Prepared by**Olalekan Ihinkalu Ebenezer, Alexander Chefranov, 21.02.2019**

#### 2.5.4.1 The System Setup Design

This phase is also known as the key generation phase for both the original signer and the proxy signer. The scheme is as follows: Where 𝑔 and ℎ are two parameters also are

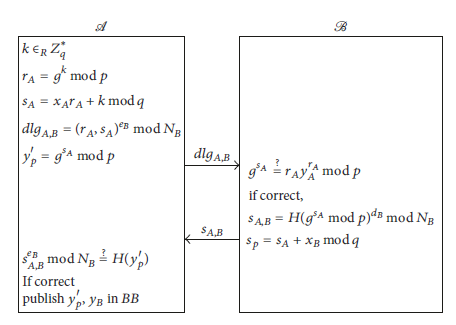


Figure 12: Proxy Phase Design [19]

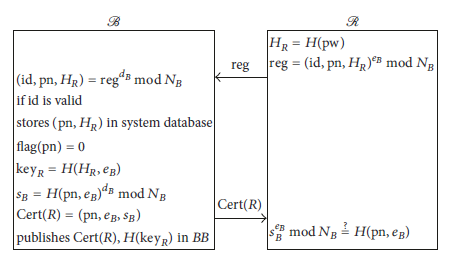


Figure 13: Registration Phase Design [19]

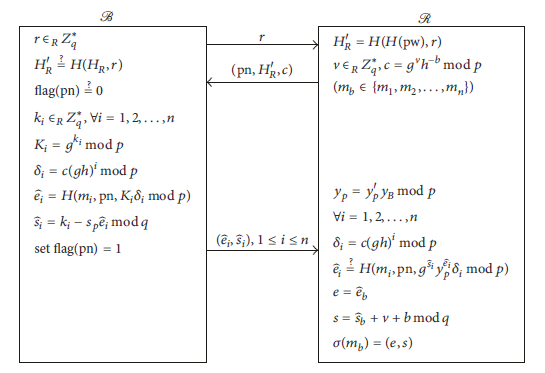


Figure 14: Circling Phase Design [19]

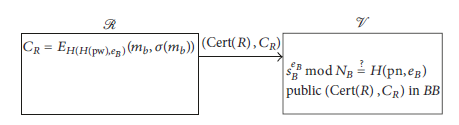


Figure 15: Voting Phase Design [19]

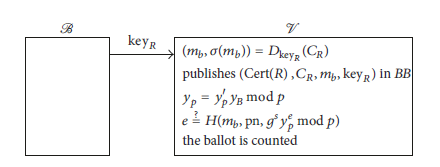


Figure 16: Counting Phase Design [19]

chosen in of order q, (The original signer) A chooses a random number say ∈ as his private to compute the public key

yA= mod 𝑝 (2.25)

also (The proxy creator) B also chooses ∈ as his private key to make a computation of the public key, RSA model of cryptography is used here to do encryption and decryption.

yB= mod 𝑝 (2.26)

The proxy creator (B) chooses two large primes,.

and the proxy creator (B) calculates = × . (2.27)

The proxy creator (B) calculates 𝜙 ( ) = (− 1)(−1). (2.28)

The proxy B chooses such that GCD( , 𝜙()) = 1. (2.29)

,for encryption in RSA.

the proxy creator B calculates = mod () (2.30)

this will is used for the decryption of RSA.

For our numerical example for the system set up phase:

From (2.1) to (2.4), p=11, q =5.

Refer to Table 1 and Table 2, g=9, h=5.

p =11, ={1,2,3,4}

refers (2.25) x**A= 2**, yA= 92 mod 11=4.

Refer(2.26), xB=4 , yB= 94 mod 11= 5.

=11, =5.

refer to (2.27), =11 x 5 =55.

refer to (2.28), 𝜙 ( )= (11-1)(5-1)= 40.

refers to (2.29), =3 .

refer to (2.30), .

#### 2.5.4.2 Proxy Phase Design

The original signer (A) randomly chooses and calculates

= mod 𝑝 (2.31)

(2.32)

and (2.33)

, see (figure 12)

The original signer (A) therefore encrypts the pair of (, ) using (,) (2.34)

and then send it securely to the proxy creator (B), and is then published.

The proxy creator (B) receives it and then decrypts (, ) using (,) and checks whether

are equal, if they are equal, the proxy creator (B) accepts the proxy and calculates

as the secret proxy signature key.

The proxy creator (B) generates a signature

mod

and forwards it to the original creator (A).

the original creator (A) checks whether () mod (2.38)

is true. If it is true then , yB published by A to the bulletin (BB).

For our numerical examples for proxy phase we have:

Recall (2.5)

= 93 mod 11 = 3

‘For (2.21) all values from (2.1), (2.2), (2.3) and (2.5)

𝑠A = 2(3) + 3 mod 5= 9

(, 𝑠A)= (3,9)

= mod 𝑝=94mod 11= 5

the original creator (A) encrypted pair (, ) (= (3,9)) using (,) (=(3, 55)),and =5 will be forwarded to the proxy creator(B)

the proxy creator (B) must decrypt the message using (, ) (= (3,9)) using (,) =(37,55) it checksB) accepts computes

From (2.2) – (2.6), which hold

mod=H(25)= 103

from (2.38)

Using Numerical values:

=H (5) = 53

() mod53

#### 2.5.4.3 Register Phase Design

R (Voter), picks pseudoname (pn) and a password pw, computes =H (pw) (2.39)

by a hash function H(pw), and encryption is done to (id, pn,𝐻(pw)) using (,, and sends it to the proxy creator (B), id is the voter’s identification number.

The proxy creator (B) decrypts (id, pn,) using (,) and checks whether the voter(R) is a legal voter. If the voter is a legal voter, the proxy creator (B) stores (pn,) in the system database, sets flag (pn) = 0, and calculates key𝑅 =(, ) (2.40) and =mod

, returns Cert(𝑅) to the voter (R), and the Cert(𝑅) is published to (BB) the bulletin,

where Cert(𝑅) =(pn, , ). (2.42)

the voter (R) checks whether = H (pn, ) mod (2.43) are equal. If they are equal, the voter (R) has the right to vote.

For the numerical examples for our register phase we have:

From (2.39)

R enters Password as “FEYI” and pseudoname as “OLA”

Each of the Ascii value for each letter is taken added up mod 256

= H(pw) = 301 mod 256 = 45

Encryption is done for each of them id, pseudoname and password, i.e (id, pn,(pw))

(e.g.,=(2,OLA,45)) using (,)=(=(3, 55)) which when encrypted,

we have (8, 49110, 45)

The encrypted (id, pn,(pw)) is decrypted by B to give us (2,OLA,45) then R is checked to be a legal voter and the flag (pn) value is set to be zero.

From (2.40), calculated to give (e.g. KeyR= (45,3)=156)

From (2.41) = mod mod= 5

This is forwarded to R as certificate, Cert(𝑅) =(pn, , ) = (ola,3,5)

Some verification is done by (voter)R from (2.43)

mod 55= 15

H (ola, ) mod =15 which holds

#### 2.5.4.4 Circling Phase

The proxy creator (B) forwards a random number 𝑟 to the voter(R) , this is done after receiving a login request from the voter (R).

The voter (R) calculates = H(𝐻(pw), 𝑟) (2.44)

, then the voter (R) picks a random number v , and computes mod 𝑝 refer (2.9), ( ∈ {}), and forwards values gotten with the following notations (pn,, 𝑐) to B.

The proxy creator (B) examines whether = H (, 𝑟) (2.45)

is correct. If so, the proxy creator (B) checks whether flag(pn) = 0. If true, B chooses

, calculates refer to (2.10), refer (2.11),

, and mod q refer to(2.13), ∀𝑖 =1, 2, . . . , 𝑛, returns () 1 ≤ 𝑖 ≤ 𝑛, to R, and sets flag(pn) =1.

R computes mod 𝑝 refer (2.20), and, for every 𝑖 =1, 2, . . . , 𝑛, he or she calculates and checks whether

is correct. If so, R computes s = b mod q refers (2.17) and e = refers(2.16).

The final signature is 𝜎 () = (𝑒, 𝑠) refer (2.18)

For our numerical values for our circling phase we have:

The proxy creator (B) sends a random chooses number 𝑟=3 to the voter (R) after which the voter (R) requested a login.

The voter (R) does some computations From (2.44)

= H(𝐻(pw), 𝑟)= H(45,3)=156

Also From (2.9)

For our values, n=6, {}={10, 3, 2, 1,4,7}, b=5,

v=2, mod 𝑝=mod 11=

3125 \* x = 1 mod 11= 3152 \* x mod 11=1, x=1, then

= =4

( ∈ {}), He then forwards the message (pn,, 𝑐)= (ola,156,4) to the proxy creator (B)

For (2.45)

= H (, 𝑟)= H(45,3)=156

From (2.10)

, , ,

,

From (2.11)

=4, = 4, =4

=4, =4, =4

From (2.16) it used two variables for the hash function, but here we are introducing for three variables for this hash function

Using a hash function that has three inputs introduced in (2.14).

, (2.45)

= (49+48+79+76+65+51) mod 256=112

= 68

= 63

= 70

= 67

= 122

From (2.13),

recalling (2.2), (2.10),

mod 5 =0, mod 5 = 3, mod 5 = 4

mod 5 =4, mod 5 = 0, mod 5= 1

() = (122,0),(68,3),(63,4),(70,4),(67,0),(122,1)

#### 2.5.4.5 Voting Phase

The voter (R) calculates H(H(pw), ) and uses it as a symmetric key to encrypt (, 𝜎()), produces a cipher , and sends (Cert(𝑅),) to the voting center(V).

The vote center (V) first examines whether (pn, ) mod (Refers to (2.43) holds. If so, V publishes (Cert(𝑅),) to the bulletin (BB).

Every voter (R) can check whether their vote is received by the voting center (V) via the bulletin(BB). If it is not, the voter (R) resends (Cert(𝑅),).

For the numerical values for our voting phase we have:

R uses this symmetric key H(45,3)= 156 to encrypt ([5], [67,2]) which will produce this result for us =(26,1,31)

it will now send (Cert(𝑅),) as ([ OLA, 3, 5 ]),([26,1,31]) to the vote center(V).

the vote center (V) checks if mod 55= 15

is true, for our values is true, the vote center (V ) publishes (Cert(𝑅),) as

for voting phase: each and all voters can check whether their vote was seen by the vote center through the bulletin (BB)

(Cert(𝑅),) as ([ OLA, 3, 5 ]),([26,1,31])

#### 2.5.4.6 Counting Phase

The proxy creator (B) forwards key, key𝑅 = H(𝐻𝑅, 𝑒𝐵) (2.47),

to the voting center (V).

The vote center (V) decrypts using the symmetric key, key𝑅, publishes (Cert(𝑅), ,, key𝑅) to the bulletin (BB) and calculates mod 𝑝 refer to (2.20)

and verifies whether

𝑒 =H(, mod 𝑝) (2.48)

is equal. If so, the signature is valid and the vote is counted.

The vote center (V) publishes the voting result. Everyone can verify and count the ballots via the bulletin.

And lastly for our numerical example for the counting phase we have:

The proxy phase (B) forwards the key, key𝑅=H(45,3)= to (v)the voting center.

the vote center decrypts =([26,1,31]) using key𝑅=, we will then have

([5], [67,2]) then we publish (Cert(𝑅), ,, key𝑅)= ((OLA,3,5) ,(26,1,31),(5,156))\

From (2.19) also adding a final variable

mod 𝑝= mod 11=3

67=𝑒 =H(, mod 𝑝)= H(4, ola,

which is correct.

The vote center (V) publishes the result of the election. Therefore, every voter can verify and count the votes via (BB) the bulletin.

This tested value makes the proposed system valid for our proposed electronic voting system using Novel oblivious and proxy signature.