way to M having the same number and length of pseudorandom sequences as matrix M but based on a different distinct set of derived primes. The new set of primes $\{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k\}$ is determined from the primes generated by the method described in Sec. 2.2 by calculating

$$\hat{p}_j = \left\lceil p_j^2 \pmod{\sum_{s=1}^{k} p_s} \right\rceil_{\mathbb{P}} \tag{16}$$

for $j = 1, 2, \ldots, k$. Sorting the primes $\{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k\}$ by the SHA-256 hash indices of primes $\{p_1, p_2, \ldots, p_k\}$, then for each j the integer $\hat{v}_j = F(\hat{p}_j, l_j)$ with l_j hexadecimal digits is calculated. This integer is concatenated with itself to produce an infinite l_j-periodic pseudorandom sequence $\hat{\mathbf{v}}_j$. Just as for matrix M a new $k \times \infty$ matrix L is created where the jth row of L is the infinite sequence $\hat{\mathbf{v}}_j$. Matrices L and M have the same number of rows and for both matrices the jth row is periodic with period l_j. Just as for matrix M, pairs of columns of L can be grouped so that L is thought of as having entries which are non-negative integers less than 256.

$$L = \begin{bmatrix} l_{1,1} & l_{1,2} & l_{1,3} & l_{1,4} & \cdots \\ l_{2,1} & l_{2,3} & l_{2,3} & l_{2,4} & \cdots \\ l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ l_{k-1,1} & l_{k-1,2} & l_{k-1,3} & l_{k-1,4} & \cdots \\ l_{k,1} & l_{k,2} & l_{k,3} & l_{k,4} & \cdots \end{bmatrix} \tag{17}$$

3.3. **Generation of a Pseudorandom Byte Stream.** Assuming the pseudorandom matrix M described in Sec. 3.2 has the form below, a byte stream can be calculated using its entries.

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & \cdots \\ m_{2,1} & m_{2,3} & m_{2,3} & m_{2,4} & \cdots \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ m_{k-1,1} & m_{k-1,2} & m_{k-1,3} & m_{k-1,4} & \cdots \\ m_{k,1} & m_{k,2} & m_{k,3} & m_{k,4} & \cdots \end{bmatrix} \tag{18}$$

Each entry of M is a two-digit hexadecimal number. The matrix is column periodic with a typically large period.

A pseudorandom offset into L and M is used to begin construction of the byte stream. Referring once again to $d = F(p; 4050)$, take the next five digits $d_j d_{j+1} d_{j+2} d_{j+3} d_{j+4}$ and calculate $s = 1 + [d_j d_{j+1} d_{j+2} d_{j+3} d_{j+4} \pmod{45}]$ and likewise with the following five digits find $t = 1 + [d_{j+5} d_{j+6} d_{j+7} d_{j+8} d_{j+9} \pmod{45}]$. The particular index j will depend on the number of digits of d used in earlier calculations. From matrix H in Eq. (7) if $H_{s,t}$ is even then let $(w)_{16}$ be the integer represented by the concatenation of the three entries on row s of matrix H beginning in column t (wrapping around the matrix if necessary). Otherwise if $H_{s,t}$ is odd let $(w)_{16}$ be the integer represented by the concatenation of the three entries in column t of matrix H beginning in row s (again, wrapping around if necessary). Note that $0 \leq w < 2^{24}$. The entries of M used to generate the byte stream will start in the (s', t') position where

$$(s', t') = g(w) \equiv \left(1 + [w \pmod{k}], 1 + \left\lfloor \frac{w}{k} \right\rfloor\right). \tag{19}$$

The first byte of the key stream is then calculated as

$$b_1 = G(s', t') \equiv \left(\bigoplus_{j=s'}^{k} m_{j,t'} \right) \oplus \left(\bigoplus_{j=1}^{s'} m_{j,t'+1} \right) \oplus l_{s',t'+1}. \tag{20}$$

where $\oplus$ denotes the bitwise XOR operation. Thus b_1 depends on a single entry from L and on a value from each row of M and in the case of row s', two values. The jth byte of the key stream b_j, for $j \in \mathbb{N}$ is found in a similar fashion using $b_j = G(g(w + j - 1))$ from Eq. (19) in the right-hand side of Eqs. (20). In this way an infinite sequence $\{b_j\}_{j=1}^{\infty}$ of pseudorandom bytes (or integers in $\{0, 1, \ldots, 255\}$) is generated. The key stream thus produced is infinite, but periodic with a period at least as large as the $\operatorname{lcm}(l_1, l_2, \ldots, l_k)$. Alice and Bob each generate the same infinite sequence since they use the same private and public initialization values This infinite sequence can serve as a one-time stream cipher for cryptographic purposes.

3.4. **Pseudorandomness Issues and Questions Relating to the Key Stream.** This section opened with the assumption that a set of prime numbers was available and has described a method for creating an infinite pseudorandom key stream from the primes. The procedure described in Sections 2 and 3 has been implemented in computer code in order to test the randomness of the generated key stream. The randomness of the key stream was assessed according to the NIST Statistical Test Suite [15] and was found to score favorably when compared with the true random source at the Australian National University [8, 19].

The method of construction of the key stream requires further discussion and poses questions. Some of the issues and questions are outlined in the remainder of this section.

(1) Function f defined in Eq. (4) maps prime numbers (a countable set) into the irrational numbers (an uncountable set). Can the set of all possible prime numbers which may be generated by the procedure outlined in Sec. 2 be determined by an attacker so that the set of all possible irrational numbers used in the construction of the key stream can be known as well?

(2) It is possible to select private and public initialization values which produce the minimum and maximum prime integers which can result from this cryptographic algorithm. A question of some mathematical interest is whether there are private and public initialization values which yield all the primes between these minimum and maximum primes.

(3) Is function f a one-to-one or many-to-one function?

(4) The mantissa of the output of function f is converted into an integer having between 900 and 2399 hexadecimal digits. From a knowledge of those limited number of digits, is it possible to determine the prime integer used as input to f?

(5) The periods of the pseudorandom vectors which make up matrix M are determined by a combination of the public and private initialization values. Does knowledge of these periods impart any information about the initialization values?

(6) The SHA-256 algorithm used to sort the input set of primes is a one-way function. Changing one prime in the input set would change a row in matrix M at an unpredictable location - as would inserting a new prime - and all the pseudorandom sequences which follow it. What are the implications to the security of the system vis-a-vis attacks on the additional authentication factors?

(7) Calculation of each byte of the key stream depends on entries from each row of matrix M and for each byte two entries from one row are used. Modification (including addition or deletion) of a row of M will alter the entire generated key stream. Differential bit analysis testing finds that any given bit of key stream has equal probability of changing or remaining the same. What are the implications of this to the security of the system?

(8) Would interception of a finite segment of the key stream enable an attacker to reconstruct matrix M?

(9) Does possession of a portion of the key stream enable an attacker to determine the dimensions of matrix M?

4. Encryption Strength and Security

The cryptographic method described above for generating a pseudorandom key stream is novel and as such, it is difficult to determine the appropriate metric of its cryptographic strength. The strength of many cryptographic algorithms is dependent on the size of the cryptographic keys (for this algorithm, the key can be thought of as the private initialization values $((c, \alpha, m), i, (r, x_s)))$ and on the algorithms' resistance to cryptanalytic attacks. However, due to the use of the periodic, pseudorandom key stream by this algorithm, the cryptographic strength may lie in the pseudorandom key stream. In this section, the size of the key space will be described, possible cryptanalytic attacks will be set forth, and issues and questions related to the security of this algorithm will be delineated.

4.1. **Key Length.** The private initialization values $((c, \alpha, m), i, (r, x_s))$ function as a key for the cryptographic process. The first component c is an integer where $(c)_2$ contains between 256 and 1160 binary digits. The angle α is a real number which will be represented on a digital computer as a floating point number. The number of bits involved in the representation of α will depend upon the floating point format in use. Since $\pi/12 \leq \alpha \leq 5\pi/12$ then if the IEEE-754 double precision (64 bits) format [20] is used, then the significand occupies 52 bits and two bits of the exponent field may change. Thus a total of 54 binary digits are involved in the representation of α. If an extended precision format or custom format is used, then more bits may be involved. The multiplier m is also a real number where $4 \leq m \leq 65535$. Thus in IEEE-754 double precision format, a total of 56 bits may be involved in the representation of m. Therefore if the basic form of the cryptographic procedure is used (without iteration or translation) the size of the key space (c, α, m) in bits is

$$1160 + 54 + 56 = 1270.$$

The iteration extension (nonnegative integer i) controls the number of rows in matrix M and in many usage cases is kept small so as to reduce the computer memory requirements of storing M. Thus the iteration count does not significantly increase the size of the private initialization value key space.

The translation extension involves two additional real numbers (r, x_s). Since $0 < x_s < r < \sqrt{2}/2$ then specification of each of these real numbers involves modifications of up to 62 binary digits, again assuming the IEEE-754 double precision format. Thus if iteration and translation is employed in the cryptographic algorithm, the size of the private initialization value space is at least

$$1160 + 54 + 56 + 62 + 62 = 1394 \quad \text{bits.}$$

The cryptographic strength of the algorithm can be considered the number of bits on the key space, provided there is no cryptanalytic attack requiring the search of a smaller space.

4.2. **Key Stream.** An eavesdropper in possession of the key stream can decrypt a ciphertext encrypted with the same key stream. A question of concern in this section is whether an eavesdropper in possession of a sample of the key stream can generate the remainder of the key stream from the sample. Since the key stream will be periodic, then possession of a complete period of the key stream is sufficient to decrypt a ciphertext. The minimum period of the key stream is great enough to encrypt approximately one million average length books before repeating. Each generated byte b_j of the pseudorandom key stream depends on $k+1$ bytes found in matrix M and a byte from matrix L. Thus given b_j there are $2^{8(k+1)}$ bit patterns which can produce b_j. For $k = 15$ this corresponds to $2^{128} \approx 3.403 \times 10^{38}$ bit patterns. The next byte of key stream depends on some of the same matrix entries as did b_j. One byte (an entry of M) is dropped from the calculation while a new byte comes into use. Similarly one entry of L is retired from the calculation and a new entry takes its place. Thus given b_j and b_{j+1} there are 2^{16} bit patterns which can be XORed with b_j to produce b_{j+1}. Each new byte of the key stream would also require a search of 2^{16} bits to determine.

Perhaps the most interesting question related to this cryptographic procedure is that of determining the most appropriate measure of its cryptographic strength. A perhaps naive estimate based on the comments made in the previous paragraph would suggest that the first byte of key stream (in the case of k=15) requires a search of 2^{128} bytes while each additional byte of key stream requires a search of 2^{16} bytes. Thus for a key stream of minimal period

$$\operatorname{lcm}(900, 1000, \ldots, 2300) = 535422888000$$

an attacker attempting a brute force search must consider

$$2^{128} \left(2^{16}\right)^{535422887999} \approx 10^{2578853594483}$$

possible solutions. The authors expect other mathematicians and cryptology experts to refine the idea of the correct measure of the cryptographic strength of this algorithm.

4.3. **Cryptanalytic Attacks.** The vulnerabilities of a two-time pad are well known [13]. Since the key stream generated by the algorithm under discussion is periodic, there exists the possibility that different segments of a single ciphertext may be encrypted with the same key stream segment (offset by a multiple of the period). There are reasons this vulnerability may be difficult to exploit in practice. First, the period of the key stream is typically large and can be made even larger by the use of the iteration value i and the additional authentication factors discussed in Sec. 2.4. Thus except for lengthy communications, less than one period of the key stream may be used during encryption. Second, an eavesdropper does not know the period of the key stream and has potentially as many as $\binom{1500}{15(i+1)}$ possibilities for the period. Third, determination of the key stream from a single ciphertext enables an attacker to recover only a single plaintext. Following best cryptologic practices, Alice and Bob would use a different public initialization value n (Sec. 2.2) for their next communication, which would result in a new key stream.

In order to learn the private initialization values $((c, \alpha, m), i, (r, x_s))$ in use by Alice and Bob, an eavesdropper in possession of an intercepted ciphertext (assumed to be of sufficient quantity that a period of the key stream may be deduced) would have to achieve the following goals.

1. Determine the dimensions of matrices L and M (in essence the number of rows in each).
2. Determine the period of each row of L or M. This challenge is linked to that of determining the order of the rows in L and M. The period of the key stream is unchanged by a reordering of the rows in these matrices, but the contents of the key stream is changed by a reordering.

(3) Determine the entries of matrices L and M from a byte of the key stream. Issues related to this challenge were outlined in Sec. 4.2.

(4) From a row of matrix M (which represents a periodic, truncated approximation to an irrational number) determine the prime integer p_j which produced this row. Similarly from a row of matrix L determine the prime $\hat{p}_j$ which produces this row. This touches on the questions raised about functions f and F raised in Sec. 2.5.

(5) Assuming no additional authentication factors are in use, then from the list of prime integers $\{p_1, p_2, \ldots, p_{15(i+1)}\}$ determine the i sets of line segment lengths $\{\|\overline{AZ_2}\|, \|\overline{BZ_2}\|, \ldots, \|\overline{X_2Z_4}\|\}$. Recall that the primes are all strictly greater than the line segment lengths.

(6) Determine the two-dimensional geometric form described in Sec. 2.1 from the line segment lengths $\{\|\overline{AZ_2}\|, \|\overline{BZ_2}\|, \ldots, \|\overline{X_2Z_4}\|\}$.

(7) Determine the private initialization values $((c, \alpha, m), i, (r, x_s))$ from the two-dimensional geometric form.

Only the last two of the challenges in this list would seem to have solution procedures. The other challenges listed provide ample opportunity for cryptanalytic and mathematical research.

5. Concluding Remarks

The designers of this cryptosystem, like those of every other cryptosystem, were faced with a number of architectural choices, some of which were decided arbitrarily. Other choices were made as a result of real or perceived vulnerabilities in earlier iterations of the design. The overall operation of the algorithm remains largely unchanged if modifications are made to some of its design elements. For example, the choice of deriving fifteen prime integers from each triangle in the xy-plane was driven by the need for speedy execution of the code on modern personal computers. This number could easily be changed. The choice of a triangle as the basic geometrical building block of the algorithm is arbitrary. Other geometrical figures such as quadrilaterals, ellipses, or cones could be used as well. There is no necessity for the two-dimensional figure derived from the private initialization values to lie in the xy-plane. The boundary values of the private and public initialization values are easily changed. The primes which generate matrix M are sorted by their SHA-256 digests, which is yet another arbitrary choice. Alternatives to expanding the public initialization value as described in Sec. 2.2 are also possible. These and other design elements of the cryptosystem could be modified and the algorithm still allows secure communication between Alice and Bob so long as they agree to use the same set of modifications.

The authors hope that the research community will analyze this cryptosystem and answer some of the questions raised in this article. Should any weaknesses be found, it is possible that one of the design changes mentioned in the previous paragraph may mitigate them.

Appendix A. Passphrase Encryption

The normal mode of operation for this cryptographic algorithm requires Alice and Bob to exchange private initialization values $((c, \alpha, m), i, (r, x_s))$ and optional additional authentication factors. The creation of the private initialization values can be simplified by generating them from another piece of information shared (only) by Alice and Bob. In this section the process by which the private initialization values are generated from a shared passphrase will be outlined. Other generation procedures are certainly possible.

Let x be the passphrase shared by Alice and Bob and let s be the SHA-512 digest of x. The digest s can be represented in hexadecimal as

$$(s)_{16} = h_1 h_2 \cdots h_{255} h_{256} h_{257} h_{258} \cdots h_{511} h_{512}. \tag{21}$$

The digest is bisected between the 256th and 257th digits and a prime number p' is calculated from it.

$$p' = \lceil \lceil h_1 h_2 \cdots h_{255} h_{256} \rceil_{\mathbb{P}} \oplus \lceil h_{257} h_{258} \cdots h_{511} h_{512} \rceil_{\mathbb{P}} \rceil_{\mathbb{P}} \tag{22}$$

From p' an integer q' with 2800 hexadecimal digits (which implies approximately 3372 decimal digits) is found using function F of Eq. (5).

$$q' = F(p', 2800) = d_1 d_2 d_3 d_4 \cdots d_{3372} \tag{23}$$

In order to generate the private initialization value c let $l = 79 + (d_1 d_2 d_3 d_4) \pmod{60}$. Consequently $79 \leq l < 139$ and let

$$c = d_5 d_6 \cdots d_{l+4}, \tag{24}$$

that is, c is an integer consisting of the next l decimal digits of q'. Since q' is formed from the truncated mantissa of an irrational number, the digits of q' and thus the integer c should be pseudorandom. The multiplier m is calculated as

$$m = (d_{l+5} d_{l+6} \cdots d_{l+14}) \times 10^{-8} \tag{25}$$

provided $m \geq 4$ (by construction this choice of $m \leq 65535$). If not increase l in steps of one until an appropriate m is determined. Similar to c, the value of m should be pseudorandom. The remaining components of the private initialization values are arbitrarily set so that the complete private initialization value tuple is $((c, \pi/6, m), 2, (0, 0))$. From this set of private initialization values the two-dimensional geometric form is created as outlined in Sec. 2.1.

References

[1] IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture. *IEEE Std 802-2014 (Revision to IEEE Std 802-2001)*, pages 1–74, June 2014.

[2] M. Becker and A. Desoky. A study of the DVD content scrambling system (CSS) algorithm. In *Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International Symposium on*, pages 353–356, Dec 2004.

[3] David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701 – 716, 2005. Web Security.

[4] H.S.M. Coxeter. *Introduction to Geometry*. Wiley Classics Library. Wiley, 1989.

[5] H.S.M. Coxeter and S.L. Greitzer. *Geometry Revisited*. New Mathematical Library. Mathematical Association of America, 1967.

[6] W. Dunham. *Journey Through Genius: the Great Theorems of Mathematics*. Wiley science editions. Penguin Books, 1991.

[7] Morris J. Dworkin, Elaine B. Barker, James R. Nechvatal, James Foti, Lawrence E. Bassham, E Roback, and James F. Dray Jr. *FIPS PUB 197: Advanced Encryption Standard (AES)*. November 2001.

[8] Secure Quantum Communication group. ANU quantum random number server. `http://qrng.anu.edu.au/NIST.php`. Accessed: 2014-09-23.

[9] R.A. Johnson and J.W. Young. *Modern Geometry: An Elementary Treatise on the Geometry of the Triangle and the Circle*. Houghton, Mifflin Company, 1929.

[10] J. Jonsson and B. Kaliski. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, 2003.

[11] C. Kimberling. Encyclopedia of triangle centers. `http://faculty.evansville.edu/ck6/encyclopedia/ETC.html`. Accessed: 2014-07-14.

[12] C. Kimberling. Central points and central lines in the plane of a triangle. *Mathematics Magazine*, 67(3):pp. 163–187, 1994.

[13] Joshua Mason, Kathryn Watkins, Jason Eisner, and Adam Stubblefield. A natural language approach to automated cryptanalysis of two-time pads. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 235–244, New York, NY, USA, 2006. ACM.

[14] National Institute of Standards and Technology. *FIPS PUB 46-3: Data Encryption Standard (DES)*. March 1999.

[15] National Institute of Standards and Technology. *SP 800-22 Rev. 1a: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. April 2010.

[16] National Institute of Standards and Technology. *FIPS PUB 180-4: Secure Hash Standard (SHS)*. March 2012.

[17] D. Pedoe. *Circles, a Mathematical View*. International series of monographs on pure and applied mathematics. Dover Publications, 1957.

[18] M. L. Stein, S. M. Ulam, and M. B. Wells. A visual display of some properties of the distribution of primes. *The American Mathematical Monthly*, 71(5):pp. 516–520, 1964.

[19] T. Symul, S. M. Assad, and P. K. Lam. Real time demonstration of high bitrate quantum random number generation with coherent laser light. *Applied Physics Letters*, 98(23), 2011.

[20] D. Zuras et al. IEEE Standard for Floating-Point Arithmetic. *754-2008: IEEE Standard for Floating-Point Arithmetic*, pages 1–70, 2008.

URL: `www.UberCrypt.com`