On the use of continued fractions for stream ciphers

Amadou Moctar Kane
Département de Mathématiques et de Statistiques, Université Laval, Pavillon Alexandre-Vachon,
1045 av. de la Médecine, Québec G1V 0A6 Canada.
amadou-moctar.kane.1@ulaval.ca

May 25, 2013

Abstract

In this paper, we present a new approach to stream ciphers. This method draws its strength from public key algorithms such as RSA and the development in continued fractions of certain irrational numbers to produce a pseudo-random stream. Although the encryption scheme proposed in this paper is based on a hard mathematical problem, its use is fast.

Keywords: continued fractions, cryptography, pseudo-random, symmetric-key encryption, stream cipher.

1 Introduction

The one time pad is presently known as one of the simplest and fastest encryption methods. In binary data, applying a one time pad algorithm consists of combining the pad and the plain text with XOR. This requires the use of a key size equal to the size of the plain text, which unfortunately is very difficult to implement. If a deterministic program is used to generate the keystream, then the system will be called stream cipher instead of one time pad. Stream ciphers use a great deal of pseudo-random generators such as the Linear Feedback Shift Registers (LFSR); although cryptographically weak [37], the LFSRs present some advantages like the fast time of execution. There are also generators based on Non-Linear transitions, examples included the Non-Linear Feedback Shift Register NLFSR and the Feedback Shift with Carry Register FCSR. Such generators appear to be more secure than those based on LFSR. A new type of generators was introduced by the European project NESSIE which has spawned many algorithms like BMGL, Leviathan and Lili. More recently the eSTREAM project produced some promising algorithms like HC-128, Rabbit, Salsa20/12 and Sosemanuk; however, the effectiveness of these algorithms will be better known in the future.
Another approach consisting in building algorithms around a hard mathematical problem often yields cumbersome and slow generators. In that class, we count generators based on RSA algorithm as described in Alexi *et al.* [64], or on the difficulty of computing discrete logarithms (Blum-Micali [9]). The best algorithm in this class remains the Blum-Blum-Shub generator [11], algorithm which is based on quadratic residues.
Here we present an algorithm based on the difficulty of retrieving an irrational number from the sole knowledge of a part of its continued fraction expansion, hence our algorithm is built around a hard mathematical problem.
In [53] Pieprzyk *et al.* have proposed the use of transcendental numbers in cryptography; they proposed to use the digits of some transcendental numbers as a pseudo-random generator. Although their solution has been criticized by some cryptanalysts, we are confident that some irrational numbers can be used in cryptography.
The aim of this paper is to introduce the use of some continued fractions in cryptography, knowing that the continued fraction expansions of most irrational numbers are unpredictable.

Continued Fractions : An expression of the form

$$\alpha \quad = \quad a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{\ddots}}}$$

is called a generalized continued fraction. Typically, the numbers $a_1, \ldots, b_1, \ldots$ may be real or complex, and the expansion may be finite or infinite.
We will avoid the use of the continued fraction expansions involving $b_i = 1$ for most i's. However, in order to simplify our explanation we will use in some cases the classical continued fraction expansion, namely $b_i = 1$ for any i:

$$\begin{aligned} \alpha &= a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}} \\ &=: [a_0, a_1, a_2, a_3, a_4, \ldots]. \end{aligned}$$

In this paper we denote by Γ the combined sets of algebraic irrationals of degree greater than 2 and transcendental numbers. Our algorithm, will use the irrational numbers which are in Γ, but we will avoid the use of transcendental numbers having a predictable continued fraction expansion (some examples of irrational numbers given a predictable continued fraction expansion are presented in [17, 29]).
To calculate the classical continued fraction expansion of a number α, write down the integer part of α. Subtract this integer part from α. If the difference is equal to 0, stop; otherwise find the reciprocal of the difference and repeat. The procedure will halt if and only if α is rational.
We can enumerate some continued fraction properties:

1. The continued fraction expansion of a number is finite if and only if the number is rational.
2. The continued fraction expansion of an irrational number is unique.
3. Any positive quadratic irrational number α has a continued fraction which is periodic from some point onward, namely a sequence of integers repeats. (Lagrange Theorem)
4. The knowledge of the continued fraction expansions of α and β cannot determine simply that of $\alpha + \beta$, or $\alpha\beta$.

Continued fractions were widely studied by C. Olds [51] and O. Perron [52], but cryptographic views are not explored by number theory specialists except in some fields like RSA cryptanalysis.

This paper is organized as follows. In section 2, we show some links which exist between continued fraction and cryptology; in section 3, we will propose and demonstrate some results concerning the keystream; in section 4, we will introduce and analyze our algorithm; section 5 shows some possible attacks in a weak version of our scheme. The section 6 illustrates some of the advantages given by this algorithm and, before concluding, we will detail some possible applications of this algorithm.

2 Continued fraction and cryptology

The use of continued fractions in cryptography dates from the late 80s, and essentially covered, before the introduction of our algorithm, the study of pseudo random generators security and the cryptanalysis of RSA algorithms.

2.1 Continued fraction, pseudo random numbers and linear complexity of sequences

This domain has been highly developed by Harald Niederreiter [46], since he established the relation between the linear complexity profile of a keystream sequence and the continued fraction expansion of its generating function. He also presented relations between continued fraction expansions for formal power series and investigations on pseudorandom sequences. Another way to see the relationship between continued fractions and linear complexity of sequences can be observed through the summary of Lauder in [33]: Any sequence over a finite field may be encoded as a Laurent series. The linear complexity profile of the sequence, which is of interest from a cryptographic viewpoint, may then be read off from the continued fraction expansion of the associated Laurent series. If all the partial quotients of the continued fraction have small degree, then the Laurent series may be considered difficult to approximate. Sequences with desirable profiles correspond to Laurent series which are difficult to approximate. More information in this area can be found in [46, 47, 33, 19].

2.2 Continued fraction and RSA cryptanalysis

In 1990, Wiener published in [65] an attack against RSA, using the classical continued fractions. This attack can recover the private key with the sole knowledge of the public key e and n.

His technique is based on approximations using continued fraction. For (e, n) the public key and d the private key of RSA, the Wiener attack can recover the private key d by computing the continued fraction of e/n. Among convergents obtained from the continued fraction expansion of e/n, we will find that one has d as denominator. This wiener attack was possible if d is less than $n^{0.25}$.

2.3 Continued fraction and chaos theory

Chaos theory is used in cryptography since the 90s [5], however, in recent years, contributions using the links between continued fractions and chaos theory have been published in cryptography [24, 41]. Unfortunately, their paper does not mention our works [20, 21] that predate their contribution.

2.4 Continued fraction and approximation

Continued fractions can also be linked to Diophantine approximations used in cryptography. However, we must point out that our approach in the use of continued fractions is different because we are interested in the randomness of partial quotients and not on the approximation aspect.

2.4.1 On the cryptanalysis

The approximation problems are used since long time in cryptanalysis, for example in 1988, Kannan et al. [23] by using the lattice basis reduction algorithm have showed that the binary expansions of algebraic numbers do not form secure pseudorandom sequences. Shamir [56], by using the integer programming method, gave a cryptanalysis of a version of knapsack [43]. Invented by Lenstra, Lenstra and Lovasz, in 1982, the LLL algorithm [34] is a polynomial time lattice reduction algorithm which has allowed the cryptanalysis of several encryption systems such as many versions of knapsack [43]. The LLL is also used in cryptanalysis of RSA since the attack showed by Coppersmith[13]; many other improved attacks try to use the LLL algorithm in order to break RSA.

2.4.2 On the encryption scheme

The lattices reduction algorithms are also used in encryption scheme as in Ajtai-Dwork, and in the NTRU. The Ajtai-Dwork [3] encryption is based on the hard mathematical problem (an instance of the shortest vector problem (SVP) showed by Ajtai [1, 2]). The shortest vector problem (SVP), aim is to find a nonzero lattice vector of smallest norm, given a lattice basis as input.

The NTRU Public Key Cryptosystem is also based on a hard mathematical problem (finding very short vectors in lattices of very high dimension). The process of solving this problem is called "Lattice Reduction".

More information between lattices and cryptology can be found in [49, 48].

Recently a new type of encryption system has been introduced, Elsner and Schmitt had published in [14] a paper presenting a Symmetric Cryptosystem KronCrypt using continued fraction, Kronecker symbol, and Feistel Cipher in a S-box.

In another paper Armknecht, Elsner and Schmitt [6] tried to formalize the Decisional Inhomogeneous Simultaneous Approximation Problem (DISAP) and show that this problem is NP-complete. They also exhibit a bit commitment scheme based on DISAP.

3 Results

A keystream must satisfy the basic condition of key-recovery resistance. And as defined by Rueppel in [55], a keystream generator is defined to be perfect if it is unpredictable or indistinguishable by all polynomial time statistical tests. In this study, we consider that a keystream is unpredictable if it passes the next-bit and previous-bit tests.

In this section we will establish the key-recovery resistance and the unpredictability of our keystream. The unpredictability results from some properties of continued fractions and is covered in *result 5* & *6*. The *Results 1* establishes the property of key-recovery resistance.

Although we do not study the indistiguishability in this paper, we believe that our keystream is indistinguishable from a truly random stream for two reasons: the first reason is that there is an equivalence for bit sequences between the unpredictability and the indistinguishability [45] and the second reason is that our pseudo-random stream passes the NIST test suite [50] with success.

Notation 1 Let $\alpha \in \Gamma$ and $[a_1, \ldots, a_m, \ldots, a_{m+n}, \ldots]$ be the continued fraction expansion of α; m and n are two integers such that $m > 20$ and $n \geq 1$. We denote δ the vector made with the n partial quotients following the m first partial quotients in the continued fraction expansion.

Result 1. *It is not possible to find α out of the knowledge of δ.*

PROOF. Let $\alpha \in \Gamma$. We suppose that we know a given part $[a_{m+1}, \ldots, a_{m+n}]$ of α's continued fraction expansion. Can we find α with the knowledge of these n partial quotients? The answer is negative, because there exist an infinite number of irrationals with these same partial quotients. For instance we can exhibit infinitely many irrational numbers α_ρ which are different from α and which have the property that $[a_{m+1}, \ldots, a_{m+n}]$ appears as a sequence of n consecutive partial quotients. As a matter of fact, when θ is an irrational number, it suffices to consider any sequence of m integers $(r_1, r_2, \ldots, r_m)$ and to define α_ρ to be

$$\begin{aligned}\alpha_\rho &= r_1 + \cfrac{1}{\ddots\, r_m + \cfrac{1}{a_{m+1} + \cfrac{1}{\ddots\, a_{m+n-1} + \cfrac{1}{a_{m+n} + \cfrac{1}{\theta}}}}} \\ &=: [r_1, r_2, r_3, \ldots, r_m, a_{m+1}, \ldots, a_{m+n}, \theta].\end{aligned}$$

Result 2. *For an integer r such that $r \geq 3$ and a real algebraic number A ($A > 1$), the number $\sqrt[r]{\log(A)}$ is transcendental.*

PROOF. Assume that A is a real algebraic number such that $A > 1$, then $\log A$ is transcendental number by Corollary 3.6 of [12]. If we suppose that $X = \sqrt[r]{\log(A)}$ is an algebraic number, then X^r is a algebraic number, which is absurd because $X^r = \log(A)$ and $\log(A)$ is transcendental.

REMARK 1. The irrational number $\sqrt[n]{\log(A)}$ used in this paper is not a standard which we impose. It is an example which we choose in order to illustrate our scheme.

Result 3. *Let $\beta \in \Gamma$ and let $a = [a_r, \ldots, a_{r+n}]$ be a part of the continued fraction expansion of β. The knowledge of a does not allow to know any other partial quotients of continued fraction expansion.*

REMARK 2. From the proof of *Result 1*, we can deduce the proof of *Result 3*.

Notation 2 Let $A_1 \in \Gamma$, $b_i \in \mathcal{R}$ $(\forall\, i \in \mathcal{N})$, and $e_1, \ldots, e_{p+n}, \ldots$ be the partial quotients of A_1 $(p, n \in \mathcal{N}\backslash\{0, 1\})$. If we consider that the binary expression of the concatenation of this sequence $[e_n, \ldots, e_{n+p}]$ is $q_1, \ldots, q_l$ $(l > 322)$, then we define our keystream x to be $q_{162}, \ldots, q_{l-161}$.

Result 4. Levy shows in [36] that if k is the partial quotient and f_k is the frequency of its occurrence, then

$$f_k = \frac{1}{\log(2)} \log(1 + \frac{1}{(k(k+2))})$$

.

REMARK 3. This formula is valid for the classical continued fraction expansion ($b_i = 1\ \forall i$).

Conjecture. Our algorithm is cryptographically secure under the assumption that the distribution of partial quotients in the continued fraction expansion is indistinguishable by all polynomial-time statistical tests from the uniform distribution of integers in the interval $[S, P]$.

In our case we estimate that S is greater or equal to the smallest partial quotient which means $2^{16610} < S < 2^{16611}$ and we can add that P is difficult to estimate. This estimation for S is valid when the b_i's used to compute the partial quotients have 5000 digits. More explanations on this conjecture can be find at the end of this paper.

Result 5. *The binary expansion of x passes the next-bit test.*

PROOF. Let $\gamma_1, \ldots, \gamma_q$ be the binary expansion of x. Let i be a number chose randomly in $[1, \ldots, q]$ and $\gamma_1, \ldots, \gamma_i$ be the i first bits of x. If we suppose that we know the i first bits of x, then the quantities $P(\gamma_{i+1} = 1)$ and $P(\gamma_{i+1} = 0)$ cannot be greater than 1/2. Otherwise the probability of having some numbers in the partial quotients would be higher than others. We recall that

in the previous conjecture we have supposed that the distribution of partial quotients is indistinguishable by all polynomial-time statistical tests from the uniform distribution of integers in the interval $[S, P]$, then $P(\gamma_{i+1} = 0) = 1/2$.

Result 6. *The binary expansion of x passes the previous-bit test.*

REMARK 4. The proof of *Result 6* looks like the proof of *Result 5.*

4 Design

4.1 Continued fraction algorithm

Because a stream cipher usually requires a secret-key agreement, we will use the RSA scheme to exchange the seed (other public key encryptions can be used as well). In our design an entity Bob can receive his messages with a stream cipher from any entity having his public key. We assume that we are in a classical RSA [54] scheme, where Bob has a public RSA key (n, e) and a private key d. In order to avoid a Man in the Middle attack we have to authenticate or to add the sender's signature on the first message.
Suppose that Alice wants to send a message to Bob. She selects randomly and uniformly a large integer $z \in \mathcal{N}$, computes $t \equiv z^e \bmod n$ and sends t to Bob. Bob finds z by using his private key d; namely $z \equiv t^d \bmod n$. Now the two entities are ready to begin their stream communication.
At each round i:

1. The two parties compute $X_i = \sqrt[a]{g(h_{z_i}(t_i))}$ where a, g, h_{z_i} and t_i can be chosen freely as long as X_i is in Γ.
2. Each of the two entities computes the continued fraction expansion of the irrational number X_i.
3. Each of the two parties selects the n partial quotients following the p first partial quotients of the continued fraction expansion.
4. These n partial quotients are concatenated, transformed in bits and treated. They will be a part of our keystream.
5. If the keystream produced is lower than the plain text, then the two parties move to the round $i + 1$(namely return to step 1).

Below, in table 1 and figure 1, we illustrate the first round of the algorithm; we will restrict ourselves on the irrational number X, the parameter a will correspond here to Bob's public key e and $h_{z_0}(t_0)$ will be equal to z. The function g will correspond to the log.

Table 1: **Continued fraction cipher.**

Alice		Bob
computes $t \equiv z^e \bmod n$	$\stackrel{t}{\Longrightarrow}$	computes $z \equiv t^d \bmod n$.
Computes $X = \sqrt[e]{\log(z)}$		Computes $X = \sqrt[e]{\log(z)}$
Computes the CFE of X		Computes the CFE of X.
Concatenates some PQ's		Concatenates some PQ's.
Produces the keystream k_1		Produces the keystream k_1.
Computes $m_1 := m \oplus k_1$	$\stackrel{m_1}{\Longrightarrow}$	receives m_1.
		Computes $m := m_1 \oplus k_1$

Alice	**Attacker**	**Bob**
(1) *Computes* $X = \sqrt[e]{\log(z)}$.		(1) Computes $X = \sqrt[e]{\log(z)}$.
(2) *Takes* $r = 100$ *and* $s = 20$.		(2) Takes $r = 100$ and $s = 20$.
(3) *Computes the continued fraction expansion of* X*; this process does not stop until the end of this algorithm. Example: the* $2r$ *first partial quotients are* $X = [a_1, \dots a_r, \dots, a_{2r}]$.		(3) Computes the continued fraction expansion of X; this process does not stop until the end of this algorithm. Example: the $2r$ first partial quotients are $X = [a_1, \dots a_r, \dots, a_{2r}]$.
(4) *Concatenates all partial quotients whose indices are between* $s+1$ *and* $s+r$ $x_1 = a_{s+1} \dots a_{s+r}$.		(4) Concatenates all partial quotients whose indices are between $s + 1$ and $s + r$ $x_1 = a_{s+1} \dots a_{s+r}$.
(5) *Fromthe binary expression of* x_1*, extracts the part located between the first 161 bits and the last 160 bits. This extracted part* key_1 *is a part of our keystream.*		(5) From the binary expression of x_1, extracts the part located between the first 161 bits and the last 160 bits. This extracted part key_1 is a part of our keystream.
(6) *From the message* m*, computes the ciphertext* c *and sends it.* $c = m \oplus key_1$	Receives c	(6) Receives c and computes (XOR) $m = c \oplus key_1$.
(7) *Continues to produce the keystream and to send the stream cipher as in the previous steps until the end of the message.*		(7) Continues to produce the keystream and to decrypt the cipher as in the previous steps until the end of the cipher text.

Figure 1: Example of our stream cipher.

4.2 Security and efficiency analysis

4.2.1 Efficiency analysis

Let $a, b \in \mathcal{N}\backslash\{0, 1\}$, $s = \max(\log_2(a), \log_2(b))$ and $\epsilon \in]0, 1[$ a small number. We estimate that the product ab can be obtained by the Fourier transform method with a cost $O(s^{1+\epsilon})$, and the division cost of a/b is $O(s^{1+\epsilon})$ with the Newton algorithm.
Key setup cost: The RSA encryption cost is about $O(\log_2^2(n))$ and the RSA decryption cost will be about $O(\log_2^3(n))$(here n is the RSA public key). With the shifting n-th root algorithm, we evaluate the time to compute $X = \sqrt[e]{\log(z)}$ to be $O(k^{2+\epsilon}e^{1+\epsilon})$ where k is the number of digits of $\log(z)$ (we assume that the computer takes the k first digits of the irrational $\log(z)$ for its computations). The computation time of $\log(z)$ is $O(t^{\frac{4}{3}+\epsilon})$ where $t = \log_2(z)$, for more details in the computation of logarithm see [59].
Keystream cost: The keystream cost is obtained by adding the time used for the computation of the partial quotients and the cost of the concatenations. We neglect the cost of the concatenations and we evaluate the cost of calculating a partial quotient to be: $O(\delta^{1+\epsilon})$, where $\delta = \max(\log_2(t_1), \log_2(b_i))$. We suppose that the computer takes the first k digits of our irrational X for its computations and t_1 is the number composed by the first k digits of X. Hence, the keystream cost is $O(r\delta^{1+\epsilon})$ where r is the number of partial quotients to compute.

4.2.2 Security Analysis

In some case this algorithm can be vulnerable to the Timing Attack; thus the implementation will have to be done in such a way as to prevent such an attack. In addition, we believe that the cryptanalysis would be more difficult with a b_i secret, even if we do not find an attack on the generalized continued fraction where the b_i's are public. For example, the b_i's can be choose at random, encrypted and sent before the beginning of the stream communication.

Cryptanalysis

In this section we will study some possible attacks on our algorithm, and we consider that a successful attack on our keystream is realistic if the attacker can predict a bit in our keystream with a probability higher than $\frac{1}{2}$.

Static analysis

A static analysis takes the keystream and tries to reconstruct the initial key. The aim of *result 1* is to prove that the attacker is not able to reconstruct the initial irrational number even if he succeed in obtaining many of the partial quotients produced by our irrational.

Period Length

If we concatenate partial quotients having more than 5000 digits into 100-letter words, then for each keystream block to produce, we will have a number x with more than 500000 digits. It is known that for a generalized continued fraction of $\alpha \in \Gamma$ the probability of meeting the same x twice is around zero. Because if we suppose that we have a sample of one hundred keystreams blocks $x_1, \ldots, x_{100}$, where each $x_i (i \in 1, \ldots, 100)$ has more than 500000 digits, then the probability of finding two identical elements in $\{x_1, \ldots, x_{100}\}$ is

$$1 - (1 - \frac{1}{10^{500000}})(1 - \frac{2}{10^{500000}}) \ldots (1 - \frac{99}{10^{500000}}) \approx 0.$$

We can conclude that the period length is high enough to protect our keystream.

Collision

As we have seen it in the period length section, the probability of finding a keystream block twice is very small when our partial quotients are composed by more than 5000 digits.

Weak keys

We strongly advise against the use of some irrationals like the quadratics irrationals, or some numbers having a predictable expansion such $e, \phi, \ldots$ because without countermeasures the use of these numbers can create a weakness in this algorithm.

Chosen plain text attacks

Even if we show in *results 1 & 5 & 6* that the chosen plain text attacks will fail, additional protections against such attacks are detailed in the following steps.

1. The attacker must be able to identify the part of the keystream coming from x_1 which is a concatenation of one hundred partial quotients. For example if the attacker see this part 0101011100101010110101 on the keystream, then he must be able to say that 0101011100101010 comes from x_1 and 110101 comes from the following bloc which is x_2. We believe that this identification is not easy because the lengths of $x_1, x_2, \ldots$ are not fixed.

2. The second security level is based on the impossibility of finding back x_1 from the sole knowledge of its keystream contribution. Since the binary expression of x_1 is truncated by 321 bits, the attacker needs to find the 320 unknown bits which is very difficult even if the birthday attack is used.

3. If we suppose that the attacker moves past the two previous steps, then he needs to find the partial quotients present in x_1. We know that the length of the partial quotients varies, for example the six first partial quotients of π are 3, 7, 15, 1, 292, 1. These partial quotients are concatenated into 371512921, hence it will not be easy to recover all the partial quotients from the sole knowledge of 371512921.

4. If we admit that the attacker found the partial quotients in the previous step, he will not be able to go further as proved in *results 1& 5 & 6*.

Distinguisher attacks

We have two reasons to think that our keystream cannot be attacked by a distinguisher. The first one is that our keystream passed the NIST test suite with success. The second one is this algorithm pass the bit test as shown previously in *result 5 & 6*.

Linear cryptanalysis

The current knowledge on continued fractions does not allow a linear cryptanalysis of this algorithm and we believe that it would be very difficult to find a linear behavior for some generalized continued fraction.

Statistical tests

The statistical test on this algorithm was performed using the NIST test suite [50]. We tested 20 millions of bits for each of the following generators Blum-Blum-Shub, Micali-Schnor, DES and our algorithm. All the four algorithms passed the test with success, and the results obtained for our algorithm were much closer to those obtained with Blum-Blum-Shub. No weaknesses were found in our keystream.

4.3 Implementation aspects

4.3.1 Irrational numbers in computer

We chose to base our work on irrational numbers because our aim was to show theoretically the utility of continued fraction expansions of some irrational numbers. There is another advantage given by the irrational numbers which is their infinite number of digits, (depending on the needs and the users the irrationals permit to take enough digits). We can add that our results obtained for the irrational numbers can be easily extended to the rationals.
Consider x an irrational number, $n \in \mathcal{N}^*$ and $k_n(x)$ the exact number of partial quotients in the continued fraction expansion of x, given by the n first decimals of x. Lochs [38] in 1964 showes that

$$\lim_{n \to +\infty} \frac{k_n(x)}{n} = \frac{6 \log(2) \log(10)}{\pi^2} \approx 0.9702.$$

Lochs's result is valid for almost all x, with respect to the Lebesgue measure. This result allows us to say that our computer needs to take the first 10000 digits of our irrational number in order to produce its first 9702 partial quotients. If each partial quotient produced has more than 10000 digits, then we can produce with these 9702 partial quotients more than 40215 Megabytes of keystream.

4.3.2 Attack on rational numbers

We suppose that the attacker knows the number of digits taken by the computer which for example is the first 10000 digits in the irrational numbers. At the same time we suppose that the attacker knows this sequence of partial quotients $[a_n, \ldots, a_m]$ given by the continued fraction expansion of the irrational number ($9702 > m > n > 2$).
Let x be the rational number composed by the first 10000 digits of our chosen irrational number. We assert that we can extend the *results 1 & 3 & 5 & 6* on x. The fact is that the converse of Lochs's theorem permits to construct the first 10000 digits of an irrational number when we know its first 9702 partial quotients. Hence if we take randomly $(n-1)$ d_i and $(9702-m)$ c_i ($d_i \in \mathcal{N}^*, c_i \in \mathcal{N}^*$), we can construct many rational number x_i producing the same sequence of partial quotient $[a_n, \ldots, a_m]$.

$$x_i \quad =: \quad [d_1, d_2, \ldots, d_{n-1}, a_n, \ldots, a_m, c_{m+1}, \ldots, c_{9702}].$$

4.3.3 Protocol agreement

Due to the rounding errors, the use of numbers involving a lot of digits must obey some rules. For example the two correspondents must agree on their multiple precision library, on the rounding error, on the software used and on the architecture.
The problem of rounding errors can be an advantage against the attackers. If Alice and Bob have the same architecture, software ..., and the attacker Eve does not know their precision, architecture, and software, then Eve would have more difficulty in the cryptanalysis of Alice's messages.

5 Attack in a weak version of this algorithm

Because of their properties, the classical continued fraction expansions can introduce some weakness in the cryptosystem where they are used. One of the differences with our design is the use of generalized continued fractions producing big numbers (more than 5000 digits). We suppose that Alice and Bob compute the continued fraction expansion of $X_0 = \sqrt[e]{\log(z)}$ and transform it into a binary number, partial quotients by partial quotients. For example if $X_0 = [a_1, \ldots a_r, a_{r+1}, \ldots], a_1 = 00101101, a_2 = 10110110 \ldots$ then the keystream $k = \underbrace{0010110110110000\ldots}_{keystream}$ can be attacked by the following way.

5.1 Description of Khinchin's attack

Aleksandr Khinchin proved in [25] that for almost all real numbers x, the partial quotients a_i of the continued fraction expansion of x have an astonishing property: their geometric mean is a constant, known as Khinchin's constant, which is independent of the value of x. That is, for

$$x \quad = \quad a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}$$

$$\lim_{n\to\infty}\left(\prod_{i=1}^{n} a_i\right)^{1/n} = K \approx 2.6854520010\ldots$$

where K is Khinchin's constant.

The attacker Eve needs the cipher only to find a part of the message in the following steps:

1. Eve eavesdrops a long cipher text T_n.
2. She splits the cipher in bytes (because in a classical continued fraction expansion a lot of partial quotients can be coded by using a single byte).
3. Let d_i be the integer corresponding to the byte i.
4. Eve computes
$$K_1 = \lim_{n\to\infty}\left(\prod_{i=1}^{n} d_i\right)^{1/n}.$$
5. If K_1 is near the Khinchin's constant, Eve might make some hypothesis like that in the plain text structure there are a lot of zeros in the plain text. Hence, the keystream has some similarity with the cipher text.
6. Eve then builds a keystream r with the information obtained in the previous step.
7. Eve computes $q = r \oplus T_n$. The q thus found will have some similarities with the plain text.

6 Comparison and advantages

6.1 Comparison

Because our algorithm can be linked to the family of hard mathematical problems, we choose to compare our scheme to some algorithms like the ones of Blum-Blum-Shub [11] or Alexi *et al.* [64] in terms of speed, although our result can compete with others algorithms like AES in mode OFB.

Our algorithm cost is $O(t^{1+\epsilon})$ when we produce approximately θ bits of keystream (θ is the number of bits of b_i, ℓ is the number of digits extracted from our irrational number X by the computer and $t = \max(\log_2(\ell), \theta)$).

Comparison with Blum-Blum-Shub: In summary Blum-Blum-Shub algorithm computes $x^2 \bmod n$ with a cost of $O(\log_2^{1+\epsilon}(n))$ to produce a bit.

The performances were measured on a 2GHz AMD Processor and the two algorithms were programmed in Maple. As the programs were implemented without specific attention to optimization issues, the performance of these two algorithms could certainly be improved. We worked on the 6000 first digits of our irrational X, and the number of digits of b_i's was 5000. For BBS, n had 949 digits, the results are listed below.

Table 2: **Comparison with Blum-Blum-Shub.**

	Number of bits products	Computing time in seconds
BBS	150000	2.358
Our algorithm	150000	0.007

If we suppose that the number of digits taken by the computer is around 6000 digits, then the number of digits of $\max(\ell, b_i)$ is close to the number of digits of n^6, however our algorithm is quicker because we produce a big number (around 5000 digits in this test) when BBS produces one bit. Even if we use some variations of Blum-Blum-Shub which gives more bits output, our algorithm is more efficient.

Comparison with the algorithm of Alexi *et al.*: This algorithm produces a bit with a cost of $O({\log_2}^2(N)\log_2(e))$ where e and N are the RSA public keys. The algorithm of Alexi *et al.* is more expensive than Blum-Blum-Shub, hence it is clear that our algorithm is more efficient.

Likewise our algorithm could also compete with some systems like AES in mode OFB if we take large b_i's in the calculation of the partial quotients. Suppose that all b_i's have more than 10000 digits. Our algorithm will produce partial quotients having a number of digits greater than 10000.

6.2 Advantages

We can enumerate some advantages given by our cryptosystem.

1. The security of generators obtained from hard mathematical problems seems to be more interesting than that given by the other types of generators. And our algorithm is built around a hard mathematical problem.
2. This algorithm seems to be faster than those based on hard mathematical problems.
3. One of the most important aspects of this algorithm is the fact that all the parameters can be changed, so the cryptanalysis will depend partly on each implementation.
4. As we have seen it in the security analysis section, the period length of our keystream is very high.
5. The calculations can be done in parallel because there are no links between the X_i's. Hence the computing time can be less important.
6. When the entities want to begin a new exchange, Alice does not need to send a new $t \equiv z^e \bmod n$. If they stop the conversation at $X_i = \sqrt[a]{g(h_{z_i}(t_i))}$, they need to compute X_{i+1}.
7. There are other solutions when the entities want to begin a new exchange. For instance, Alice and Bob can use the parameter time in the computing of $X_i = \sqrt[e]{\log((z + day + time) + padding)}$.

7 Applications

This algorithm can be adapted to different needs. For instance in pay-per-view videos, expensive attacks on the stream cipher are not current.

This algorithm can improve the safety of communications between servers and clients. For instance, we can use our scheme in order to improve the DHCP security. The Dynamic Host Configuration Protocol (DHCP) is a protocol used by networked devices (clients) to obtain IP addresses. Thus this protocol can stand two potentials attacks: unauthorized DHCP clients and unauthorized DHCP servers. For example, we will focus on the security of DHCP clients against unauthorized DHCP servers. We suppose that we install the same private key RSA in all the DHCP servers; the public key will be installed in all the DHCP clients. Before sending its first request, the DHCP client will send $t \equiv z^e \bmod n$ in broadcast mode. The DHCP servers can compute z and finish their communication with the client. The only entities that can give a response to the DHCP client requests will be the DHCP servers.

In addition, this system can open a lot of opportunities, like an encrypted television selling its program to cell phones or computers holders. Suppose that Bob wants to watch a football match, he encrypts a message m with the public key of the television company and sends it. The television and Bob compute the keystream, the TV company will broadcast the encrypted match live. Bob will be the single entity who can decrypt this stream cipher (the keystream depends on the message m sent at the beginning), so Bob can watch this match live because the encryption and decryption can be fast.

We can apply this algorithm to protect movie industries by preventing illegal downloading. The illegal downloading will prove itself to be more difficult than it presently is even if Bob shares his keystream and the encrypted movie, because the illegal downloaders cannot take different parts of the same movie from different sources.

We can speculate on a new generation of music or movie player. In order to reduce illegal copies, we can use a simple RFID (radio frequency identification) on the compact disc music. For each compact disc the music company chooses a big number $x \in \mathcal{Z}$ and a positive integer n such that $n \geq 3$. As in our scheme the company encrypts the sound with the continued fraction expansion of $\sqrt[n]{\log(x)}$; in parallel, the company writes in the RFID the numbers x and n. When someone buys the compact

disc, the seller activates the RFID (an attacker could steal an encrypted music with an inactivate RFID). When the buyer puts the compact disc in the player, the black box of the music player reads from the RFID the numbers n, x and compute the keystream. This proposition can reduce the illegal copies and the exchange in peer to peer website because each compact disc has its own key and the simple user will not be able to copy or to activate the RFID.

8 Conclusion

In this paper we introduced an algorithm using a public key cryptosystem to start up a stream cipher. This system is easy to modify and implement. The use of generators from the theory of continued fractions can be a good alternative to the family of pseudo-random generators. Most of the members of this family are slow, impracticable or weak. It could be interesting to study in the future the best irrationals which we can use in this algorithm. Likewise it may prove useful to generalize some properties of classical continued fractions to the generalized continued fractions along the lines of the works of Khinchin [25], Levy [36], Lochs [38]. In addition to the benefits we foresee for the stream ciphers systems, this encryption scheme could give rise to new methods of cryptanalysis.

References

[1] M. Ajtai *Generating hard instances of lattice problems.* In Proc. of 28th STOC, 99-108, 1996.

[2] M. Ajtai *The shortest vector problem in L_2 is NP-hard for randomized reductions* In Proc. of 30th STOC, 10-19, 1998.

[3] M. Ajtai, AND C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence* In Proceedings of the 29th ACM Symposium on Theory of Computing, 284-293, 1997.

[4] R. Alvarez, J.J. Climent, L. Tortosa, A. Zamora *An efficient binary sequence generator with cryptographic applications*, Applied mathematics and computation , vol. 167, no1, 16-27, 2005.

[5] G. Alvarez and S. Li *Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems*,International Journal of Bifurcation and Chaos, vol. 16, no. 8, 2129-2151, 2006.

[6] F. Armknecht, C. Elsner, M. Schmidt *Using the Inhomogeneous Simultaneous Approximation Problem for Cryptographic Design*, IACR Cryptology ePrint Archive 2010: 302, 2010

[7] K. Ayadi , M. Hbaib and F. Mahjoub *Family of Formal Power Series with Unbounded Partial Quotients*, International Journal of Algebra, Vol. 4, no. 28, 1383 - 1392, 2010.

[8] M. Bellare , P. Rogaway *Random oracles are practical: a paradigm for designing efficient protocols* , Conference on Computer and Communications Security,62-73, 1993.

[9] M. Blum, S. Micali *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput., vol 13, No 4, 1984, issn 0097-5397, 850–864.

[10] M. Boesgaard , M. Vesterager, and E. Zenner *The Stream Cipher Rabbit. New Stream Cipher Designs*. Lecture Notes in Computer Science, 69-83, 2008.

[11] L. Blum, M. Blum, M. Shub *A simple unpredictable pseudo-random number generator*, SIAM J. Comput., vol 15, No 2, 1986, issn 0097-5397, 364–383.

[12] E.B. Burger, R. Tubbs *Making transcendence transparent: An intuitive approach to classical transcendental number theory*, Springer-Verlag, 2004.

[13] D. Coppersmith *Small solutions to polynomial equations, and low exponent RSA vulnerabilities.* J. Crypt. 10, 4, 233-260, 1997.

[14] C. Elsner, M. Schmidt *KronCrypt - A New Symmetric Cryptosystem Based on Kronecker's Approximation Theorem*, IACR Cryptology ePrint Archive 2009: 416, 2009.

[15] D. Barbolosi, C. Faivre *Sur les grands quotients partiels du développement en fraction continue*, Journal of Number Theory, Vol 106, 299-325, 2004.

[16] H. C. Geon and C. Kim *The Khintchine constants for generalized continued fractions*, Applied mathematics and computation , vol. 144, no2-3, pp. 397-411, 2003.

[17] M. Beeler, R.W. Gosper, R. Schroeppel *MIT Artificial intelligence memo 239*, 1972.

[18] J. Hastad, R. Impagliazzo, L.A. Levin, M. Luby *A pseudo-random generator from any one-way Function*, SIAM Journal on Computing, vol 28, Nř 4, 1999, 1364-1396.

[19] Q. Hu, *Stream Ciphers and Linear Complexity*, Master thesis, National University of Singapore, 2008.

[20] A. M. Kane *On the use of Continued Fractions for Stream Ciphers*, In Proceedings of Security and Management 2009, Las Vegas, USA.

[21] A. M. Kane *On the use of continued fractions for electronic cash*, in International Journal of Computer Science and Security, Vol 4, Issue 1, , (2010), 136-148.

[22] A. M. Kane *On the use of continued fractions for mutual authentication*, in International Journal of Information Security Science, Vol 1, Issue 3, 88-99, 2012.

[23] R. Kannan, A.K. Lenstra and L. Lovasz *Polynomial Factorization and Nonrandomness of Bits of Algebraic and Some Transcendental Numbers* Mathematics of Computation, 50, 235-250, 1988.

[24] A. Kanso *Cryptosystems based on continued fractions*, Security and Communication Networks, Vol 4, 1199Ű1211, 2011.

[25] A. Ya. Khinchin *Metrische Kettenbruchprobleme*, Compositio Mathematica, 1 (1935), 361-382.

[26] A. Khintchine, *Continued fractions*, Gosudarst. Isdat. Tecn.-Teor. Lit, Mosow-Liningrad, second ed., (1949) (MR 13_=274f).

[27] A. Khintchine, *Kettenbruche*, B. G. Teubner, (1956) (MR 18_=274f).

[28] S-J. Kim , K. Umeno , A. Hasegawa *On the NIST Statistical Test Suite for Randomness*,IEIC Technical Report Vol.103;N.499, 21-27, 2003.

[29] D. E. Knuth *The art of computer programming Volume 2: Seminumerical algorithms (3rd Edition)*, Addison-Wesley, 1997.

[30] L.Kocarev, P. Amato , and G. G.Rizzotto , *Method of generating a chaos-based pseudo-random sequence and a hardware generator of chaos-based pseudo random bit sequences* United States Patent No. US7779060B2, Date of Patent 17 Aug 2010.

[31] P. C. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA,DSS, and Other Systems*,CRYPTO, 104-113, 1996.

[32] Kuzmin, R. O. *Sur un problème de Gauss.* Anni Congr. Intern. Bologne 6, 83-89, 1928.

[33] A. G. B. Lauder, *Continued fractions and sequences*, Ph.D. thesis, University of London, 1999.

[34] A. K.Lenstra, H. W. Lenstra Jr , and L. Lovasz *Factoring polynomials with rational coefficients* Math. Ann., 261(4), 515-534, 1982.

[35] D. Lekkas, D. Spinellis *Implementing regular cash with blind fixed-value electronic coins*, Computer Standards & Interfaces, vol 29 No 3, 2007, 277-288.

[36] P. Levy *Sur les lois de probabilite dont dependent les quotients complets et incomplets d'une fraction continue*, Bull. Soc. Math. 57 (1929) 178-194.

[37] S. Burman, D. Mukhopadhyay and K. Veezhinathan *LFSR Based Stream Ciphers Are Vulnerable to Power Attacks,* Progress in Cryptology - INDOCRYPT 2007, 384-392.

[38] G. Lochs *Vergleich der Genauigkeit von Dezimalbruch und Kettenbruch*, Abh. Math. Sem. Univ. Hamburg 27, 1964, 142-144.

[39] M. H. Au, W. Susilo, Y. Mu *Practical Anonymous Divisible E-Cash From Bounded Accumulators*, Financial Cryptography and Data Security 2008, Lecture Notes in Computer Science 5143, Springer-Verlag, 2008, 287 - 301.

[40] P. Marquardt , P. Svaba and T. V. Trung, *Pseudorandom number generators based on random covers for finite groups*, Design, Codes and Cryptography, 64(1-2):209-220 , 2012.

[41] A. Masmoudi, W. Puech, M.S. Bouhlel, *An Efficient PRBG Based on Chaotic Map and Engel Continued Fractions*, Communications and Network, Vol. 2, 1141, 2010.

[42] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone *Handbook of applied cryptography*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1997.

[43] R. Merkle and M. Hellman, *Hiding Information and Signatures in Trapdoor Knapsacks*, IEEE Trans. Information Theory, 24(5), 525-530, 1978.

[44] S. Murphy *The power of NIST's Statistical Testing of AES Candidates, The Third AES Candidate Conference* , 2000. http://csrc.nist.gov/aes/

[45] M. Naor, O. Reingold *From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs*,Advances in Cryptology - CRYPTO '98, LNCS, Springer-Verlag, 1998, 267-282.

[46] H. Niederreiter, *Continued fractions for formal power series, pseudorandom numbers, and linear complexity of sequences*, General algebra, Proc. Conf., Salzburg/Austria 1986, 221-233 1987.

[47] H. Niederreiter, *Sequences with almost perfect linear complexity profile*, Advances in Cryptology - EUROCRYPT' 87: Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, vol. 304, 37-51, 1988.

[48] P. Q.Nguyen, *Lattice Reduction Algorithms: Theory and Practice*,EUROCRYPT, 2-6, 2011.

[49] P. Q.Nguyen and J. Stern *The Two Faces of Lattices in Cryptology*,CaLC, 146-180, 2001.

[50] *A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications*, NIST Special Publication 800-22, National Institute of Standards and Technology, 2001, http://csrc.nist.gov/rng.

[51] C. D. Olds *Continued Fractions*, Random House, 1963.

[52] O. Perron *Die Lehre Von Den Kettenbrüchen*, 3rd ed. 1954, (Stuttgart: Teubner, 1954).

[53] J. Pieprzyk, H. Ghodosi, C. Charnes, R. Safavi-Naini *Cryptography based on transcendental numbers,* Information Security and Privacy, Lecture Notes in Computer Science, Vol. 1172, Proceedings, First Australasian Conference on Information Security and Privacy, ACISP'96, Wollongong, Australia, 1996.

[54] R. Rivest, A. Shamir, L. Adlemnan *A method for obtaining digital signature and public-key cryptosystems*, Commun. ACM 21, 1978, 120-126.

[55] Rueppel, R. *Stream Ciphers, Chapter 2 of Simmons, G., Contemporary Cryptology: The Science of Information Integrity*, IEEE Press, 1994, 65-134. Piscataway, NJ, USA.

[56] A. Shamir *A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem* proc of 23rd FOCS, 145-152, 1982.

[57] B. Schneier *Applied cryptography (2nd ed.): protocols, algorithms, and source code in C*, John Wiley Sons, Inc., (1995).

[58] C. Shine, *Method and apparatus of using irrational numbers in random number generators for cryptography* United States Patent, Application No. 10/190455, Application Date: Jul 3 2002.

[59] D. M. Smith *Efficient multiple-precision evaluation of elementary functions*, Mathematics of Computation, vol 52, No 185, 1989, 131-134.

[60] J. Soto *Randomness Testing of the Advanced Encryption Standard Candidate Algorithms* 1999. http://csrc.nist.gov/aes/

[61] J. Soto and L. Bassham *Randomness Testing of the Advanced Encryption Standard Finalist Candidates*, NIST (2000). http://csrc.nist.gov/aes/

[62] M. Trott, *Finding Trott Constants*, The mathematica Journal Volume 10, Issue 2 , September 2006.

[63] H. Wang, and Y. Zhang, *Untraceable off-line electronic cash flow in e-commerce*,Proceedings of the 24th Australasian conference on Computer science, 191-198, 2001.

[64] A. Werner, B. Chor, O. Goldreich, and C. P. Schnorr *RSA/Rabin bits are 1/2 + 1/poly(log N) Secure,* 25th Annual Symposium on Foundations of Computer Science, Singer Island, Florida, USA, 1984, 449-457.

[65] M. J. Wiener *Cryptanalysis of short RSA secret exponents*, IEEE Transactions on Information Theory, 36, 553-558, 1990.

[66] Implementation AES *WIKI AES* http:http://en.wikipedia.org/wiki/File:Cbc128192256.jpg.

9 Remarks on this paper

There are essentially two problems that make it difficult to prove the security of algorithms based on generalized continued fractions. The first one is due to the fact that most of the existing conjectures or theorems are based on classical continued fractions.

The second problem is that any irrational number admits an infinity of generalized continued fraction expansion (following the partial numerators) which make it difficult to have common rules on the distribution of the partial quotients.

9.1 Remark 1: Statistical tests presented in this paper

The results presented in this paper are obtained with the function $gfc()$ defined below. Throughout this section, $gfc()$ defined the function that we used to perform our tests on sagemath. We obtained the partial quotients by using sagemath with the following code.

$$sage: def\ \ gcf(x,\, n,\, b):$$

$$...v = [\,]$$
$$...v.append(floor(x))$$
$$...r = x$$
$$...for\, i\, in\, range(1, n):$$
$$...r = (b/(r - v[i-1]))$$
$$...v.append(floor(numerical\_approx(r, prec = 100000)))$$
$$...return\, v$$

In order to generate the five millions bits we used the following irrationals: $log(10^{45})$, and $(log(7^{12}))^{(1/5)}$.
Which corresponds to the following lines:
$sage: time\, h1 = gcf(log(10^{45}), 200, (10^{20000}) * log(17))$
$sage: time\, h1 = gcf((log(7^{12}))^{(1/5)}, 5, (10^{20000}) * log(17))$

9.2 Remark 2: Sizes of the partial quotients

In our tests we chose as partial numerator a large irrational number.

The choice of the size of the digits is due to several factors including the aim to avoid the classical continued fractions (when the numerator is equal to 1) and their inherent defects but also the desire to reduce the computation time.

For example, the existence of the Khinchin constant or the Gauss Kuzmin distribution [32] can permit to launch attacks on the partial quotients obtained from the classical continued fraction expansion. When all the partial numerators are equal to 1, we have for example in [15] the following distribution 47% of 1, 17% of 2, 9% of 3 ...

The current state of knowledge in this area does not allow us to exhibit a Gauss Kuzmin distribution for the generalized continued fraction; however if there is a frequency of occurrence when the partial numerator is always equal to 1, we can think that there is necessarily a frequency of occurrence when the partial numerator is greater.

That is why we refused to take a partial numerator equal to 2, 3, ... and we opted for a large irrational number taken randomly, which would protect us from attacks using the generalization of the Khinchin's constant or the Gauss Kuzmin distribution.

Without being able to prove it, we conjecture that the frequency of occurrence of partial quotients obtained from a generalized continued fraction expansion (when the partial numerator is a large number) will tend towards zero for all partial quotients.

This means that even if we had at our disposal a frequency of occurrence for partial quotients obtained from a generalized continued fraction expansion (when the partial numerator is a large irrational number), this frequency could not be used to attack our algorithm since the frequency of occurrence is the same for many partial quotients (hence the use of the interval $[S, P]$ in my paper, since we affirm that all the numbers that are in this interval have approximately the same frequency of occurrence).

In summary, our conjecture is: If the partial numerator is a large irrational number chosen randomly and if the partial quotients are obtained from the generalized continued fraction expansion of an irrational number x taken randomly in Γ then:

1. There exists a large interval $[S, P]$ which contains most of the partial quotients and where an exhaustive search is not possible in polynomial time.

2. The probability of having in polynomial time a collision in the continued fraction expansion is close to zero, which contributes to the good distribution of partial quotients in the interval $[S, P]$ (the more the size of the partial quotients is large, the less we have risk of collision).

3. All elements of the interval $[S, P]$ have nearly the same probability of occurrence as partial quotient (probability close to zero).

Conclusion of our conjecture: There exists a large interval $[S, P]$, such that the distribution of partial quotients in the continued fraction expansion of x is indistinguishable by all polynomial-time statistical tests from the uniform distribution of integers in the interval $[S, P]$.

There are two remaining questions: how do we choose the interval $[S, P]$ and what can we say about the partial quotients which are not in the interval $[S, P]$?

In practical cases, there is no need to determine the interval $[S, P]$ when we choose a large irrational number as partial numerator. Presumably, some partial quotients will not be in this interval, since the Khinchin's conjecture [7, 27, 26] for classical continued fractions asserts that: if x is a real algebraic number of degree > 2 then x has a continued fraction expansion whose the sequence of partial quotients is unbounded.

This implies that there are partial quotients which may not be in the interval $[S, P]$ (that is to say greater than P) but these will not affect the security of our algorithm because of their negligible number and that these are also unpredictable.

In our paper the partial numerator b_i is composed by 5000 digits on its floor part, then, we estimate that the smallest partial quotients is greater than 2^{16610} and most of partial quotients are smaller than 2^{16623}. However we would like to improve this assertion here because by our conjecture, we are sure that this interval $[S, P]$ exists, we know that the lower bound is slightly higher than the value of the partial numerator. On the other hand, the upper bound (2^{16623}) which we gave in our paper can be refined.

Our objective to base the proof of the safety of our algorithm on partial quotients belonging to a specific interval is pretty close to the approch taken in [14] published after our paper and where the authors take their partial quotients over a interval of length K where K is equal to $2^{m-1} - 2$, m denotes the minimal number of additional bits coming from the process of solving an inhomogeneous diophantine inequality.

In our advice for the implementation of our algorithms, the choice of the size of the partial numerators can be as follows:

- For the stream cipher: irrational number with at least 5000 digits in the integer part.
- For e-cash: irrational number with 100 digits in the integer part.
- For authentication: irrational number with 100 digits in the integer part.

In addition to the reasons mentioned above on the size of the digits, these advices are also guided by the aim to increase the efficiency of algorithms in computation time.

We may note that the choice of large partial quotients can improve the computation time by comparing the following calculations:

If we work on sagemath with partial quotients composed by approximately 30000 digits, with this command:

$sage : timeh1 = gcf((log(7)), 2, (10)^{(30000)} * log(17))$, we obtain in 0.01s one partial quotient with more than 30000 digits.

While if we work with partial quotients composed by approximately two digits, we need 6.85s in order to obtain 500 partial quotients with this command:

$sage : timeh1 = gcf((log(7), 500, (10) * log(17))$. Hence, it would take approximately 205.5s to obtain the 30000 digits.

Note

1. The conjecture stated above is intended to demonstrate that the pseudo random generator is unpredictable. However, in view of the statistical tests, the user does not need to verify if the conjecture is true since the statistical tests showed that we will get a pseudo random stream when we choose randomly a large irrational number as partial numerator and when $x \in \Gamma$.

2. We did not choose partial quotients greater than 5000 digits by a necessity of compromise between time and space.

3. It will be very difficult to determine the minimum size of partial numerators which I call "large irrational number taken randomly" due to the fact that there are an infinite number of generalized continued fraction expansions for one irrational number. However, after having tested several partial numerators composed by at least 5000 digits, we think that an irrational number with 5000 digits in its integer part may be considered as a "large irrational number".

9.3 Remark 3: Competition with AES

Our algorithm could compete with AES.

By reducing the precision in our program (see Appendix), we obtained a partial quotient composed by more than 840,000 digits in 0.01s with this following command:

$sage : timeh1 = gcf((log(7)), 2, (10)^{(840000)} * log(17))$.

If the time of calculating a partial quotient composed by 840,000 digits is 0.01 s on sagemath then we have approximately 2,700,000 bits in 0.01s (given that RSA 576 bits corresponds to 174 digits) which correspond to 270 million bits/s.

If we admit that on average, the AES produces 60MiB/s, which is equivalent approximately to 60 * 8 million bits = 480millionbits/s.

We can say that AES is not twice faster than our algorithm, so we can conclude by the following statement: "Properly optimized our algorithm can be competitive with AES".

Our calculations were obtained with sagemath on the website http://www.sagenb.org with the program indicated in the appendix. The results of AES were obtained from Wikipedia [66].

9.4 Remark 4 : Unpredictability and next-bit test

In order to remain in line with the definition of the unpredictability given in [28, 30], we consider in this study, that a Pseudo Random Bit Generator is unpredictable if it passes the next-bit test.

As presented in [28, 30]: A pseudo-random bit generator is said to pass the next-bit test if there is no polynomial-time algorithm which, on input of the first l bits of an output sequence s, can predict the $(l+1)$st bit of s with probability significantly greater than $1/2$.

9.5 Remark 5: Elimination of partial quotients in our stream cipher

Why did we eliminate first partial quotients in our stream cipher?

We chose to cut some partial quotients (20) to ensure that our algorithm cannot be attacked, one day, by a generalization of the Khinchin's constant [16] or by this type of attack introduced by Kannan et al. [23].

Kannan, Lenstra and Lovasz have showed that the binary expansions of algebraic numbers do not form secure pseudorandom sequences; given sufficiently many initial bits of an algebraic number, its minimal polynomial can be reconstructed, and therefore the further bits of the algebraic number can be computed.

However, as we said it before, the number of partial quotients eliminated does not have a real influence on the security of this algorithm because any partial quotient can take infinitely many values. And we know that it is impossible to find an unknown which can take an infinite number of values, so we could cut less than 20 partial quotients.

9.6 Remark 6: One the use of irrational numbers in cryptography

We state that we are confident that some irrational numbers can be used in cryptography. It is due to the fact there are already some contributions (even some patents) in this area [58].

9.7 Remark 7: The Lochs result

Why did we use the Lochs result in our paper?

The problem is due to the fact that the result of Lochs is valid for classical continued fractions while in our study we propose the use of generalized continued fractions. The way in which we used the Lochs result is dictated by the need to show that there is a relationship between the size of the digits taken and the number of partial quotients which we can obtain from these digits. This relationship is not very important in our paper and doesn't impact on the security of our keystream. In our study, we consider that the Lochs constant for generalized continued fractions is approximately the same as for classical continued

fractions, which is also said in this contribution [62]: "Lochs' theorem holds in its classic form only for simple continued fractions. But experience shows that non simple continued fractions behave similarly".

Otherwise if the Lochs constant for generalized continued fractions is different, then the length of the keystream which we presented should simply be refined.

9.8 Remark 7: The Khinchin Attack

Why did we present the Khinchin attack?

There are two reasons for that:

1. If one excludes the theory of dynamical systems, this is perhaps the first use of this amazing constant in a practically-oriented algorithm.

2. In our paper, we chose to present an attack using the Khinchin constant because the other attack that we have identified already exist (we could use a frequency attack which exists since the 9th century). In addition, an attack using Khinchin's constant can be used to complete the most obvious attacks.

9.9 Remark 8: The choice of parameters

The choice of parameters a, g, h_{z_i} and t_i (in section 2.3.1) will depend on the user, since any irrational number in Γ can be selected, for example one can take (as we do it) the parameter $a = e$ (a will correspond here to Bob's public key), $h_{z_i}(t_i)$ will be equal to z and the function g will correspond to the log.

9.10 Remark 9: Comparison with Blum Blum Shub

Why did we compare our stream cipher with Blum Blum Shub?

Following the example of recent algorithms which were compared with Blum Blum Shub [4, 40], our goal was to create an algorithm as secure as BBS and much faster than BBS.

9.11 Remark 10: The Timing attack

Introduced by Kocher [31], the timing attack uses the computation time of cryptographic operations to try to discover the encryption key. The timing attack cannot be effective in this algorithm when the partial numerator is a large irrational number taken randomly.

9.12 Remark 11: The NIST test suite

Issued by the National Institute of Standards and Technology (NIST), the NIST test suite consists on a battery of statistical tests designed to detect non-randomness in binary sequences constructed using random number generators and pseudo-random number generators utilized in cryptographic applications. It is composed by a sequence of 16 statistical tests. We can see the utility of this test through validation of AES [60, 61, 28] or the various algorithms which use it for validation [40, 10]. The 16 statistical tests are:

Frequency (Monobits) test

The purpose of this test is to determine whether that number of ones and zeros in a sequence is approximately the same as would be expected for a truly random sequence.

Test for frequency within a block

The purpose of this test is to determine whether the frequency of ones is an M-bit block is approximately M/2.

Runs Test

The focus of this test is the total number of zeros and ones runs in the entire sequence, where a run is an uninterrupted sequence of identical bits. A run of length k means that a run consists of exactly k identical bits and is bounded before and after with a bit of the opposite value.

Test for the longest run of ones in a block

The focus of the test is the longest run of ones within M-bits blocks. The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence.

Random binary matrix rank test

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

Discrete Fourier transform (spectral) test

The focus of this test is the peak heights in the discrete Fast Fourier Transform. The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness.

Non-overlapping (aperiodic) template matching test

The focus of this test is the number of occurrences of pre-defined target substrings. The purpose of this test is to reject sequences that exhibit too many occurrences of a given non-periodic (aperiodic) pattern.

Overlapping (periodic) template matching test

The focus of this test is the number of pre-defined target substrings. The purpose of this test is to reject sequences that show deviations from the expected number of runs of ones of a given length.

Maurer's universal statistical test

The focus of this test is the number of bits between matching patterns. The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information. An overly compressible sequence is considered to be non-random.

Linear complexity test

The focus of this test is the length of a generating feedback register. The purpose of this test is to determine whether or not the sequence is complex enough to be considered random. Random sequences are characterized by a longer feedback register. A short feedback register implies non-randomness.

Serial test

The focus of this test is the frequency of each and every overlapping m-bit pattern across the entire sequence. The purpose of this test is to determine whether the number of occurrences of the 2m m-bit overlapping patterns is approximately the same as would be expected for a random sequence. The pattern can overlap.

Approximate entropy test

The focus of this test is the frequency of each and every overlapping m-bit pattern. The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a random sequence.

Cumulative sum (cusum) test

The focus of this test is the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences.

Random excursions test

The focus of this test is the number of cycles having exactly K visits in a cumulative sum random walk. The cumulative sum random walk is found if partial sums of the (0,1) sequence are adjusted to (-1, +1). A random excursion of a random walk consists of a sequence of n steps of unit length taken at random that begin at and return to the origin. The purpose of this test is to determine if the number of visits to a state within a random walk exceeds what one would expect for a random sequence.

Random excursions variant test

The focus of this test is the number of times that a particular state occurs in a cumulative sum random walk. The purpose of this test is to detect deviations from the expected number of occurrences of various states in the random walk.

Lempel-Ziv compression test

The focus of this test is the number of cumulatively distinct patterns (words) in the sequence. The purpose of the test is to determine how far the tested sequence can be compressed. The sequence is considered to be non-random if it can be significantly compressed. A random sequence will have a characteristic number of distinct patterns.

9.13 Remark 12: Modification on *Result 1*

In order to remain in accordance with the use of generalized continued fractions that we recommend, α_ρ which we used in *Result 1* can be written as follows:

$$\alpha_\rho = r_1 + \cfrac{b_1}{\ddots\, r_m + \cfrac{b_m}{a_{m+1} + \cfrac{b_{m+1}}{\ddots\, a_{m+n-1} + \cfrac{b_{m+n-1}}{a_{m+n} + \cfrac{b_{m+n}}{\theta}}}}}$$