E-Voting Authentication Preparation Scheme (EV-APS) Based on Evox-MA and REVS E-Voting Blind Signature Protocols

Reem Al-Saidi, Nidal Shilbayeh*, Ebrahim Elnahri, Khaled Alhawiti
*Corresponding Author's Email: nshilbayeh@ut.edu.sa, n_shilbayeh@yahoo.com

Abstract – **In this paper, we propose a modified and efficient E-Voting Authentication Preparation Scheme (EV-APS). The proposed new scheme acts as an improvement over the last two Evox-MA and REVS, E-voting protocols that based on the blind signature. The proposed schemes are suited for large scale E-Voting over the internet, and overcome the problems associated in these well-known protocols and achieve all e-voting security requirements. The new modified protocol applies some cryptographic technique to enhance some security aspects. Some of these modified security aspects are Kerberos authentication protocol, PVID scheme, responder certificate validation, and the converted Ferguson e-cash protocol.**

Keywords – **E-Voting, Preparation Stage, Blind Signature Protocol, Nonce Based Authentication Scheme, Kerberos Authentication Protocol, Pseudo Voter Identity Scheme PVID, Evox-MA, REVS.**

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]-[8], [17] 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). The protocols under blind signature scheme are considered as the most commonly implemented due to their practicality and applicability. The last common two blind signature protocols under E-Voting environment are Evox-MA [7] and REVS [8]

Generally, E-Voting consists of three main stages: the preparation stage, voting and counting stages [9].

Authentication is an important part at the preparation stage and thus of the overall E-Voting process, both for the E-Voting system authenticating the human as eligible voter without sacrificing secret balloting, and for the voter authenticating authority control E-Voting [10]. Voters want the capability to vote remotely, but this makes both directions of authentication more difficult. The proposed scheme combined more than scheme or modified protocol to assure the authentication requirement.

Up to now, no complete solution is provided to gurantee that only authorized voters vote either in theoretical or practical domain. Neither Evox-MA nor REVS, last two E-Voting blind signature protocols, prevented Dos attack at the prepration stage so the attacker can fill the counter buffer with garbage votes and coruupted the overall E-Voting process

This paper is organized as follows: Section 2 provide general background required to understand the proposed authentication scheme. Section 3 presents the proposed authentication E-Voting Authentication Preparation Scheme. Section 4 and 5 discussion and conclusion.

II. BACKGROUND

A. Public Key encryption

In the public key encryption, also known as asymmetric encryption, there are two keys: an encryption key Kpub (public key) and a decryption key Kpri (private key). The encryption of a message m with Kpub results in c, to recover m from c using Kpri, as follows:

c = E(Kpub(m))

m = DKpri(c) = DKpri(EKpub(m))

In E-Voting a public key cryptosystem is normally used to provide secure authentication to the voters, or to establish secure connections between the voters and the electoral servers

- *RSA public key cryptosystem*

The most known and used algorithm for public key encryption is the RSA, proposed by [11]. The security of the RSA algorithm is based on the problems of factorization and calculation of modular logarithm for large numbers. In E-Voting the use of the RSA, or some derived algorithms is common on blind signature based voting systems. It is also used in the construction some of mix-nets .The details of the algorithm are shown in Table 1.

Table 1: RSA Algorithm

Secret Values	p, q	Secret distinct large primes, also calculate $\varphi = (p-1)(q-1)$
Public value	n	n = p . q

Public key	e	$1 < e < n$, such that $\gcd(e, \varphi) = 1$
Private key	p, q, d	$d < n$ such that $1 = e.d \bmod \varphi$
Encryption	$c = m^e \bmod n$	
Decryption	$m = c^d \bmod n$	

B. Blind Signature

The concept of blind signature was introduced by David Chaum [1]. 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 [12].

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 satisfying 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

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

(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 in

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

$s = v^d \bmod n$

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

Implementation of blind signature Protocol in EVS

A blind signature protocol is similar to a digital signature except that it allows a person to get another person to sign a message without revealing the content of the message. In EVS, a ballot is blinded in order to achieve its confidentiality requirement .For simplicity, a protocol with two authorities; mainly a validator and a tailler are used to demonstrate how a blind signature is employed in EVS. A voter is required to get the signature of the validator when he votes. To ensure the secrecy of his/her ballot, a voter cast a ballot, B, blinds a vote using a random number and send it to the validator .

Let (n,e) be validators public key and (n,d) be his/her private key. A voter generates a random number r such that gcd (r, n) =1 and sends the following to the validator $B' = (r^e B) \bmod n$.

The random number r conceals the ballot from the validator. The validator then signs the blinded ballot after verifying the voter, the signed value is $S' = (B')^d = (r^e B)^d \bmod n$.

After receiving the validated ballot, the voter unblinds the ballot, to get a true signature of a validator S by computing $S = S' r^{-1} \bmod n$.

The voter then sends his/her ballot together with validator signature to the tailler. The tailler verifies that if the ballot was correctly validated, then the ballot is valid.

C. Secret Sharing

Secret sharing, as the name suggests, is called to the process of sharing a secret S among N parties so that only t or more parties can later recreate the secret. Each party Pi keeps his/her share si secret, so that just $m \geq t$ parties can recreate the secret S. Such a scheme it's called (t, N) threshold secret sharing scheme. The interest of this scheme is to prevent the ability of less than t parties to reveal the shared secret.

- *Threshold cryptosystem*

In a threshold cryptosystem the secret sharing technique is used to share a private key Kpri among N parties, in such a way that at least t parties must cooperate to decrypt EKpub(m), where m is an arbitrary message. These systems are called (t,N) threshold cryptosystems. Threshold cryptosystems usually include two algorithms [13]-[15]

- Key Generation protocol: All the N parties are involved in the generation of the share public key Kpri. At the end each one receives its share of the private key Kpri.
- Verifiable Decryption protocol: Allows t parties to cooperatively decrypt an encrypted message EKpub(m) in a way that everyone can verify that the decryption was performed correctly. This process should not give anyone the ability to decrypt alone any other messages encrypted with the same public key. In some E-Voting protocols there is an election public key, used to encrypt the ballots. The use of a threshold cryptosystem for the election's private key brings obvious improvements to

Copyright © 2014 IJEIR, All right reserved

the system security, because votes cannot be revealed without the cooperation of t election authorities.

D. Kerberos Authentication Protocol

Kerberos version 5 that specified in RFC 1510, which supported the different realm architecture as Fig. 1 shows.

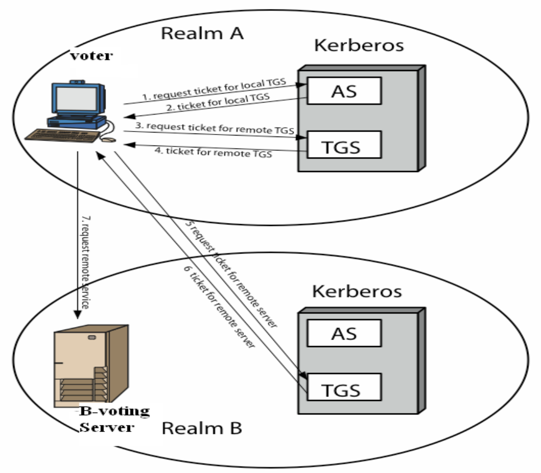

Fig.1. Kerberos Architecture supported different realm

It consists of several sub-protocols (or exchanges). There are two basic methods by which a client can ask a Kerberos server for credentials. In the first approach, the client sends a clear text request for a ticket for the desired server to the AS. The reply is sent encrypted in the client's secret key. Usually this request is for a ticket-granting ticket (TGT), which can later be used with the ticket-granting server (TGS). In the second method, the client sends a request to the TGS. The client uses the TGT to authenticate itself to the TGS in the same manner as if it were contacting any other application server that requires Kerberos authentication. The reply is encrypted in the session key from the TGT. Though the protocol specification describes the AS and the TGS as separate servers, in practice they are implemented as different protocol entry points within a single Kerberos server.

Once obtained, credentials may be used to verify the identity of the principals in a transaction, to ensure the integrity of messages exchanged between them, or to preserve privacy of the messages. The application is free to choose whatever protection may be necessary.

To verify the identities of the principals in a transaction, the client transmits the ticket to the application server. Because the ticket is sent "in the clear" (parts of it are encrypted, but this encryption doesn't thwart replay) and might be intercepted and reused by an attacker, additional information is sent to prove that the message originated with the principal to whom the ticket was issued. This information (called the authenticator) is encrypted in the session key and includes a timestamp.

The timestamp proves that the message was recently generated and is not a replay. Encrypting the authenticator in the session key proves that it was generated by a party possessing the session key. Since no one except the requesting principal and the server know the session key (it is never sent over the network in the clear), this guarantees the identity of the client.

The integrity of the messages exchanged between principals can also be guaranteed by using the session key (passed in the ticket and contained in the credentials). This approach provides detection of both replay attacks and message stream modification attacks. It is accomplished by generating and transmitting a collision-proof checksum (elsewhere called a hash or digest function) of the client's message, keyed with the session key. Privacy and integrity of the messages exchanged between principals can be secured by encrypting the data to be passed by using the session key contained in the ticket or the sub-session key found in the authenticator.

The authentication exchanges mentioned above require read-only access to the Kerberos database. Sometimes, however, the entries in the database must be modified, such as when adding new principals or changing a principal's key. This is done using a protocol between a client and a third Kerberos server, the Kerberos Administration Server (KADM). There is also a protocol for maintaining multiple copies of the Kerberos database.

III. The proposed Authentication Scheme in E-Voting Preparation Stage

A. Overview

In order to achieve voter privacy at E-Voting preparation stage, the researchers first applied the modified PVID scheme. In which, voter prepares a list of blinded identities and then obtains blind signature for each of them separately by interacting with the approval authority in one session, PVID Authority. 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. The value of PVID is only known by the voter**.**

Then, the researchers deploy the modified Kerberos authentication protocol 5 under public key cryptography. This will provide a non-repudiation service so neither the voter nor any other entities can deny such a communication. In addition to its own main entities, many others will be added such as the responder, derived from the Distributed online status Certificate protocol (D-OSCP), which is responsible for verifying the validity for the eligible voter certificate (Cert_v). For that; any attempt from the voters to supply a fake or old certificate will be easily detected. Thus; limited the DoS attack and the counter buffer will never be filled with garbage votes.

The communicating entities at the preparation stage will share a secret key based on the Nonce Based Authentication scheme rather than on the basis of combination of voter RegID and password R_v, which can detected by the attacker keeping track of the whole operation and comprise the voter associated password.

The Ferguson E-cash protocol had been modified to operate under E-Voting at the preparation stage, it combined at some point with Kerberos authentication protocol step to verify the whole voter identity and its own generated certificate (Cert_v).

B. Proposed Security Technique.

The proposed authentication protocol in E-Voting preparation stage suggested using other scheme or protocols' behind Kerberos Authentication protocol this will enhance the voters' privacy, authentication confidentiality and non repudiation service. The main are:

- *Responder from D-OSCP-KIS*

The responder entity is added at this stage, it actually derived from the OSCP-KIS and it will be operated in a distributed online election environment with hash function for a timeliness checking purpose. It will interact with the AS, one of the Kerberos Authentication protocol main entities, to verify the eligible voter issued certificate by contact the PVID authority that issue such a certificate. Implicitly, this will detect any attempt for double E-Voting. It will operate at three main stages:

1. Key generation:

To generate and distribute every responder's private key for digital signature, PVID authority chooses a master secret and calculates its corresponding public key. Then, if the number of responders is n, PVID authority generates n private keys for responders by applying KIS key generating algorithm and securely distributes the keys to each responder. In the key generation, the PVID authority will distribute private keys for every responder as shown in Fig. 2.

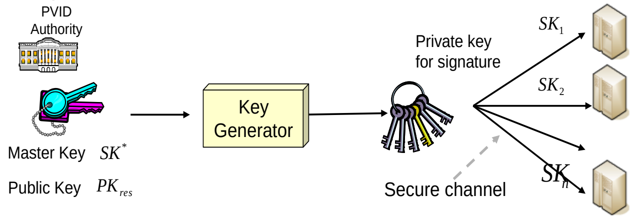

Fig.2. KIS-OSCP key Generation

Theoretically, in (A):

For public key generation: let p and q prime numbers such that p=2q+1 and g ,h be the elements of order q in Z_p. The PVID authority will generate a master key SK^* as in eq. 2 by choosing $x^*_i, y^*_i \in S_o$ randomly. SK^* is used for private key generation. Responders' public key PK_{res} as in eq. 3 is calculated by eq. 1.

$(x^*_0, y^*_0, \ldots\ldots, x^*_{n-1}, y^*_{n-1}) \leftarrow Z_q$.

$$v^*_i = g^{x^*_i} h^{y^*_i} \bmod p \; for \; 0 \le i \le n-1 \quad (1)$$

$$SK^* = (x^*_0, y^*_0, \ldots \ldots, x^*_{n-1}, y^*_{n-1}) \quad (2)$$

$$PK_{res} = (g, h, v^*_0, \ldots \ldots, v^*_{n-1}) \quad (3)$$

In (B): A private key will be generated :a different private key is assigned to each responder with the initial value of $SK_0=(x_0,y_0)=(x^*_0, y^*_0)$,the responder Ro's private key Sk_i is generated according to the eq. 4-8:

$$x'_i = \sum_{k=1}^{n-1} x^*_k \left(i^k - (i-1)^k\right) \quad (4)$$

$$y'_i = \sum_{k=1}^{n-1} y^*_k \left(i^k - (i-1)^k\right) \quad (5)$$

$$x_i = x_{i-1} + x'_i \quad (6)$$

$$y_i = y_{i-1} + y'_i \quad (7)$$

$$SK_i = (x_i + y_i) \quad (8)$$

2. Hash chain

The PVID authority will deliver the private key Sk_i to R_i anonymously. After all private keys are derived, intermediate values including the master key SK^* as in eq.2 are deleted.

Then, PVID authority generates hash chains to be used for timeliness checking. If the total time periods are T, PVID authority generates T chained hash values for each responder and keeps the first elements securely. Each hash value is used for given time period. If the time period is one day, 365 hash values are generated per responder. AS checks the timeliness of a responder by checking (hash chain) at the given time period.

PVID authority issues the certificate for all responders. This certificate includes KIS public key and the first hash values in the hash chain of all responders.

$X_1=H(X_2) = H^2(X_3)=\ldots\ldots\ldots H^{t-1}(X_t)$

For total T time period and n responders :

$$X^1_T \rightarrow X^1_{T-1} \rightarrow \ldots\ldots\ldots X^1_t \rightarrow \ldots\ldots X^1_1$$

$$X^2_T \rightarrow X^2_{T-1} \rightarrow \ldots\ldots\ldots X^2_t \rightarrow \ldots\ldots X^2_1$$

……..

$$X^n_T \rightarrow X^n_{T-1} \rightarrow \ldots\ldots\ldots X^n_t \rightarrow \ldots\ldots X^n_1$$

PVID authority keeps them securely. PVID authority provides X^i_t at time period $t \in T$ to i-th responder, the validity checks at $t \in T$ for i-th responder ,the value to be checked ($X^i_1 = H^{t-1}(X^i_t)$) is true (in signing and verification phase).

3. Signing /Verification Algorithm:

I. Signing Algorithm :When R_i sends a response to AS , R_i generates a digital signature (i,w,a,b) by using $SK_i=(x_i,y_i)$ as follows in eq. 9-12 :

$$r_1 r_2 \leftarrow Z_q$$

$$w = g^{r1} h^{r2} \bmod p \quad (9)$$

$$r = H(i, M, w) \quad (10)$$

$$a = r_1 - \tau x_i \quad (11)$$

$$b = r_2 - \tau y_i \quad (12)$$

Where H(.) ,is a cryptographic hash function.

II. Verification Algorithm : The AS will verify the Ri's signature (i, w, a, b)by using eq. 3 to calculate PKres as follows in eq.13-15:

$$v_i = \prod_{k=0}^{n-1} (v^*_i)^{i^k} \bmod p \quad (13)$$

$$\tau = H(i, M, w) \quad (14)$$

$$w == g^a h^b v_i^{\tau} \bmod p \quad (15)$$

As shown in fig. 3.

Copyright © 2014 IJEIR, All right reserved

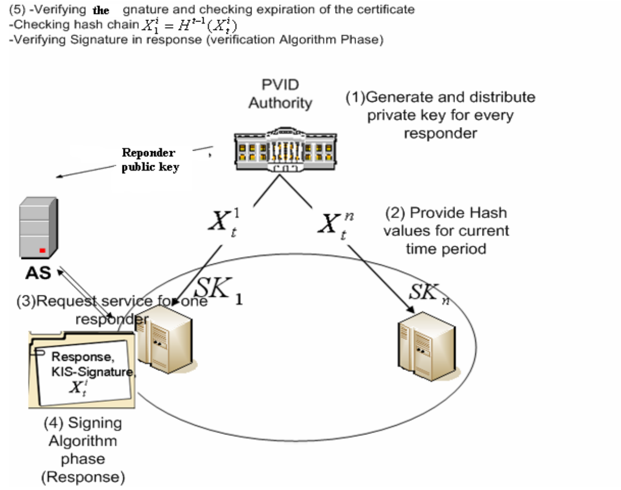
(5) -Verifying the gnature and checking expiration of the certificate
-Checking hash chain $X_1^i = H^{t-1}(X_t^i)$
-Verifying Signature in response (verification Algorithm Phase)
PVID Authority
(1)Generate and distribute private key for every responder
Reponder public key
AS
X_t^1
X_t^n
(2) Provide Hash values for current time period
(3)Request service for one responder
SK_1
SK_n
Response, KIS-Signature, X_t^i
(4) Signing Algorithm phase (Response)

Fig. 3 D-KIS-OSCP phases

- *Nonce Based Authentication Scheme*

If the communicating entities at the preparation stage depend on the combination of voter RegID and password Rv that both knows, it can be easily detected by the attacker. So, by based on the Nonce Based Authentication scheme to share a secret key will be better. All the exchanged message between the communicating entities will be encrypted with the shared secret key. For instance, in order to AS and voter to authenticate each other and agree on a session key to be used between them based upon Nonce Authentication scheme [16] the following is performed:

(1) V → AS: Request (ID_v ,Nc)

The user generates a random number Nc and sends Request(ID_v ,Nc). To generate such N_c.

(2)AS → V: Challenge (realm, Ns (XOR) h(R_v || Nc), h(R_v || Ns || Nc)) .

As the AS receives the Request message, the AS generates a random Ns and uses Ns, Nc, R_v to compute Ns (XOR) h(R_v || Nc). Then, the server uses R_v, Ns, Nc to compute h(R_v || Ns || Nc) and sends Challenge (realm, Ns(XOR) h(R_v || N_c), h(R_v || Ns || Nc)) to the voter.

(3) V→AS : Response (ID_v ,realm , h(Ns || R_v ||Nc))

When the voter receives the response message ,this voter uses N_c, R_v to compute h(R_v ||Nc) and uses h(R_v ||N_c), Ns (XOR) h(R_v ||Nc) to compute h(R_v ||N_c) (XOR) Ns (XOR) h(R_v ||Nc) to get N_s. Then, the voter uses R_v, Ns, Nc to compute h (R_v || Ns || Nc)).

If the computed h(R_v || Ns || Nc)) isn't the same as challenge (h(R_v || Ns || Nc)), the voter will be rejected by AS request $[A]_{PR\text{-}AS}$. Otherwise, the voter uses Ns, R_v and Nc to compute h(Ns || R_v ||Nc) and sends response (ID_v , realm, h(Ns || R_v ||Nc)) to the AS server.

(4) When the AS receives response message, the server uses Ns, R_v, Nc to compute h(Ns || R_v ||Nc). If the computed h (Ns || R_v ||Nc) isn't the same as response (h(Ns || R_v ||Nc)), the AS will reject such vote $[A]_{PR\text{-}AS}$. Otherwise the server accepts voter request $[O]_{PR\text{-}AS.}$

(5) After the AS and the remote voter authenticate each other, they use N_s as a session key between them $SK_{V\text{-}AS}$.

By this way both the AS and voter authenticated each other and agree on the session key to be used between them.

- *The converted Ferguson E cash protocol*

In order to verify the whole voter identity and issued certificate. The Ferguson E cash protocol is converted to operate under E-Voting environment in the voter and B-voting interaction at the preparation stage.

The voter will select two blind factors b_1 and b_2 and three random numbers $x_1,x_2 \in Z^*_{e'BV}$ and $s \in Z^*_{e'BV}$ and compute A ,A' ,B,w_1,w_2 as follows in eq. 16-20:

$$A= g_1^{u_v} g_2 \bmod n_{BV} \qquad (16)$$

$$A'=A^s \bmod n_{BV} \qquad (17)$$

$$B= g_1^{x_1} g_1^{x_2} \bmod n_{BV} \qquad (18)$$

$$w_1=B\, b_1^{e'BV} \bmod n_{BV} \qquad (19)$$

$$w_2 =(A'+B)\, b_2^{e_{BV}} \bmod n_{BV} \qquad (20)$$

Then, the voter send

{$Cert_v$, A, w_1, w_2, t , ((A||w_1||w_2||t $)^d{}_v$) mod n_v}

to B-voting Server.

(2) As the B-voting Server receive this message, again it will verify the validity of the voter certificate, but at this time it won't contact a responder, rather it will contact a PVID Authority database to check if this voter is legal to participate in voting and thus he /she has a certificate. Also the B-voting Server will contact an AS database, as AS verify the validity of the voter obtained certificate. After B-voting server verify the validity of the certificate, timestamp and value of A by using certificate, identity of the voter and public information. It also validates the signature (A||w_1||w_2||t$)^d{}_v$) mod n_v). After passing all the verification, B-voting server will compute the following equation:

$$w_3= A^{\frac{1}{e_{BV}}} \bmod n_{BV} \qquad (21)$$

$$w_4= w_1^{\frac{1}{e'_{BV}}} \bmod n_{BV} \qquad (22)$$

$$w_5= w_2^{\frac{1}{e_{BV}}} \bmod n_{BV} \qquad (23)$$

(3) Finally the message {$(w_3,w_4w_5)^{e_v} \bmod n_{BV}$ } is sent to V

(4) Decrypting the received value, V will get access to the signature of B-Voting server on A and blinded signature of B-voting server on B and A` +B. Voter compute the signature of B-voting server on A` ,B and A`+B as follows:

$$s_1= w_3^s \bmod n_{BV} = A'^{\frac{1}{e_{BV}}} \qquad (24)$$

$$s_2= \frac{w_4}{b_1} \bmod n_{BV} = B^{\frac{1}{e'_{BV}}} \qquad (25)$$

Copyright © 2014 IJEIR, All right reserved

$$s_3=\frac{w_5}{b_2} \bmod n_{BV} =(A'+B)^{\frac{1}{e_{BV}}} \qquad (26)$$

- *Architecture*

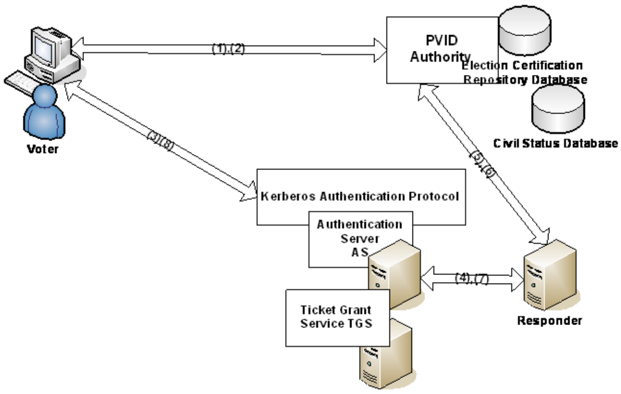

Fig.4. The proposed Authentication Scheme

In the preparation stage shown in fig. 4 (step 1– step 8), the modified PVID scheme will be operated. In step 1, the voter will send a set of blinded identities M_b, after the ID generation and blinding PVID stages applied, to the PVID authority, it will never sign a non eligible voter as it will check the voter RegID against country election law. As the voter is eligible the PVID authority will sign a set of the voter blinded identities (M_{bs}) via a PVID signing stage and send them to the voter accompanied with the issued voter certificate in step 2. The optional step for the voter to contact a password generator (PG) to generate a unique password for each voter, instead of using the traditional voter password, as an attacker may keep track of the voters' password and compromise it. From (step 3- step 8), the modified Kerberos authentication protocol will be operated with the converted Ferguson E cash protocol. In step 3, the voter will send a message encrypted with the AS public key consist of the voter certificate and a set of the signed blinded identities (PVID-list). As the AS receives this message at step 3, it will send to the responder to check its status in step 4, the OSCP-KIS will be applied to operate in a distributed environment. The responder will contact a PVID authority to check a certificate status in step 5; the PVID authority will send the voter certificate status to the AS in step 6, to the voter via AS in step 7, 8. A Kerberos authentication protocol consists of other steps that eventually end with the generated voter authenticate ticket that will be used in the E-Voting stage, administrators will never sign a voter without the Kerberos authenticated ticket.

IV. Evaluation and Analysis

In order to guarantee the authentication and privacy requirement were met in the proposed E voting preparation stage. the researcher evaluates the proposed scheme by first introducing a formal definition for these requirements, then mathematically prove each of them. Finally, for each requirement a checklist items is given below for a requirement brief summary proof.

1) Authentication: guarantee that the counter buffer will be never with garbage votes and thus only eligible and authorized voters were permitted to vote:

let $f: V \rightarrow B, f(v_i)=b_j andg: B \rightarrow A, g(b_j)=a_j$.

$If \forall v \in V[f_{ae}(g(f(v))) \in E]$ for a voting scheme *VS*, then *VS* satisfies Authentication.

Proof: By relying on the Kerberos authentication protocol infrastructure, the researcher guarantees that only authorized voters' casting votes by the generation of the issued voter ticket.

(1) Also the voter can't forge such a ticket without any detection

Proof (1): it can be proved by a contradiction. Let us suppose that a voter can forge the ticket. This means that the forged ticket is provided by changing in values of one of the signed amount s_1=sign $_{BV}$ (A'), s_2=sign $_{BV}$ (B), s_3=sign $_{BV}$ (A'+B). As the value of s_3 depend on the two previous value of s_1 and s_2, changing the value of s_3 is impossible. As well as the value of B is optimal, B forging isn't valuable. So forging a ticket without detection is impossible.

(2) It becomes impossible to forge an extra ticket to vote with

Proof(2): This requires a forgery of the PVID-list signature which is impossible as the PVID authority issues blind signature on voters blinded ID too after checking against country election registration laws (e.g. above 18 years old). Let prove by a contradiction method too, assuming there exit a function $f: P \rightarrow E, f(p_i)=e_i$ that known only by the voter.

Then the proposed scheme satisfies $\forall v \in V[\exists ! e \in E \mid f(p)=e]$. Furthermore, depending on the prove in (1), the voter alone is unable to forge A. However, if voter colludes together for such extra ticket forgery, the forgery one is identified by dealer in the voting stage as a case of double voting. Finally the issued PVID authority certificate will never be forging due to the additional entity (responder) that verifies the certificate.

V. Conclusion

In this paper, a new modified authentication protocol has been proposed in the preparation stage of the e-voting scheme. The proposed modification act as an improvement over the last two blind signature protocols, Evox-MA and REVS. The new modified protocol apply some cryptographic technique to enhances some security aspects. Some of these modified security aspects are Kerberos authentication protocol, PVID scheme, responder certificate validation, and the converted Ferguson e-cash protocol. These modified aspects will help in filtering the counter buffer from unauthorized votes by ensuring that only authorized voters are permitted to vote. Applying mechanisms such as the converted Ferguson E-Cash protocol and, the Modified PVID scheme and the voter certificate helps in detecting the

Copyright © 2014 IJEIR, All right reserved

double voting issued by the voters. Kerberos provides the tools of authentication and strong cryptography over the network to help you secure your information systems across your entire enterprise within a mutual authentication under the assumption that the underlying internet infrastructure is insecure. Kerberos has been invaluable to our e-voting proposed scheme.

ACKNOWLEDGMENT

The authors would like to acknowledge financial support for this work from the Deanship of Scientific Research (DSR), University of Tabuk, Tabuk, Saudi Arabia, under grant no. 0144/1435/S

REFERENCES

[1] Chaum D. (1981,1983): "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM journal*, vol 24, pp 84-90

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

[3] Cohen J. and Fischer M. (1985): "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, *IEEE Press*

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

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

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

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

[8] Joaquim R., Zúquete A., Ferreira P. (2002): "REVS –a roubst electronic voting system", *Instituto Superior Técnico (Technical Univ. of Lisbon) / INESC ID* , Lisboa, Portugal.

[9] "E-vote: Election markup language 5.0 approved as OSAIS standard" (2008), *New Report government technology magazine*, (On-Line) ,available:http://www.govtech.com/e-government/E-Vote-Election-Markup-Language-50-Approved.html. Last access on 1 June 2014

[10] Paul N, Evans D, Rubin A, Wallach D (2004): "Authentication for remotE-Voting", Charlottesville, VA 22903 USA

[11] Rivest R., Shamir A., and Adelman L.M (1977): "A method for obtaining digital signatures and public-key cryptosystems, *MIT LCS Technical Report MIT/LCS/TM.*

[12] Wen X., Niu X., Liping J and Tian Y. (2009): "A weak blind signature scheme based on quantum cryptography", Volume 282, Pages 666-669, *IEEE*

[13] Desmedt Y.(1993): "Threshold Cryptosystems", *Advances in Cryptology-ASIACRYPT92*, Old Coast, Queensland

[14] Baek J., Zhen Y.(2004): "Identity-Based Threshold Decryption", Cryptology ePrint Archive, Report 2003/164, available at http://eprint.iacr.org/2003/164, last access 14th July 2011

[15] Libert B. and Quisquarter J.(2003): "Efficient Revocation and Threshold Pairing Based Cryptosystems", *Symposium on Principles of Distributed Computing PODC*, pp. 163-171

[16] Tsai J(2008): " Efficient Nonce-based Authentication Scheme for Session Initiation Protocol", *International Journal of Network Security*, Vol.9, No.1, PP.12{16, July 2009

[17] Shilbayeh, N. Aqel, M., Al-Saidi, R., "A Modified Pseudo-Voter Identity (PVID) Scheme for e-Voting Preparation Stage", Innovations on Communication Theory Conference, INCT 2012, Istanbul, TURKEY, October 3-5, 2012.

AUTHOR'S PROFILE

Reem Al-Said

Miss. Reem A.Al-Saidi was born in Jordan on 8th Nov 1988. She worked at the University of Jordan. She received her master degree from Middle East University-Jordan with honor degree and her bachelor's degree from the University of Jordan. Her research interest include Cryptography and Network security, Image Processing, Watermarking Techniques and Steganography.
Email: r.saidi@ju.edu.jo

Nidal Shilbayeh

Received the BSc degree in computer science from Yarmouk University, Irbid, Jordan in 1988, the MS degree in computer science from Montclair State University, New Jersey, USA in 1992, and the PhD in computer science from Rajasthan University, Rajasthan, India in 1997. He is a Professor and the Vice Dean at university of Tabuk, Saudi Arabia; He was the Vice Dean of Graduate Studies and Scientific Research at Middle East University, Amman, Jordan. His research interests include Security (Biometrics, Identification, Privacy, Authentication, and Cryptography), Information Security (e-payment, e-voting, and e-government), Face Recognition, Digit Recognition, Watermarking, Embedding, Nose System, Neural Network, Image Processing, and Pattern Recognition.

Ebrahim Elnahri

Assistant Professor, Faculty of Engineering, Department of Computer and Control System, Manager of Quality Assurance Unit, Faculty of Engineering, University of Port Said. Egypt, He was working at University of Tabuk, Faculty of computers and Information Technology for 6 years.

Khaled Alhawiti

Bio Khaled M. Alhawiti received his BSc in 2005 from Amman, University and received his Ph.D. at the University of Wales, Bangor in 2014. He is currently acting as the dean of Faculty of Computers and Information Technology at Tabuk university.

Copyright © 2014 IJEIR, All right reserved