SHADOW NUMBERS PUBLIC KEY ENCRYPTION

John Almeida
john.almeida@unoweb.com

UnoWeb Inc.
www.unoweb.com
P.O. Box 8743
Emeryville, CA 94662

This paper is part of a pending US patent: US 2011/0026711 A1

Abstract

The present public key encryption in this paper involves the use of two values and they are the shadow's values of a base value, and the base value is derived from the two shadows' values. Whenever two positive whole values (*first shadow* value and *second shadow* value) are multiplied producing a product value and the value of **1** is subtracted from the product value, a first base value is derived and it is the first base value of the two shadows values. The derived first base value may be divided by any divisor that it can be divided with which produces a positive quotient result and zero for the remainder. All values that are used in the division and the quotient result are bases values for the chosen shadow value-pair. Then one of the base values is chosen along with the two chosen shadows values and they comprise a triplet values that represent the public key to encrypt a message and the private key to decrypt the encrypted message.

Keywords: Shadow number, shadow number system, shadow number algorithm, public key encryption, algorithm, public key, private key, encryption.

1 Introduction

Since the dawn of times humans felt the need to protect sensitive information that only the sender and the intended recipient could've known its contents. Various form of encryption have been devised and used throughout the ages and some common ones involved the scrambling of letters of the message by the sender and only the recipient possessing the knowledge of the method the sender used was able to reassemble the message to its original state. These methods had their weaknesses because the message's deliverer had to be trusted as well. If the message's deliverer knew the method that was used to scramble the message then the message could've been read while in transit from the sender to the recipient. These methods were based on a common encryption and decryption key, private key. Based on the weakness of private key methods, there always had a need to devise a method that the key used to encrypt the message would've been readily available to the public but the key used to decrypt the message would've only be available to the message's recipient.

In 1976, Diffie and Hellman introduced the concept of two-key cryptosystems [1] . They proposed a method of public-key encryption, wherein each user has both a public and private key, and two users could communicate knowing only each other's public keys. In the public-key system devised by Diffie and Hellman, secrecy and authenticity were provided by two separate transformations. Suppose user A wishes to send a message M to another user B. If A knows *B's* public transformation Eb, A can transmit M to B in secrecy by sending the ciphertext $C = Eb(M)$. On receipt, B deciphers C using *B's* private transformation DB, equation (1):

$$DB(C) = DB(EB(M)) = M \tag{1}$$

For authenticity, M must be transformed by *A's* own private transformation DA. A sends $C = DA(M)$ to B. On receipt, B uses *A's* public transformation EA to compute, equation (2):

$$EA(C) = EA(DA(M)) = M \tag{2}$$

Authenticity is provided because only A can apply the transformation DA. [2]

In 1978, Pohlig and Hellman [3] published an encryption scheme based on computing exponentials over a finite field. At about the same time, Rivest, Shamir, and Adleman [4] published a similar scheme, but with a slight twist, a twist that gave the MIT group a method for realizing public-key encryption as put forth by Diffie and Hellman [1].

The Pohlig-Hellman and RSA schemes both encipher a message block by computing the exponential, equation (3):

$$C = Me \bmod n \tag{3}$$

Where e and n are the key to the enciphering transformation. M is restored by the same operation, but using a different exponent d for the key, equation (4):

$$M = Cd \bmod n \tag{4}$$

The enciphering and deciphering transformations are based on Euler's generalization of Fermat's Theorem [5].

In 1993, Michio Shimada devised an algorithm [6] that uses linear transformation. To encrypt data word of plaintext, it takes the values of *0, 1, 2, ... , (N- 1)*, the modulus is set to *N*. The enciphering is implemented as shown in equation (5):

$$C = ak(...(a2(a1.M + b1 \bmod N) + b2 \bmod N)...) + bk \bmod N \tag{5}$$

where *a1 ... ak* and *b1... bk* are cryptographic keys. Above equation can be rewritten as in equation (6):

$$C = A.M + B \bmod N \tag{6}$$

In 1999, the CayleyPurser algorithm [7] was a public-key cryptography algorithm published in early 1999 by 16-year-old Irishwoman Sarah Flannery [8], based on an unpublished work by Michael Purser, founder of Baltimore Technologies, a Dublin data security company.. Flannery named it for mathematician Arthur Cayley. It was found to have flaws as a public-key algorithm, but was the subject of considerable media attention.

Since this algorithm uses *2 x 2* matrices and ideas due to Purser it is called the Cayley-Purser Algorithm. The matrices used are chosen from the multiplicative group *G = GL(2, Zn)*. The modulus $n = pq$, where p and

q are both primes of 100 digits or more, is made public along with certain other parameters which will be described presently, equation (7).

$$|GL(2, Zn)| = nphi(n)2(p + 1)(q + 1) \tag{7}$$

we note that the order of G cannot be determined from a knowledge of n alone.

Plaintext message blocks are assigned numerical equivalents as in the RSA and placed four at a time in the four positions (ordered on the first index) of a *2 x 2* matrix. This message matrix is then transformed into a cipher matrix by the algorithm and the corresponding ciphertext is then extracted by reversing the assignment procedures used in the encipherment [9].

2 Prior solutions

All the prior solutions require the use of large prime numbers and requiring a great deal of processing power to locate them on the fly. Also, most of these algorithms require prime numbers of the same length and therefore making it easier to find one of the prime number and compromising the security of the encryption. Furthermore, some of the currently in use algorithms require exponentiation which is computational extensive and making them less appealing for use in smaller devices with limited computational speed and low battery life.

3 Shadow numbers public key

The solutions offered by prior encryption algorithms requiring the use of prime numbers and repetitive exponentiation limit their use in computing devices with limited processing speed and low battery life, e.g. personal mobile devices.

The proposed solution in this paper uses any kind of positive whole values and not necessarily being prime numbers, thus making the encryption easy of use by simply randomly generating the two shadow's values on the fly. Then using the generated two shadows' values to derive a base value. And finally, using one of the shadow value with the base value as the public encryption key, and the other shadow value with the base value as the private encryption key. The proposed solution is easy to implement and does not require high computational power as compared to some of the other solutions, thus enabling it to be used in any kind of personal device without slowing it down, without draining the device's battery and without requiring the randomly chosen shadow's value to be of the same length, therefore increasing the speed and the security of the encryption.

In terms to use the shadow system, a two shadows' values are necessary and they may be any positive whole value and not necessarily prime numbers. After the two chosen values are multiplied, a product is derived and once the value of **1** is subtracted from the product, a first base value is obtained.

The first shadow value may be any whole value that is greater than **1** and it may not necessarily be a prime number. The second shadow value may be any positive whole value that is greater than **2** and it may not necessarily be a prime number either. Once the first shadow value is multiplied by the second shadow's value, a product value is obtained. After the product value is obtained and the value of **1** is subtracted from the product value, a first base value is obtained.

The base value may be the obtained first base value or any other value that it can be divided with and producing a positive quotient value and zero for the remainder. In case the first base value is divisible by any other value, then the divisor value that is used in the division of the first base value and the quotient value of the division, are also bases' values.

For our example let's choose two shadows values: **5** and **3**. Once they are multiplied the product value of **15** is obtained. After the value of **1** is subtracted from the product **15** the result value of **14** is obtained. And **14** is the first base value for the shadows' value-pair of **5** and **3**. The first base value **14** is divisible by **2** and producing a quotient result value of **7**. As in this example, the shadows' value-pair of **5** and **3** has three bases values: **14**, **7** and **2**.

The value to encrypt must be from **1** to the chosen base value minus **1**. For instance, as in our example, if we choose the base value of **14**, then the value that can be encrypted is from **1** to **13**. If we choose the base of **7**, then the value that can be encrypted is from **1** to **6**. If we choose the base value of **2**, only the value of **1** can be used for encryption. Basically, the chosen base value is of a high value to accommodate large encrypting values. For our example we'll be using the first base value of **14**. The shadows value-pair and its corresponding bases values are illustrated at *Fig. 1*.

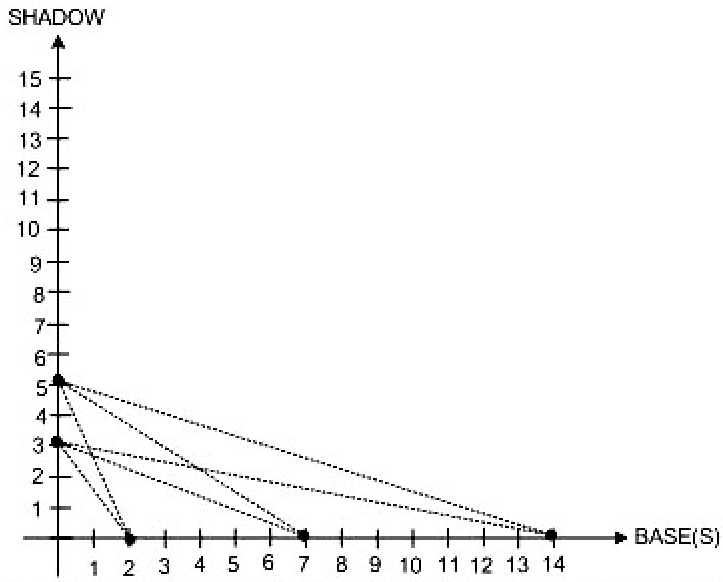

Fig. 1 – The shadows' values of 3 and 5 and their corresponding bases' values of 2, 7 and 14

3.1 First example

The base value is derived with equation (8):

$$B = (Sa \, . \, Sb) \; - \; 1 \tag{8}$$

The encryption is derived with equation (9):

$$e = (M \, . \, Sa) \; mod \; B \tag{9}$$

The decryption is derived with equation (10):

$$d \; = \; (e \, . \, Sb) \; mod \; B \tag{10}$$

Let's use an example by applying the values to the prior three equations.
Sa = first shadow
Sb = second shadow
M = message to encrypt

e = encrypted message
d = decrypted message

Sa = 5
Sb = 3
M = 13

The base value:
B = (Sa . Sb) - 1 $\rightarrow$ *(5 . 3) - 1 = 14*

Public key: *Sa = 5* and *B=14*
Encryption: *e = (M . Sa) mod B* $\rightarrow$ *(13 . 5) mod 14 = 9*

Private key: *Sb = 3* and *B=14*
Decryption: *d = (e . Sb) mod B* $\rightarrow$ *(9 . 3) mod 14 = 13*

3.2 Second example (Exponentiation)

The prior example is of limited use in the science of encryption and decryption since anyone possessing one key pair will be able to easily derive the other key pair and there is remedy.

The solution is to add the base value to the two shadows' values and then applying a common exponentiation positive whole value to both shadows' values and to the base value, let's use the value of **3** for the exponentiation.

First shadow + Base $\rightarrow$ *5 + 14 = 19* $\rightarrow$ $(19)^3$ *= 6859*
Second shadow + Base $\rightarrow$ *3 + 14 = 17* $\rightarrow$ $(17)^3$ *= 4913*
Raised Base value $\rightarrow$ $(14)^3$ *= 2744*

Now we have two new raised shadows and two bases: the new base and the original base.
Raised first shadow: *Sar = 6859*
Raised second shadow: *Sbr = 4913*
Raised Base: *Br = 2744*
Base: *B = 14*

We now have a new public key: *Sar = 6859* and *Br = 2744*, and a new private key: $Sbr = 4913$ and $B = 14$. In our prior equations, after deriving the product between the value to encrypt and the public key's shadow value, a modulus was taken between the product and the chosen base value, we might do this as well before performing any encryption and we'll perform the same process for the public key and the private key.

Public key is the raised first shadow value and the raised base value: $Sar = 6859$ and $Br = 2744 \rightarrow 6859 \ mod \ 2744 = 1371$

The new public key is: $Sar = 1371$ and $Br = 2744$

The private key is the second shadow $Sbr = 4913$ and the original Base: $B = 14$ and we may proceed and take the modulus between the second shadow and the base value.

Private key is the raised second shadow value and the original base value $(Sbr = 4913 \ and \ B = 14) \rightarrow \ 4913 \ mod \ 14 = 13$

The new private key is: $Sbr = 13$ and $B = 14$

Now we may proceed and perform encryption and decryption with the new derived shadow's values and base value using similar equations.

Value to Encrypt: $M = 13$

Public key is: $Sar = 1371$ and $Br = 2744$. The encryption is performed with equation (11):

$$e = (M \ . \ Sar) \ mod \ Br \tag{11}$$

Encryption: $(13 \ . \ 1371) \ mod \ 2744 = 1359$

Private key is: $Sbr = 13$ and $B = 14$. The decryption is performed with equation (12):

$$d = (e \ . \ Sbr) \ mod \ B \tag{12}$$

Decryption: $(1359 \ . \ 13) \ mod \ 14 = 13$

The prior example using a common exponent is much better than the our first one, but it is still has a flaw, since the raised base value may be reversed by using a logarithm, e.g. $\log_3(2744) = 14$, and there is a remedy as we'll see in our next example.

3.3 Third example (Exponentiation + addition)

In our second example the public key was: $Sar = 1371$ and $Br = 2744$; and the private key was: $Sbr = 13$ and $B = 14$. What we'll do next is to take the original base **7** and multiply it by any whole positive value that is greater than **0** and then add the product to both: the raised base value and the raised first shadow value. For our example let's multiply the original base **7** by **3** and we get: $14 \,.\, 3 = 42$. Next, we'll add 42 to both: the raised first shadow value and to the raised base value. Now we have the raised and added first shadow $Sara = 1371 + 42 \rightarrow Sara = 1413$ and the first raised and added first base value $Bra = 2744 + 42 \rightarrow Bra = 2786$. The new public key is: $Sara = 1413$ and $Bra = 2786$. The private key is: $Sbr = 13$ and $B = 14$.

Now we may proceed and perform encryption and decryption with the new derived shadow's values and base value using the similar equations.

Value to Encrypt: $M = 13$

Public key is: $Sara = 1413$ and $Bra = 2786$. The encryption is performed with equation (13):

$$e = (M \,.\, Sara) \; mod \; Bra \tag{13}$$

Encryption: $(13 \,.\, 1413) \; mod \; 2786 = 1653$

Private key is: $Sbr = 13$ and $B = 14$. The decryption is performed with equation (14):

$$d = (e \,.\, Sbr) \; mod \; B \tag{14}$$

Decryption: $(1653 \,.\, 13) \; mod \; 14 = 13$

The third example fixes the flaw found with the second one and if a logarithm is applied to the raised and added base value $Bra = 2786$ it will not work, therefore hiding the actual base value from discovery.

The private key-pair is the one that has the original base value along with the shadow value that was part of the modulus taken between it's raised value and the value of the original base, it can be either one of the two raised shadow's value. The public key-pair is the one with the raised shadow value with the added product of the original base value into it, and the raised base value with the added product of the original base value into it. The security of the shadow encryption relies in the fact that the private key-pair being the smallest of the two key-pairs, and the public key-pair having the highest value of the two key-pairs, it is quite difficult to derive the private key-pair from the public key-pair. The most probable method to use while trying to break the encryption will be the use of brute force and trying each combination until the private key-pair is found. The use of brute force is time consuming and since the shadow's values used may be in length of hundreds of numbers and not being prime numbers, the combination will be quite large, thus requiring a great deal of computer processing power.

4 Conclusion

Whenever two positive whole numbers are chosen, one is the first shadow value and the other is the second shadow value. After the first shadow value and the second shadow value are multiplied and a product is derived and after the value of **1** is subtracted from the derived product value, the first base value is derived. The first base value is added to both the first shadow value and the second shadow value, next, a common exponent is used to raise the first shadow value, the second shadow value and the first base value. After raising the first shadow value, the second shadow value and the first base value to the common denominator, there will be a raised first shadow value, a raised second shadow value and a raised first base value. Finally, multiplying the base value to a positive whole value and deriving a product, then adding the product to the raised first shadow value and to the raised first base value and deriving a raised and added first shadow value, and raised and added first base value. The public key is the raised and added first shadow value and the raised and added first base value and the private key is the raised second shadow value and the original first base value.

The shadow number public encryption algorithm presented in this paper solves the problems that are found in the currently in use public key algorithms without compromising security and at the same time making it possible to offer public key encryption to broad base of electronic devices without high processing power.

References

[1] Diffie, W. and Hellman, M., "New Directions in Cryptography," IEEE Trans. On Info. Theory Vol. IT-22(6) pp. 644-654 (Nov. 1976).

[2] Dorothy Elizabeth and Robling Denning, Cryptography and Data Security, Addison-Wesley, pp. 11-12. (1982)

[3] Pohlig, S. and Hellman, M., "An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance," IEEE Trans. on Info. Theory Vol. IT-24(1) pp. 106-110 (Jan. 1978).

[4] Rivest, R. L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM Vol. 21(2) pp. 120-126 (Feb. 1978).

[5] Dorothy Elizabeth and Robling Denning, Cryptography and Data Security, Addison-Wesley, pp. 101. (1982).

[6] United States Patent US 5,301,235, Arrangement for transforming plaintext into ciphertext for use in a data communication system, column 4, Apr. 5, 1994.

[7] http://en.wikipedia.org/wiki/Cayley

[8] Sarah Flannery with David Flannery, In Code, Algonquin Books of Chapel Hill, pp. 274-282. (2002).

[9] http://cryptome.org/flannery-cp.htm