A Modified Pseudo-Voter Identity (PVID) Scheme for e-Voting Preparation Stage

Nidal F. Shilbayeh
Faculty of computers and Information Technology
University of Tabuk
Tabuk, Saudi Arabia
n_shilbayeh@yahoo.com

Musbah M. Aqel
Faculty of information Technology
AlZarqa University
Amman, Jordan
Aqelm06@yahoo.com

Reem Ali Al-Saidi
Faculty of Information Technology
Middle East University
Amman – Jordan

***Abstract*— In this paper, a modified PVID has been proposed and analyzed. The modified PVID scheme includes some improvements in the used PVID Scheme. In the proposed scheme, the PVID authority issues a voter certificate obtained only once for each eligible voter. This certificate will increase the security in any e-voting scheme that uses the PVID scheme. In addition to that, this certificate only issued once and verified by responder (an additional entity has been added to the PVID) in an attempt to prevent double voting. The issued voter certificate will be multi-encrypted with voter public key (e_v) and PVID authority private key ($PR_{PVID\text{-}Authority}$). A Password Generator (PG) has been added for the voter to generate a unique password for each eligible voter, instead of using the traditional voter password, as an attacker may keep track of the voters' password and compromise it. Security requirement analysis has been given**

Keywords-component; e-voting, PVID, Pseudo Voter Identity, Blind Signature, e-voting Scheme

I. INTRODUCTION

In the recent two decades E-voting became a hot research topic in advanced cryptography, posing several new challenges to fulfill voting general requirements. The challenge arises primarily from the needs to convince the voters that security and democracy requirements such as privacy, accuracy, receipt-freeness and verifiability were achieved and thus reduced their fear towards using E voting by providing them with a trusted E voting that they can rely on.

Many scientists and researchers [1, 5, 6, 8, 9, 10, 11]; explored in E voting cryptographic field in order to overcome the security issues in the election process. Each made his/her own contribution towards a trusted E voting but all agree about the major schemes that can be classified into three main categories: A blind signature scheme, the homomorphic encryption scheme and the mixing net scheme. Each of the above mentioned schemes underlies many protocols, these protocols try to achieve some general security requirements (e.g. by using a blind signature, the voter privacy will be guaranteed). Also, a combination between these schemes is possible depending on the requirements.

The protocols under blind signature scheme are considered as the most commonly implemented due to their practicality and applicability, at which the voter first obtains a token, which has been blindly signed by the administrator and which is only known to the voter her/himself. Later, the voter sends her vote anonymously, with this token as proof of eligibility to the auditing for counting. While, in the homomorphic scheme the voter cooperates with the administrator in order to construct an encryption of his/her vote. Then, the administrator exploits homomorphic prosperities of encryption algorithm to compute the encrypted tally directly from the encrypted votes. For the mixing net scheme is the most common approach to achieving anonymity. The general concept of mix nets is based on permuting and shuffling the messages in order to hide the relation between the message and its sender. However, the details, as to the implementation of mixing protocols, change depending on configurations and arrangements of mix-nets. In the existing voting protocols [1, 5, 6, 8, 12], voter generally uses his/her real identity while communicating with the authorities. While, in PVID scheme introduces by [3], voter uses pseudo identities, which have no relation with the voter's real identity and are unlinkable to it.

In PVID scheme, voter prepares a list of blinded identities and then he/she obtains blind signature for each of them separately by interacting with the approval authority in one session. Later, voter extracts anonymous pseudo identities (PVIDs) which are unlinkable to voter registration identity. Each of PVID is selected by the voter and blindly signed by the approval authority after verifying voter eligibility. These list of blindly signed pseudo identities is only known by the voter and uses them throughout the entire communication while interacting with the authorities.

Some modification had been applied to the used PVID scheme. PVID authority issues a voter certificate obtained **only**

once for each eligible voter and verified by an additional entity call the responder. This certificate will increase the security in any e-voting scheme that uses the PVID scheme. In addition to that, this certificate only issued once in an attempt to prevent double voting. Here a PVID authority acts as a Certificate authority (CA). In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users ("endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together. The issued voter certificate will be multi-encrypted with voter public key (e_v) and PVID authority private key ($PR_{PVID\text{-}Authority}$). Also, An optional step has been added for the voter to contact a password generator (PG) to generate a unique password for each eligible voter, instead of using the traditional voter password, as an attacker may keep track of the voters' password and compromise it.

II. BACKGROUND

A. Blind Signatures

The concept of blind signature was introduced by [5, 6]. Chaum demonstrated the implementation based on RSA signatures. It allows the realization of secure voting schemes, protecting the voter privacy.

Initially the blind signature is used within e-cash system (e-cash) to guarantee owner anonymity, as in E voting scheme the motivation is to keep the voters anonymity as well, so this technique can be applied [15].

The idea of blind signature allows a signer to sign a document without revealing its contents similarly in a real life world to sign a carbon paper lined envelopes. Writing a signature on the outside of such envelope leaves a carbon copy of the signature on a slip of paper within the envelope. When the envelope is opened, the slip will show the carbon image of the signature.

A distinguishing feature of blind signatures is their unlinkability: The signer cannot derive the correspondence between the signing process and the signature, which is later made public.

The blind signatures can be accomplished by the following steps:

1) The authority key is given:

   (e, n) public key of the signer

   (d, n) private key of the signer

2) The voter's purpose is to let the authority to sign the vote, say v, without revealing its content (Blind Signature).

The voter generates a random number, r that satisfy the following formula :

$$gcd(n,r)=1$$

The voter using this random variable r and authority public key component e to blind his/her vote and calculates x as in the following formula:

$$x = (r^e v) \bmod n$$

3) The voter asks the authority to sign the vote using its private key. Noted that the authority cannot derive any useful information from x.

$$t = x^d \bmod n$$

4) The authority sends the signed vote to the voter:

$$t = x^d \bmod n$$

$$t = (r^e v)^d \bmod n$$

$$t = (r^{ed} v^d) \bmod n$$

$$t = r v^d \bmod n$$

5) As the voter know the random value r, she /he can remove it from the signed vote by taking r^{-1} to both side :

$$r^{-1} t= v^d \bmod n$$

$$s= v^d \bmod n$$

Where s is the vote v signed by the use of the authority private key preventing the authority from learning the signed vote v.

B. Pseudo Random Number Generator

Pseudo random number generator (PRNG) [13] is a deterministic algorithm to generate a sequence of numbers with little or no discernible pattern in the numbers. The sequence is not truly random since it is determined solely by a relatively small set of initial values. Although sequences that are closer to truly random ones can be generated using hardware random number generators, most pseudo random generator algorithms produce sequences which are uniformly distributed. Getting truly random data is typically based on nondeterministic physical phenomena. In the deterministic environment of computer systems, people often use deterministically generated pseudorandom data. The truly random data are used only for deterministic pseudorandom number generators and after seeding, an arbitrary amount of pseudorandom data is always available. The PRNG is in fact a deterministic finite state machine, which implies that it is at any point of time in a certain internal state.

This PRNG state is kept confidential since the PRNG output must be unpredictable. Many classes of PRNGs exist, but the goal of a PRNG in cryptography is the production of pseudo random data that are computationally indistinguishable from statistically ideal random data. A PRNG is cryptographically secure, on condition that it is computationally infeasible to predict the next output even if all the previous outputs and the complete algorithm are given. Basic types of PRNGs utilize linear feedback shift registers, NP hard problems of number and complexity theory and typical cryptographic functions/primitives. Mechanisms necessary for recovering from the state compromise are used only in the last category.

C. Public key certificate

In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic

document which uses a digital signature to bind a public key with an identity, information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual. In a typical public key infrastructure (PKI) the signature will be of a certificate authority (CA). Here a PVID authority acts as a CA. In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users ("endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together.

D. Trapdoor commitment scheme

A trapdoor commitment scheme [7] is a function with associated a pair of matching public and private keys. The main property that want from such a function is collision-resistance: unless one knows the trapdoor, it is infeasible to find two inputs that map to the same value. On the other hand, knowledge of the trapdoor suffices to find collisions easily. The trapdoor commitment scheme [2] is based on bit commitment scheme cryptosystem. A trapdoor commitment scheme consists of key generation algorithm, commitment function, and collision-finding function.

- Key Generation

The key generation algorithm, on input a security parameter l produces a modulus N product of two safe primes of size $l/2$ together with a square h of maximal order in G. The public key is given by N and h. The factorization of the modulus is the private key(p,q).

- Committing a Message

To commit to a message $m \in \mathbf{Z_N}$ the sender chooses a random number $r \in_R Z_{N\lambda(N)/2}$ and sets

$B=C(r,m)=h^r (1+mN) \bmod N^2$, and sent

(B,r,m) to the receiver.

- Collision-Finding Function

Now given a commitment $B=C(r,m) \in G$ together with the corresponding(r,m), knowing the factorization of the modulus, one can find collisions, for any message m' as follows $r' = r + (m - m')d\lambda(N) \bmod N\lambda(N)/2$.

Thus the receiver can get

$$B = C(r,m) = C(r',m') \in G$$

i. Trapdoor commitment scheme in e-voting

Trapdoor bit commitments were introduced in voting schemes as a means of solving the problem of coercion. As well as the convenience for the voters is an important property, schemes using bit commitments do not seem practical for use in large scale elections. In a trap-door bit commitment scheme, where a voter v has committed to a message M, it is possible for v to open M in many different ways. This may seem to contradict the purpose of commitment schemes, but the following scenario shows how this property can be useful in E voting:

1. A voter v commits to a voting intention B. v then provides the authority A with the information necessary to open the commitment on B, but keeps a secret value, the trap-door, to him. This enables only the voter to open the commitment in different ways.

2. In the tallying phase, A opens the commitment on B. No interaction from the voter is required.

3. If a coercer forces v to demonstrate how he has voted, v can use the secret trap-door to claim a voting intention different from his/her actual intention, without the coercer being able to detect it. But the main difficult requirement to achieve within a trap door commitment scheme is the secret keeping for the trap door value as one generally assumes the coercer has access to the same information as the voter..

E. Elliptic Curve Cryptography

The Elliptical curve cryptography (ECC) is a public key encryption technique based on *elliptic curve theory* that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large primes. The technology can be used in conjunction with most public key encryption methods, such as RSA, and Diffie-Hellman. ECC can yield a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. ECC was developed by [14] a mobile e-business security provider, and was recently licensed by Hifn, a manufacturer of integrated circuitry (IC) and network security products. RSA has been developing its own version of ECC.

ECC is based on properties of a particular type of equation created from the mathematical group (a set of values for which operations can be performed on any two members of the group to produce a third member) derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse.

III. THE MODIFIED PVID SCHEME

Before Voter has a registration identity (RegID) which can be any widely used identity such as national identity number or social security number. RegID can be a government-issued voter ID as well. On the Election Day, voter uses his/her RegID to authenticate himself to the system. In almost all blind signature based voting protocols, voter tries to obtain blindly signed ballot and/or his/her cast or part of them. In PVID scheme, voter obtains a list of blindly signed anonymous pseudo identities and uses them instead of real RegID while interacting with the authorities.

The PVID, the responsible authority, issues a blind signature on voters PVID-list after checking voter eligibility.

The trust of this authority is very important as it can blindly sign ineligible voters PVID list. As soon as the voter obtains a PVID-list, he/she can use in later communication instead of using the voter RegID (public key) as this will be vulnerable for attack. By applying the PVID scheme in the proposed scheme, the privacy degree will be increased. Whatever, PVID is consider as one of the most practical scheme as it apply only blind signature to obtain the authority signature.

It provides as well privacy without requiring any complex mechanisms and computational operation. RSA is used as a public key cryptosystem. A pseudo random number generator is used to feed PVID with random number. By using the *elliptic key cryptography*, the voter will generate his/her associated key pairs, public and private keys (d_v, e_v) as the following:

$E_q(a, b)$: elliptic curve with parameter a, b and q ,where q is a prime or an integer of the form 2^m G point on elliptic curve whose order is large value n

Voter pair key generation	
Select private d_v	$d_v < n$
Calculate public e_v	$e_v = d_v \times G$

By using cryptographically DES cyclic encryption random number generation which generates random numbers as shown in figure 1. A counter with period N provides input to the encryption logic. For example, if a 56-bit DES keys are to be produced, then a counter with period 2^{56} can be used. After each key is produced the counter is incremented by one .Thus, the pseudorandom numbers produced by this scheme cycle through a full period: Each of the output $X_0, X_1, \ldots\ldots\ldots\ldots X_{N-1}$ is based on a different counter value and therefore $X_0 \neq X_1 \neq \ldots\ldots\ldots\ldots \neq X_{N-1}$. Because the master key is protected, it's not computationally feasible to deduce any session keys (random numbers) through knowledge of one or earlier session keys.

PVID scheme has four stages: ID generation stage, blinding stage, signing stage and PVID obtaining stage. The detailed descriptions of these stages will be described as the following:

- **ID Generation Stage**

Voter generates *k* pseudo identity numbers and prepares ID-list. Each ID contains the election data, authority data and a big random number (generated by a PRNG shown in Figure 1), so it is constructed as follows; for each ID, the authority data should be different whereas the random number should be same. Using same random number provides that IDs belong to one voter.

ID_i = (Election Data, Authority Data, Random Number)

ID-list = $\{ID_1, ID_2, \ldots..ID_k \mid IDi \text{ is } i^{th} \text{ pseudo identity}\}$.

Now, voter has an ID-list that he wishes to have signed each ID_i in the list by PVID Authority. Voter does not want PVID Authority to learn anything about ID_i. More details are indicated about ID-List in figure 2.

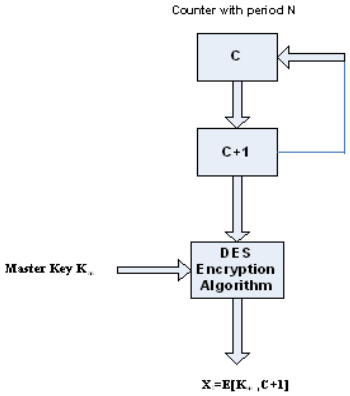


Figure 1 Pseudo Random Number Generation (PRNG) from a counter

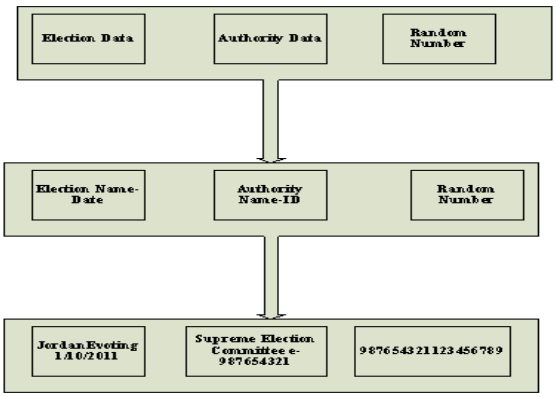


Figure 2 ID-List Details [3]

- **Blinding Stage**

Voter generates a random blinding factor number *r* (using PRNG shown in figure 1) and calculates blinded message m_b for each ID_i, and obtains a list of blinded IDs which is M_b as shown in eq. 1, 2, and figure 3.

$$m_{bi} = (r^e [ID_i] \bmod n \quad \text{where } \gcd(n, r) = 1 \qquad (1)$$

$$M_b = \{m_{b1}, m_{b2}, m_{b3}, \ldots\ldots\ldots, m_{bk}\} \qquad (2)$$

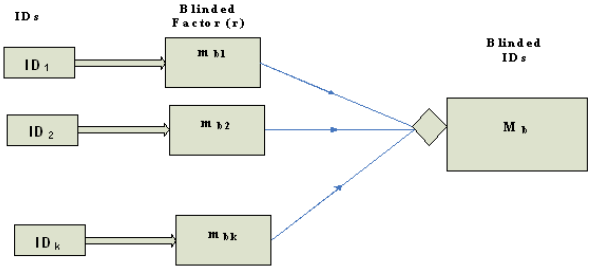
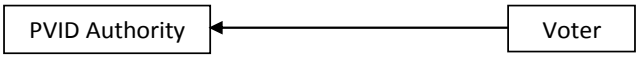


Figure 3 Blinding stage [3]

Voter signs the list M_b and obtains $d_v(M_b)$. Then, he/she will encrypt his/her RegID and $d_v(M_b)$ with PVID authority public key ($PU_{PVID\text{-}Authority}$) and obtain $E(PU_{PVID\text{-}Authority}(RegID, d_v(M_b))$. Voter will send this message to PVID authority (see figure 4). As the value m_b is blinded by the random value r, it can't derive any useful information from it. This message will accompany with another message that contains the following $\{e_v, E(d_v((RegID)_v)))\}$. Noted that the voter send his/her RegID to let the PVID authority check the RegID against country election registration laws. The voters' public key (e_v) is sending in clear for two later purposes. First, for checking voter signature. Second, for encrypting the list of blinded voter identities if he/she permitted to vote.



$E(PU_{PVID\text{-}Authority}((RegID)_v , d_v(M_b)+ e_v, E(d_v((RegID)_v)))$

Figure 4 Voter-PVID authority interaction

- **Signing Stage:**

PVID authority will decrypt the received message, and obtains the voters *RegID* and $d_v(M_b)$. It will verify the voters' eligibility by checking his/her RegID against the civil status data base. If the voter is eligible and hasn't made any request yet, PVID uses voter public key (e_v) and check the voter signature on M_b. For each eligible voter, PVID authority signs each blinded message m_b in the list of M_b and calculates m_{bs}. Subsequently; PVID authority obtained a list of blindly signed IDs which is $M_{bs.}$ As indicated in Figure 5.

$$m_{bs_i} = m_{b_i}^d \bmod n \quad (3)$$

$$M_{bs} = \{m_{bs_1}, m_{bs_2}, \dots\dots\dots, m_{bs_k}\} \quad (4)$$

Then, PVID authority encrypt the list M_{bs} with the voter public key (e_v (M_{bs}), for PVID authority to supply only one PVID for each eligible voter it will change the voter status and issue a voter certificate (review the literature for public key certificate), $Cert_v$= E($PR_{PVID\text{-}Authority}$ (e_v ($Time_1$ || RegID || ElectionData || e_v])))) these will be send to the voter (as shown in figure 6 (step(1)), also this will be accompanied with PK_{voting}, that the PVID authority received from the commissioner (as shown in figure 6 (step(2)). The PK_{voting}, the key used in overall voting and know to each involved entity, was encrypt with the commissioner private key E (PR-Commissioner(PK_{voting}), as it received the PVID authority will decrypt it with the commissioner public key ($PU_{commissioner}$), confidentiality and authentication between communicating entities will also achieved here by such an encryption and decryption operations. Furthermore; the commissioner will send a hash for voting public key (PK_{voting}) for verification purpose to the Electronic Ballot Generator (EBG), (as shown in figure 6 step (3)). Noted that all hash values will be introduced using SHA-1 cryptographic algorithm.

As the voter status had been changed and the certificate was issued to the eligible voter, this will achieve the e-voting requirements of democracy and completeness, a copy of the certificate will keep in the repository. So if a voter try to vote again, any such attempt will be easily detected either by checking voter status (vote or unvote) in civil status database or by checking the issued voter certificate in election certification repository database, depend on that the certificate obtains only once (the issued certificates are kept in database) for authorized voters only, which permit voter to participate in election during the specified election period, and send back a component [A] $_{PR\text{-}PVIDAuthority,}$ so the voting process is canceled. Otherwise (in case the voter hadn't voted before) the value component [O] $_{PR\text{-}PVIDAuthority}$ is sent and voting registration continued.

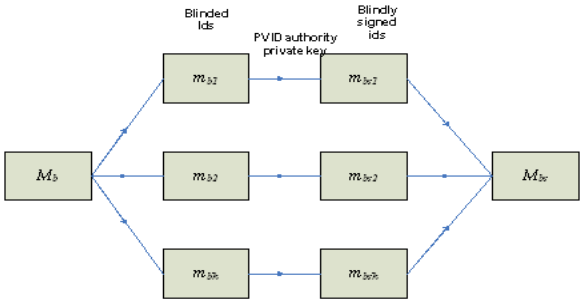


Figure 5 Signing Stage [3]

Another way is provided here in order to detect a double voting attempt, the PVID authority supply only one PVID for each eligible voter and doesn't make any sign for the blinded identities if the voter had been signed before. The issued voter certificate will be multi-encrypted with voter public key (e_v), public key encryption (asymmetric encryption) will be used here, and PVID authority private key ($PR_{PVID\text{-}Authority}$), as shown in figure 6.

- **PVID Obtaining Stage**

As the voter received the blindly signed ID list M_{bs}. He/she will decrypt them and can easily now obtain PVIDs, the true sign of IDs, by removing the blinding factor *r* from each m_{bs}. Voter carries out the following operations for each m_{bs} in the list M_{bs} in order to obtain $PVID_i$ for each ID_i. Also the voter will obtain his/her certificate now, by decrypting it using the PVID Authority public key ($PU_{PVID\text{-}Authority}$) and make sure about the decrypting information ($Time_1$ || RegID || ElectionData ||e_v) by this he can trust such authority.

(1) (e_v(M_{bs} , PK_{voting}) +$Cert_v$= E($PR_{PVID\text{-}Authority}$ (e_v ($Time_1$ || RegID || ElectionData ||e_v])))+[O] $_{PR\text{-}PVIDAuthority}$

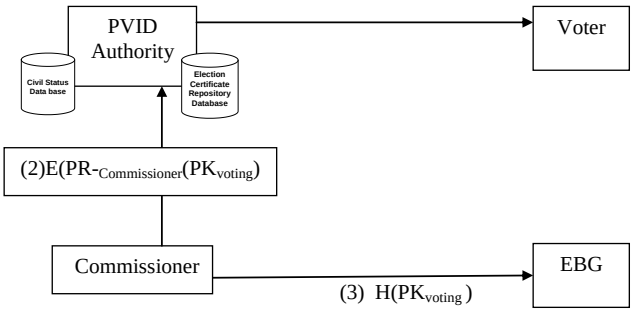


Figure 6 Message passing in signing stage

$$m_{bs_i} = m_{b_i}^{d} \bmod n = (r^{e}[ID_i])^{d} \bmod n \quad (5)$$

$$m_{bs_i} = r^{ed}[ID_i]^{d} \bmod n = r[ID_i]^{d} \bmod n \quad (6)$$

$$PVID_i = r^{-1} m_{bs_i} \bmod n = [ID_i]^{d} \bmod n \quad (7)$$

$PVID_i$ is the sign of PVID authority on the voters selected ID_i. Then, the voter will calculate PVID-list with PVID as shown in figure 7(PVID-list={ PVID1, PVID2,……. PVIDk}).

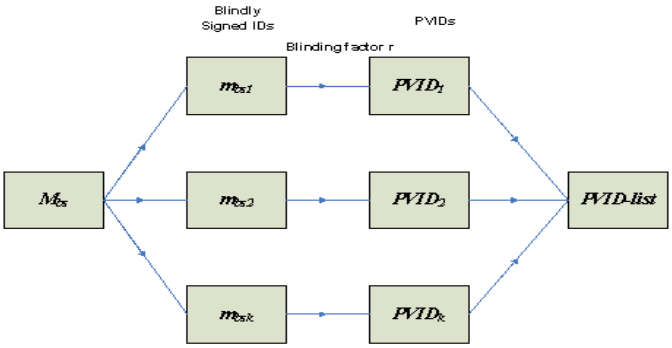


Figure 7 PVID obtaining stage [3]

Now, voter has valid and signed pseudo identities that are unlinkable to his/her real RegID. Voter can use them in the proposed E voting scheme without providing his/her RegID to the voting authorities. Moreover, he/she can directly communicate with the authorities without requiring any anonymous channel since PVIDs aren't linkable to his/her real identities.

When voter uses his/her PVID, the authority only verifies the signature on PVID by unsigning it with PVID authority public key and simply checking the election data and authority data. Noted that the same strategy had used under E cash environment to assure a non-repudiation service as (spognardi 2006) indicated in his survey. Here; it had been used according to PVID scheme [3] in E voting environment. As explained above some modification had been applied (e.g. the voter issue certificate) and other more in order to provide the secure E voting scheme.

The optional step associated with the proposed scheme that preferred by the voter to contact a password generator (PG) that is responsible to generate a unique password for each eligible voter, instead of using his /her own traditional password that usually he/she used in other website (attacker may keep track of a user password and compromise the voter password). The voter will send {e_v, *PVID-list*} to password generator (PG) as shown in figure 8.

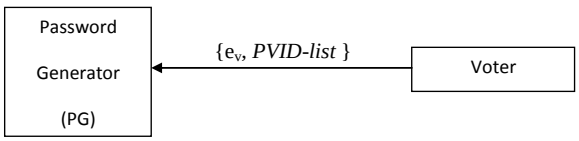


Figure 8 Voter-PG interaction

The password generator is responsible to generate a unique password for each eligible voter, the following algorithm [13] can be used to generate such passwords under E voting environment, as well as the PG received the signed PVID authority pseudo identity (*PVID-list*), it will verify the PVID authority sign and signed it again with a PG private key (PR_{PG}), for non-repudiation goal $[PVID\text{-}list]_{PR\text{-}PG}$, so the voter can trust a such generator.

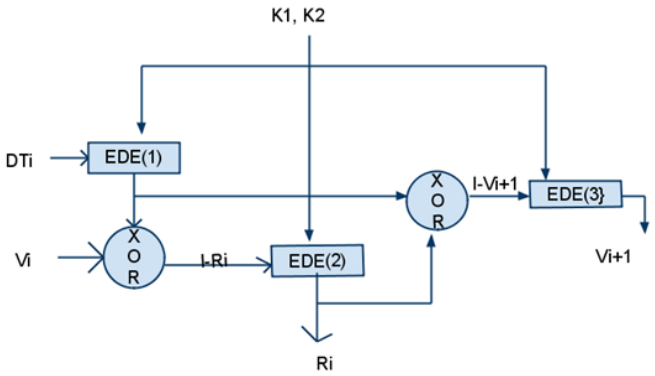


Figure 9 ANSI X9.17 for pseudorandom number generator [13]

The password generator algorithm will depend on the ANSI X9.17 PRNG cyclic generated random number encryption infrastructure (as shown in figure 9 and table 1). It will use triple DES for encryption . The ingredients are as follows :

- **Input :**

✓ 64 bit representation of current date and time, which is updated at each generation.

✓ 64 bit representation which is a combination of voter public key and signed blinded identities (e_v +PVID-list) that differs at each round (each vote has a different identity).

- **Keys :**

Making use of the three triple DES encryption modules, with a 56 bit keys, which must be kept secret and are used for password generation.

- **Output :**

The output consist of a 64-bit for password (R_i) and a 64-bit seed value .

Several factors contribute to the cryptographic strength of the proposed approach. It involve a 112-bit key and three Encryption Decryption Encryption (EDE) for a total of nine DES encryption. It is driven by two input, the date and time values and the voters public key with the PVID-list, which will be differ at each round. So, the amount of material that must be compromised by an opponent is overwhelming even if R_i is compromised it would be impossible to deduce V_{i+1} from the R_i because an additional EDE operation is used to produce the V_{i+1}.

Table 1 Password generator algorithm Description

DT_i	Datre /time at the beginning of ith generation stage
$(V)_i$	Combination of voter identity and public key at round i
R_i	Finally generated password
K_1,K_2	DES key used at each round
Then : R_i=EDE([k_1,k_2], [V_i [XOR] EDE ([K_1,K_2], DT_i)]) V_{i+1}=EDE([k_1, k_2], [R_i [XOR] EDE([k_1,k_2], DT_i) Where EDE ([k_1,k_2] ,X) refers to the sequence encrypt-decrypt-encrypt using two key triple DES to encrypt X .	

After the password is generated, it will be encrypted with the voter public key, and sent to the voter, this will also be accompanied with the password generator (PG) sign for the signed PVID authority ($[PVID\text{-}list]_{PR\text{-}PG}$), as shown in figure 10.

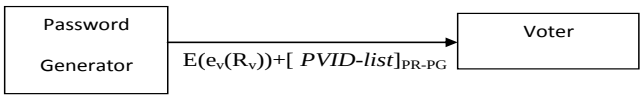


Figure 10 PG –Voter interaction

As the voter recives the generated password, encrypted with the voter generated public key (e_v), it will be decrypted using voter private key $D(d_v(R_v))$ and get the generated password. Also the voter will verifiy the PG sign $[PVID\text{-}list]_{PR\text{-}PG}$ using the password generator public key (PU-PG). By this way the confiednialty and a trust between communicating entities is achieved. Till now the voter securly has the generated $Cert_v$,e_v,d_v,and the signed blinded voters identites and R_v.

IV. Analysis and Discussion

After In this study, we do not propose an e-voting scheme. However, any e-voting scheme that is use the modified PVID scheme can fulfill some of the e-voting requirement without requiring any complex mechanism and computational operations. In comparison with the used PVID, the proposed modifications will improve the privacy and the security.

In the following details, we will evaluate the modified scheme according to the security requirement definitions given by [4]:

- ***Privacy*** (Voter-Vote relationship cannot be revealed): If

  $\forall d \in D \forall v \in V \forall e \in E[\neg(\exists f(S,W,d,v) = e)]$

  for a voting scheme *VS*, then *VS* satisfies privacy.

Since a blind signature scheme is used, there is no particular link between the voter real identity and the pseudo identity especially when generated using our password random generator (PG). In order to find any relation between them, the generated random number used to create blinded message should be known. Otherwise, adversary should break RSA cryptosystem because the PVID scheme uses blind signature based on RSA public key cryptosystem, which is infeasible.

Frankly Speaking, after the voter obtains PVID list, the voter no more use his/her RegID, thus no adversary, including all authorities can find a function *f* such that $\forall v \in V \forall e \in E[\exists f(S,W,D,v) = e]$. so nobody can break the voter-vote unlinkability.

Additionally, by relying on the blind signature and according to its definition ,there is no function f satisfying $\forall v \in V \forall e \in E[\exists f(p) = e]$ in the proposed scheme guarantee that all votes will kept secret due to the blindness property under blind signature and thus no participant other than a voter should be able to determine the value of the vote cast by that voter as the voter sign his/her blinded vote without the sign authority (administrators) know the actual vote (see table 1 that summarize the case related to privacy).

Table 2 Privacy requirement details

Main Requirement	Requirement Details	How it is satisfied	Assumption
Privacy	Voter-vote unlinkability	Applying PVID scheme + Blind signature protocol	
	Voter-voteIP untraceability	There is no point in trying to trace the voter IP since nobody can guarantee whether or not the voter accesses over a dynamic IP, he uses the voting pool or any other public network, and he employs any IP anonymizer application. In case of a voter having a static IP and not taking any care about it, then IP untraceability may fail if authorities corrupt.	none of the authorities keeps IP of the voters and releases them

- *Eligibility* (Each vote counted in the tally should be cast by an eligible voter): let

$f: V \to B, f(v_i) = b_j and g: B \to A, g(b_j) = a_j . If \forall v \in V[f_{ae}(g(f(v))) \in E]$

for a voting scheme *VS*, then *VS* satisfies eligibility.

This requires a forgery of the PVID-list signature which is impossible as the PVID authority issues blind signature on voters blinded ID after checking against country election registration laws (e.g. above 18 years old). Also, the issued PVID authority certificate will never be forging due to the additional entity (responder) that verifies the certificate.

- *Uniqueness* (There should be at most one valid vote for each eligible voter in the final tally). Let:

$f: V \to B, f(v_i) = b_j and g: B \to A, g(b_j) = a_j$

$. If \forall v_i \in V \forall v_j \in V[f_{ae}(g(f(v_i)) = f_{ae}(g(f(v_i)) = f_{ae}(g(f(v_j)) \leftrightarrow i = j]$

for a voting scheme *VS*, then *VS* satisfies uniqueness.

(1) Since the *PVID-list* values (e.g. $PVID_1$ differ from $PVID_2$ and both are unique in the same list) and can be verified using the PVID authority public key ,there is only one vote counted for each voter. There exit such a function $f: V \to P, f(v_i) = p_j$,so uniqueness is satisfied as there is a true value for

$\forall v_i \in V \forall v_j \in V[f(v_i) = f(v_j) \leftrightarrow i = j]$

(2) Depending on the uniqueness of the PVID authority issued voter certificate ($Cert_v$) obtained only once for each eligible voter and permitted them to participate in election during the specified election period, under the assumption that the PVID authority is trusted and thus can't forge certificate, if it forged a responder will verify it (responder and PVID authority can't collude, disjoint set with direct communication only)

- *Receipt-freeness (Uncoercibility)* (Voters cannot prove their votes and thus No coercer can figure out a voters vote by forcing him):

If $\forall a \in A \forall v \in V[\neg(\exists f(D,W,a) = v)]$ for a voting scheme *VS*, then *VS* is receipt-free.

By applying the trapdoor commitment scheme, a vote recasting due to the fault tolerant is possible. If someone coerces a voter, even by only being physically next to him, the voter will cast in a way the coercer influences. Later, he/she can change his/her vote, by recasting a new vote which will automatically discard the old one in the counting stage. Even if the voter records his/her voting activity, still he cannot convince the coercer of the content of his/her vote due to recasting. That is, practically it is not possible to coerce or vote buy, since nobody can know whether the current vote will be the final one or not ; due to the trapdoor commitment and as the voter can provide $C(_{fake}\ r\ ,_{fake}\ C_j)$ and can be verified but can't find that a $C_{j\text{-}fake}$ is a fake credential by a coercer, at the same time in the counting stage the voter will send the committed ballot by applying a trapdoor commitment scheme and related to the property that the voter can also find collision: $B_c=C(r_1,C_j)=C(_{fake}r_1,_{fake}C_j)$ $||B_B=C(r_1,B_t)=C(_{fake}\ r_1,_{fake}B_t)$ and provide $B_c=C(_{fake}\ r_1\ ,_{fake}C_j)$ $||B_B=C(_{fake}\ r_1,_{fake}\ B_t)$. As the coercer has the ability to monitor the communication between voter and counter through anonymizer and verify without finding that $C_{j\text{-}fake}$ is a fake credential .Thus, there is no function *f* satisfying $\forall a \in A \forall v \in V[(\exists f(D,W,a) = v)]$ for the proposed scheme. In practice, there is no point in coercing either physically or socially. Therefore, uncoercibility is achieved.

V. CONCLUSION

In order to achieve voter privacy at e-voting scheme, the PVID scheme provides PVIDs which are anonymous pseudo identities and blindly signed by the PVID authority in the registration stage of any implemented e-voting system. By applying the proposed modification suggested in modified PVID scheme, the privacy degree will be increased. It provides as well privacy without requiring any complex mechanism and computational operations. In addition to that, this certificate only issued once in an attempt to prevent double voting. Also, the password generator (PG) used in the modified version of the PVID increase the security and cannot be predicted by the attacker in comparison with the traditional one generated by the voter.

REFERENCES

[1] Benaloh J.C , A verifiable secret-ballot elections, *PhD thesis* (published), New Haven, Yale University, and Institute of information Technology, USA, 1987.

[2] Bresson E ,Catalano D, and Pointcheval, "A simple public key cryptosystem with a double trapdoor decryption mechanism and its applications". *In Laih CS, ed. Aciacrypt 2003*. LNCS 2894, pp.37-54 , Berlin: Springer-Verlag.

[3] Cetinkaya O. & Doganaksoy, "Pseudo-Voter Identity (PVID) Scheme for E voting Protocols" , *In Proceedings of the International Workshop on Advances in Information Security(WAIS'07) in conjunction with ARES'07, Vienna, Austria*, pp. 1190-1196,2007.

[4] Cetinkaya O, and Koc M.L, "Practical Aspects of DynaVote E Voting Protocol.", *Electronic Journal of E Government*, vol. 7, no., pp327-338, 2009. available online at www.ejeg.com.

[5] Chaum D., "Untraceable electronic mail, return addresses, and digital pseudonyms", *Comm. of the ACM journal*, vol 24, pp 84-90, 1981.

[6] Chaum D., "Blind signatures for untraceable payments", *In Proceedings of Crypto 82*, pp. 199 – 203, 1983, Plenum Press, New York.

[7] Chen X, Wub O, Zhang F, Tian H, Wei B, Lee B, Lee H, Kim K, "New receipt-free voting scheme using double-trapdoor commitment", *Information Science Magazine*, pp.1493-1502, 2011.

[8] Cohen J. and Fischer M., "A robust and verifiable cryptographically secure election scheme", *In Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp 372 – 382, 1985, IEEE Press.

[9] Cramer R., Gennaro R., Schoenmakers B., and Yung M. , *Multi-Authority Secret-Ballot Elections with Linear Works*, Springer-Verlag, Vol. 1070 of Lecture Notes in Computer Science, pp. 72-83, 1996.

[10] Davenport B., Newberger A, and Woodar J., "Creating a Secure Digital Voting Protocol for Campus Elections", *Princeton University, Department of computer engineering and computer Science*, UK, 1996.

[11] DuRette B.W, "Multiple Administrators for Electronic Voting", *Msc. Thesis* (published), Massachusetts Institute of Technology MIT, Cambridge, USA, 1999.

[12] Fujioka A., Okamoto T., & Ohta K., "A practical secret voting scheme for large scale elections", *proceedings on the theory and application of cryptographic techniques*, pp.244-251, 1992, Springer Verlag, Australia.

[13] Kelsey J., Schneier B., Wagner D. & Hall C., " Cryptanalytic Attacks on Pseudorandom Number Generators", *LATEX macro package with LLNCS style*, 1997.

[14] Menezes A., *Elliptic Curve Pub1ic Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[15] Wen X., Niu X., Liping J and Tian Y., "A weak blind signature scheme based on quantum cryptography", Optics Communications, Vo. 282, pp. 666-669, 2009.