A Secure End-to-End Verifiable Internet-Voting System Using Identity-Based Blind Signature

Mahender Kumar, Satish Chand, and C. P. Katti

***Abstract*—The end-to-end (E2E) verification enables a voter to check if his ballot is recorded as he intended and the public to check if the system has correctly counted all of the recorded ballots. The Internet voting systems based on the principle of E2E verifiability have many challenges; the most important is its security. Several E2E voting systems have been discussed in the last decade in terms of analyzing the e-voting system and formalizing its security requirements. This article presents an E2E verifiable internet voting system that provides mobility to a voter and allows him to cast his vote secretly in public computer with the benefit of early voting. The proposed system aims to support the election process universally by using the voter's unique identification and biometric features. We propose a new identity-based blind signature scheme that ensures the voter's anonymity. We adopt the Boneh–Lynn–Shacham short signature scheme that ensures the vote privacy with the least ballot size. The system provides a digital witness to a voter that enables him to check whether his vote is recorded as he meant and the public to check if all the recorded ballots are counted correctly. The privacy of the proposed system is achieved under the well-known elliptic curve discrete logarithm and gap Diffie–Hellman assumptions.**

***Index Terms*—Batch verifiability, early voting, elliptic curve cryptography, end-to-end verifiable, functional digital signature, Internet-voting system.**

I. Introduction

THE Internet-voting system (IVS) [1] empowers an electorate to formulate his opinion through the election process and permits him to cast a ballot of his choice over the internet. The IVS conducts a smooth and fair election with the high participation rate of voters, but the primary concern with the IVS system includes the integrity of a ballot, voter anonymity, ballot stuffing, and transparency of election process [2]. Recently, the end-to-end verifiable internet voting (E2E-VIV) system has introduced that guaranteed transparent by providing a receipt to each voter at the end of the voting process. In the E2E-VIV system, a receipt allows a voter to check if his vote was cast as intended, recorded as cast, and tabulated as recorded using an audit mechanism [3]. The receipt does not allow the voter to prove to others how he voted or to whom he voted for. The notion of E2E-VIV is a decade old but designing and building its architecture with high security remains a challenge.

The first electronic voting system (EVS) was achieved by Chaum [4] accepting the mix-net approach where the number of machines reorganized the encrypted votes to conceal the relationship between a voter and his vote. Other cryptographic tools used for implementing the EVS system include the blind signature scheme [5]–[7], homomorphic cryptosystem [8], proxy servers [9], and secret sharing mechanism [9], threshold encryption, and zero-knowledge proof [10]. Recently, cryptographers and academicians have shown good interest in the hash-chain and blockchain technology [2], and verifiable computing [11] for the e-voting construction.

Several practical voting systems based on the principle of E2E verifiability [12]–[16], [3], [17], [18] have cherry-picked the above discussed cryptographic tools. These systems are partially aware of some security requirements but still, they have numerous inconsistencies, e.g., voter secrecy, ballot stuffing, costly infrastructure and equipment, auditing, accessibility, and usability. Some systems do not support receipt freedom to the voter and the malfunctioning of a voting device. The security requirements and voter's infeasibility to claim his receipt as a proof of mean that he voted it, make E2E-VIV system a challenge to both voting system architect and cryptographer.

In this article, we present an end-to-end verifiable internet voting system (E2E-IVS) that provides the mobility to a voter and enables him to cast his vote secretly without revealing any information about the vote. Our system employs the fingerprint and iris scan technology-enabled device (smartphone as voting device) that collects the voter's biometric features, process and transmit these features to the election officials over the secure medium. Based on the voter's biometric information and unique identification, election official authenticates the voter and provide a blank ballot to him. We consider the fuzzy extractor method [19] to preserve the original biometric data from the public domain.

In order to ensure the voter's anonymity, we extend the works [6], [7] and improve a functional digital signature variant, i.e., the identity-based blind signature (IDBS) scheme. The IDBS schemes [6], [7] are based on the bilinear pairing by practicing the Boldyreva's blind signature scheme [20] and Cha-Cheon's identity-based signature scheme [21]. On comparing with [6], [7], the proposed IDBS scheme generates a blank ballot for the voter in the least cost, without leaking any information of the voter's credentials. Since an IDBS scheme has inherent key

Manuscript received March 1, 2019; revised August 14, 2019; accepted September 5, 2019. This research work was supported in part by the Council of Scientific and Industrial Research and in part by the Department of Science and Technology PURSE. *(Corresponding author: Mahender Kumar.)*

The authors are with the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India (e-mail: mahendjnu1989@gmail.com; schand@mail.jnu.ac.in; cpaktti@mail.jnu.ac.in).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/JSYST.2019.2940474

escrow problem in which the private key generator controls the private key of each entity, it is not suitable for a large network. Our system can address this problem by distributing the master key among multiple authorities [22]. During vote casting, the system adopts the BLS short signature scheme [23] to achieve the vote integrity with a small ballot size. The main contribution is the transparency of the system, in which the proposed system enables the delegation of verification ability from election-official to the public, i.e., the voter obtains a receipt after the election process. The receipt enables the voter to check his vote in the ballot list and the public to check all votes are counted correctly. Under the well-known elliptic curve discrete logarithm (ECDL) and gap Diffie–Hellman (GDH) assumptions and random oracle model (ROM), the proposed system is existentially unforgeable under the chosen message and identity attack. Also, the proposed system allows a voter for early voting. We show that the proposed system performs well as compared to other related EVS systems. Additionally, we show that the proposed system supports batch verifiability, i.e., it simultaneously verifies multiple votes and ballots.

The rest of this article is organized as follows. Section II presents the related work. The proposed system architecture, and security notions are given in Section III. The construction of the proposed system is discussed in Section IV. Section V provides security and performance analysis. Finally, Section VI concludes this article.

II. Literature Survey

Several practical E2E-VIV systems have been proposed. The first voting system based on the principle of E2E verifiability is *Rijnland internet election system* [16]. It enables a voter to access the public algorithm and parameters to confirm his participation in the election. However, E2E verification is stronger as compared to the traditional postal system but it is far weaker than individual verifiability. Chaum *et al.* [12] invented the *Pret a Voter* system that uses a list of shuffled ordering candidate, a vote marking space and traditional cryptographic method. It allows a voter to verify his vote based on the shuffled ordering candidate. Similar to *Pret a voter's* election experience, Popoveniuc and Hosp [13] proposed *PunchScan* that allows voting process on paper ballots and its security is achieved by the use of an optical scanner.

Scantegrity [3], [17] generates a random number and distributes it to the election officials using a secret sharing scheme. The system computes a three-letter code using a random number for each printed ballot and generates a table for the postelection verification. During the election process, the voter marks a choice with invisible ink on the paper to obtain the three-letter code. Voter can record the code and ballot ID to check if his vote was tabulated. In Remotegrity [24], the voter obtains a lottery style code and authentication card with serial numbers. Voter uses both the serial number and code against his candidate's selection and authentication code to cast his vote. Adida presents a web-based voting system, known as Helios, which achieves E2E verifiability and privacy [15]. This method has a unique design mechanism in the voting device to help the voter in verification during the vote casting. A voter would cast the encoded vote only if he is convinced that the voting device is not cheating him. Corteir and Smyth [25] pointed out that the repetition of voter's choice, in the Helios, will leak any information about his vote. In [18], e-voting system achieves E2E verifiability in the standard model and is secure under the well-known decisional Diffie–Hellman problem. Joaquim *et al.* [26] presented an E2E-VIV system that provides the advantage of mobility and vote privacy to the voters. In this system [26], each voter has a security token that encrypts the vote to transmit his encoded candidate choice to the officials.

Recently, Kumar *et al.* [6] proposed a new IDBS scheme using bilinear pairing that ensures the security of the e-voting system. The scheme is based on the Boldyreva's blind signature scheme [20] and Cha-Cheon's IDBS scheme [21]. Inspired from the architecture of Lopez-Garcia *et al.*'s e-voting system [5] and proposed IDBS scheme, Kumar *et al*. [6] designed a framework for e-voting scheme, but could not provide its implementation. In [7], Kumar *et al.* improved the IDBS scheme and implemented the e-voting system (EVS-ID-BS) based on the proposed IDBS scheme and Boneh's short signature scheme [23]. The proposed EVS-ID-BS scheme [7] provides batch verifiability for a large number of voters, least communicated parameter size, and needs the least interaction with election authority as compared to other e-voting systems. Recently, Yu *et al.* [27] proposed a secure and practical platform-independent verifiable voting system, in which the blockchain technology supports the verifiability architecture and cryptographic primitives, such as ring signature and proof-of-knowledge provide the security to the system. Yang *et al.* [28] gave the first decentralized ranked-choice e-voting system based on the smart contract on Ethereum blockchain. The system uses homomorphic encryption to protect voter privacy. Li *et al*. [29] proposed an efficient and transparent e-voting system that is independent of any central authority. The proposed e-voting system is a decentralized multirole system that generates multiple candidates for different positions using a multisecret sharing scheme.

Most of the above discussed E2E-VIV systems are paper ballot based systems. Recently, a team of experts has suggested substituting the election process online, as a voter is particularly interested in an online system [26]. It has been shown that an Internet voting system has a lack of security and transparency, where the E2E verifiability is one of the tools to promise the integrity and transparency of the system [30]. In this article, we extend the work [7] and present an E2E-VIV scheme based on the functional blind signature scheme.

III. System Model

A. Architecture Design

Here, we first summarize the notations and abbreviations used throughout the paper in Table I. We then discuss the architecture of the proposed E2E-VIV system that involves five entities: key generation center (KGC), voters, candidates, voting device, and election commission authority (ECA).

1) Key Generation Center: The KGC initializes the system, authenticates each entities (voters, candidates, and ECA) and

TABLE I
SYMBOLS USED IN THIS ARTICLE

Notations	***Meaning***
s_0, P_0	*Master and public key of KGC*
a_1, a_2	*Random numbers known to ECA*
$ID_V/\{d_V, R_V\}$	*Identity/Private key of Voter*
$ID_C/\{d_C, R_C\}$	*Identity/Private key of Candidate*
$ID_E/\{d_E, R_E\}$	*Identity/Private key of ECA*
R	*Pseudonym public key of voter*
a, b, c, d, e, f	*Pseudonym private key of voter*
absc(P)	*x coordinate of point P on elliptic curve*
$BB = \{s, R, h\}$	*Pseudonym Certificate or blank ballot*
C_name	*Candidate name to whom a voter support*
B	*Ballot*
$Rcpt/S_{Rcpt}$	*Acknowledgement/ signed acknowledgment by ECA*
n_c	*Random number for the uniqueness in each* $Rcpt$
List^{RV}	*List of biometric details of registered voters.*
List^{RC}	*List of biometric details of registered candidates.*
List^{VV}	*List of biometric details of registered voters to whom ballot has been issued by the ECA.*
List^{B}	*List of valid/invalid ballots of registered voters*
List^{VB}	*List of valid ballot list of registered voters*
List^{IB}	*List of invalid ballots list of registered voters*

issues the private keys to them. Additionally, the KGC maintains a record of the registered voters and registered candidates.

2) Voter: A voter is a person in any organization or a citizen of the country having unique ID and registered with KGC to obtain his private key.

3) ECA: The ECA, in the proposed system, is an election conducting server that performs the following three tasks. First, it authenticates the voter and distributes a secret digital blank ballot to him without knowing his ID. It also maintains the record of the valid voters for whom blank ballots are already issued. Second, it receives and verifies the vote during vote casting phase and issues a witness to him. Third, it filters the duplicate ballots, counts them, and announces the winner.

4) Voting Device: Voting device could be a smartphone or computer that contains fingerprint and iris sensor device, which is either implanted or attached with the voting device.

5) Candidate: A person nominated by any group or political party who can participate in the election process. He must be first registered with KGC and obtain a private key.

Definition 1. (E2E-verifiable internet-voting system): The proposed E2E-IVS system consists of the following six algorithms.

1) Initialization: The KGC initializes the system, computes its master key s_0 and public parameter param. The KGC keeps s_0 secret and publishes param.

2) Registration: Before the election process, each participating entities are required to register themselves with the KGC. The KGC authenticates the entities against their IDs and biometric information, and issues the private keys to them over an insecure channel. Further, KGC prepares two lists: $\mathrm{List}^{\mathrm{RV}}$ and $\mathrm{List}^{\mathrm{RC}}$ that contain the biometric details of all the registered voters and registered candidates, respectively, and passes them to the ECA after registration.

3) Authentication and Ballot Distribution: In the PKI-based cryptography, the user's private key is directly linked to his ID. In proposed system, we cannot use the voter's private key for signature purpose, as it may lose the voter's anonymity. In order to preserve the anonymity, the system enables the voter to generate the pseudonym public and pseudonym private keys. During the ballot distribution phase, a voter blinds and signs the pseudonym public key using his pseudonym private key and asks the ECA for a blind blank ballot against his pseudonym public key. To authenticate a voter, the ECA maintains two lists ($\mathrm{List}^{\mathrm{RV}}$ and $\mathrm{List}^{\mathrm{VV}}$) for the registered voters and voters for whom the ballots have previously distributed. The ECA first checks if the voter is registered (i.e., found in list $\mathrm{List}^{\mathrm{RV}}$), and the blank ballot is not issued to him previously (i.e., not found in list $\mathrm{List}^{\mathrm{VV}}$), only then ECA issues a fresh pseudonym certificate, known as blank ballot, against the pseudonym public key to the voter and adds an entry in the list $\mathrm{List}^{\mathrm{VV}}$; otherwise, it raises an error. Now, the voter extracts the blank digital ballot using his secret values.

4) Vote Casting: An IVS system has several inconsistencies, such as network congestion due to the voter traffic, election server failure, and people with no internet facility during election period. In order to overcome these inconsistencies, the proposed system allows early voting, in which a voter may cast a vote before the election date. During the election and/or pre-election, a voter chooses a candidate from the list $\mathrm{List}^{\mathrm{RC}}$, computes the electronic ballot B using BLS short signature scheme and passes it to the ECA. The ECA authenticates the voter and validates the ballot B, and issues an electronic receipt if both the conditions are satisfied; otherwise it rejects the vote, and adds B in List^{B} (initially empty). The ECA includes a unique random value in each receipt that makes it fresh and unique.

5) Vote Counting: When the voting time is over, the ECA stops to receive additional ballots. It makes sure that there are no invalid or duplicate electronic ballots. To identify the duplicate votes in the list List^{B}, the ECA manages two lists: List^{V} to store the details of valid voters, and $\mathrm{List}^{\mathrm{IV}}$to store the details of invalid voters. The ECA separates the invalid votes from the list List^{B} by identifying two ballots having the same signature. Finally, the ECA publishes the both lists.

6) Auditing: After the election process, the voter verifies if his vote corresponding to his receipt is in the ballot list $\mathrm{List}^{\mathrm{V}}$ or $\mathrm{List}^{\mathrm{IV}}$. Also, anyone can verify if all ballots are counted correctly.

B. Security Requirements

Inspired from [26], [30], and [31], we discuss the security requirements for the E2E-IVS system.

1) *Voter's anonymity:* During election process, it ensure that the vote remains anonymous against the voter's ID who is able to vote.
2) *Vote integrity:* The system must secure against an adversarial attack on an individual voter's device and on its architecture. It must also ensure the vote privacy.
3) *Voter's eligibility:* The system must permit only the eligible/authentic voters to vote.
4) *Uniqueness:* It must ensure that a voter gets only one ballot, and can cast only one vote.

5) *Resistant to bribery and coercion:* In any election process, the collusion between a voter and a candidate may give two main inconsistencies: first, candidate may give some bribe to the voter (bribery), or second, candidate may threaten or force a voter (coercion). During as well as after vote counting process, a voter can freely cast his vote under coercion and bribery situations and cannot prove whom he supported.
6) *Ballot stuffing:* An illegal practice of voter, i.e., casting multiple ballots during vote should not be possible, whereby one ballot is allowed.
7) *Individual verifiability:* Any voter can verify that his ballot is correctly involved in the ballot list.
8) *Universal verifiability:* Anyone can verify that all the ballots in the list have been correctly tallied.
9) *Batch verifiability:* In order to optimize the cost of verification, the system must verify the massive ballots simultaneously.

C. Security Threat

The proposed E2E-IVS system is considered to be secured if it ensures the voter's anonymity, ballot stuffing, and vote integrity. Ensuring the voter's anonymity is equivalent to blindness property of an IDBS scheme, whereas ensuring ballot stuffing and vote integrity is equivalent to the existential unforgeable attack of digital signature under the chosen message and ID attack.

Unforgeability: The unforgeability can be defined by the following game playing between a forger F that acts as a malicious user and challenger Ch that acts as the signer [32].

Setup: The Ch computes the master key s_0 and public parameter param. It keeps s_0 secret and responds param to F.

Oracles: Following oracles are executed by F:

1) H_1 *Oracles:* Ch asks query H_1 on ID and sends it to F;
2) H_2 *Oracles:* Ch asks query H_2 on M and sends it to F;
3) *Key extract oracles:* Ch asks key extract query on ID and s_0, and responds d_i to F;
4) *Bling signature oracles:* F queries for (M, d_i), Ch responds with signature σ_i to F, where d_i is the response of key extract query;

Forgery: At the end, the forger F responses $< M^*, \sigma^* >$ against ID^*. The forger F will win the game if

1) $< M^*, \sigma^* >$ is a valid message-signature pair against ID^*;
2) the blind signature oracle has not been queried on $< M^*, \sigma^* >$;
3) the key extract oracle has not been queried on ID^*.

Definition 2: Suppose H_1 and H_2 are two random oracle models and a forger F wants to forge a signature on message *M* that executes at most q_E queries of key extraction, q_1 queries of H_1, q_2 queries of H_2 and runs at most *t* times with advantage at least ε. Under these assumptions and chosen message and ID attacks, an IDBS scheme is said to be secure against $F(t, q_1, q_2, q_E, q_B, \varepsilon)$ that is unforgeable against the chosen message and identity attack if no forger $F(t, q_1, q_2, q_E, q_B, \varepsilon)$ exists.

Algorithm 1: System Initialization.

```
1. Given a security parameter k, choose three groups
   $G_1$, $G_2$, and $G_T$ of order q (k-bit), a generator P of
   $G_1$, a generator Q of $G_2$, and bilinear map
   $e : G_1 \times G_2 \to G_T$.
2. Choose five hash functions: $H_1 : \{0,1\}^* \times G_1 \to Z_q$,
   $H_2 : \{0,1\}^* \times G_1 \to Z_q$, $H_3 : \{0,1\}^* \to Z_q$,
   $H_4 : \{0,1\}^* \to G_2$, and $H_5 :$
   $Z_q \times G_1 \times Z_q \times G_1 \times G_2 \times \{0,1\}^* \times G_1 \to Z_q$.
3. Choose a random element $s_0 \in Z_q$ (its master key)
   and compute the public key $P_0 = s_0 P$.
4. Publish the public parameter $\mathrm{param} =$
   $< q, e, P, P_0, G_1, G_2, G_T, H_1, H_2, H_3, H_4, H_5 >$,
   and keep $s_0$ secret.
```

Blindness: The blindness property is defined by the following game playing between the adversary Adv that acts as a malicious signer and challenger Ch that acts as the honest user [32].

Setup: The Ch computes the master key s_0 and public parameter param. It keeps s_0 and responds param to F.

Phase1: Adv picks two different messages, say, M_0 and M_1 identity ID, and outputs $\{M_0, M_1, ID\}$ to Ch.

Challenge: Ch picks a random bit $b \in \{0, 1\}$ and asks Adv for signature on M_b and M_{1-b}. Finally, Ch generates the signature $\{\sigma_b, \sigma_{1-b}\}$ on message $\{M_b, M_{1-b}\}$ and gives $\{\sigma_b, \sigma_{1-b}\}$ to Adv.

Response: For given tuple $\{M_0, M_1, \sigma_b, \sigma_{1-b}\}$, Adv predicts a bit $b' \in \{0, 1\}$ and wins the game if $b = b$' with advantage

$$|Pr\,[b = b']| > 1/2 + \varepsilon. \tag{1}$$

Definition 3: Suppose an adversary Adv that acts as a malicious signer and an honest user. Let the user have two distinct messages $\{M_b, M_{1-b}\}$ engaged in the proposed IDBS scheme and produces the signatures $\{\sigma_b, \sigma_{1-b}\}$, where $b \in \{0, 1\}$. A scheme is said to be blind if it is secured against Adv with advantage atleast ε.

IV. Proposed E2E Internet-Voting System

Here, we construct our E2E-IVS system that consists of six algorithms. In Algorithm 1, the KGC initializes the system, computes its master key s_0 and public parameter param. It keeps s_0 secret and publishes param.

In Algorithm 2, KGC registers every entity (voter, ECA and candidate) against their IDs and biometric information h_{3X}. It issues a private key (d_X, R_X) to the entity *X*, where *X* = {voter, candidate, ECA} over a secure channel and adds an entry of entity's h_{3X} in the respective list: $\mathrm{List}^{\mathrm{RV}}$ or $\mathrm{List}^{\mathrm{RC}}$. The KGC outputs both lists to ECA and $\mathrm{List}^{\mathrm{RC}}$ to voter. The private key (d_X, R_X) can be verified using

$$d_X P = H_1\left(\mathrm{ID}_X || R_X\right) P_0 + R_X. \tag{2}$$

In Algorithm 3, the voter requests to the ECA for digital ballot, where ECA authenticates the voter by checking his biometric parameters and makes a list $\mathrm{List}^{\mathrm{VV}}$ for valid voters to whom

Algorithm 2: Registration and Key Generation.

```
1. KGC prepares two lists: $\mathrm{List}^{\mathrm{RV}}$ and $\mathrm{List}^{\mathrm{RC}}$.
2. ECA asks KGC for its private key on its identity $\mathrm{ID}_E$. The KGC pick a random element $r_E \in Z_q$, compute the private key $R_E = r_E P$, $Q_E = H_1(\mathrm{ID}_E||R_E)$, $d_E = r_E + s_0 Q_E$ and give $< d_E, R_E >$ to ECA over a secure channel.
3. Candidate scans his biometric data and computes $h_{3C} = H_3(\mathrm{binfo}_C||\mathrm{ID}_C)$, using his identity $\mathrm{ID}_C$.
4. Given $h_{3C}$ and $\mathrm{ID}_C$, candidate asks KGC for its private key. The KGC picks a random element $r_C \in Z_q$, compute $R_C = r_C P$, $Q_C = H_1(\mathrm{ID}_C||R_C)$, $d_C = r_C + s_0 Q_C$, and give the private key $< d_C, R_C >$ to candidate and add an entry $h_{3C}$ to $\mathrm{List}^{\mathrm{RC}}$.
5. Similarly, the voter obtains his private key $< d_V, R_V >$ from KGC, where KGC maintains list $\mathrm{List}^{\mathrm{RV}}$for registered voters.
6. KGC sends lists ($\mathrm{List}^{\mathrm{RV}}$ and $\mathrm{List}^{\mathrm{RC}}$) to the ECA and candidate, and $\mathrm{List}^{\mathrm{RC}}$ to voter.
```

Algorithm 3: Authentication and Ballot Distribution.

```
1. ECA chooses two secret integers $a_1, a_2 \in Z_q$, computes $\{A_1 = a_1 P, A_2 = a_1 P, R_E\}$ and gives them to voter.
2. Ballot request: Voter performs the following steps:
   – Select six elements $a, b, c, d, e, f \in Z_q$ such that $\gcd(c, d) = 1$ and $ec + fd = gcd(c, d)$. The selection of elements e and f is done using the extended Euclidean algorithm.
   – Compute a pseudonym public key R as follows:
     $R_1 = aA_1 + cP$ and $r_1 = \mathrm{absc}(R_1) \mathrm{mod}\ q$
     $R_2 = bA_2 + dP$ and $r_2 = \mathrm{absc}(R_2)\ \mathrm{mod}\ q$
     $r = r_1 r_2 \mathrm{mod}\ q$
     $R = R_1 Z_q + R_2 + r(R_E + Q_E P_0)$
     $h_{3V} = H_3(\mathrm{binfo}_V, \mathrm{ID}_V)$
     $h_V = a h_{3V} + c$
     $b_{R1} = e a^{-1} c(H_2(R, h_V) - r) \mathrm{mod}\ q$
     $b_{R2} = f b^{-1} d(H_2(R, h_V) - r) \mathrm{mod}\ q$
     $U_1 = (a b_{R1} + b b_{R2})P$, $U_2 = rP$
   – Send $< b_{R1}, b_{R2}, R, h_V, h_{3V}, U_1, U_2 >$ to ECA for requesting a blank ballot.
3. Authentication: ECA authenticates the voter, if $U_1 == (H_2(R, h_V)P - U_2)$, $h_{3V} \in \mathrm{List}^{\mathrm{RV}}$ and $h_{3V} \notin \mathrm{List}^{\mathrm{VV}}$.
4. Ballot issuing: ECA computes $s'_1 = (d_E b_{R1} -\ a_1) \mathrm{mod} q$ and $s'_2 = (d_E b_{R2} - a_2)$ mod$q$, issues a blind ballot $< s'_1, s'_2 >$ to voter and add $h_{3V}$ to $\mathrm{List}^{\mathrm{VV}}$.
5. Unblinding: The voter unblinds the blind ballot and computes the blank ballot $\mathrm{BB} =< s,\ R, h_V >$, where $s_1 = (s'_1 a - c\ ) \mathrm{mod}\ q$, $s_2 = (s'_2 b - d)\ \mathrm{mod} q$, and $s = (s_1 + s_2) \mathrm{mod} q$.
```

Algorithm 4: Vote Casting.

```
1. The voter chooses a candidate $C\_\mathrm{name} \in \mathrm{List}^{\mathrm{RC}}$, timestamp T and, sets $V_1 = aH_4(C\_\mathrm{name}||T)$ and $V_2 = a(R_E + Q_E P_0)$.
2. Sets ballot $B = \{\mathrm{BB},\ V_1, V_2,\ C\_\mathrm{name},\ T\}$ and sends it to ECA.
3. ECA validates the vote ballot B using the following:
     $d_E H_2\left(R, h_V\right) P ? = sP + R \quad (3)$
     $e\left(d_E P, V_1\right) ? = e\left(V_2, H_4\left(C\_\mathrm{name}||T\right)\right). \quad (4)$
4. If (4) and (5) are valid,
   – ECA picks a random element $n_c \in Z_q$, and sets receipt as $P_{n_c} = n_c P$, $\mathrm{Rcpt} = H_5(B||P_{n_c})$ and $s_{\mathrm{Rcpt}} = n_c + d_E \mathrm{Rcpt}$ and adds $< B, P_{n_c}, \mathrm{Rcpt},\ s_{\mathrm{Rcpt}} >$ in $\mathrm{List}^B$.
   – Sends receipt $< P_{n_c}, \mathrm{Rcpt},\ s_{\mathrm{Rcpt}} >$ to voter
5. Otherwise, raises an error
```

Algorithm 5: Vote Counting.

```
1. ECA prepares two list: $\mathrm{List}^{\mathrm{VB}}$ and $\mathrm{List}^{\mathrm{IB}}$.
2. Picks two tuples
   $B_i =< s_i,\ R_i, h_{Vi}, V_{1i}, V_{2i}, C\_\mathrm{name}_i >$ and
   $B_j =< s_j,\ R_j, h_{Vj}, V_{1j}, V_{2j}, C\_\mathrm{name}_j > \in \mathrm{List}^B$.
3. On given two ballots, ECA filters them into two lists $\mathrm{List}^{\mathrm{VB}}$ and $\mathrm{List}^{\mathrm{IB}}$ according to Table II.
4. Publishes $\mathrm{List}^{\mathrm{VB}}$ and $\mathrm{List}^{\mathrm{IB}}$
```

a ballot is issued. If the valid voter has not received any ballot previously, the ECA issues a fresh blind blank ballot to him without knowing any information about his identity. Finally, the voter extracts the blank ballot. Algorithm 4 defines the vote casting process, in which a voter picks a candidate that he supports, computes the signature on his choice of support using the short signature scheme [23] and attaches it with the blank ballot to produce the electronic ballot B. The voter sends the electronic ballot to ECA that verifies the ballot and vote and acknowledges the receipt to the voter. An ECA prepares a list List^B where it stores the ballot's information $< B, P_{n_c},\ S_{\mathrm{Rcpt}}, \mathrm{Rcpt} >$.

Algorithm 5 counts the valid ballot from the premaintained list List^B. Suppose two ballots in list List^B are $B_i =< R_i, s_i,\ h_{Vi}, C\_\mathrm{name}_i, V_{1i}, V_{2i} >$ and $B_j =< R_j, s_j, h_{Vj}, C\_\mathrm{name}_j, V_{1j}, V_{2j} >$. The ECA creates lists $\mathrm{List}^{\mathrm{VB}}$, which contains the valid ballot details, and $\mathrm{List}^{\mathrm{IB}}$, which contains the invalid ballot details. The ECA filters the duplicate ballot and separates it in $\mathrm{List}^{\mathrm{VB}}$ or $\mathrm{List}^{\mathrm{IB}}$. The ECA performs $B_i \oplus B_j$, where $\oplus$ denotes the exclusive or operation that outputs the set of bits for each ballot's parameters. The resultant bits match with tuple of Table II that decide which corresponding action has to be performed with the two ballots. At the end, the ECA publishes both lists: $\mathrm{List}^{\mathrm{VB}}$ and $\mathrm{List}^{\mathrm{IB}}$. Algorithm 6 enables a voter to check if his vote is present in the ballot list. If a voter finds that his receipt is present in $\mathrm{List}^{\mathrm{VB}}$ and, using (5) he checks the ballot that he intended. Also, anyone can find

TABLE II
FILTRATING DUPLICATE BALLOTS AND SEPARATES THEM INTO VALID AND INVALID BALLOT LISTS, WHERE WE DENOTE 0 AND 1 IF THE CORRESPONDING PARAMETER IN TWO BALLOTS HAS THE SAME VALUES AND DIFFERENT VALUES RESPECTIVELY, C: CANDIDATE VOTE AND × DENOTES DO NOT CARE

R	**s**	$\boldsymbol{h_V}$	**C**	$\boldsymbol{V_1}$	$\boldsymbol{V_2}$	**Actions**
0	0	0	0	×	×	add one ballot in $List^{VB}$ and reject other
0	0	0	1	×	×	add one ballot in $List^{VB}$ and other in $List^{IB}$
×	1	×	×	×	×	both ballot are add in $List^{IB}$
1	×	×	×	×	×	both ballot are add in $List^{IB}$
×	×	1	×	×	×	both ballot are add in $List^{IB}$
1	1	1	×	×	×	both ballot are add in $List^{VB}$

Algorithm 6: Auditing.

```
1. On given $< s_{\mathrm{Rcpt}},\ P_{n_c} >$, voter finds his ballot in lists
   $\mathrm{List}^{\mathrm{VB}}$ and $\mathrm{List}^{\mathrm{IB}}$.
2. Any entity, say X, including voter can check whether
   the ballot is correctly recorded as the voter intended
   without knowing the actual vote, using
       $Q_X s_{\mathrm{Rcpt}} P = Q_E H_5 (B, P_{n_c}) (Q_E^{-1} Q_X R_E$
           $+ s_X P - R_X) + Q_X P_{n_c}.$     (5)
```

that a receipt is recorded in $\mathrm{List}^{\mathrm{VB}}$ and can check if the vote is recorded as the voter intended using (5). Further, anyone can count the ballots from the published list $\mathrm{List}^{\mathrm{VB}}$ and can verify that all ballots are correctly counted.

V. SYSTEM ANALYSIS

A. Security Analysis

Theorem 1. (Consistency): The proposed E2E-VIV system is consistent for the election process.

Proof: The consistency of (3) is verified as follows. From (3), we have $s = (s_1 + s_2)$ that gives

$$\begin{aligned} sP + R &= (s_1 + s_2) P + R \\ &= (s'_1 a - c + s'_2 b - d) P + R \\ &= ((d_E b_{R1} - a_1) a - c + (d_E b_{R2} - \ a_2) b - d) P + R \\ &= (d_E ec (H_2 (R, h_V) - r) \\ &\quad - a_1 a - c + d_E fd (H_2 (R, h_V) - r) - a_2 b - d) P + R \\ &= d_E H_2 (R, h_V) (ec + fd) P \\ &\quad - (ec + fd) r d_E P - a_1 a P - a_2 b P - cP - dP + R. \end{aligned}$$

Using EEA, we have $ec + fd = 1$

$$\begin{aligned} &= d_E H_2 (R, h_V) P - (r d_E P + a_1 a P \\ &\quad + a_2 b P + cP + dP) + R \\ &= d_E H_2 (R, h_V) P - (r (R_E + Q_E P_0) + R_1 + R_2) + R \\ &= d_E H_2 (R, h_V) P - R + R = d_E H_2 (R, h_V) P. \end{aligned}$$

This proves the consistency of (3). The consistency of (4) is verified as follows. From LHS, we have

$$\begin{aligned} e (d_E P, V_1) &= e ((r_E + Q_E s_0) P, a H_4 (C\_\mathrm{name})) \\ &= e (a (R_E + Q_E P_0), H_4 (C\_\mathrm{name})) \\ &= e (V_2, H_4 (C\_\mathrm{name})). \end{aligned}$$

This proves the consistency of (4). □

Theorem 2. (EF-ID-CMA): Suppose H_1 and H_2 are the two random oracles model and forger F wants to forge a signature on message M. Suppose forger F executes at most q_E extract oracles, q_B Blind signature oracles, q_1 H_1 hash oracles, q_2 H_2 hash oracles, runs at most t times with advantage at most k^{-n}. Under the assumption of ROM and intractable to solve the ECDLP, our proposed IDBS scheme is existentially unforgeable secured against the adaptive chosen message and identity attacks (EF-ID-CMA). Under the given assumptions and chosen message and ID attacks. Forger $F(t, q_1, q_2\ q_E,\ q_B, k^{-n})$ have the following advantage to breaks the proposed internet-voting scheme:

$$|\Pr \left[F \left(t, q_1, q_2\ q_E,\ q_B, k^{-n} \right) \right] | \geq \varepsilon (1 - q_1/k)^{q_2 + q_E}.$$

Proof: Consider forger F wish to forge any signature in the proposed ID-BS scheme and let there exist an algorithm *B* which helps F. We design an algorithm B that helps F to solve the ECDLP.

Setup: B considers two cryptographic hash functions $H_1 : \{0,\ 1\}^* \times G_1 \rightarrow Z_q$ and $H_2 : \{0,\ 1\}^* \rightarrow Z_q$ and is accountable to simulate these oracles. B picks $a \in Z_q$, set $P_0 = aP$ and gives public parameter $\mathrm{param} = \{G_1, q, P, P_0, H_1, H_2\}$ to F.

Oracles: Forger F can perform the following oracles.

$\boldsymbol{H}_1$ *oracle: B* prepares an empty list H_1^{List} having tuple $(R_{Xi}, \mathrm{ID}_i,\ H_1(\mathrm{ID}_i, R_{Xi}), *)$. When F queries to H_1^{List} on (ID_i, R_{Xi}), *B* responds F in the following way.

1) B gives $H_1(\mathrm{ID}_i, R_{Xi})$ to F and adds the tuple $(R_{Xi}, \mathrm{ID}_i, H_1(\mathrm{ID}_i, R_{Xi}), *)$ to list H_1^{List}, if $\mathrm{ID}_i = \mathrm{ID}^*$.
2) Otherwise, B chooses randomly $m_i \in Z_q$ and gives $H_1(\mathrm{ID}_i, R_{Xi}) = -m_i$ to F and adds tuple $(R_{Xi}, \mathrm{ID}_i,\ H_1(\mathrm{ID}_i, R_{Xi}), m_i)$ to list H_1^{List}.
3) B gives $H_1(\mathrm{ID}_i, R_{Xi})$ to F, if (ID_i, R_{Xi}) found in the H_1^{List} in the tuple of $(R_i, \mathrm{ID}_i,\ H_1(\mathrm{ID}_i, R_{Xi}), m_i)$ or $(R_{Xi}, \mathrm{ID}_i,\ H_1(\mathrm{ID}_i, R_{Xi}), *)$.

Note, $H_1(\mathrm{ID}, R_X)$ gives no information to F until he queries the H_1 oracle on ID as H_1 is the random oracle.

H_2 oracle: On given parameters R and h, B runs H_2 oracle and gives the output to the F. Suppose $z_1,\ z_2$, and z_3 are outputs when H_2 oracle execute three times.

Key extract oracle: B simulate the extract oracles. It pick $n_i \in Z_q$ and set $R_{Xi} = m_i P_0 + n_i P$ in such that $d_{Xi} = m_i$ and $H_1(\mathrm{ID}, A_i) = a_i$, and add $(R_{Xi}, \mathrm{ID}_i, H_1(\mathrm{ID}_i, R_{Xi}), d_{Xi})$ in list. We set these parameters in such that they satisfies the following:

$$d_{Xi} P = R_{Xi} + H_1 (\mathrm{ID}_i, R_{Xi}) P_0. \quad (6)$$

Ballot issuing oracle: F queries the blind ballot issuing algorithm to get a blind signature on message M_i with identity ID_i.

Let F give the blinded ballot $(b'_{R1},\ b'_{R2})$ to B. Then, B responses the following oracles.

1) If $\mathrm{ID}_i \neq \mathrm{ID}^*$, using ID_i corresponding to H_1^{List}, B executes the extraction oracles and signs the ballot using corresponding private key.
2) If $\mathrm{ID}_i = \mathrm{ID}^*$, B executes H_2 oracles on ballot R_i and picks up the tuple $(R_{Xi}, \mathrm{ID}_i,\ H_1(\mathrm{ID}_i, R_{Xi}), d_{Xi})$ from H_1^{List} to sign ballot R_i and outputs the corresponding signature $< \sigma_i^* = (s_i^*, R^*,\ h^*), R_X^* >$ to Adv.

Forgery: F responds a signature $< \sigma_i^* = (s_i^*, R^*,\ h^*), R_X^* >$ pair against identity ID^*. In order to forge a signature, suppose the forger F creates three distinct signatures $(\sigma_1^*, \sigma_2^*, \sigma_3^*)$ on same message *M*, where $< \sigma_1^* = (s_1^*, R^*, h_V^*), R_X^* >, < \sigma_2^* = (s_2^*, R^*,\ h_V^*), R_X^* >$, and $< \sigma_3^* = (s_3^*, R^*,\ h_V^*), R_X^* >$.

We consider $s_0, r_X,$ and u as the discrete logarithm of $P_0, R_X,$ and R, respectively, i.e., $P_0 = s_0 P$, $R_X = r_X P$ and $R = uP$. From $sP = H_2(R, h_V)(R_X + H_1(\mathrm{ID}_X, R_X)P_0) - R$, we get $s_i^* = (z_i(r_X + s_0 H_1(\mathrm{ID}, R_X)) - u\ )\mathrm{mod} q$ for $1 \leq i \leq 3$. The parameters $s_0, r_X,$ and u from these equations are unknown to F. Thus, F solves three linear equations to obtain these values, which is equivalent to solving the ECDL problem.

Analysis: The following two events define the probability that B does not abort the game:

$$\Pr[\neg E_1\ \wedge \neg E_2]$$

E_1*:* the extract oracle fails if H_1 oracle outputs the inconsistent outputs with probability at most q_1/k. The simulation is completed in q_E time, which happens with probability at least $(1 - q_1/k)^{q_E}$.

E_2*:* the execution of H_2 oracle fails if H_2 oracle gives the inconsistent outputs with probability at most q_1/k. The simulation is completed in q_2 time, which happens with probability at least $(1 - q_1/k)^{q_2}$.

From these two events, we obtain the probability that Adv can break the scheme with

$$\Pr[\neg E_1\ \wedge\ \neg E_2] \geq \varepsilon(1 - q_1/k)^{q_2 + q_E}.$$

Theorem 3: The voter remains anonymous in ballot distribution phase of the proposed E2E-VIV system.

Proof: Suppose an adversary Adv that plays a role of ECA and challenger *Ch* that plays a role of an honest voter runs the blind signature algorithm. Suppose Adv obtains the parameters $< b_{R1}, b_{R2}, s'_1, s'_2 >$ by executing the ballot issuing oracle. Let the corresponding signature be $< R,\ h_V, s >$. There exists a tuple of values $< a, b, c, d, e, f >$ that links $< b_{R1}, b_{R2}, s'_1, s'_2 >$ to $< R, h_V,\ s >$.

From $b_{R1} = ea^{-1}c(H_2(R, h_V) - r)\mathrm{mod}\ q$ and $b_{R2} = fb^{-1} d(H_2(R, h_V) - r)\mathrm{mod} q$, we get $a = eb_{R1}^{-1}c(H_2(R, h) - r)$ and $b = fb_{R2}^{-1}d(H_2(R, h_V) - r)$, respectively. Similarly, from $s_1 = (s'_1 a - c\ )\ \mathrm{mod}\ q$ and $s_2 = (s'_2 b - d\ )\mathrm{mod}\ q$, we get $c = (s'_1 a - s_1)$ and $d = (s'_2 b - s_2)$, respectively. By substituting these values in $R = aA_1 + bA_2 + (c + d)P + r(R_E + Q_E P_0)$, it can be noted that Adv must know the value of r to compute R. The value r depends on $< a, b, c, d >$ whose production is equivalent to solving the ECDL problem. Thus, the proposed E2E-VIV system provides voter anonymity. □

B. Security Requirements

1) Voter Anonymity: During and after the election process, the voter's identity must be hidden from other entities in the system. In ballot request and vote casting, a voter uses the pseudonym public-private key pair $(R, < a, b, c, d, e, f >)$ instead of using his real private key (d_V, R_V). Thus, ECA verifies the blank ballot BB without knowing the information about the voter's real identity. In order to protect the voter's biometric information, we use preimage collision-resistant hash function H_3 that encrypts the biometric information. So, the use of pseudonym public and private key pair, preimage resistant hash function H_3 and Theorem 3 prove that our system preserves the voter's anonymity.

2) Resistant to Bribery/Coercion: After election, the ECA issues a receipt $< P_{n_c}, \mathrm{Rcpt}, s_{\mathrm{Rcpt}} >$ to each voter, from which voter can check that he has cast his vote to the intended one. The ECA also adds a random number n_c to each receipt to ensure the freshness of the ballot. Any curious voter can show his choice of vote to others only if he guess the correct value of n_c, which is equivalent to compute the ECDLP. Thus, our proposed scheme is resistant to the bribery and coercion attack.

3) Voter's Eligibility/Authentication: In registration process, KGC authenticates a voter against his ID_V and biometric details h_{3V} and stores h_{3V} in registered voter list $\mathrm{List}^{\mathrm{RV}}$ and an issue private key to him. During ballot distribution, the ECA maintains a list $\mathrm{List}^{\mathrm{VV}}$ for those valid voters who have been already issued a ballot. On a given blank ballot BB $< b_{R1}, b_{R2}, R, h_V, h_{3V}, U_1, U_2 >$, the ECA authenticates the voter only if $U_1 == (H_2(R, h_V)P - U_2)$, h_{3V} is present in $\mathrm{List}^{\mathrm{RV}}$, and not present in $\mathrm{List}^{\mathrm{VV}}$. The consistency of first condition is proved as follows. From LHS

$$\begin{aligned} U_1 &= (ab_{R1} + bb_{R2})\,P \\ &= (ec\,(H_2\,(R, h_V) - r) + fd\,(H_2\,(R, h_V) - r))P \\ &= (H_2\,(R, h_V) - r)\,(ec + fd) \\ P &= H_2\,(R, h_V)\,P - U_2 = \mathrm{RHS}. \end{aligned}$$

4) Vote Integrity: Theorem 2 proves the integrity of blank ballot generated in ballot issuing phase. During vote casting, the integrity of the vote is ensured by the short signature scheme whose security is equivalent to solving the GDH problem [23]. Thus, the proposed system achieves the vote integrity during and after election process.

5) Uniqueness and Fairness (Ballot Stuffing): In the proposed system, a voter uses a pair of pseudonym public/private keys to request a blank ballot from the ECA. In order to authenticate a voter against his voter's pseudonym public key R and biometric information h_{3V}, ECA checks if $U_1 = (H_2(R, h_{3V})P - U_2)$ holds and h_{3V} is present in lists $\mathrm{List}^{\mathrm{RV}}$, not present in $\mathrm{List}^{\mathrm{VV}}$. Now, the ECA issues a fresh blank ballot to the voter and adds h_{3V} in $\mathrm{List}^{\mathrm{VV}}$. Thus, a voter obtains only one blank ballot and can cast only one vote. Therefore, the proposed system is resistant to ballot stuffing.

6) Verifiability: After election, the ECA publishes the list of invalid ballots $\mathrm{List}^{\mathrm{IB}}$ and valid ballot list $\mathrm{List}^{\mathrm{VB}}$. From given $< P_{n_c}, \mathrm{Rcpt},\ S_{\mathrm{Rcpt}} >$, an eligible voter can check if

$< P_{n_c}, \mathrm{Rcpt}, S_{\mathrm{Rcpt}} >$ belongs to $\mathrm{List}^{\mathrm{IB}}$ or $\mathrm{List}^{\mathrm{VB}}$. The voter ensures that his vote is counted in the election process if the parameters $< P_{n_c}, \mathrm{Rcpt}, S_{\mathrm{Rcpt}} >$ satisfy (5). The consistency of (5) is verified as follows from RHS, we have:

$$\begin{aligned}
&Q_E H_5\left(B, P_{n_c}\right)\left(Q_E^{-1} Q_V R_E + d_V P - R_V\right) + Q_V P_{n_c}\\
&= H_5\left(B, P_{n_c}\right) Q_V R_E + Q_E H_5\left(B, P_{n_c}\right) d_V P\\
&\quad - Q_E H_5\left(B, P_{n_c}\right) R_V + Q_V P_{n_c}\\
&= Q_V\left(H_5\left(B, P_{n_c}\right) r_E P + Q_E H_5\left(B, P_{n_c}\right) s_0 P + P_{n_c}\right)\\
&= Q_V\left(H_5\left(B, P_{n_c}\right)\left(r_E + s_0 Q_E\right) P + P_{n_c}\right)\\
&= Q_V\left(H_5\left(B, P_{n_c}\right) d_E + n_c\right) P = Q_V S_{\mathrm{Rcpt}} P = \mathrm{LHS}.
\end{aligned}$$

This proves the consistency of (5). Since a voter is allowed to check the ballot's status, it can bring the coercion problem, in which the voter can show the value of the vote to anyone. In order to overcome this inconsistency, the proposed system publishes two lists: invalid ballot $\mathrm{List}^{\mathrm{IB}}$ and valid ballot $\mathrm{List}^{\mathrm{VB}}$, so that he can check in which list his ballot belongs to. In order to count the valid voter, anyone can verify that only valid votes were counted. Thus, the ECA proves that all receipts of the votes have been counted corresponding to the valid ballots that were previously verified and stored. Therefore, the proposed system achieves the individual verifiability and universal verifiability.

C. Performance Analysis

This section provides the comparison of our system with the related electronic voting systems and examines their performance. Suppose the Weil pairing is defined over the Type-F curve (BN curve) of PBC library [33] with 256 bit of one group and 512 bit of another group, and embedding degree is 12, whose security level is identical to 3072-bit RSA. For 128 bit of AES security of BN curve $E/F_P : y^2 = x^3 + b$, where $b \neq 0$, gives the group size $|Z_q| = 256$ bit, the field size of G_1, i.e., $|G_1| = 160$ bit, the field size of G_2, i.e., $|G_2| = 512$ bit, and the field size of G_T, i.e., $|G_T| = 3072$ bit. To compare our scheme with the non-ECC-based schemes [9], [34]–[36], we consider two large prime numbers of 3072 bit, i.e., $|p_1| = |p_2| = 3072$ bit, such that their product is hard to factorize, and consider a large prime p of 1024 bit and a prime factor q of 160 bits, i.e., $|p| = 1024$ bit, and $|q| = 160$ bit, such that $(p-1)|q$. Further, we assume that $|C\_\mathrm{name}| = |ID| = |T| = 80$ bit.

Implementation and benchmarks: Here, we compare the performance of our proposed internet-voting system with related schemes [5], [8], [9], [34]–[36], in terms of machine cycles obtained by simulating experiment on Intel(R) Core(TM) i7-2600K CPU @ 3.4 GHz, and 8 GB of RAM, using GCC 4.6. We consider the same methodology as discussed in [8] to estimate the machine cycles consumed by operations (for example, private and public operations based on RSA, operations on elliptic curve and pairing). Table III summarizes the notations and number of machine cycles consumed by the required cryptographic operations.

TABLE III
NUMBER OF MACHINE CYCLES REQUIRED BY CRYPTOGRAPHIC OPERATIONS

Notation	*Operations*	*# Cycles (in Kilo)*
#Pr	*RSA public key*	*766*
#Sk	*RSA private key*	*51,805*
#Exp	*DSA exponentiation*	*17,334*
#Sm	*ECC scalar point multiplication*	*300*
#Pm	*Pairing multiplication*	*450*
#MTP	*Map-To-Point hash function*	*310*
#Bp	*Bilinear pairing*	*2,100*

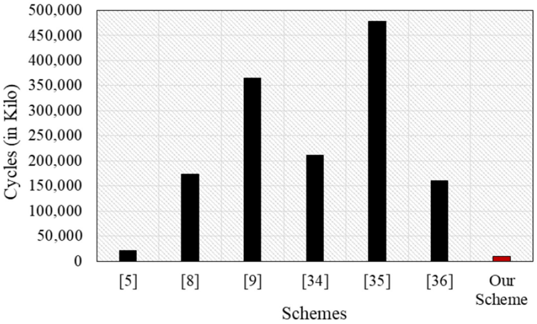

Fig. 1. Computation costs of our scheme and other schemes.

Table IV lists the performance of our system with the existing systems [5], [8], [9], [34]–[36]. The proposed system and system [5] use elliptic curve cryptography, and their security is based on ECDL and GDH problems. The systems [8], [9], [34]–[36] are based on the traditional RSA public key cryptosystem and their security is based on the discrete logarithm problem (DLP) and integer factorization problems (IFP). It can be seen from Table III that the ECC-based operations (scalar multiplication and addition) are efficient than the RSA based operations.

For computation cost, our proposed system needs $16 * \#Sm + 2 * \#Bp + 2 * \#MTP = 9{,}620$ K cycles, whereas the schemes [5], [8], [9], [34]–[36] need $20{,}940$, $173{,}340$, $364{,}935$, $211{,}053$, $477{,}746$, and $160{,}015$ K cycles, respectively. The computation cost of our system is much better, i.e., it takes 46%, 5%, 2%, 4%, 2% and 6% of the computation costs of the schemes [5], [8], [9], [34]–[36], respectively, shown in Fig. 1. For a blank ballot, our system needs $3{*}32 = 96$ bytes, whereas the schemes [5], [8], [9], [34]–[36] need 128, 1152, 1152, 768, 1152, and 188 bytes, respectively. For a vote ballot, our system needs $4{*}32{+}1{*}64 + (80)/8 + (80)/8 = 212$ bytes, whereas the schemes [5], [8], [9], [34]–[36] need 202, 2098, 2304, 1930, 1152, and 296 bytes, respectively. From Fig. 2, we observe that the ballot bandwidth in our scheme is much better, i.e., it takes 75%, 8%, 8%, 12%, 8% and 51% of ballot bandwidth size of the schemes [5], [8], [9], [34]–[36], respectively, and the vote-ballot bandwidth in our scheme is 105%, 10%, 9%, 10%, 18% and 68% of the vote-ballot bandwidth of schemes [5], [8], [9], [34]–[36], respectively.

D. Batch Verifiability

Suppose ECA obtains $BB_i = < s_i,\ R_i,\ h_{Vi} >$ from ith voter, $1 \leq i \leq n$. The ECA verifies n valid blank ballots

TABLE IV
EFFICIENCY COMPARISON OF OUR SCHEME WITH RELATED SCHEMES (WHERE IFP: INTEGER FACTORIZATION PROBLEM, DLP: DISCRETE LOGARITHM PROBLEM)

Scheme	**#Cycles**	**Blank Ballot Size (in Bytes)**	**Vote Ballot size (in Bytes)**	**Cryptographic primitives**	**Security Assumption**	**Security Parameter size (in bit)**
[5]	20,940,500	128B	202B	Short signature Blind signature	GDHP DDHP	|q| = 254, k = 12 |q|= 254, k = 12
[8]	173,340,000	1152B	2098B	Homomorphic Enc. Zero Knowledge proof	DDHP	$\|p_1\| = \|p_2\| = 2048$ $\|p_1\| = \|p_2\| = 5120$
[9]	364,935,431	1152B	2304B	Digital signature Blind signature	IFP IFP	$\|p_1\| = \|p_2\| = 3072$ $\|p_1\| = \|p_2\| = 3072$
[34]	211,053,653	768B	1930B	Digital signature Blind signature	IFP IFP	$\|p_1\| = \|p_2\| = 3072$ $\|p_1\| = \|p_2\| = 3072$
[35]	477,745,803	1152B	1152B	Digital signature Blind signature	IFP IFP	$\|p_1\| = \|p_2\| = 3072$ $\|p_1\| = \|p_2\| = 3072$
[36]	160,015,290	188B	296B	Digital signature Mix Net	IFP DLP	$\|p_1\| = \|p_2\| = 3072$ $\|p\| = 1024, \|q\| = 160$
Our	9,620,000	96B	212B	Short signature IDBS	GDHP ECDLP	$\|q\| = 254, k = 12$ $\|q\| = 254, k = 12$

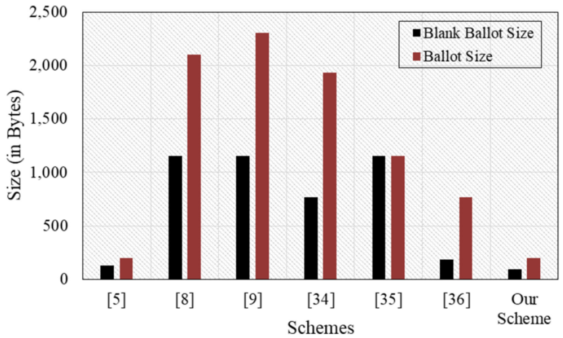

Fig. 2. Ballot and vote-ballot size of our scheme and other schemes.

using

$$\left(\sum_{i=0}^{n} H_2\left(R_i, h_{Vi}\right)\right) d_E P = \left(\sum_{i=0}^{n} s_i\right) P + \sum_{i=0}^{n} R_i. \quad (7)$$

Similarly, the ECA verifies multiple ballots using

$$e\left(\sum_{i=0}^{n} V_{1i}, d_E P\right) = e\left(\sum_{i=0}^{n} H_4\left(C\_\text{name}_i\right), \sum_{i=0}^{n} V_{2i}\right). \quad (8)$$

The consistencies of (7) and (8) are verified as similar to verification of (3) and (4), proved in Theorem 1.

Analysis: (7) and (8) verify the multiple blank ballots and ballots, respectively, if all the blank ballots and ballots are valid, i.e., they individually verify (3) and (4), respectively. If any ballot is invalid or modified, the (7) and (8) will not be satisfied. For the batch of valid and invalid ballots, the proposed system uses the divide and conquer mythology to optimize the computation cost. The proposed system divides n ballots into two $n/2$ ballots, and checks (7) and (8). If the ballots satisfy these equations, stop the process; otherwise, the process is repeated until the size of ballot becomes one. Thus, the proposed system has $O\log(n)$ time complexity for batch of n invalid and valid ballots, the complexities for n valid ballots and single ballot are $O(1)$ and $O(n)$, respectively, as listed in Table V.

TABLE V
BATCH VERIFICATION ANALYSIS OF BLANK BALLOTS AND BALLOTS

Verification	**Blank ballots**		**Ballots**	
	Computation cost	**Complexity**	**Computation cost**	**Complexity**
Single	*2n*#Sm*	*O(n)*	*2n*#Bp*	*O(n)*
n valid	*2*#Sm*	*O(1)*	*2*#Bp*	*O(1)*
n invalid &valid	*Log(n)*2*#Sm*	*Olog(n)*	*Log(n)*2*#Bp*	*Olog(n)*

VI. CONCLUSION

In this article, we have discussed an end-to-end verifiable Internet-voting system (E2E-VIV), in which each voter is authenticated using a unique identifier issued by the appropriate authority and his biometric information. First, we have discussed the architecture of proposed E2E-VIV system, and then provided its implementation. We have presented a functional digital signature for anonymously issuing a blank ballot to a voter and used the BLS short signature scheme for protecting the vote from any modification. The proposed system is secured against the existential forgery attack under chosen message and ID. It needs least machine cycles as compared to the existing schemes. Further, it requires the least bandwidth cost for the blank ballot and vote ballot. It also allows the batch verifiability to verify multiple ballots and vote-ballots simultaneously.

REFERENCES

[1] K. Vassil, M. Solvak, P. Vinkel, A. H. Trechsel, and R. M. Alvarez, “The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015,” *Government Inf. Quart.*, vol. 33, no. 3, pp. 453–459, 2016.

[2] R. C. Silva, “Blockchain technology for end-to-end verifiable elections on internet voting system,” in *Proc. 3rd Int. Joint Conf. Electron. Voting*, 2018, Paper no. 351.

[3] D. Chaum *et al.*, “Scantegrity: End-to-end voter-verifiable optical-scan voting,” *IEEE Secur. Privacy*, vol. 6, no. 3, pp. 40–46, May/Jun. 2008.

[4] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[5] L. López-García, L. J. D. Perez, and F. Rodríguez-Henríquez, “A pairing-based blind signature e-voting scheme,” *Comput. J.*, vol. 57, pp. 1460–1471, 2014.

[6] M. Kumar, C. P. Katti, and P. C. Saxena, "An identity-based blind signature approach for E-voting system," *Int. J. Modern Educ. Comput. Sci.*, vol. 9, no. 10, 2017, Art. no. 47.

[7] M. Kumar, C. P. Katti, and P. C. Saxena, "A secure anonymous E-voting system using identity-based blind signature scheme," in *Proc. Int. Conf. Inf. Syst. Secur.*, 2017, pp. 29–49.

[8] X. Yang, X. Yi, S. Nepal, A. Kelarev, and F. Han, "A secure verifiable ranked choice online voting system based on homomorphic encryption," *IEEE Access*, vol. 6, pp. 20506–20519, 2018.

[9] C.-L. Chen, Y.-Y. Chen, J.-K. Jan, and C.-C. Chen, "A secure anonymous e-voting system based on discrete logarithm problem," *Appl. Math. Inf. Sci.*, vol. 8, no. 5, pp. 2571–2578, 2014.

[10] R. Kusters, T. Truderung, and A. Vogt, "Clash attacks on the verifiability of e-voting systems," in *Proc. IEEE Symp. Secur. Privacy*, 2012, pp. 395–409.

[11] T. S. Robbie Simpson, "Third-party verifiable voting systems: Addressing motivation and incentives in e-voting," *J. Inf. Secur. Appl.*, vol. 38, pp. 132–138, 2018.

[12] D. Chaum, P. Y. A. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2005, pp. 118–139.

[13] S. Popoveniuc and B. Hosp, "An introduction to PunchScan," in *Towards Trustworthy Elections*, vol. 6000, New York, NY, USA: Springer, 2010, pp. 242–259.

[14] K. Gjøsteen, "The Norwegian internet voting protocol," in *Proc. Int. Conf. E-Voting Identity*, 2011, pp. 1–18.

[15] B. Adida, "Helios: Web-based open-audit voting," in *Proc. USENIX Secur. Symp.*, 2008, vol. 17, pp. 335–348.

[16] E. Hubbers, B. Jacobs, and W. Pieters, "RIES-internet voting in action," in *Proc. 29th Annu. Int. Comput. Softw. Appl. Conf.*, 2005, vol. 1, pp. 417–424.

[17] D. Chaum *et al.*, "Scantegrity II: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes," in *Proc. USENIX/ACCURATE Electron. Voting Workshop*, 2008, vol. 8, pp. 1–13.

[18] N. Chondros *et al.*, "D-DEMOS: A distributed, end-to-end verifiable, internet voting system," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst.*, 2016, pp. 711–720.

[19] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[20] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," in *Proc. Int. Workshop Public Key Cryptography*, 2003, pp. 31–46.

[21] J. C. Choon and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *Proc. Int. Workshop Public Key Cryptography*, 2003, pp. 18–30.

[22] M. Kumar and S. Chand, "ESKI-IBE: Efficient and secure key issuing identity-based encryption with cloud privacy centers," *Multimed. Tool Appl.*, vol. 78, pp. 19753–19786, 2019.

[23] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2001, pp. 514–532.

[24] F. Zagórski, R. T. Carback, D. Chaum, J. Clark, A. Essex, and P. L. Vora, "Remotegrity: Design and use of an end-to-end verifiable remote voting system," in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.*, 2013, pp. 441–457.

[25] V. Cortier and B. Smyth, "Attacking and fixing Helios: An analysis of ballot secrecy," *J. Comput. Secur.*, vol. 21, no. 1, pp. 89–148, 2013.

[26] R. Joaquim, P. Ferreira, and C. Ribeiro, "EVIV: An end-to-end verifiable Internet voting system," *Comput. Secur.*, vol. 32, pp. 170–191, 2013.

[27] B. Yu *et al.*, "Platform-independent secure blockchain-based voting system," in *Proc. Int. Conf. Inf. Secur.*, 2018, pp. 369–386.

[28] X. Yang, X. Yi, S. Nepal, and F. Han, "Decentralized voting: A self-tallying voting system using a smart contract on the Ethereum blockchain," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2018, pp. 18–35.

[29] J. Li, X. Wang, Z. Huang, L. Wang, and Y. Xiang, "Multi-level multi-secret sharing scheme for decentralized e-voting in cloud computing," *J. Parallel Distrib. Comput.*, vol. 130, pp. 91–97, 2019.

[30] S. Dzieduszycka-Suinat *et al.*, "The future of voting: End-to-end verifiable internet voting-specification and feasibility study," US Vote Found., Arlington, VA, USA, 2015.

[31] D. A. Gritzalis, "Principles and requirements for a secure e-voting system," *Comput. Secur.*, vol. 21, no. 6, pp. 539–556, 2002.

[32] M. Kumar, C. P. Katti, and P. C. Saxena, "An untraceable identity-based blind signature scheme without pairing for E-cash payment system," in *Proc. Int. Conf. Ubiquitous Commun. Netw. Comput.*, 2017, pp. 67–78.

[33] B. Lynn, "The Stanford pairing based crypto library," *Privacy preservation scheme for multicast communications in smart buildings of the smart grid* vol. 324, 2013.

[34] Y.-F. Chung and Z.-Y. Wu, "Approach to designing bribery-free and coercion-free electronic voting scheme," *J. Syst. Softw.*, vol. 82, no. 12, pp. 2081–2090, 2009.

[35] C.-T. Li, M.-S. Hwang, and Y.-C. Lai, "A verifiable electronic voting scheme over the internet," in *Proc. 6th Int. Conf. Inf. Technol., New Gener.*, 2009, pp. 449–454.

[36] Z.-Y. Wu, J.-C. Wu, S.-C. Lin, and C. Wang, "An electronic voting mechanism for fighting bribery and coercion," *J. Netw. Comput. Appl.*, vol. 40, pp. 139–150, 2014.