World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:3, 2016

Evaluation and Analysis of the Secure E-Voting Authentication Preparation Scheme

Nidal F. Shilbayeh, Reem A. Al-Saidi, Ahmed H. Alsswey

***Abstract*—**In this paper, we presented an evaluation and analysis of E-Voting Authentication Preparation Scheme (EV-APS). EV-APS applies some modified security aspects that enhance the security measures and adds a strong wall of protection, confidentiality, non-repudiation and authentication requirements. Some of these modified security aspects are Kerberos authentication protocol, PVID scheme, responder certificate validation, and the converted Ferguson e-cash protocol. Authentication and privacy requirements have been evaluated and proved. Authentication guaranteed only eligible and authorized voters were permitted to vote. Also, the privacy guaranteed that all votes will be kept secret. Evaluation and analysis of some of these security requirements have been given. These modified aspects will help in filtering the counter buffer from unauthorized votes by ensuring that only authorized voters are permitted to vote.

***Keywords*—**E-Voting preparation stage, blind signature protocol, nonce based authentication scheme, Kerberos authentication protocol, pseudo voter identity scheme PVID.

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 and accuracy.

Many scientists and researchers [1]-[9], [17] explored 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

The protocols under blind signature protocol 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]. A new secure modified EV-APS scheme has been has been added to the blind signature protocols [18], [19]. The new modified scheme applies some cryptographic techniques 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.

Authentication is an important part at the preparation stage 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. Neither Evox-MA nor REVS, last two E-Voting blind signature protocols, prevented Dos attack at the preparation stage so the attacker can fill the counter buffer with garbage votes and corrupted the overall E-Voting process.

This paper is organized as follows: Section II provides general background required to understand the proposed authentication scheme. Section III presents the proposed the architecture and operations of the Authentication E-Voting Preparation Scheme. Sections IV and V are discussion and conclusion.

II. Literature Survey and Related Works

A. 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 E-Voting schemes, protecting the voter privacy. The blind signature is used within E-cash system to guarantee owner anonymity [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).

Nidal F. Shilbayeh is with Information Systems Department, University of Tabuk, Tabuk 71491, Saudi Arabia, (phone: +966595978903; fax: +966 4 4223642; e-mail: nshilbayeh@ut.edu.sa).

Reem Al-Saidi is with Computer Science Department, University of Jordan, Amman, Jordan.

Ahmed H. Alsswey is with Information Systems Department, University of Tabuk, Tabuk 71491, Saudi Arabia

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 knows the random value r, she/he can remove it from the signed vote by taking r^{-1} to both sides in.

(6) r^{-1} t= v^d mod n; s = v^d mod 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.

1. 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 sends 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) mod 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} mod 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} mod 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.

B. 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 S_i secret, so that just m 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 K_{pri} among N parties, in such a way that at least t parties must cooperate to decrypt $EK_{pub}(m)$, where m is an arbitrary message. These systems are called (t, N) threshold cryptosystems. Threshold cryptosystems usually include two algorithms [11]-[15]

- Key Generation protocol: All the N parties are involved in the generation of the share public key K_{pri}. At the end each one receives its share of the private key K_{pri}.
- Verifiable Decryption protocol: Allows t parties to cooperatively decrypt an encrypted message $EK_{pub}(m)$ 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 the system security, because votes cannot be revealed without the cooperation of t election authorities.

C. *Maintaining the Integrity of the Specifications*

Kerberos version 5 is specified in RFC 1510, which supported the different realm architecture. 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 does not 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 it 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.

D. Related Works

A robust E voting system designed [8] for distributed and faulty environments, namely the Internet. The goal of REVS is to be an E-voting system that accomplishes the desired characteristics of traditional voting systems, such as accuracy, democracy, privacy, and verifiability.

The authors [18] propose a modified and efficient EV-APS. They proposed a 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.

The authors [19] proposed operations and modifications acts as an enhancement over the latest blind signature protocols and support the new secure EV-APS proposed by the authors in [18]. The new scheme and its operations apply many modified mechanisms and security aspects that support the authentication as one of the important requirements in the preparation stage of any E-Voting protocol. Some of these modified schemes are the modified PVID scheme, Kerberos authentication protocol and the Ferguson E-Cash protocol in addition to many others modified techniques and protocols. These modified schemes and protocols will enhance the security measures and adds a strong wall of protection, confidentiality, non-repudiation, and authentication requirements.

The authors [20] propose a secure E voting protocol. Their suggested scheme does not require a special voting channel and communication can occur entirely over the current internet. This method integrates internet convenience and cryptology. Thus, the proposed scheme satisfies the more important requirements of any E voting scheme: completeness, correctness, privacy, security, and uniqueness.

III. The Proposed Architecture

Fig. 1 shows the conceptual point of view for the modified authentication scheme in the preparation stage of the e-voting protocol.

The details of these steps done in this conceptual view can be explained as the following:

- Steps 1 and 2 will use the modified PVID Scheme as in [17]

Step 1: The voter will send a set of blinded identities Mb to the PVID authority.

Step 2: As the voter is eligible after checking his or her real identity the PVID authority will sign a set of the voter blinded identities (Mbs) via a PVID signing stage and send them to the voter accompanied with the issued voter certificate.

- Step 3-8 will use the modified Kerberos authentication protocol and the Ferguson E-Cash protocol

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) to the Authentication Server (AS).

Step 4: The Authentication Server (AS) will send to the Responder to check the voter certificate status.

Step 5: The Responder will contact the PVID Authority to verify the validity for the eligible voter certificate (Certv).

Step 6: The PVID Authority will send the voter certificate status to the Authentication Server (AS).

Step 7: The Authentication Server (AS) will receive the voter certificate status from the PVID authority.

Step 8: The Authentication Server (AS) will send the voter certificate status to the voter.

A Kerberos authentication protocol consists of other steps that eventually end with the generated voter authenticate ticket that will be used in the voting stage, administrators will never sign a voter without the Kerberos authenticated ticket.

IV. The Proposed Operations

The following detailed operations of the proposed scheme are shown in Fig. 2.

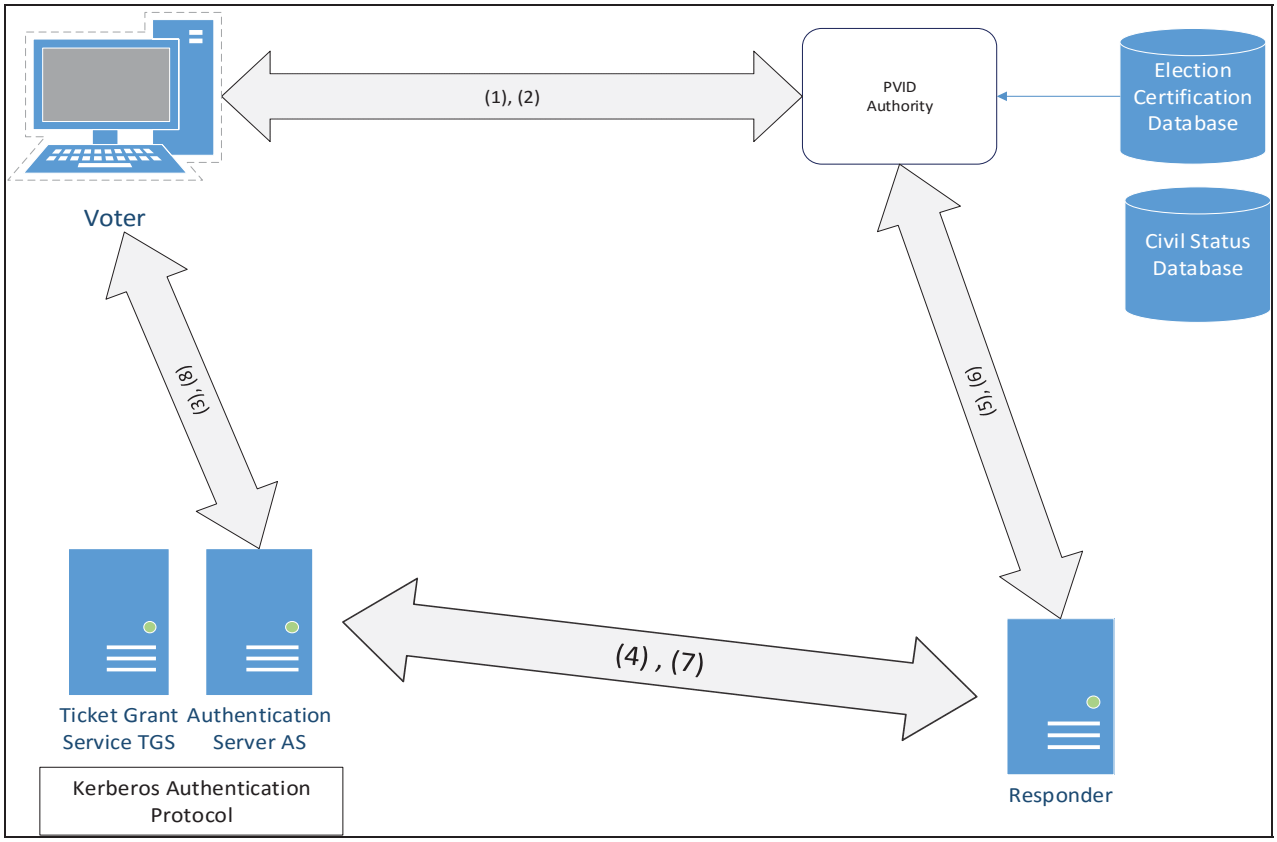

Fig. 1 The conceptual point of view for the modified authentication scheme

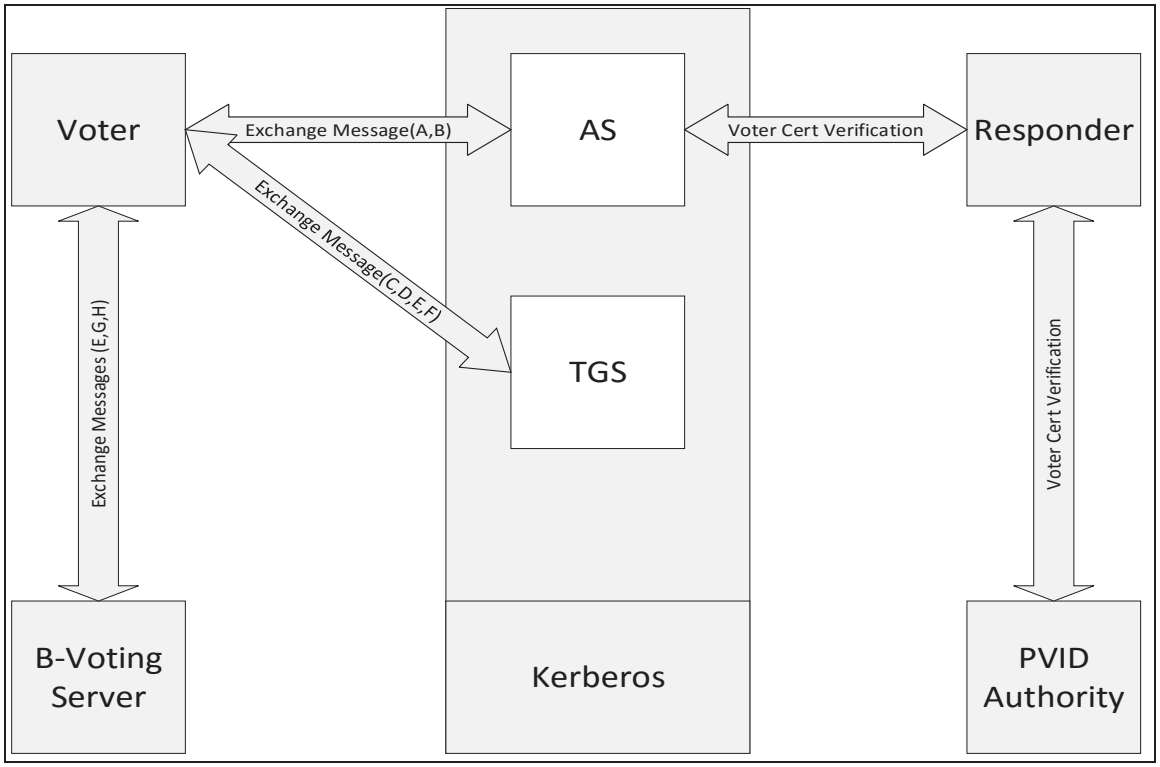

Fig. 2 Block Diagram for the main operation in the proposed authentication scheme

A. Step 1

First, the voter will send a message which consists of his/her own generated certificate ($Cert_v$) and identities signed with the PVID authority to the AS encrypted with the AS public key (PU_{AS}) as shown in Fig. 3. As the AS receives this message, it will decrypt it using its own private key and obtain the associated encrypted information (PVID-list), $Cert_v$. Then, AS will verify the signed identities. Later, the AS will verify the $Cert_v$ by contact a responder. The AS will send a request containing $Cert_v$, encrypted with responder public key (PUres), to the responder for $Cert_v$ verification purpose. As the responder receives the message, it will be decrypted using the responder private key (PRres) as shown in Fig. 4. Then, the responder will contact the PVID authority that issues such a certificate for verification purpose. Noted that the responder will operate in a distribute E-Voting election environment under OSCP-KIS protocol as illustrated in the proposed scheme.

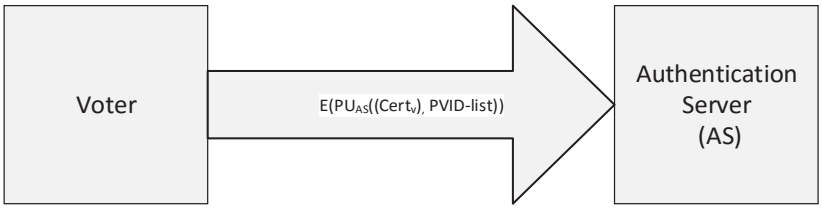

Fig. 3 Voter-AS interaction

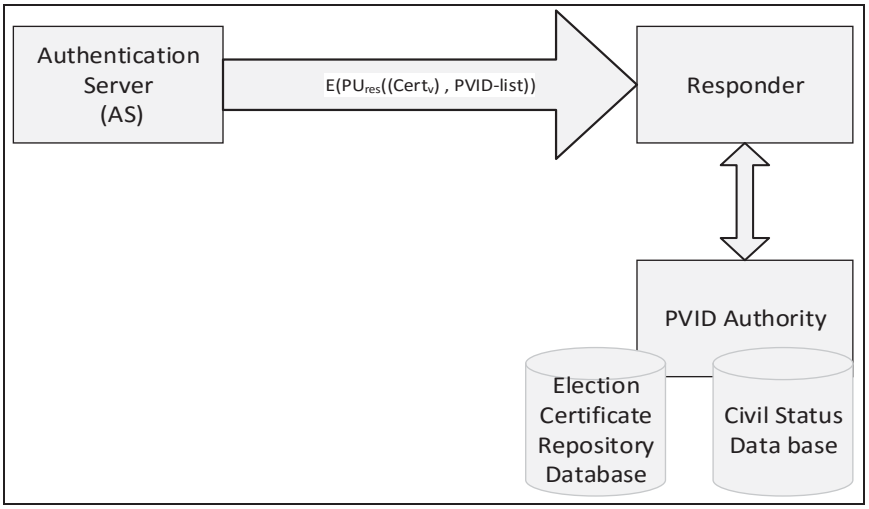

Fig. 4 AS-Responder interaction

B. Step 2

As the AS verifies the $Cert_v$ validity, it will send the voter certificate again to the voter, after keep a copy of it in its own database. This is another way for detecting any attempt for double voting by a voter, with the update time and signed PVID authority identities signed again with AS private key, all entities of this message will be encrypted with AS private key $E(PR_{AS}(Cert_v, time +1, (PVID\text{-}list)_{PR\text{-}AS}))$ as shown in Fig. 5. After the voter receives the previous message, it will be decrypted using PU_{AS} and the voter check the time from his/her own certificate and the received update one so he/she can judge any attempt of forgery, also he/she will verify the signed of PVID-list.

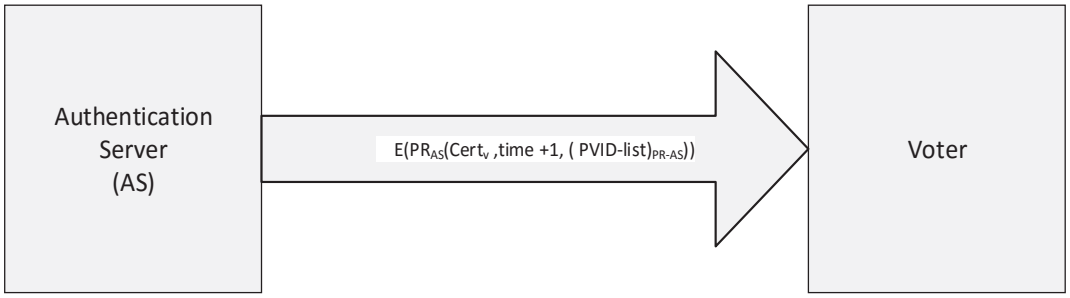

Fig. 5 AS-Voter Interaction-1

C. Step 3

As the proposed technique illustrate before the AS and voter authenticated each other and agree on the session key based on the Nonce based authentication scheme [16]. The AS will send to the voter the following:

- Message A: The TGS session key that will be used between voter and TGS encrypted with the agreement on voter and AS session key.
- Message B: Ticket granting Ticket (TGT) that consist of the PVID-list and voter network address (NW address$_v$), ticket validity period and voter TGS session key) encrypted with TGS secret key, which mean that the only one which can decrypt is the TGS as shown in Fig. 6. As the voter receives these two messages, he/she can deal only with message A, voter can decrypt it using the agree on voter and AS session key (SKV-AS) as the following:

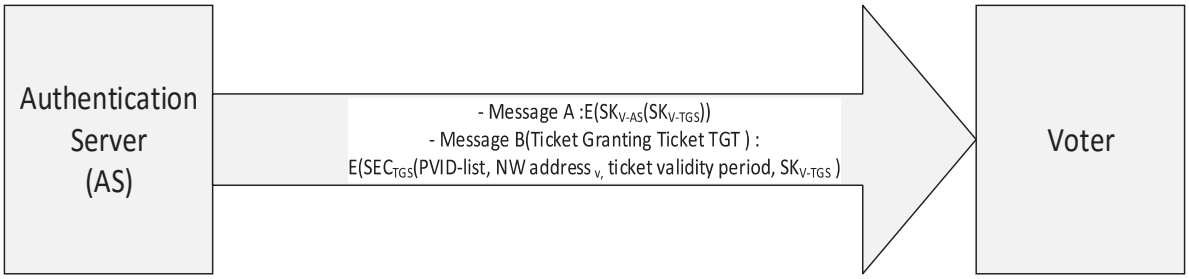

Fig. 6 AS-Voter Interaction-2

D. Step 4

The voter will send two messages to TGS as shown in Fig. 7.

- Message C: Same as the Message B, this received from AS by voter. With an election data (date in which election take place as state in PVID scheme this will be act as a voting service ID, known publicly to each eligible voter).
- Message D: An authenticator which consist of PVID-list and a timestamp (e.g. may indicated the current date and time) encrypt with the voter TGS session key retrieved from message A, in the step before.

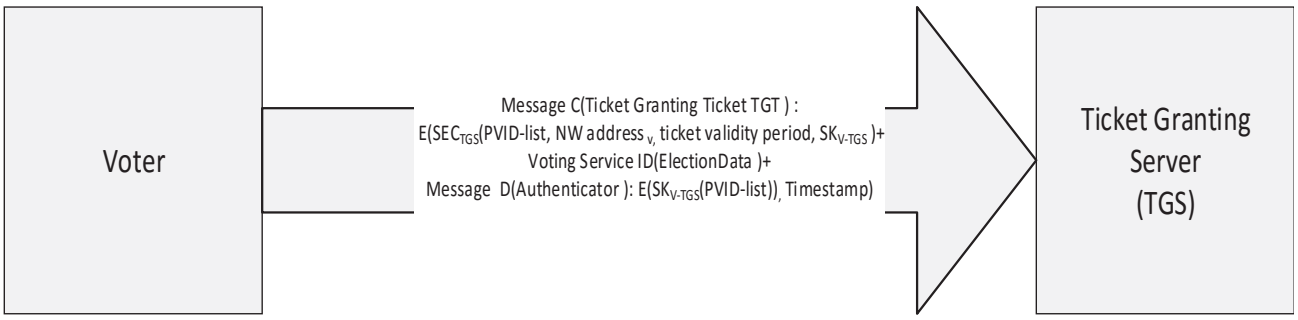

Fig. 7 Voter-TGS interaction

E. Step 5

As the TGS received these two message (Message C and D), the TGS can decrypt message C using TGS secret key and therefore obtain the associated TGT, D (SECTGS(TGT)). As it decrypted TGS can obtain the associated voter TGS session key. In this way, both voter and TGS securely obtain the voter TGS session key and can talk with each other using SKV-TGS.

Additionally, TGS will decrypt the authenticator (Message D) as it has the associated voter TGS session key, SKV-TGS) from message C, that will be used in decryption operation D (SKV-TGS(authenticator)) by TGS. After TGS decrypt these two message (Message C, D), it will make a match (if (PVID-list from Message C== PVID-list from Message D && Timestamp.Message D<= Ticket Validity period.Message C)). Also it will verify the PVID authority signed pseudo identities.

F. Step 6

The TGS will now send two messages to the voter as shown in Fig 8.

- Message E: Is the voter to B-Voting server ticket that consists of (PVID-list, NW address v, ticket validity period, SKV-B-VotingServer) Encrypted with B-Voting server secret key, the only entity that can decrypt is B-Voting Server itself
- Message F: Is a voter B-Voting server session key encrypted with the voter TGS session key from A (SKV-TGS). Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".

As the voter receives these two message (Message E, F), he /she can't decrypt message E as it encrypted with B-voting server secret key, so the only one that can decrypt it is the B-voting server itself. The voter can only decrypt message F as he/she already had the associated voter TGS session key from A (SKV-TGS), so the decryption is performed D(SKV-TGS(Message F)) and the voter can obtain now the voter B-voting server session key(SKV-B-VotingServer).

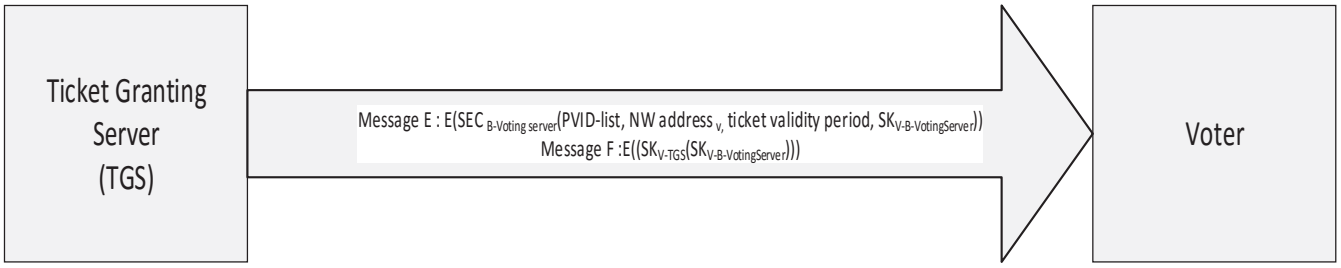

Fig. 8 TGS-Voter interaction

G.Step 7

The voter now will contact the B-Voting server and send two messages (Message E, G) as shown in Fig. 9. The converted Ferguson E cash protocol will operate here beside the Kerberos authentication protocol at this step.

- Message E: Same one in the previous step
- Message G: Is an authenticator that consist of the PVID-list and a timestamp (e.g. may indicated the current date and time) encrypted with voting B-voting server session key from message F (SKV-B-VotingServer).

As the B-Voting Server receive these two messages (Message E, G), it can decrypt message E, using the B-voting server secret key and thus retrieve the associated information from message E: D (SEC B-Voting server (Message E)). As the B-Voting server get SKV-B-VotingServer from message E decryption, the voter can decrypt message G as the following D(SKV-B-VotingServer(Message G)). As B-voting server decrypt these two messages (Message E, G), it can make a match between them as the following ==> if (PVID-list from Message E== PVID-list from Message G && Timestamp.Message G <=Ticket Validity period.Message E)). If all above steps in both Ferguson and Kerberos were successfully passed, the B-Voting server will confirm the voter's true identity.

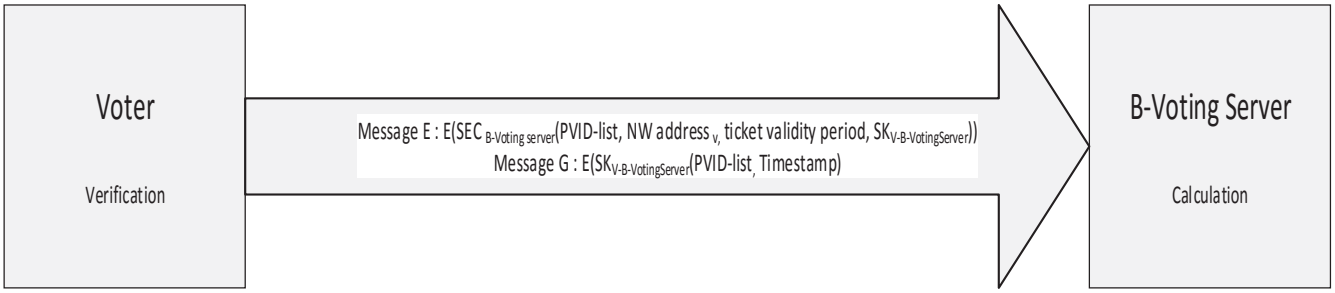

Fig. 9 Voter-B-Voting Server interaction

H.Step 8

The B-Voting server now will send a message H to voter as shown in Fig. 10.

- Message H:

It contains the authenticated ticket and the timestamp+1 (timestamp that found in message G in the previous step increment by one) encrypted with the voter and B-voting server session key (SKV-B-VotingServer). Additionally, it will send a [E(SKV-B-VotingServer (PKvoting) +h(PKvoting)), so the voter can verify the received PKvoting at step 1 from PVID with the PKvoting received here. Also, verify the received PKvoting by computing the same hash function for it.

As the voter receive message H, he/she can decrypt using SKV-B-VotingServer and check if the timestamp (voter send in message G) is updated by one. Furthermore, as the voter has the PKvoting from the beginning, it can compare it with the received PKvoting from B-voting server if they are the same or not. On other side, the voter will decrypt D(SKV-B-VotingServer (PKvoting) +h(PKvoting)), and thus get the (PKvoting) +h(PKvoting). The voter is now able to calculate the same hash function for the decrypted PKvoting and make such a match.

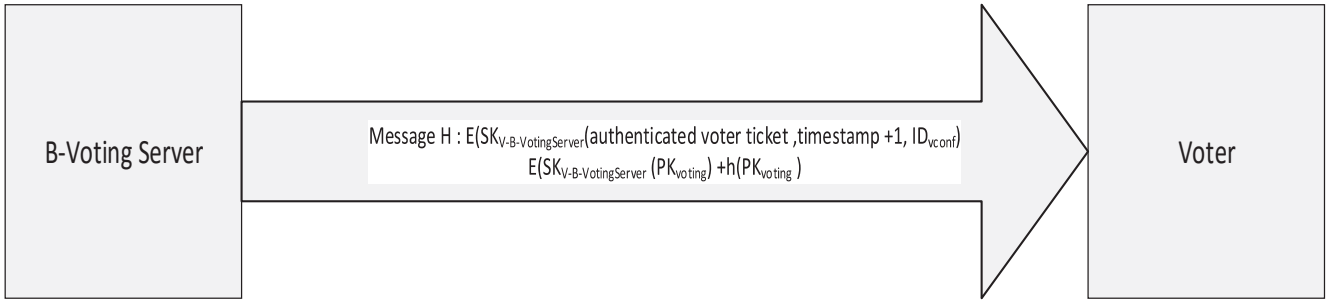

Fig. 10 B-Voting Server-Voter Interaction

V. Evaluation and Analysis

In order to guarantee the authentication and privacy requirement were met in the proposed E-voting preparation stage. The researchers evaluate the proposed scheme by first introducing a formal definition for these requirements [21], and 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:

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

For a voting scheme *VS*, then *VS* satisfies Authentication.

Proof: By relying on the Kerberos authentication protocol infrastructure, the researcher guarantees that only authorized voter' 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 values 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 $f : P \to E$, $f(p_i) = e_i$ 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 proof given 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 as shown in Table I that summarizes the case related to Eligibility proof.

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

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

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

Proof: This requirement is met by applying a PVID scheme that relies on the unlinkability between voter's pseudo identity and real identity. In order to prove any relation between them, the random number used to create blinded message should be known. Otherwise, adversary should break RSA cryptosystem since PVID scheme uses blind signature based on RSA public key cryptosystem, which is infeasible. The random number is generated by voter and nobody knows it.

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. The blindness property under the blind signature will guarantee that all votes will be kept in secret. 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 (the administrator) know the actual vote. Table II summarizes the case related to privacy proof.

TABLE I
AUTHENTICATION REQUIREMENT DETAILS

Main Requirement	Requirement Details	Satisfied	Not satisfied	Not applicable	How it is satisfied	Assumption
Eligibility	Eligible voters can vote	yes	-	-	Kerberos authentication protocol with an authenticated ticket +*PVID-list* signature	PVID is a trusted authority
	Ineligible voters cannot vote	yes	-	-	As PVID checking against voter, a blinded voter identities will never be signed for ineligible voter + Kerberos authentication ticket guarantee only authorized voters were vote, can't be forged (see proof (1),(2))	

TABLE II
PRIVACY REQUIREMENT DETAILS

Main Requirement	Requirement Details	Satisfied	Not satisfied	Not applicable	How it is satisfied	Assumption
Privacy	Voter-vote unlinkability	yes	-	-	Applying PVID scheme + Blind signature protocol	

VI. CONCLUSION

E-voting system is an important tool which allows voters to cast their votes over the Internet without the geographical restrictions with considers important criteria in evaluating electronic voting schemes such as the mobility, democracy, authentication, and privacy. The most important E-voting stage is the preparation one at which eligible voter registers themselves to cast their votes remotely at the specified election period time. Preparation stage needs to be arranged securely to guarantee that all over the E-voting process running smoothly. Authentication is the most important requirement at this stage utilizes the advantages of Kerberos to be operated behind other modified schemes and protocols. Authentication and privacy have been evaluated and proved. Other security requirements will be evaluated in the future work.

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. 0011/1437/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, (Online), 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.

[18] Reem Al-Saidi, Nidal Shilbayeh, Ebrahim Elnahri, And Khaled alhawiti, "E-Voting Authentication Preparation Scheme (EV-APS) Based on Evox-MA and REVS E-Voting Blind Signature Protocols." International Journal of Engineering Innovations and Research 3.5 (2014):590.

[19] Nidal F. Shilbayeh, Reem Al-Saidi, Sameh T. Khuffash, Ebrahim Elnahri, " Efficient and Secure Operations of the New Secure E-Voting Authentication Preparation Scheme (EV-APS)", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 5, Issue 1, January - February 2016, pp. 035-041, ISSN 2278-6856.

[20] Kalaichelvi. V, Chandrasekaran R. (2011): "Secured single transaction E voting protocol design and implementation", *European Journal of Scientific Research*, Vol.51 No.2 (2011), pp.276-284.

[21] Cetinkaya O, and Koc M.L (2009): "Practical Aspects of DynaVote E Voting Protocol", *Electronic Journal of E Government*, Volume 7 Issue 4, (pp327 - 338), available online at www.ejeg.com.