On the Use of Continued Fractions for Mutual Authentication

Amadou Moctar Kane

Département de Mathématiques et de Statistique, Université Laval, Québec G1V 0A6 Canada

e-mail: amadou-moctar.kane.1@ulaval.ca

Abstract- The purpose of this paper is to present an improvement of the Needham-Schroder public key protocol. This new protocol will use partial quotients issue from the continued fraction expansion of some irrational numbers to secure the authentication between two principals. We introduce a new approach in the use of pseudo-random numbers, because besides using these numbers to provide uniqueness and timeliness guarantees, we use them to ensure that nobody can guess the identity of the sender. We also keep this new protocol secure against the Lowe attack, without taking the solution suggested by Lowe. This protocol remains fast although we compute some partial quotients during the authentication process.

Keywords- Authentication, continued fraction, cryptography, Needham-Schroeder protocol.

1. Introduction

The alarming increase in victims of impersonation and the need to secure emerging tools as cloud computing imply the necessity to improve existing authentication protocols.

As defined by Menezes et al [11], entity authentication is the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (i.e., is active at, or immediately prior to, the time the evidence is acquired).

The mutual authentication, also called two-way authentication, is a process in which both entities authenticate each other. In this paper, it is the definition that we will adopt.

There exits several authentication protocols including: Kerberos [16], Needham-Schroeder [12], Wide Mouthed Frog [3], Woo-Lam [18]. Some protocols are based on others such as Kerberos which is based on Needham-Schroeder.

The Needham-Schroeder protocol has two variants, the first one is based on symmetric cryptography and the second one is based on public key cryptography. In this paper, we will focus on the version based on the public key cryptography.

This protocol has been widely studied [5] since 1978 but the greatest improvement was made in 1995, when Lowe [10] proved that this protocol was sensitive to the impersonate attack. The improved version Needham-Schroeder-Lowe seems to be strong until now and currently most studies, on this protocol are oriented on the security proof.

The improvement of the Needham-Schroeder protocol introduced in this paper will be partly based on the fact that the continued fraction expansion of an irrational number is unique.

Also, it will be based on the difficulty of retrieving an irrational number from the sole knowledge of a part of its continued fraction expansion.

Continued Fractions: An expression of the form

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

is called a generalized continued fraction. Typically, the numbers $\mathrm{a}_1, \mathrm{b}_1, \ldots$ may be real or complex and the expansion may be finite or infinite.

We will avoid the use of the continued fraction expansions involving $\mathrm{b}_\mathrm{i} = 1$ 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:

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}$$

$$\alpha \coloneqq a_0, a_1, a_2 \ldots$$

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 with a given predictable continued fraction expansion are presented in [1]-[8]).

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 fractions properties:

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

Continued fractions were widely studied by Olds [13] and Perron [14], but cryptographic views are not explored by number theory specialists except in some areas like RSA cryptanalysis. In addition to the RSA cryptosystem, continued fractions are used to build a stream cipher [6] or to set up a e-cash scheme [7].

This paper is organized as follows: In Section 2 we will propose and demonstrate some results concerning continued fractions; in Section 3, we will introduce the Needham-Schroeder protocol. In Section 4 we present our new protocol, and before the conclusion, we will compare to the two algorithms.

Preliminaries

The Result 1 and the Result 2 have already been presented in [6]-[7].

The Result 2 will exhibit an example of irrational number which we can use in our protocol.

The Result 1 shows that the intruder will not succeed if he tries to impersonate the principal in the last attack of *Section 4.5*.

Notation

In this paper we shall use the notation $E(X{:}\,Y)$ to denote the result of encrypting message plaintext X with key K.

We have three principals namely the first entity Alice (A), the second entity Bob (B), and the Intruder (I), we also have a server (trusted third party) (AS). PK_A will be the public key of Alice, PK_B will be the public key of B, and PK_I will be I's public key. In the same manner SK_A will be A's private key, SK_B will be B's private key and I's private key will be SK_I. The notation I(A) denotes the principal I acting in the role of A.

A message may have several components and message components will be separated by commas. Thus $E(\{N_a, A\}{:}\,PK_B)$ denotes that the message encrypted by the key PK_B is composed by a nonce N_a and a principal identifier A.

Let $\alpha \epsilon \Gamma$ such that $a_1, \dots, a_m, \dots, a_{m+n}, \dots$ is the continued fraction expansion of α; m and n are two integers such that $m > 1, n \geq 1$. We denote by δ the vector made with the n partial quotients following the m first partials quotients in the continued fraction expansion.

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

Proof. Let $\alpha \epsilon \Gamma$. We suppose that we know a given part $a_{m+1}, \dots, a_{m+n}$ of α's continued fraction expansion. Can we find α with the knowledge of these n partial quotients?

The answer is negative, because there exists 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 $a_{m+1}, \dots, a_{m+n}$ 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 $(r_1, r_2, \dots, r_m)$ and to define α_ρ to be

$$\alpha_\rho = r_1 + \cfrac{1}{\ddots_{r_m} + \cfrac{\ddots_1}{a_{m+1} + \cfrac{1}{\ddots_{a_{m+n-1}} + \cfrac{\ddots_1}{a_{m+n} + \cfrac{1}{\theta}}}}}$$

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

Proof .Assume that A is a real algebraic number such that $A > 1$, then $\log_e(A)$ is transcendental number by Corollary 3.6 of [2].

If we suppose that $X = \sqrt[r]{\log_e(A)}$ is an algebraic number, then X^r is a algebraic number, which is absurd because $X^r = \log_e(A)$ and $\log_e(A)$ is transcendental.

Remark

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

Remark

Due to the rounding errors, the use of continued fractions must obey some rules. For example Alice and Bob must agree on their multiple precision library, on the rounding error, on the software used and on the architecture.

2. The Needham-Schroeder Protocol

2.1 The Needham-Schroeder Protocol

As defined in [12], the public key protocol consists on the following seven steps:

Step 1: A → AS

The exchange opens with A consulting the authentication server to find B's public Key.

Step 2: AS responds with: $E(\{PK_B, B\}: SK_{AS})$.

Where SK_{AS} is the authentication server's secret key, PK_B is B's public key and B is B's identity.

Step 3: A sends to B the following
$E(\{N_a, A\}: PK_B)$.

This step is for the communication with B to be initiated. This message, which can only be understood by B indicates that someone purporting to be A wishes to establish communication with B. B decrypts the message with his private key and then finds the nonce N_a chosen by A.

Steps 4 & 5: B finds A's public key (PK_A) with steps similar to 1 & 2.

Step 6: At this point B return the nonce N_a, along with a new nonce N_b, to A, encrypted with A's public key (E ($\{N_a, N_b\}: PK_A$)).

Step 7: At the end, A returns the nonce N_b to B, encrypted with B's public key.

The protocol can be described as follows:

1. A → AS : A, B;
2. AS → A : $E(\{PK_B, B\}: SK_{AS})$;
3. A → B : $E(\{N_a, A\}: PK_B)$;
4. B → AS : B, A;
5. AS → B : $E(\{PK_A, A\}: SK_{AS})$;
6. B → A : E $(\{N_a, N_b\}: PK_A)$;
7. A → B : E $(\{N_b\}: PK_B)$.

2.2 The Needham-Schroeder-Lowe Protocol

In [10], Lowe shows that an attack on the protocol allows an intruder I to impersonate another agent A to set up a false session with B. In this attack, we can ignore the interaction with the server because this does not have a real influence on this attack. The attack involves two simultaneous runs of the protocol: in run 1, A establishes a valid session with I; and in run 2, I impersonates A to establish a fake session with B.

In Step 1.3, A starts to establish a normal session with I, sending him a nonce N_a.

In Step 2.3, the intruder impersonates A to try to establish a false session with B, sending it the nonce N_a obtained in the previous message.

B responds in the message 2.6 by selecting a new nonce N_b, and trying to return it along with N_a, to A. The intruder therefore forwards the message to A in the step 1.6. A decrypts the message to obtain N_b, and returns this to I in message 1.7.

I can then decrypt this message to obtain N_b which he returns to B in message 2.7. Hence B believes that A has correctly established a session with him.

This attack can be described as follows:

1.3 A → I : $E(\{N_a, A\}: PK_I)$;
2.3 I(A) → B: $E(\{N_a, A\}: PK_B)$;
2.6 B → I(A):$E(\{N_a, N_b\}: PK_A)$;
1.6 I → A: $E(\{N_a, N_b\}: PK_A)$;
1.7 A → I $E(\{N_b\}: PK_I)$;
2.7 I(A) → B $E(\{N_b\}: PK_B)$.

In the same paper Lowe showed that it is easy to change the protocol so as to prevent the attack; for this purpose he included the responder's identity in message 6 of the protocol.

Hence, the step 2.6 of the attack would become $E(\{B, N_a, N_b\}: PK_A)$ and the intruder cannot successfully replay this message in Step 1.6.

3. Our contribution

The improvement proposed here is based on the work of Lowe, since we have solved the previous attack without his solution described above.

As for the Needham-Schroeder algorithm we suppose that communications are carried on an insecure channel.

We denote by $\mathrm{FC(X, Y)}$ the first ten partial quotients issue from the continued fraction expansion of the irrational number X and where Y is a vector of ten b_i ‘s (we recall that the b_i ‘s are used during the computation of the generalized continued fraction).

We denote by $\mathrm{FC}'(\mathrm{X}, \mathrm{Y})$ the nine partial quotients following the first one in the continued fraction expansion of X (the first partial quotient is ignored in the authentication protocol).

We denote by Y_A, Y_B, or Y_I the vectors used in the computation of the generalized continued fraction as described in the introductory paragraph.

Y_A is computed as follows:

- We apply the hash function SHA1 on A’s public key and we obtain SHA1(PK$_A$).
- We divide the string obtain in the previous step in ten part, and we obtain $Y_A = (b_1, b_2, \dots, b_{10})$.

Y_B is computed like Y_A but we apply the hash function on B’s public key instead of A (SHA1(PK$_B$)).

Y_I is obtained in the same manner as we apply the hash function on I’s public key SHA1(PK$_I$).

We denote by $Y_u Y_v$ the concatenation of Y_u and Y_v, for example, if $Y_u = (a_1, \dots, a_{10})$ and $Y_v = (b_1, \dots, b_{10})$ then $Y_u Y_v = (a_1 b_1, \dots, a_{10} b_{10})$.

The vector $Y_u Y_v$ will be used in the computation of the partial quotients if the sender of the nonce is u and the receiver is v. For example, if the sender of the nonce is A and the receiver is B, then B will use the vector $Y_A Y_B$ to compute the partial quotients.

3.1 The new protocol

The new protocol is conducted in accordance with the following steps:

Step 1:

A chooses randomly a nonce N_a, encrypt it with B's public key (PK$_B$) and sends it to B.

Step 2:

B calculates the first 10 partial quotients $FC(\sqrt[3]{\log_e N_a}\,; Y_A Y_B)$, ignores the first partial quotient, composes a message with the 9 remaining partial quotients $(FC'(\sqrt[3]{\log_e N_a}\,; Y_A Y_B))$, adds a nonce N_b (chosen randomly) encrypts the message with the public key of A and sends it to A.

Step 3:

A computes $FC'(\sqrt[3]{\log_e N_a}\,; Y_A Y_B)$ and verifies that the nine partial quotients received from B are correct. If these partial quotients are correct, A computes the first 10 partial quotients $FC\left(\sqrt[3]{\log_e N_b}\,; Y_B Y_A\right)$, ignores the first partial quotient and sends the 9 remaining in a message encrypted with B’s public key.

The new protocol can be described as follows:

B →A :“ Hi I am B”;

A → B : $E(\{N_a\}: PK_B)$;

B → A : $E\ (\{FC'\left(\sqrt[3]{log_e N_a}\,; Y_A Y_B\right), N_b\ \}: PK_A)$;

A →B : $E\ (\{FC'\left(\sqrt[3]{log_e N_b}\,; Y_B Y_A\right)\ \}: PK_B)$.

Remark:

I. We suppose that we are in the case of an identity-based cryptosystem in which KC issues a private key to a registering user and uses the user’s identity as his public key.
II. The status of the key (revoked or not) will depend on the security given by identity-based cryptosystem.
III. The first partial quotient is ignored because it does not change regardless the chosen b_i.

IV. Although we did not find any attack when $Y_u Y_v$ is the concatenation of Y_u and Y_v, we believe that the security of continued fractions obtained can be strengthened if we used $Y_u Y_v = SHA1(PK_U || PK_V)$.

V. We use the third root of log because it corresponds to the example exhibited in Result 2.

VI. The SHA1 used in this protocol is just an example and it can be replaced by any hash function which has a pre-image resistance property.

VII. We conjecture 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 $[S;\ P]$. S and P will be determined by the value of b_i's (see [6]).

3.2 Attack using Lowe's method.

Alice wants to talk to the intruder I, hence she chooses N_a and sends $(\{N_a\})$ encrypted with I's public key.

Step 2:

We recall that I(A) is the attacker who tries to impersonate A. I(A) decrypts the message received from A with its private key and transfers to B N_a encrypted with the public key of B.

Step 3:

B computes $(FC(\sqrt[3]{\log_e N_a}, Y_A Y_B))$, chooses a nonce N_b and sends $\{FC'(\sqrt[3]{\log_e N_a}, Y_A Y_B)),\ N_b\}$ to A after having encrypted it with A's public key (PK_A).

Step 4:

I(A) cannot decrypt the message encrypted with the public key of A, then he transfers the message to A.

Step 5:

To check if $FC(\sqrt[3]{\log_e N_a}, Y_A Y_B))$ is correct, A computes $FC'(\sqrt[3]{\log_e N_a}, Y_A Y_I))$ since the nonce N_a was sent to I. The $FC'(\sqrt[3]{\log_e N_a}, Y_A Y_I))$ will not match with $FC'(\sqrt[3]{\log_e N_a}, Y_A Y_B))$ since the first one was calculated with Y_I and the second one was computed with Y_B.

A can then conclude that an attack is underway.

Summary:

A → I : $E(\{N_a\}: PK_I)$

I(A) → B : $E(\{N_a\}: PK_B)$

B → I(A): $E(\{FC'(\sqrt[3]{\log_e(N_a)}, Y_A Y_B)), N_b\}: PK_A)$

I → A : $E\ (\{FC'(\sqrt[3]{\log_e(\mathrm{N_a})}, Y_A Y_B)), N_b\}: PK_A)$

A computes and verifies if

$FC'\left(\sqrt[3]{\log_e(N_a)}; Y_A Y_I\right) \neq FC'\left(\sqrt[3]{\log_e(N_a)}; Y_A Y_B\right)$.

Remark

The usefulness of continued fractions is noticeable at this level because without continued fractions the Lowe attack would be effective on this algorithm.

3.3 Example of the new Authentication Protocol

Before sending any message Alice and Bob will calculate the following elements in order to speed up the authentication protocol.

Let's suppose that Alice's public key is: PK_A =12345678910111213, then we apply the hash function SHA1 on that string and we obtain SHA1(PK_A)= A3 BF AE 33 E7 3F 0C A318 0D 8B FA 5C AB EA 4E F1 39 C3 6D (in hexadecimal).

The vector Y_A used in the computation of the generalized continued fraction will be Y_A= (b_{1A}=A3BF=41919, b_{2A}=AE33=44595, b_{3A}= E73F=59199, b_{4A}=0CA3=3235, b_{5A}=180D=6157, b_{6A}=8BFA=35834, b_{7A}=5CAB=23723,

b_{8A}=EA4E=59982, b_{9A}=F139=61753, b_{10A}=C36D= 50029).

Let’s suppose that Bob’s public key is: PK_B = 9876543210, then we obtain after applying the hash function SHA1 on the public key, SHA1(PK_B) = 9C D6 56 16 96 00 15 7E C1 72 31 DC F0 61 3C 94 93 2E FC DC.

The vector Y_B used in the calculation of the generalized continued fraction will be Y_B= (b_{1B}=9CD6=40150, b_{2B}=5616=22038, b_{3B}= 9600=38400, b_{4B}= 157E=5502, b_{5B}= C172=49522, b_{6B}=31DC=12764, b_{7B}=F061=61537, b_{8B}=3C94=15508, b_{9B}= 932E=37678, b_{10B}=FCDC=64732).

The intruder public key will be: PK_I = 76543210123, hence SHA1(PK_I)= E0 68 09 F6 BB 62 9F 10 F8 48 A0 9A A5 3F E1 5C 0E D3 A5 2D.

Table 1. Example of the new authentication protocol

Alice	*Intruder*	*Bob*
Chooses randomly N_a, N_a=456576890 and Sends $E(N_a: PK_B)$.	Sees $E(N_a: PK_B)$	Receives $E(N_a: PK_B)$.
		Decrypts $E(N_a: PK_B)$ with SK_B
Computes $FC((\sqrt[3]{\log_e(N_a)}, Y_AY_B))$= ($a_1$=2, a_2=4106694278, a_3=11096573017, a_4=285849649, a_5=455761744, a_6=3431553022, a_7=2199137734, a_8=8018839091, a_9=4822767116, a_{10}=4987704341).		Computes $FC((\sqrt[3]{\log_e(N_a)}, Y_AY_B))$= ($a_1$=2, a_2=4106694278, a_3=11096573017, a_4=285849649, a_5=455761744, a_6=3431553022, a_7=2199137734, a_8=8018839091, a_9=4822767116, a_{10}=4987704341).
Receives $E(a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b: PK_A)$.	Sees $E(a_2,a_3,a_4,a_5,a_6,a_7, a_8,a_9,a_{10}, N_b: PK_A)$	Chooses randomly N_b, N_b=4567387 and sends $E(a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b:PK_A)$.
Decrypts with SK_A $E(a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b: PK_A)$. Verifies that a_2,..,a_{10} are correctly calculated.		$FC(\sqrt[3]{\log_e(N_b)}, Y_BY_A)$ = (c_1=2, c_2=2981639062, c_3=9232230473, c_4=578035971, c_5=9394835548, c_6=6724620718, c_7=15782459451, c_8=2048625312, c_9=3728834053, c_{10}=7375783858).
Computes $FC(\sqrt[3]{\log_e(N_b)}, Y_BY_A)$ = (c_1=2, c_2=2981639062, c_3=9232230473, c_4=578035971, c_5=9394835548, c_6=6724620718, c_7=15782459451, c_8=2048625312, c_9=3728834053, c_{10}=7375783858) and sends $E(c_2,c_3,c_4,c_5,c_6,c_7,c_8,c_9,c_{10}; PK_B)$.	Sees $E(c_2,c_3,c_4,c_5,c_6,c_7, c_8,c_9,c_{10}; PK_B)$.	Receives $E(c_2,c_3,c_4,c_5,c_6,c_7,c_8,c_9,c_{10}; PK_B)$.
		Decrypts $E(c_2,c_3,c_4,c_5,c_6,c_7,c_8,c_9,c_{10}: PK_B)$ with SK_B. Verifies that c_2,..,c_{10} are correctly calculated.

The vector Y_I used in calculation of the generalized continued fraction will be Y_I= (b_{1I}=E068=57448, b_{2I}=09F6=2550,

b_{3I}=BB62=47970, b_{4I}=9F10=40720, b_{5I}=F848=63560, b_{6I}=A09A=41114, b_{7I}= A53F=42303, b_{8I}=E15C=57692, b_{9I}= 0ED3=3795, b_{10I}=A52D=42285).

The choice of the vector Y_A, Y_B or Y_I used in the computation of the generalized continued fraction will depend on the origin of the challenge. For example, if Bob has to respond to a challenge sent by Alice, then the vector used will be Y_AY_B, if Alice sends a challenge to the intruder, then the vector used will be Y_AY_I, and if the intruder sends a challenge to Bob the vector used by Bob will be Y_IY_B. In Table I, we present the example of the new protocol.

3.4 Example of the Lowe's attack in the new protocol

This attack is detailed, in Table II.

3.5 Example of attack (using the lack of identity)

This attack is detailed below, in Table III.

Table 2. Example of the Lowe's attack in the new protocol

Alice	*Intruder*	*Bob*
Chooses randomly N_a, N_a=456576890 and Sends $E(N_a : PK_I)$	Receives $E(N_a: PK_I)$	
	Decrypts $E(N_a: PK_I)$ with SK_I and sends $E(N_a: PK_B)$ to Bob	Decrypts $E(N_a : PK_B)$ with SK_B
Computes $FC(\sqrt[3]{\log_e (N_a)}, Y_AY_I)$ = (a'_1=2, a'_2=4106666894, a'_3=5059764476, a'_4=309131866, a'_5=481976420, a'_6=3404891394, a'_7=3070712358, a'_8=4494633197, a'_9=30367444710, a'_{10}=3998170756).		Computes $FC(\sqrt[3]{\log_e (N_a)}, Y_AY_B)$ (a_1=2, a_2=4106694278, a_3=11096573017, a_4=285849649, a_5=455761744, a_6=3431553022, a_7=2199137734, a_8=8018839091, a_9=4822767116, a_{10}=4987704341).
Receives E(a_2 ,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b: PK_A).	Receives E(a_2,a_3,a_4,a_5,a_6, a_7,a_8,a_9,a_{10}, N_b: PK_A) and transfers it to Alice	Chooses randomly N_b, N_b=4567387 and sends E(a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b :PK_A)
Decrypts with SK_A E(a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}, N_b: PK_A). Verifies if a_2=a'_2,..,a_{10}= a'_{10} and then concludes that someone is trying to impersonate her.		

Remark:

In the last exchange of this attack, the intruder is not able to find the correct $a'_2,..,a'_{10}$ because he cannot find N_a (by Result 1) and he is not able to guess $a'_2,..,a'_{10}$ in the sole knowledge of $a_2,..,a_{10}$.

3.6 Others attacks

Replay, Interleaving, and Reflection attack

The nonces play an essential role in this algorithm, they are chosen randomly ($\in_R$), and they change with each message since the principal chooses a new nonce at each exchange. Thus, it is impossible to have a replay, an interleaving or a reflection

Table 3. Example of attack (using the lack of identity)

Alice	*Intruder*	*Bob*
	Sends to Alice (Hi I am Bob).	
Chooses randomly N_a, N_a= 456576890 and Sends $E(N_a: PK_B)$	Receives $E(N_a : PK_B)$.	
	Transfers $E(N_a : PK_B)$ to Bob saying (I am Intruder).	Decrypts $E(N_a : PK_B)$ with SK$_B$.
Computes $FC(\sqrt[3]{\log_e(N_a)}, Y_A Y_B)$= ($a'_1$=2, a'_2=4106694278, a'_3=11096573017, a'_4=285849649, a'_5=455761744, a'_6=3431553022, a'_7=2199137734, a'_8=8018839091, a'_9=4822767116, a'_{10}=4987704341).		Computes $FC(\sqrt[3]{\log_e(N_a)}, Y_I Y_B)$= ($a_1$=2, a_2=234855306, a_3=4078278518, a_4=3802145145, a_5=4630995005, a_6=24219050134, a_7=20475451123, a_8=3813152404, a_9=286637470, a_{10}=6468778119).
Receives E(a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9,a_{10}, N_b: PK_A).	Decrypts E(a_2,a_3,a_4,a_5,a_6,a_7, a_8,a_9,a_{10}, N_b: PK_I) and transfers to Alice E(a_2,a_3,a_4,a_5,a_6,a_7, a_8,a_9,a_{10}, N_b: PK_A)	Chooses randomly N_b, N_b=4567387 and sends E(a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, N_b: PK_I)
Decrypts with SK$_A$ E(a_2,a_3,a_4,a_5, a_6, a_7, a_8, a_9, a_{10}, N_b: PK_A) Verifies if a_2=a'_2,..,a_{10}= a'_{10} then concludes that someone is trying to impersonate her.		

attacks. We can add that the calculations of the partial quotients are based on the sender and the receiver of the challenge, which is a kind of a signature.

The forced delay attack

This attack does not have a real influence on our algorithm because there is no timestamp. Hence, the behaviour of the principals will be the same even if someone intercepts the message and relays it later.

Remark

We recommend the use of the generalized continued fraction instead of the classical continued fraction; because the classical continued fraction produces a several partial quotients with only one digit [9], while the partial quotients obtained from some generalized continued fraction seems to be indistinguishable by all polynomial-time statistical tests from the uniform distribution of integers [6].

4. Comparison with Needham-Schroeder-Lowe

4.1 Comparison

The Needham-Schroeder-Lowe protocol used nonces where we use continued fraction expansions and nonces, so it is clear that our algorithm is more expensive in terms of computation time, however we strongly believe that our scheme is more secure than the algorithm of Needham-Schroeder-Lowe. Our protocol increases the level of security with the introduction of continued fraction and if we consider that the calculations are done in two phases, the preparation phase can reduce the time needed for the authentication itself.

Similarly in the Needham-Schroeder-Lowe protocol, if the intruder sees one of the two nonces (N_a or N_b), it is risky to use this part of secret in the symmetric encryption key while in our protocol the properties of continued fractions allow to use this nonce without any detrimental effect on the security of the symmetric encryption. We recall that the knowledge of the continued fraction expansions of α and β cannot determine simply those of $\alpha+\beta$, or $\alpha\beta$ which imply that N_aN_b can be used as seed for the symmetric encryption as defined in [6].

Hence, we can add that this protocol introduced some aspects of the zero knowledge system in this algorithm, because seeing the partial quotients cannot give any idea of the composition of the nonce (see Table 3 (Example of attack using the lack of identity) & Result 1).

We have greatly simplified the Needham-Schroder-Lowe protocol, because in addition to the number of steps which we have reduced, we also removed the identities of the principals in messages. It is a great progress to remove the identity of the principal in the protocol because if the secret key of the principal A fell into the wrong hands, the attacker could use this key to impersonate A, while in the new protocol, the intruder will not be able to identify the other principal.

Let's suppose that the Intruder I has the secret key of A and he intercepts these following messages
$E(N_a{:}\,PK_B)$;
$E\ (\{FC'(\sqrt[3]{log_e N_a}\,; Y_AY_B), N_b\ \}{:}\ PK_A)$;
$E\ (\{FC'(\sqrt[3]{log_e N_b}\,; Y_BY_A)\ \}{:}\ PK_B)$.

He will not be able to decrypt $E(N_a : PK_B)$ and $E(\{FC'(\sqrt[3]{log_e N_b}\,; Y_BY_A)\ \}{:}\ PK_B)$. He will be able to decrypt $E\ (\{FC'(\sqrt[3]{log_e N_a}\,; Y_AY_B), N_b\ \}{:}\ PK_A)$ with SK_A, however he cannot know who is the recipient of the message in order to continue the authentication.

4.2 Efficiency analysis

Let $b_T = \max(b_i)\ \forall\, b_i \in (Y_AY_B\ \cup\ Y_BY_A \cup\ Y_AY_I\ \cup Y_IY_A \cup Y_BY_I \cup Y_IY_B)$ and $t_1 = \max(\ (\sqrt[3]{\log_e N_a}\ ), (\sqrt[3]{\log_e N_b}\ ))$ we evaluate the cost of calculating a partial quotient to be $O\ (\delta^{1+\epsilon})$ where $\delta = \max(log_2\ (t_1)\ , log_2(b_T))$ and $\epsilon \in]0,1[$. Hence, we can conclude that the time needed for computing partial quotients is low.

5. Identity

The problem of removing identities should be more studied in authentication protocols because for security reasons, it is not conceivable to let the identity of the principal anywhere (web merchant, webmail, website, forum ...).

Although, it will be difficult to remove completely the identity, we are sure that we can reduce it to a minimum.

Several solutions such as the anonymous credential systems or group signatures have been proposed in the past, unfortunately, these systems have not yet been deployed on a large scale [19].

We add to these existing solutions, two new proposals.

The first one is the complete removal of the explicit identity if it can be guessed.

We note here that identity will always be present in the authentication protocol but implicitly.

The identity may be inferred, for example, when a company has only one client at King Street, it would be easy to omit the identity of that client.

Or if a company has five customers on King Street, and among them one who connects every day at 7 am, we can also omit the identity of this customer on the authentication protocol.

Finally, we can try to find from artificial intelligence tools, some solutions which will allow us to guess who is the entity who wants to connect instead of mentioning explicitly its identity.

The second solution which we propose is the separation between the explicit identity and authentication protocol itself.

We would use two encryption keys, the first key would be used to send this message "HI I am B", and the second key would be used for the following steps (the authentication protocol itself). If one key is cryptanalysis, the identity of the principal entity would still be preserved.

6. Conclusions

In this paper, we presented a mutual authentication protocol which introduces the use of continued fractions in authentication schemes. We also improve the Needham-Schroeder-Lowe protocol by eliminating the identity of the principal in the authentication messages.

The rounding errors presents in the computation of the partial quotients could be an advantage, since the absence of agreement on the rounding errors between the principal and the intruder will increase the probability of failure of any attack.

It could be interesting to see in the future, which properties of continued fractions may help to reduce the cost of partial quotients calculations.

Due the computer limitation, the use of irrational numbers can be theoretical, but as proved in [6], we can use an approximation of irrational numbers.

References

[1] M. Beeler, R.W. Gosper, and Schroeppel, R. Hakmen, “MIT Artificial intelligence memo 239”, Feb. 29, 1972.

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

[3] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication” Technical Report 39, Digital Systems Research Center, 1989.

[4] J. Clark and J. Jacob, “A survey of authentication protocol literature”, 1997.

[5] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communication of the ACM*, 24(8):533-536, August 1981.

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

[7] A.M. Kane, “On the use of continued fractions for electronic cash” in International Journal of Computer Science and Security, Vol: 4, No:1, pp.136-148, (2010).

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

[9] P. Levy, “Sur les lois de probabilité dont dépendent les quotients complets et incomplets d'une fraction Continue”, Bull. Soc. Math. 57 (1929) 178-194.

[10] G. Lowe, “Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR”, in: Proc. TACAS’96, Springer LNCS 1055, 1996, pp. 147–166.

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

[12] R. Needham, M. Schroeder, “Using Encryption for Authentication in Large Networks of Computers”, Comm. ACM 21 (12) (1978) 993–999.

[13] C. D. Olds, “Continued Fractions”, Random House, 1963.

[14] O. Perron, “Die Lehre Von Den Kettenbrüchen”, 3rd ed. (1954).

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

[16] J. G. Steiner, B. C. Neuman, and J. I. Schiller, Kerberos: An authentication service for open network systems. In *Proceedings of the Winter 1988 Usenix Conference*, pages 191-201, (1988).

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

[18] T.Y.C. Woo, S. S. Lam, "Authentication for Distributed Systems," *Computer*, vol. 25, no. 1, pp. 39-52, (1992).

[19] P. Bichsel, J. Camenisch, and D. Sommer, “A calculus for privacy-friendly authentication”, Proceedings of the 17th ACM symposium on Access Control Models and Technologies, pages 157--166, 2012.