Use of Personal Rhythm to Authenticate Encrypted Messages

Carlos Gonzalez

***Abstract*—**When communicating using private and secure keys, there is always the doubt as to the identity of the message creator. We introduce an algorithm that uses the personal typing rhythm (keystroke dynamics) of the message originator to increase the trust of the authenticity of the message originator by the message recipient. The methodology proposes the use of a Rhythm Certificate Authority (RCA) to validate rhythm information. An illustrative example of the communication between Bob and Alice and the RCA is included. An algorithm of how to communicate with the RCA is presented. This RCA can be an independent authority or an enhanced Certificate Authority like the one used in public key infrastructure (PKI).

***Keywords*—**Personal rhythm, public-key encryption, authentication, digital signature, keystroke dynamics.

I. Introduction

A digital signature is the public-key equivalent of message authentication codes. As mentioned by Ferguson [7], the main problem is that (Alice) most of the time does not sign the message herself, but her computer automatically signs the messages for her. Thus, the signature only means that the message comes from Alice's account but it may not be Alice who is the one producing the message. Since the signature is automatic, Alice may not notice or be aware of this fact.

The algorithm we are presenting here forces the sender (Alice) to perform an action, which is to type a text sent by the future message recipient. Alice will have to type and send the results of the rhythm of typing such a message. We assume that Alice's computer has the application software to collect such rhythm information. With this rhythm and the use of a third party agency, the recipient will be able to trust more the authenticity of the sender. There are some previous patent efforts done by using the rhythm of typing for the purpose of authentication in electronic devices [1], [4], [8], [9], [12], and many academic papers [2], [6], [10], [11], but no patent or paper uses rhythm to enhance encryption protocols like we propose in this paper.

II. Methodology

A. The Rhythm of Typing

The rhythm of typing has been used before to authenticate users in electronic devices. The rhythm procedure includes the creation of one or more patterns.

The most well-known patterns with rhythm information are:

- Inactive time between two keystrokes (time between the end of the first and the time of pressing of the second)
- Time pressing a key (how long the key is pressed down)
- Location of the finger pressing the key
- Amount of pressure put on a pressed key
- Typing speed (in the long run)
- Time between X most used two letter words
- Time between Y most used three letter words
- Number of keys depressed by time unit
- Most Z mistakes made, and time to correct
- Rhythm and Time of the day
- Rhythm and geography

These patterns will vary depending on if the operating system and the hardware of the device support such measurement.

To make the system respond quicker, we recommend for the system to use a subset of the user's alphabet. We propose for example to use 10 to 15 of the letters more frequently used in the English language [5], and these will be: E, T, A, O, I, N, S, R, H, D, L, U, C, M, and F.

The 15 most used letters constitute approximately 88% of the total use. We may include additional filters like the most common two letter words: of, to, in, it, is, be, as, at, so, we, he, by, or, on, do, if, me, my, up, an, go, no, us, am [3].

B. Measurements

The system will save for each pattern the following information:

1. Count of number of times used
2. Minimum value recorded
3. Maximum value recorded
4. Sum of all values recorded

Table I shows the results of an experiment done for one week on a Smartphone. The measurement captured is the amount of time spent pressing a key.

C. Main Algorithm

The proposed algorithm uses the Public-key Encryption methodology. The algorithm is described in Fig. 1 and works as follows:

1. When Alice wants to send a secure message to Bob, she first sends a request for dialog to Bob. To do this, Alice encrypts the request for dialog r using Bob's public key. Alice sends message c over the channel which is the result of the described encryption, c=E(PB,r).
2. Bob receives message c from Alice and decodes using Bob's secure key, r=D(SB,c).
3. Since is a request for dialog from Alice, Bob proceeds to generate a text t to send to Alice. Alice will have to type such text and send back her typing rhythm, c=E(PA,t).

Carlos Gonzalez is with the Universidad Autonoma de Coahuila, Mexico (phone: +52-844-176-8798; e-mail: gonzalezc757@gmail.com).

4. Alice receives message c from Bob, and proceeds to decode using her secure key, t=D(SA,c).

The message contains text t that Bob wants Alice to type and send her typing rhythm. The idea here is twofold: first is to make Alice do the action of typing the text instead of the computer, and second is to collect the typing rhythm of Alice as a proof of signature. Alice proceeds to type text t and collects the rhythm of this action. This is done by software inside Alice's computer. The results of this action are data R_A which will be sent to Bob but not for him to use, but to send to a third party RCA. Alice generates a message a containing her ID at the RCA (ID_A), the text that Bob sent to type is called now t_A, and the recorded rhythm RA that is: a= {ID_A, t_A, R_A}. This information is encrypted using the public key of the RCA, ar=E(P_{CA}, a). Finally, Alice encrypts the message m that she wants to send to Bob, c=E(PB,m).

5. Bob receives data c and ar from Alice. Using Bob's secure key decrypts the message m sent by Alice, m=D(S_B,c). All Bob needs now is to make sure Alice is the one signing the message. To do this, Bob sends to the RCA the data ar that Alice sent without making any changes, and includes a copy of the original text sent to Alice t now called t_B, which will be encrypted using a private key for RCA, d=E(P_{CA},t_B).
6. The RCA receives from Bob data d and ar. The ar contains Alice's ID, rhythm and text typed. The text initially used by Bob is sent in encrypted data d. RCA proceeds to decrypt using RCA's secure key {IDA, tA, RA}=D(S_{CA}, ar) and tB=D(S_{CA},d).

With the ID of Alice ID_A, RCA proceeds to check if this ID exists in its data base, if not, the result will generate a negative result. RCA also checks that the texts from Alice and Bob are the same t_A=t_B; otherwise, the search is invalid. RCA computes the probability that the rhythm send by Bob is between the parameters (i.e. maximum and minimum values) existent for such an ID, and reports the values back to Bob (as A%) encrypting the values using the public key of Bob, c=E(PB,A%).

7. Bob receives and decrypts the results from the RCA using Bob's secure key, A%=D(SB,c).

With this information, Bob decides if the message is signed by Alice or not, and how much confidence should he have of Alice being the message sender.

8. Bob encrypts a message n for Alice using her Public key informing her of her trust, c=E(PA,n).
9. Alice decrypts message n sent by Bob about his trust in her signature, n=D(SA,c).

III. The Rhythm Certification Agency

As we can see from the previous algorithm, the Rhythm Certification Authority (RCA) is an entity that needs to exist. This RCA will be similar as the one used by Alice and Bob to make accessible publicly their public key. The certificate authority (CA) is used for PKI [7]. Each user takes his public key to the CA and identifies itself to the CA. The CA then signs the user's public key using a digital signature. We envision the RCA entity as an independent entity, or as an enhanced CA providing the rhythm services. Thus, the RCA is an entity (server on the web) where users like Alice and Bob can communicate and first ask to create an ID. This is done by encrypting this request to the RCA c=E(PCA,r), the RCA decrypts this information r=D(SCA,r) and proceeds to generate an ID for Alice and responds back to Alice with an encrypted message c=E(PA,IDA). Alice receives this information and decrypts it IDA=D(SA,c). The IDA is saved by Alice for future access to the RCA. Once Alice has the IDA, it can proceed to send all the pertinent information about her typing rhythm. It is assumed that Alice has been collecting her rhythm for some period of time, and her rhythm data has reached a steady state (i.e. no significant changes in values). This information could be as a simple, such as the depression times for any key, or the time to depress two consecutive keys, or any of the other most well-known patterns described before. The information saved may be the minimum and maximum times for Alice. This information, along any other pertinent rhythm information, is sent to RCA by and encryption message ar where a= {IDA, RA} and ar=E(PCA, a). RCA receives this information and decrypts it {IDA, RA}=D(SCA, ar). RCA first checks if IDA exists then proceeds to save the rhythm data sent by IDA. Fig. 2 shows this process.

The RCA is a single point of failure because it needs to decrypt every request and Distributed Denial of Service (DDoS) attacks can prevent every communication using the RCA. In order to avoid this scenario, we propose a slight modification to our main encryption algorithm.

This algorithm, in its communication between Alice and Bob, will include a step where Alice sends Bob the additional encrypted result of her rhythm. This data will be saved (i.e. the text sent and the rhythm results) by Bob when the RCA certifies that this is Alice's signature. When later communicating if the RCA is inaccessible, and both Bob and Alice agree on this fact, Bob will ask Alice to again send a rhythm for the saved data; thus, allowing Bob to validate Alice's signature. Two parties can continue communicating using this procedure until the RCA is available again. If the RCA becomes inaccessible without having Alice send a single corroborated message to Bob, then both can agree to continue communicating under this condition or stop until the RCA is available again.

IV. An Illustrative Example

For example, Bob will send Alice the following text: "Roses are red, violets are blue, I will always love you".

Alice proceeds to type the text and generates her rhythm. Alice's rhythm data would look something like: first Alice's ID, then a measure for each individual key depresses (milliseconds of time depressing the key).

ID: 3458873
r= 105, 95, 100, 97
o=115, 107, 108, 109
s=102, 123, 98, 102
e=103, 102, 100, 101, 100, 101
b=161
a=110, 108, 109

d=150
v=110, 100
i=107, 105, 123
l=110, 111, 170, 105, 115, 107
t=249
u=98, 105
w=124, 91
y=225, 228

These values represent in milliseconds the depression time of the keys for each of the letters in the text.

This information is passed by Bob to the RCA, which has Alice's rhythm information. RCA examines and analyzes Alice's rhythm and returns to Bob the data, as shown in Table II.

There are many statistics that one could use to compare two sets of data (i.e. the control data and the captured data). We decided to use two filters: the boundaries for each character (i.e. minimum and maximum values) and Hedge's-g values. We use Hedge's-g in this example because it is an easy formula to implement, and also easy to understand.

The Hedge's g [13] formula is:

$$g= (M_E - M_C)/SD \text{ pooled}$$

where: ME = mean for the experimental group. MC = mean for the control group. SDpooled = pooled standard deviation.

TABLE I
RHYTHM DATA

ID	Char	Count	MIN	MAX	SUM
97	a	188	3	218	20529
98	b	19	3	163	1818
99	c	37	3	149	3748
100	d	57	3	156	5760
101	e	157	3	167	15877
102	f	13	5	131	1292
103	g	25	3	152	2644
104	h	16	5	199	1835
105	i	82	3	169	8914
106	j	17	3	125	967
107	k	12	53	156	1203
108	l	90	3	234	9773
109	m	32	5	1440	4799
110	n	91	3	191	8953
111	o	116	2	209	12772

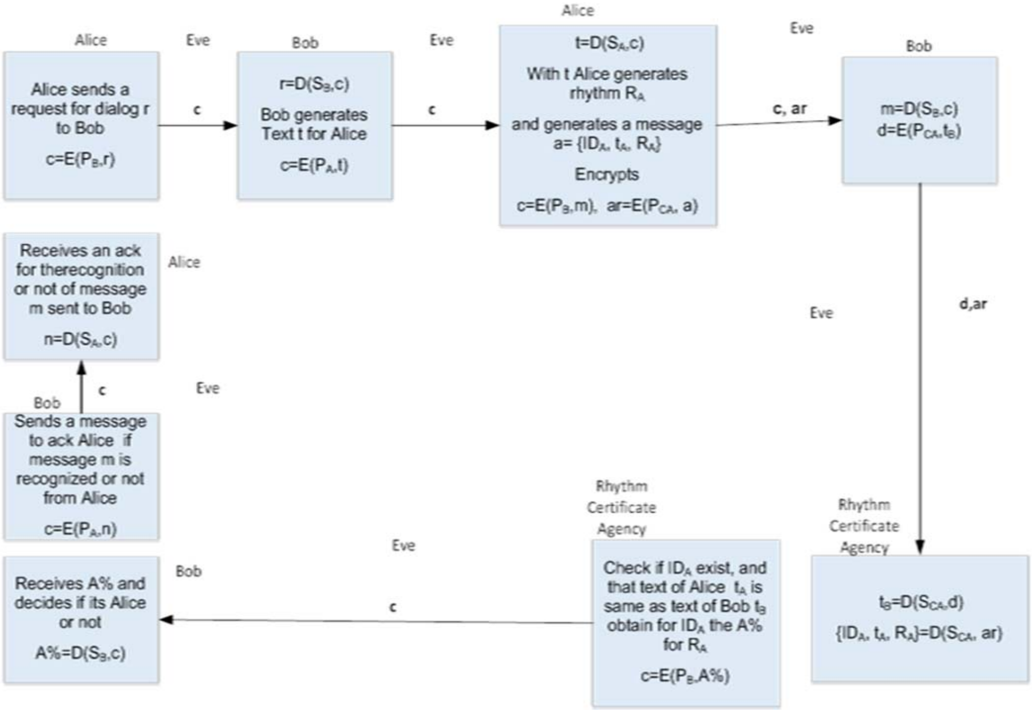
Alice
Eve
Bob
Eve
Alice
Eve
Bob
Alice sends a request for dialog r to Bob
c=E(P_B,r)
c
r=D(S_B,c)
Bob generates Text t for Alice
c=E(P_A,t)
c
t=D(S_A,c)
With t Alice generates rhythm R_A
and generates a message a= {ID_A, t_A, R_A}
Encrypts
c=E(P_B,m), ar=E(P_CA, a)
c, ar
m=D(S_B,c)
d=E(P_CA,t_B)
Receives an ack for therecognition or not of message m sent to Bob
n=D(S_A,c)
Alice
c
Eve
Bob
Sends a message to ack Alice if message m is recognized or not from Alice
c=E(P_A,n)
Eve
d,ar
Rhythm Certificate Agency
Rhythm Certificate Agency
Eve
Bob
Receives A% and decides if its Alice or not
A%=D(S_B,c)
c
Check if ID_A exist, and that text of Alice t_A is same as text of Bob t_B obtain for ID_A the A% for R_A
c=E(P_B,A%)
t_B=D(S_CA,d)
{ID_A, t_A, R_A}=D(S_CA, ar)

Fig. 1 Messages Communication Algorithm

A g of 1 indicates the two groups differ by 1 standard deviation, while a g of 2 indicates they differ by 2 standard deviations, and so on.

Hedge suggested using the following rule of thumb for interpreting results:

Small effect (cannot be discerned by the naked eye) = 0.2

Medium Effect = 0.5

Large Effect (can be seen by the naked eye) = 0.8

If the g value is less than 0.5, there is a good chance that the groups differ very little. On the other hand, if any value sent by Alice is less than her minimum or more than her maximum recorded value (in the RCA) for the character, this entry is considered a failure (it is not Alice's rhythm).

For the values with a single entry like b, d and t, we use only the boundaries filter. As we can see, all the characters are between the boundaries interval, also all have a value less than 0.5; therefore, it is safe for Bob to assume it is Alice who is the author of the sent message.

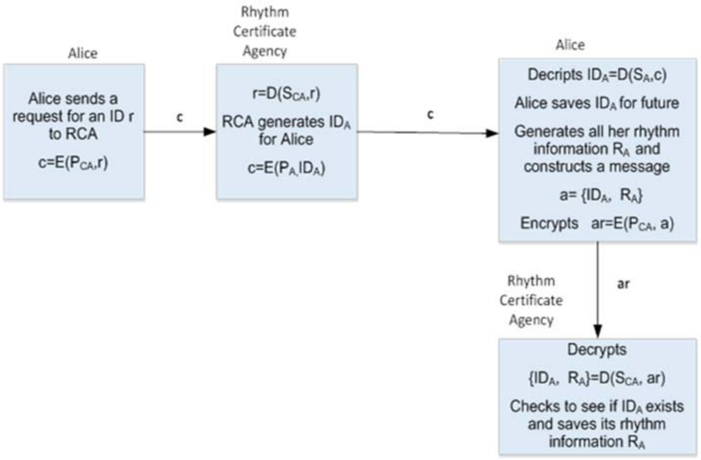
Alice
Alice sends a request for an ID r to RCA
c=E(P$_{CA}$,r)
c
Rhythm Certificate Agency
r=D(S$_{CA}$,r)
RCA generates ID$_A$ for Alice
c=E(P$_A$,ID$_A$)
c
Alice
Decripts ID$_A$=D(S$_A$,c)
Alice saves ID$_A$ for future
Generates all her rhythm information R$_A$ and constructs a message
a= {ID$_A$, R$_A$}
Encrypts ar=E(P$_{CA}$, a)
Rhythm Certificate Agency
ar
Decrypts
{ID$_A$, R$_A$}=D(S$_{CA}$, ar)
Checks to see if ID$_A$ exists and saves its rhythm information R$_A$

Fig. 2 Communicating with the RCA

TABLE II
RCA RETURN DATA

Character	Inside boundaries	Hedge's-g
r	true	-0.22975038
o	true	0.11355859
s	true	-0.4771752
e	true	-0.03680528
b	true	
a	true	0.24104021
d	true	
v	true	0.09875066
i	true	-0.36737627
l	true	-0.48740966
t	true	
u	true	-0.34200454
w	true	0.43358219
y	true	-0.20348871

V. CONCLUSION/COMMENTS

If the communication between Alice and Bob is not extremely important (i.e. not secret or top secret) then one authentication for Alice will be enough for Bob to accept her identity. On the other hand, if the information is highly classified, the authentication will have to be done on every message.

The existence of a Rhythm Certification Authority (RCA) is not going to be difficult to implement, since the RCA will not know the identity of each user, only their rhythm information. This scheme could be implemented for more than one RCA, and cross-check of the results from two or more.

Using the texts for typing rhythm by Alice from Bob by the RCA is done to avoid interference from Eve. If we do not have the checking of the texts, Eve could then remove information from the message of Bob to RCA and replace it with one of her own. With the text-checking, Eve would have to break the message from Bob to Alice, which is more difficult than just replacing data.

The author believes that the proposed methodology will increase the trust in the signature of an encrypted message, and if the messages are classified, this identification confirmation is a must.

REFERENCES

[1] Bender, et al. "Key sequence rhythm recognition system and method", U.S. Pat. No. 7,206,938, April 2007.

[2] Bergadano et al., "User authentication through keystroke dynamics", ACM Transactions on Information and System Security, Volume 5 Issue 4, November 2002 Pages 367-397.

[3] Bryce, Scott, "Cryptograms", http://scottbryce.com/cryptograms/stats.htm, accessed March 20, 2015.

[4] Cho et al., "Apparatus for authenticating an individual based on a typing pattern by using a neural network system", U.S. Pat. No. 6,151,593, November 2000.

[5] Cornell University, "English Letter Frequency" https://www.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.htm, accessed Nov 18, 2016.

[6] Deng Y. and Zhong Y., "Keystroke Dynamics User Authentication Based on Gaussian Mixture Model and Deep Belief Nets", ISRN Signal Processing Volume 2013 (2013), Article ID 565183.

[7] Ferguson N., Schneier B., Kohno T. "Cryptography Engineering. Design Principles and Practical Applications," Wiley Publishing, Inc., 2010.

[8] Garcia; John D. "Personal identification apparatus", U.S. Pat. No. 4,621,334, November 1986.

[9] Gonzalez C., Avila M., and Moreno R, "Continuous Authentication Using User's Typing Rhythm and Application Selection for Security of Mobile Electronic Devices", U.S. Patent Application #62295130, Feb 2016.

[10] Lívia et al., "User Authentication Through Typing Biometrics Features", IEEE Transactions on Signal Processing, Vol. 53, No. 2, February 2005.

[11] Rangnath Dholi P., Chaudhar K. P., "Typing Pattern Recognition Using Keystroke Dynamics", Mobile Communication and Power Engineering, Volume 296 of the. series Communications in Computer and Information Science pp 275-280, 2013.

[12] Serpa, Michael Lawrence, "System and method for user authentication with enhanced passwords", U. S. Pat No. 6,954,862, Oct 2005.

[13] Larry V. Hedges (1981). "Distribution theory for Glass' estimator of effect size and related estimators". Journal of Educational Statistics. 6 (2): 107–128.