Journal of King Saud University – Computer and Information Sciences 34 (2022) 3647–3660

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Secure Internet Voting Protocol (SIVP): A secure option for electoral processes

Cristina Satizábal [a,*], Rafael Páez [b], Jordi Forné [c]

[a] INNOVATEC, Servicio Nacional de Aprendizaje (SENA), Popayán, Colombia
[b] SiDRe, Pontificia Universidad Javeriana, Bogotá, Colombia
[c] SISCOM (Smart Services for Information Systems and Communication Networks), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

ARTICLE INFO

Article history:
Received 11 August 2020
Revised 20 November 2020
Accepted 23 December 2020
Available online 7 January 2021

Keywords:
Applied cryptography
Blind signature
Electronic voting systems
Information security
Network protocols
Public key cryptography

ABSTRACT

Colombia government wants to implement electronic voting. However, the existing electronic voting protocols only include some of the required security features and Colombia needs a protocol with all these features to ensure fraud-free elections. In this paper, we present the design of SIVP (Secure Internet Voting Protocol), a new voting protocol for electoral processes, based on blind signatures and public key cryptography. This protocol has six phases and provides: eligibility, democracy, privacy, verifiability, accuracy, fairness, robustness, receipt-freeness and coercion-resistant. Also, we compare the number of cryptographic operations per phase of SIVP with other four protocols and conclude that the computational load of our protocol is not excessively high despite including more security features.

1. Introduction

"*Electronic voting is the application of electronic technology to cast and count votes in an election*" (Collins English Dictionary, s.a). Colombia government wants to implement electronic voting to elect government representatives. However, in past electoral processes, Colombia has had: voters' coercion; alteration of the results by corrupt juries and authorities; fraud in the digitization of the results (Línea Democracia y Gobernabilidad, 2018). Due to this, Colombians mistrust the electoral processes, so it is necessary to implement a voting protocol with the enough security features to ensure transparent and fraud-free elections. In a previous work (Satizábal and Páez, 2018), we discovered that the analized voting protocols only include some of the security features that these kind of systems must have. This motivated us to design a new protocol for electoral processes in Colombia with more security features but without increasing the number of cryptographic operations too much. Therefore, our contribution in this paper is the design of a new Internet voting protocol based on blind signatures (see Appendix A.1) and public key cryptography (see Appendix A.2) that includes the security features: eligibility, democracy, privacy, verifiability, accuracy, fairness, robustness, receipt-freeness and coercion-resistant.

The paper is organized as follows: Section 2, presents the security requirements of electronic voting systems; in Section 3, we explain the notation and method used to create our protocol; Section 4 contains the description of the different phases of the Secure Internet Voting Protocol (SIVP); Section 5 includes the security analysis of SIVP and a comparison with other e-voting protocols; finally Section 6 draws the conclusions.

2. Background

The general security requirements of electronic voting systems are ((Sampigethaya & Poovendran, 2006), (Tubella i Casadevall & Vilaseca i Requena, 2005)):

- **Eligibility:** Only those who meet certain criteria can vote, so it must be possible to verify the validity of each voter.
- **Democracy:** Each voter can vote only once.

* Corresponding author at: Calle 27BN#6D-54 (Popayán, Colombia).
E-mail addresses: cristsati@hotmail.com (C. Satizábal), paez-r@javeriana.edu.co (R. Páez), jforne@entel.upc.edu (J. Forné).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.12.016
1319-1578/

- **Privacy:** It should not be possible for anyone to relate a vote with his/her voter, either in short or long term.
- **Verifiability:** Any voter can verify that his/her vote was correctly recorded and the final tally contains his/her vote.
- **Dispute-Freeness:** A mechanism must be provided to resolve disputes that arise at any stage of the process.
- **Accuracy:** The final result of the election must contain all valid votes, so they must be correctly registered and counted.
- **Fairness:** To avoid any interference in voter behavior, counting cannot begin until the election is over.

To ensure resistance against adversaries, a system must also meet the following requirements (Sampigethaya and Poovendran, 2006):

- **Robustness:** It must be robust against passive and active attacks by corrupt authorities or voters, as well as against failures (such as giving access to non-participating authorities or voters)
- **Receipt-Freeness:** The receipt must not demonstrate the intention of the vote, to avoid the loss of privacy.
- **Coercion-Resistant:** The system should not allow possible coercions.

3. Notation and method

Table 1 shows the notation used to explain SIVP, arranged alphabetically. To design this protocol, we studied the security requirements of e-voting protocols and their different types. Then, in Satizábal and Páez (2018), we analyzed the phases, entities, cryptographic operations and security features of some existing e-voting protocols, and finally we determined the features to include, entities, phases, cryptographic operations and notation of our protocol.

Table 1
Notation.

Notation	Description
B (X,r)	Message X blinded with factor r
B^{-1}	Blinded removal function
Ballot	Ballot without any mark
$CERT_i$	Digital certificate of entity i
$challenge_x$	Random number
CSR_i^j	Certificate Sign Request number j of entity i
h()	Hash operation
ID_i	Identifier of entity i
k	Number of candidates
n	Number of trustees
N	Number of registered voters
$PK_i()$	Encryption/decryption with public key of entity i
PK_i^j/SK_i^j	Public key/private key pair number j of entity i
r_A	Blinded factor of authentication
rec	Voting receipt
r_V	Blinded factor of voting
S	Secret sharing function
S^{-1}	Secret sharing composing function
$SesID_X$	Session identifier
$SK_i()$	Encryption/decryption with private key of entity i
ST_i	Security token of the entity i
t	Threshold to compose shared private key
$Total_{cVo}^{x}$	Number of coercion votes
$Total_{V}^{x}$	Total number of votes
$Total_{Vo}^{x}$	Number of valid votes
v	Number of voters
vote	Marked ballot
V_V	Facial signature of voter V.
V′v	Computed facial signature of voter V

4. Secure Internet Voting Protocol (SIVP)

This protocol includes the following entities:

- **Voter (V):** Person with the right to vote in the electoral process.
- **Certification Authority (CA):** Issues the certificates of authorities and voters.
- **Registration Authority (RA):** Registers voters prior the election.
- **Authentication Center (AC):** Authenticates voters.
- **Voting Center (VC):** Gives the ballot to voters and collects the votes.
- **Tally Center (TC):** Stores the votes and obtains the result of the election.
- **Electoral Authority (EA):** Generates the key pair of the election (PK_E/SK_E) and divides SK_E among trustees.
- **Trustees (Tr):** Collaborate to compose SK_E and decrypt the votes in the counting phase. They can be representatives of the different political parties.
- **Independent Organizations (IO):** Responsible for independently validate the truthfulness of the election.
- **Bulletin Board (BB):** To publish the public information of the electoral process, including final count.

Also, SIVP uses the following lists:

- **PVL:** Public Voters List. Fields: voter's ID (ID_V), certificates of the voter ($CERT_V^1$, $CERT_V^2$). It is published in BB before the election by CA.
- **CL:** Candidates List. Fields: number, name and party of each candidate. It is published in BB before the election by EA.
- **VL:** Voters List. Fields: ID_V, voter's name ($name_V$), $CERT_V^1$, $CERT_V^2$, $challenge_1$, V_V. It is sent to AC by CA.
- **AVL:** Authenticated Voters List. Fields: ID_V, V'_V, $challenge_2$, $h(ID_V, challenge_2)$, coercion status, authentication key (PK_V^1 or PK_V^2), authentication token ($SK_V^1(h(ID_V,challenge_2))$ or $SK_V^2(SK_V^2(PK_V^1(h(ID_V,challenge_2))))$), blinded authentication evidence ($SK_{AC}^1(B(h(PK_{Vo}, ID_{Vo}), r_A))$ or $SK_{AC}^2(B(h(PK_{Vo}, ID_{Vo}), r_A))$), voting evidence ($SK_{VC}(SK_V^1(h(ID_V,challenge_2)))$ or $SK_{VC}(SK_V^2(SK_V^2(PK_V^1(h(ID_V,challenge_2)))))$), rec. It is known only by AC.
- **VVL:** Verification Voters List. Fields: ID_V, $challenge_2$,authentication key (PK_V^1 or PK_V^2), authentication token ($SK_V^1(h(ID_V,challenge_2))$ or $SK_V^2(SK_V^2(PK_V^1(h(ID_V,challenge_2))))$), voting evidence ($SK_{VC}(SK_V^1(h(ID_V,challenge_2)))$ or $SK_{VC}(SK_V^2(SK_V^2(PK_V^1(h(ID_V,challenge_2)))))$). It is sent to IO by AC.
- **VoL:** Votes List. Fields: ID of the vote (ID_{Vo}), public key of the vote (PK_{Vo}), authentication evidence ($SK_{AC}^1(h(PK_{Vo}, ID_{Vo}))$), $challenge_3$, blinded voting evidence ($SK_{VC}(B(SK_V^1(h(ID_V,challenge_2), r_v)))$, encrypted vote ($SK_{Vo}(PK_E(vote, ID_{Vo}))$), timestamp. It is sent to TC by VC.
- **VVoL:** Verification Votes List. Fields: ID_{Vo}, PK_{Vo}, authentication evidence (SK_{AC}^1 $(h(PK_{Vo}, ID_{Vo}))$), encrypted vote ($SK_{Vo}(PK_E(vote, ID_{Vo}))$). It is sent to IO by TC.
- **CVoL:** Coercion Votes List. Fields: ID_{Vo}, PK_{Vo}, authentication evidence ($SK_{AC}^2(h(PK_{Vo}, ID_{Vo}))$), $challenge_3$, blinded voting evidence ($SK_{VC}(B(SK_V^2(SK_V^2(PK_V^1(h(ID_V,challenge_2)))),r_v))$), encrypted vote ($SK_{Vo}(PK_E(vote, ID_{Vo}))$), timestamp. It is sent to TC by VC.
- **VCVoL:** Verification Coercion Votes List. Fields: ID_{Vo}, PK_{Vo}, authentication evidence ($SK_{AC}^2(h(PK_{Vo}, ID_{Vo}))$), encrypted vote ($SK_{Vo}(PK_E(vote, ID_{Vo}))$). It is sent to IO by TC.
- **DVoL:** Decrypted Votes List. Fields: ID_{Vo}, valid decrypted vote. It is sent to IO by TC.
- **DCVoL:** Decrypted Coercion Votes List. Fields: ID_{Vo}, decrypted coercion vote. It is sent to IO by TC.
- **IVoL:** Invalid Votes List. Fields: ID_{Vo}, invalid decrypted vote. It is sent to IO by TC.

- **ReL:** Results List. Fields: results per candidate (number, name and party of each candidate, number of votes per candidate), number of valid votes ($Total^3_{Vo}$), number of invalid votes ($Total_{IVo}$), number of coercion votes ($Total^3_{CVo}$), total number of votes ($Total^1_V$)). It is published in BB at the end of the counting phase by TC.
- **CReL:** Coercion Results List. Fields: results of the coercion votes per candidate (number, name and party of each candidate, number of coercion votes per candidate), number of coercion votes ($Total^3_{CVo}$). It is known only by TC.

Before the phases of this protocol, CA has generated its certificate ($CERT_{CA}$), the two certificates of AC ($CERT^1_{AC}$ (key pair used to authenticate voters) and $CERT^2_{AC}$ (key pair used for coercion votes)) and the certificates of VC ($CERT_{VC}$), TC ($CERT_{TC}$), EA ($CERT_{EA}$) and RAs ($CERT_{RA}$). All the certificates are published in BB. The private key of each RA is inside a security token (ST_{RA}) (see Appendix A.3) protected by a PIN (Personal Identification Number). The private keys of CA, AC, VC, TC and EA must be stored in a safe place. Thus, it is necessary a PKI (Public Key Infrastructure) (see ITU-T (2000) and Housley et al. (2002)).

4.1. Announcement phase

Announcement phase is carried out prior the election. Here, the parameters of the electoral process are established (date of election, type of election, CL, open and close hour).

Three days before the election, EA generates the key pair of the election (PK_E/SK_E). This key pair will be used to encrypt and decrypt the votes. Then, EA shares SK_E among the "n" Trustees (Tr), through a secret sharing function (S) with a threshold (t) (see Appendix A.4). Each trustee receives its part of SK_E in a security token (ST_{Tr}) protected by a PIN (see Fig. 1). Each trustee must keep its ST_{Tr} in a safe place until the counting phase.

4.2. Registration phase

Registration phase is carried out prior the election. Here, V goes personally to RA and presents his/her identification document. RA generates two key pairs to the voter: voting key pair (PK^1_V/SK^1_V) and anti-coercion key pair (PK^2_V/SK^2_V). These key pairs and the certificates issued by CA are stored in the security token of the voter (ST_V). Each private key is protected by a PIN (PIN_1 and PIN_2). These PINs must be remembered by the voter. Also, RA takes a sequence of pictures of the voter, to generate the facial signature (V_V) (see Appendix A.5). Thus, the authentication of the voter during election includes three factors: something he/she has (ST_V), something he/she knows (private key) and something he/she is (facial signature) (see Fig. 2).

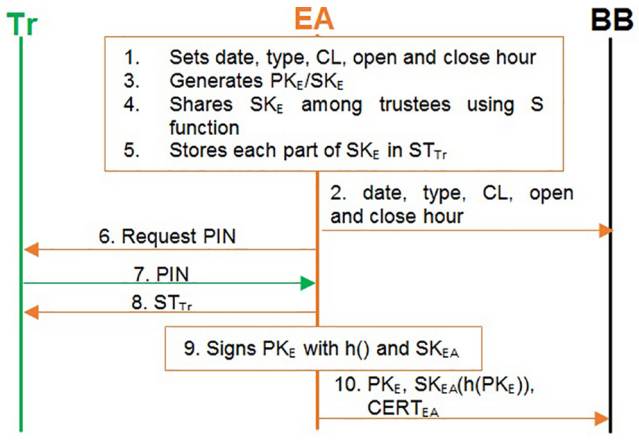

Fig. 1. Announcement Phase.

PVL is published in BB when the registration stage is closed, a month before the election, and at the same time VL is sent to the AC.

If the voter loses his/her security token (ST_V) or the PINs and the private keys were compromised, the voter must go again personally to RA, inform the incident and repeat the registration phase. Also, if the certificates of the voter expire, he/she must repeat the registration phase. CA will revoke the certificates.

4.3. Authentication phase

Authentication phase is carried out the day of the election. Here, the voter has two options: to use the voting key pair PIN (PIN_1) or to use the anti-coercion key pair PIN (PIN_2). Thus, if someone is coercing the voter, he/she must enter PIN_2. Otherwise, he/she must enter PIN_1. Note that the PIN never travels through the network; it is used by the ST_V to determine which private key it must use.

Fig. 3 shows the steps followed when voter enters PIN_1 and Fig. 4 shows the steps followed when voter enters PIN_2. Here, the voter carries out the same steps in both cases but AC and VC know about coercion and carry out the steps in a different way. Thus, if the person who is coercing the voter is with him/her, during the authentication and voting phases, this person will not realize that the voter has already informed AC about coercion.

4.4. Voting phase

Voting phase is carried out the day of the election, after the authentication phase. Fig. 5 shows the steps followed without coercion and Fig. 6 shows the steps followed with coercion. VC puts the votes without coercion in VoL and the votes with coercion in CVoL.

4.5. Counting phase

Counting phase is carried out the day of the election, after the close hour. Figs. 7 and 8 show the steps followed in this phase. Here, AC obtains $Total^1_V$ from AVL (only the voters that sent to AC the "voting evidence" are counted), and AC determines $Total^1_{Vo}$ and $Total^1_{CVo}$ using the "coercion status" field.

On the other hand, VC obtains $Total^2_{Vo}$ from VoL and $Total^2_{CVo}$ from CVoL (only the voters with the "encrypted vote" in these lists are counted). Then, VC adds $Total^2_{Vo}$ and $Total^2_{CVo}$ to obtain $Total^2_V$.

Trustees (Tr) must go personally to the TC and present their identification documents and ST_{Tr}. TC shows the numbers sent by AC and VC to the Tr and if they do not match, it can be possible that some voters did not send the "voting evidence" to AC before the close hour of the election. In this case, TC must calculate the difference between $Total^2_{Vo}$ and $Total^1_{Vo}$ and between $Total^2_{CVo}$ and $Total_{CVo}$. Then, VC must eliminate the encrypted votes in VoL and CVoL with the last timestamps, to equal $Total^2_{Vo}$ and $Total^2_{CVo}$ to the numbers of AC.

Next, TC uses S^{-1} function, ST_{Tr} and PIN of each Trustee to obtain SK_E. With this key, TC can decrypt the votes in VoL and CVoL and count them to obtain the results of the election (ReL).

4.6. Verification phase

Verification phase is carried out after the election. Fig. 9 shows the steps followed in this phase. Here, IO receives VVL from AC, VVoL and VCVoL from VC, and SK_E, DVoL, DCVoL and IVoL from TC. Also, IO downloads PVL and ReL from BB. With this information IO can verify if the authentication, voting and counting phases were carried out correctly and if the election results are correct.

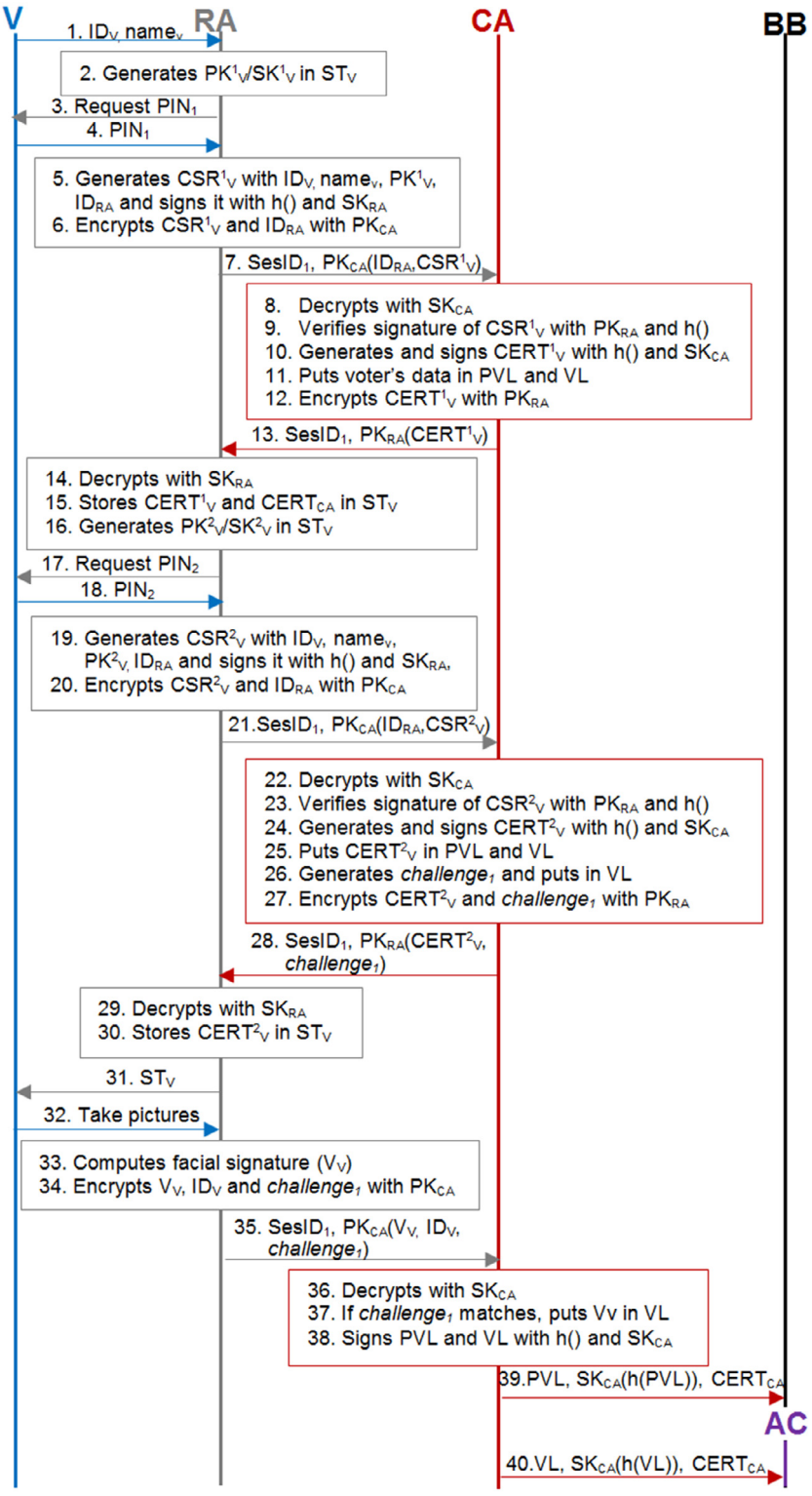

Fig. 2. Registration Phase.

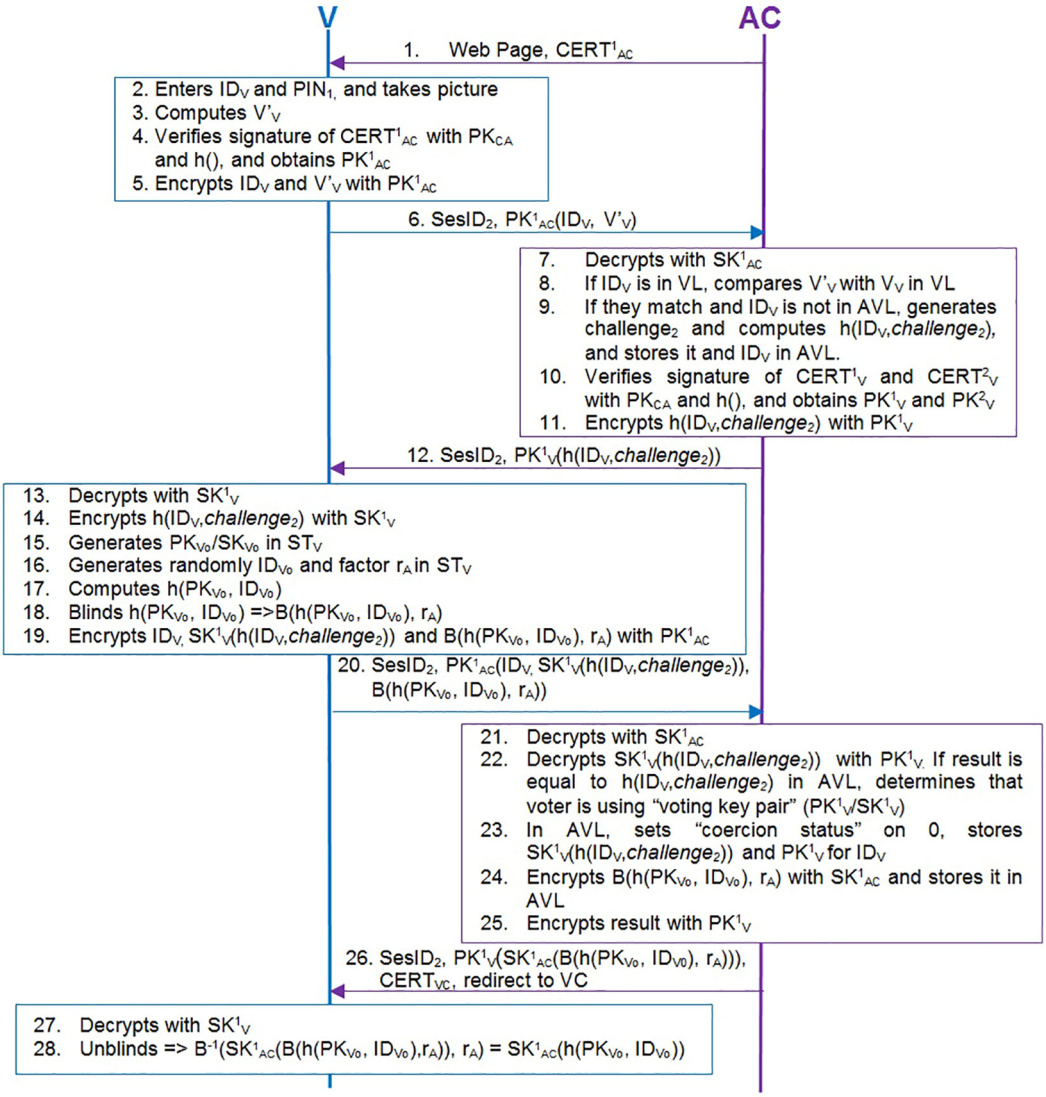

Fig. 3. Authentication Phase: Without Coercion.

Additionally, in this phase, the voter can verify if his/her vote was recorded.

5. Discussion

5.1. Security analysis

SIVP meets the following security features:

- **Eligibility:**
  Requirement 1: Only registered voters can vote.
  Proof: VL includes the information of registered voters. Thus, during authentication phase, AC verifies that ID_V is in VL (see step 8 in Fig. 3 and Fig. 4). If ID_V is not in VL, the voter will be rejected by the system. However, if an attacker tries to impersonate a voter and use the ID_V of other person, the V'_V will not match with V_V in VL (see Appendix A.5 and steps 8 and 9 in Fig. 3 and Fig. 4), so the attacker will be rejected. Even if the attacker pass the facial signature verification, he/she must prove to AC that he/she knows a secret key of the voter (SK^1_V or SK^2_V) by properly decrypting $PK^1_V(h(ID_V,challenge_2))$. However, only the voter has these secret keys in the ST_V and thanks to public key cryptography, the attacker cannot obtain the secret key from the public key (see Appendix A.2). For that reason, $h(ID_V, challenge_2)$ will not match with the one in AVL so AC will not enable the attacker to vote (see step 22 in Fig. 3 and steps 22–24 in Fig. 4) since VC will only allow to vote those voters who have the authentication evidence correctly encrypted by the AC ($SK^1_{AC}(h(PK_{Vo}, ID_{Vo}))$ or $SK^2_{AC}(h(PK_{Vo}, ID_{Vo}))$) (see steps 6 and 7 in Fig. 5 and steps 6–8 in Fig. 6).
- **Democracy:**
  Requirement 2: A voter only can vote once.
  Proof: AC and VC store in their lists (AVL, VoL and CVoL) evidences of the different steps of the authentication and voting phases. Thus, in a new attempt to vote, during authentication phase, AC verifies if ID_V is already in AVL (see step 9 in Figs. 3 and 4). If so, and AC has not already generated the blinded authentication evidence ($SK^1_{AC}(B(h(PK_{Vo}, ID_{Vo}), r_A)$ or SK^2_{AC} $(B(h(PK_{Vo}, ID_{Vo}), r_A))$ for the voter, AC will generate again $challenge_2$ and authentication will continue. Otherwise, AC will redirect the voter to VC. In addition, during voting phase, VC verifies if ID_{Vo} is in VoL or CVoL (see step 7 in Fig. 5 and step 8 in Fig. 6). If so, VC must verify if the encrypted vote (SK_{Vo} $(PK_E(vote, ID_{Vo}))$

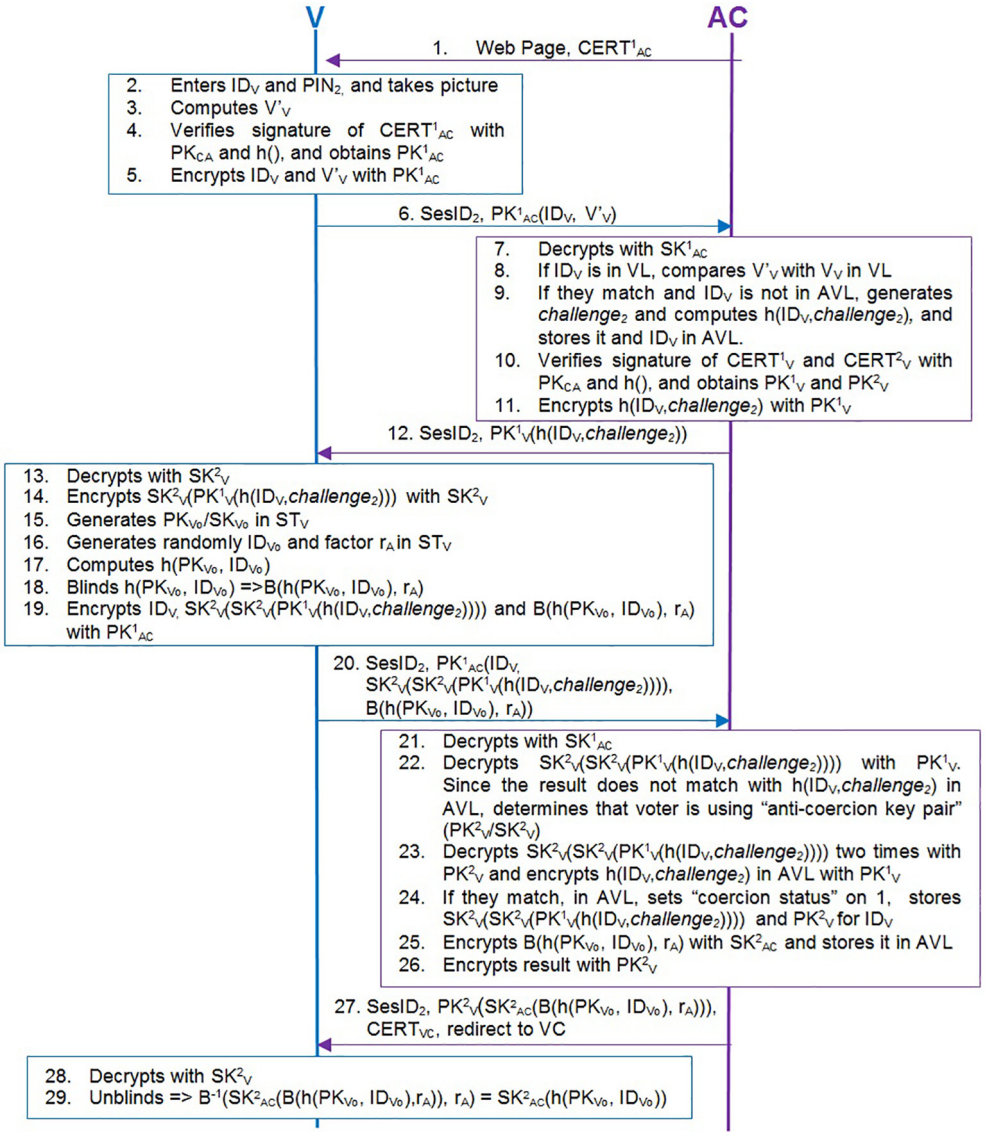

Fig. 4. Authentication Phase: With Coercion.

is in the list. If it is not, the voter can vote. Otherwise, VC will redirect the voter to AC, so AC can give her/him the voting receipt (rec).

- **Privacy:**

Requirement 3: The vote cannot be related with the voter.

Proof: The use of blind signatures (see Appendix A.1) ensures that AC and VC cannot relate the vote with the voter. Thus, AC has ID_V in VL, but it cannot obtain ID_{Vo} from the blinded authentication evidence ($SK^1_{AC}(B(h(PK_{Vo}, ID_{Vo}), r_A))$ or $SK^2_{AC}(B(h(PK_{Vo}, ID_{Vo}), r_A))$) in AVL. Although AC can decrypt the blinded authentication evidence with PK^1_{AC} or PK^2_{AC} to obtain $B(h(PK_{Vo}, ID_{Vo}), r_A)$; AC does not know r_A (only the voter knows r_A) to unblind the result.

On the other hand, VC has ID_{Vo} in VoL and CVoL but VC cannot obtain ID_V from the blinded voting evidence ($SK_{VC}(B(SK^1_V(h(ID_V, challenge_2)),r_v))$ or $SK_{VC}(B(SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2)))),r_v))$) in VoL or CVoL. Although VC can decrypt the blinded voting evidence with PK_{VC} to obtain $B(SK^1_V(h(ID_V,challenge_2)),r_v)$ or $B(SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2)))),r_v)$, VC does not know r_V (only the voter knows r_V) to unblind the result.

Thus, only the voter has all the information required to decrypt these messages but at the end of the voting phase ID_{Vo}, PK_{Vo}, SK_{Vo}, r_A and r_V are deleted from ST_V (see step 36 in Fig. 5 and step 37 in Fig. 6), so in long term it is also not possible to bind the voter with his/her vote. Therefore, IO cannot relate a vote with his/her voter, with the information it receives from AC and TC (see Fig. 9), neither an attacker with the information published at BB.

- **Verifiability:**

Requirement 4: A voter can verify that his/her vote was correctly recorded and counted.

Proof: The voter can ask the AC if he/she voted during verification phase (see steps 30–32 in Fig. 9). However, it is not possible to know the intention of the vote, since if the voter was coerced, the attacker should not know that he/she reported the coercion. Nevertheless, during verification phase, IO can verify that all the votes were included in the final tally decrypting each vote, counting them and comparing its results with those provided by TC and BB (see Fig. 9).

- **Accuracy:**

Requirement 5: The final tally contains all valid votes.

Proof: During the counting phase, Tr verify that the number of

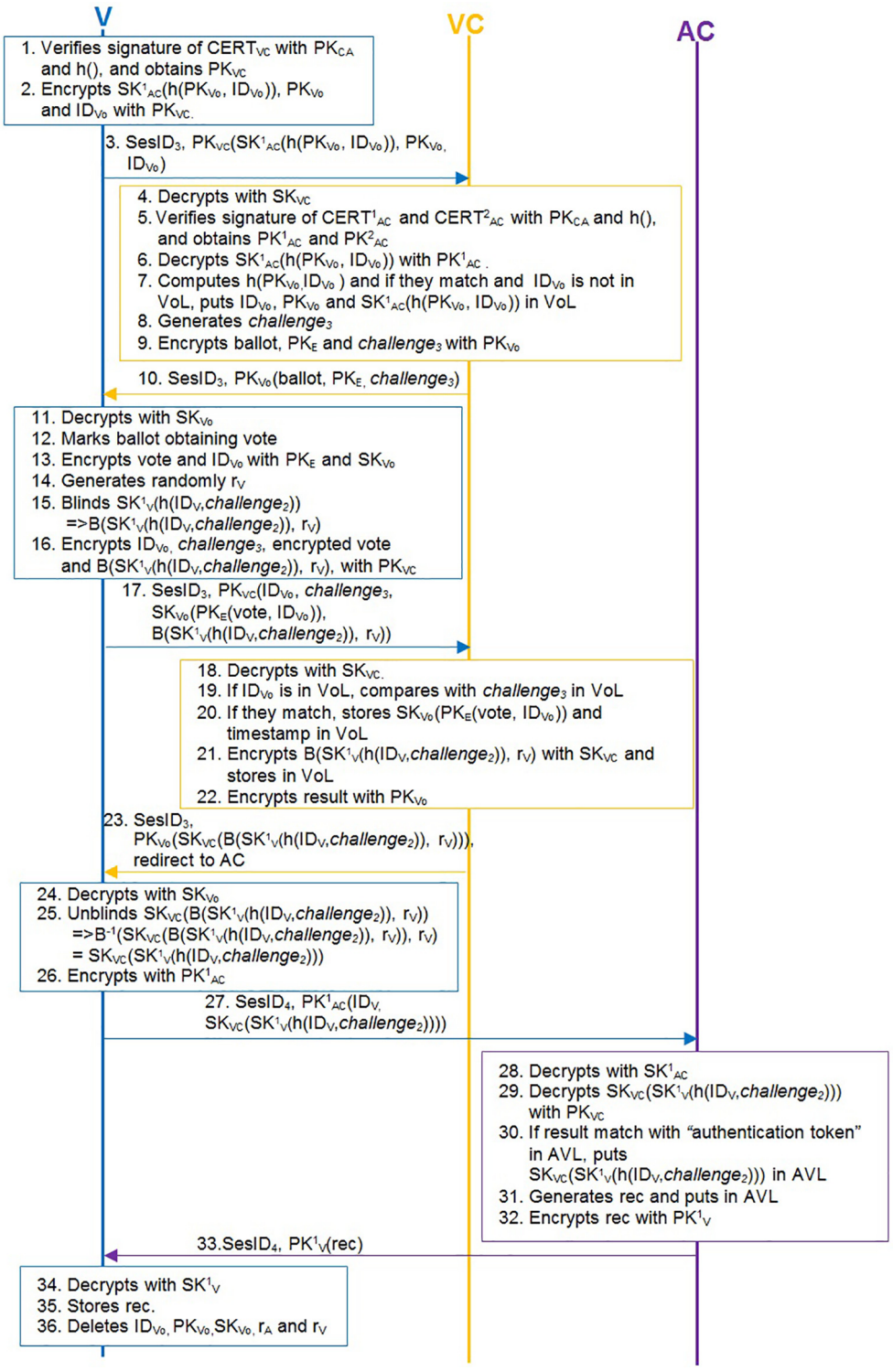

Fig. 5. Voting Phase: Without Coercion.

votes of AC and VC match (see Fig. 7). Then, TC verifies the validity of each vote, counts them and puts the results in ReL. Finally, TC shows ReL to Tr, that verify ReL contains all the votes (see Fig. 8). In addition, IO verifies the results during verification phase. First, IO verifies that each voter was correctly authenticated (see steps 16–19 in Fig. 9) and then, IO verifies that only authenticated voters voted (see steps 20–22, 25 and 26 in Fig. 9). Finally, IO decrypts each vote and verify that its results match with those provided by TC and BB (see steps 23, 24,27–29 in Fig. 9). Thus, if an attacker tries to introduce a false vote in the lists of AC and TC, this can be detected by IO during verification phase, since the attacker cannot correctly generate the voting evidence ($SK_{VC}(SK^1_V(h(ID_V,challenge_2)))$ or $SK_{VC}(SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2))))))$ because the attacker

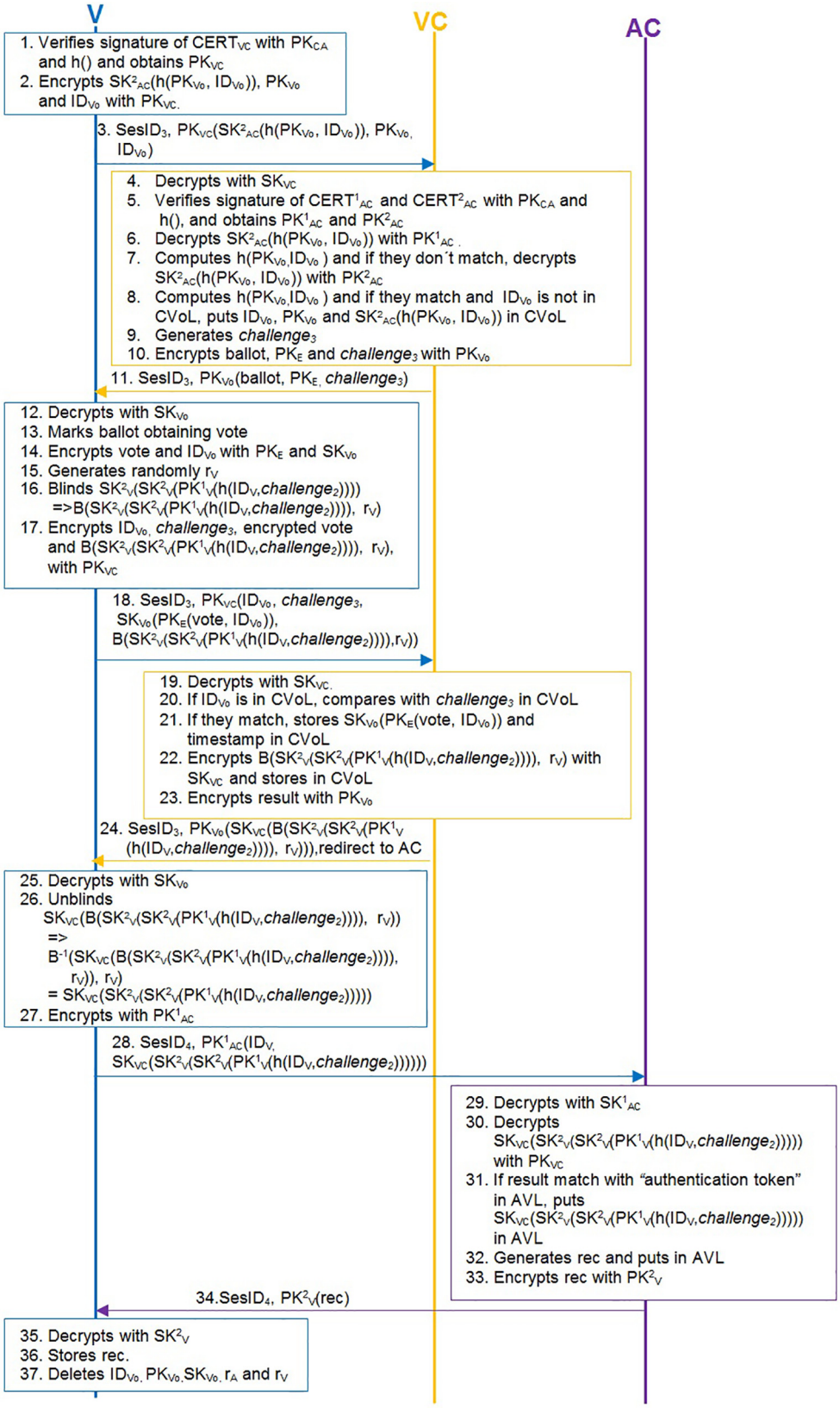

Fig. 6. Voting Phase: With Coercion.

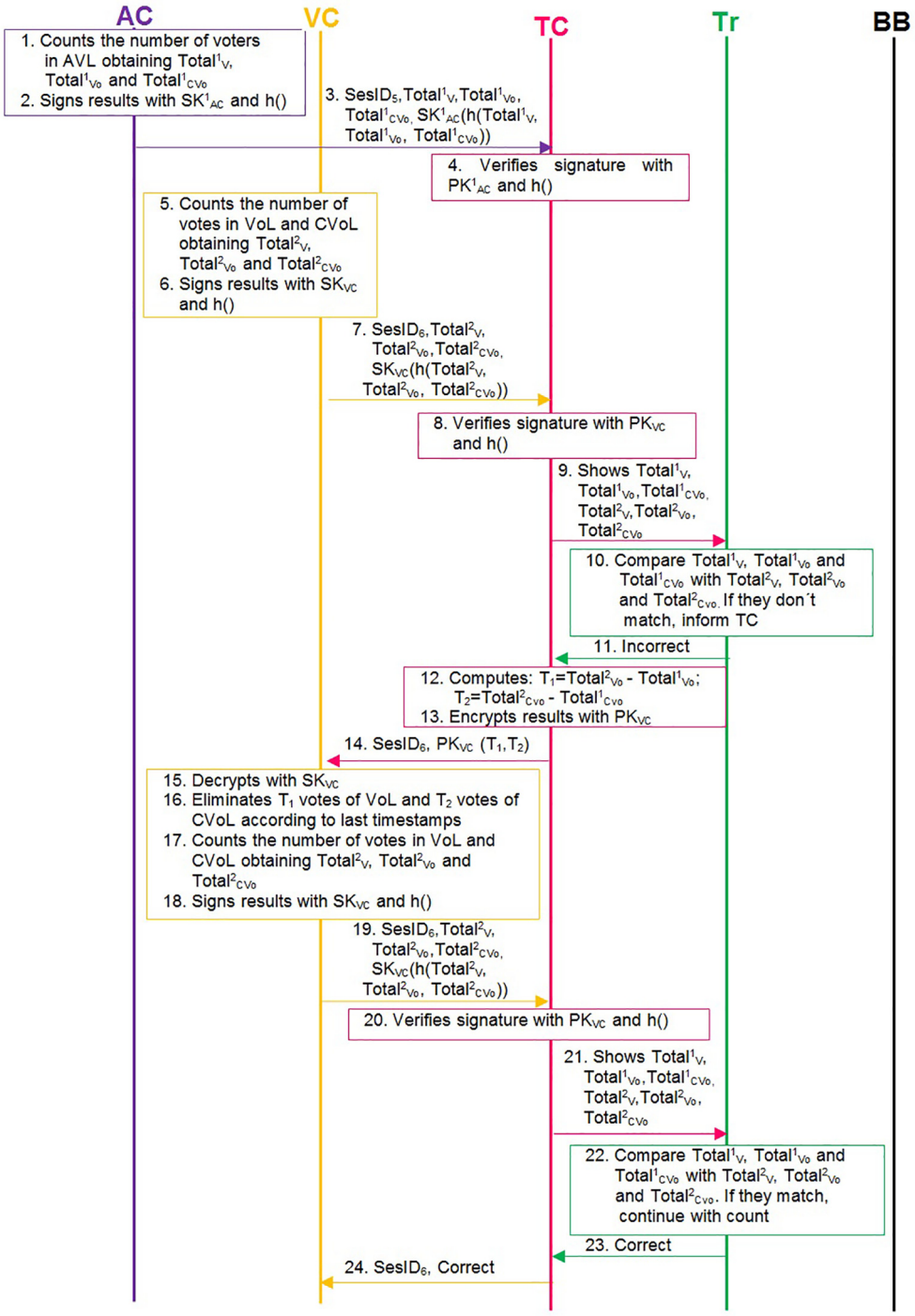

Fig. 7. Counting Phase (1).

does not know SK_{VC} and SK_V^1 or SK_V^2, and neither the authentication evidence ($SK_{AC}^1(h(PK_{Vo}, ID_{Vo}))$ or $SK_{AC}^2(h(PK_{Vo}, ID_{Vo}))$ because the attacker does not know SK_{AC}^1 or SK_{AC}^2.

- **Fairness:**
  Requirement 6: Counting cannot begin until the election is over.
  Proof: The key used to decrypt the votes (SK_E) is composed during the counting phase, when Trustees collaborate to obtain it using a secret sharing function (see steps 30–33 in Fig. 7), so the votes are counted after the close hour of election.

- **Robustness:**
  Requirement 7: The system must be robust against passive and active attacks by corrupt authorities or voters.
  Proof: If an attacker captures the messages between V and AC or VC during authentication and voting phases, the attacker cannot decrypt these messages because they are encrypted with public keys (PK_{AC}^1, PK_V^1, PK_V^2, PK_{VC}, PK_{Vo}), so only the entity with the corresponding secret key (SK_{AC}^1, SK_V^1, SK_V^2, SK_{VC}, SK_{Vo}) can decrypt each of them and the attacker cannot generate the secret key from the public key (see Appendix A.2). In addition, in each message that the voter sends to AC or VC, he/she must include

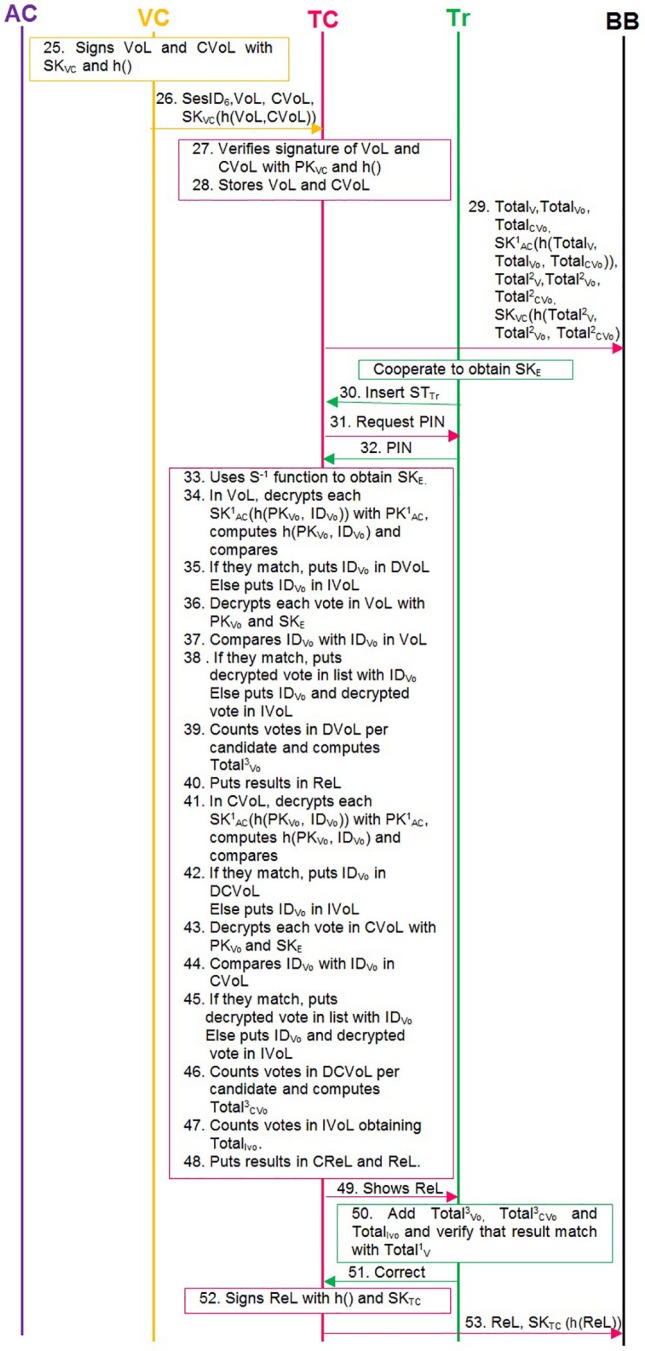

Fig. 8. Counting Phase (2).

an evidence of his/her identity, for example: V'_V in step 6 and $SK^1_V(h(ID_V,challenge_2))$ or $SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2))))$ in step 20 of authentication phase (see Figs. 3 and 4); $SK^1_{AC}(h(PK_{Vo}, ID_{Vo}))$ in step 3, $challenge_3$ in step 17 or 18, and $SK_{VC}(SK^1_V(h(ID_V,challenge_2)))$ or $SK_{VC}(SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2)))))$ in step 27 or 28 in voting phase (see Figs. 5 and 6). Thus, an attacker cannot impersonate a voter because he/she cannot decrypt the messages between V and VC or AC to obtain these evidences, so AC and VC can detect the attacker when they verify them.

If AC tries to authenticate a fake voter, it cannot generate the authentication token $(SK^1_V(h(ID_V,challenge_2))$ or $SK^2_V(SK^2_V(PK^1_V(h(ID_V,challenge_2)))))$ because it does not know the secret keys of the voter(SK^1_V and SK^2_V) . Thus, when IO decrypts the authentication token with the authentication key (PK^1_V or PK^2_V), computes $h(ID_V,challenge_2)$ and compares them, it verifies that the results do not match (see step 19 in Fig. 9). Therefore, IO decrypts again the first result ($SK^2_V(PK^1_V(h(ID_V,challenge_2))))$ with the authentication key, encrypts $h(ID_V,challenge_2)$ with PK^1_V, and compares them to determine if the voter is using the anti-coercion key pair. Otherwise, IO validates that AC became corrupted.

If VC tries to introduce a false vote, it cannot generate a valid authentication evidence ($SK^1_{AC}(h(PK_{Vo}, ID_{Vo}))$ or $SK^2_{AC}(h(PK_{Vo}, ID_{Vo}))$), because VC does not know the private keys of AC (SK^1_{AC} or SK^2_{AC}). Thus, when IO decrypts the authentication evidence with SK^1_{AC} or SK^2_{AC}, computes $h(PK_{Vo}, ID_{Vo})$ and compares them, the results do not match, so IO validates that the vote is not valid and VC became corrupted (see steps 21, 22, 25 and 26 in Fig. 9).

If TC tries to introduce false decrypted votes, during verification phase, when IO decrypts the votes, it can detect that the results are not correct, because decrypted votes in DVoL, DCVoL and IVoL do not match with the decrypted votes obtained by IO(see steps 23, 24, 27–29 in Fig. 9). Thus, IO can detect that the TC became corrupted.

- **Receipt-Freeness:**
  Requirement 8: The receipt must not demonstrate the intention of the vote.
  Proof: AC gives a receipt to the voter, at the end of voting phase (see step 33 in Fig. 5 and step 34 in Fig. 6), that has only $name_V$ and ID_V with the signature of one authority, so this does not demonstrate the intention of the vote (see step 33 in Fig. 5 and step 34 in Fig. 6).
- **Coercion-Resistant:**
  Requirement 9: The system must detect coercions.
  Proof: To avoid coercion, during the registration phase, the voter receives two key pairs: the voting key pair (PK^1_V/SK^1_V) and the anti-coercion key pair (PK^2_V/SK^2_V), each one protected by a PIN in a ST_V (see Fig. 2). Thus, if someone is coercing the voter, he/she must enter PIN_2 at the beginning of authentication phase (see step 2 in Fig. 4). Hence, AC realizes that the voter is being coerced because the voter uses SK^2_V to decrypt $PK^1_V(h(ID_V,challenge_2))$, so $h(ID_V,challenge_2)$ does not match with the one in AVL. Then, AC informs to VC about coercion using SK^2_{AC} to encrypt $B(h(PK_{Vo}, ID_{Vo}), r_A)$ (see steps 22–25 in Fig. 4).

5.2. Comparison

There are three types of e-voting protocols: based on blind signatures to protect privacy of votes; based on mix-nets to implement an anonymous channel or to cut the voter-vote link (Pereira and Rivest, 2017); and based on homomorphic encryption to protect vote's privacy and increase the speed of vote tallying (Acar et al., 2017).

In (Satizábal and Páez, 2018), we analyzed four protocols: one uses blind signatures (Li, Hwang and Lai protocol (Li et al., 2009)), other uses mix nets (Meng protocol (Meng, 2007)), other uses homomorphic encryption (EVIV protocol (Joaquim et al., 2013)) and the last is used in real electoral processes (I-Voting for Estonian Elections (Heiberg et al., 2012)).

Table 2 shows a comparison of these four protocols with SIVP, according to the security features that they include. Thus, while SIVP includes the nine security features, EVIV protocol only includes three and the others include five features.

Table 3 and Fig. 10 show the number of cryptographic operations per phase of the five protocols when the number of registered voters (N) is 36.783.940, the number of voters (v) is 19.636.714, the number of candidates (k) is 9 (blank vote is one of the options), and the number of political parties (n) is 16. These data were obtained from the results of the first round of 2018 Colombian Presidential elections (Colombia.com, 2018).

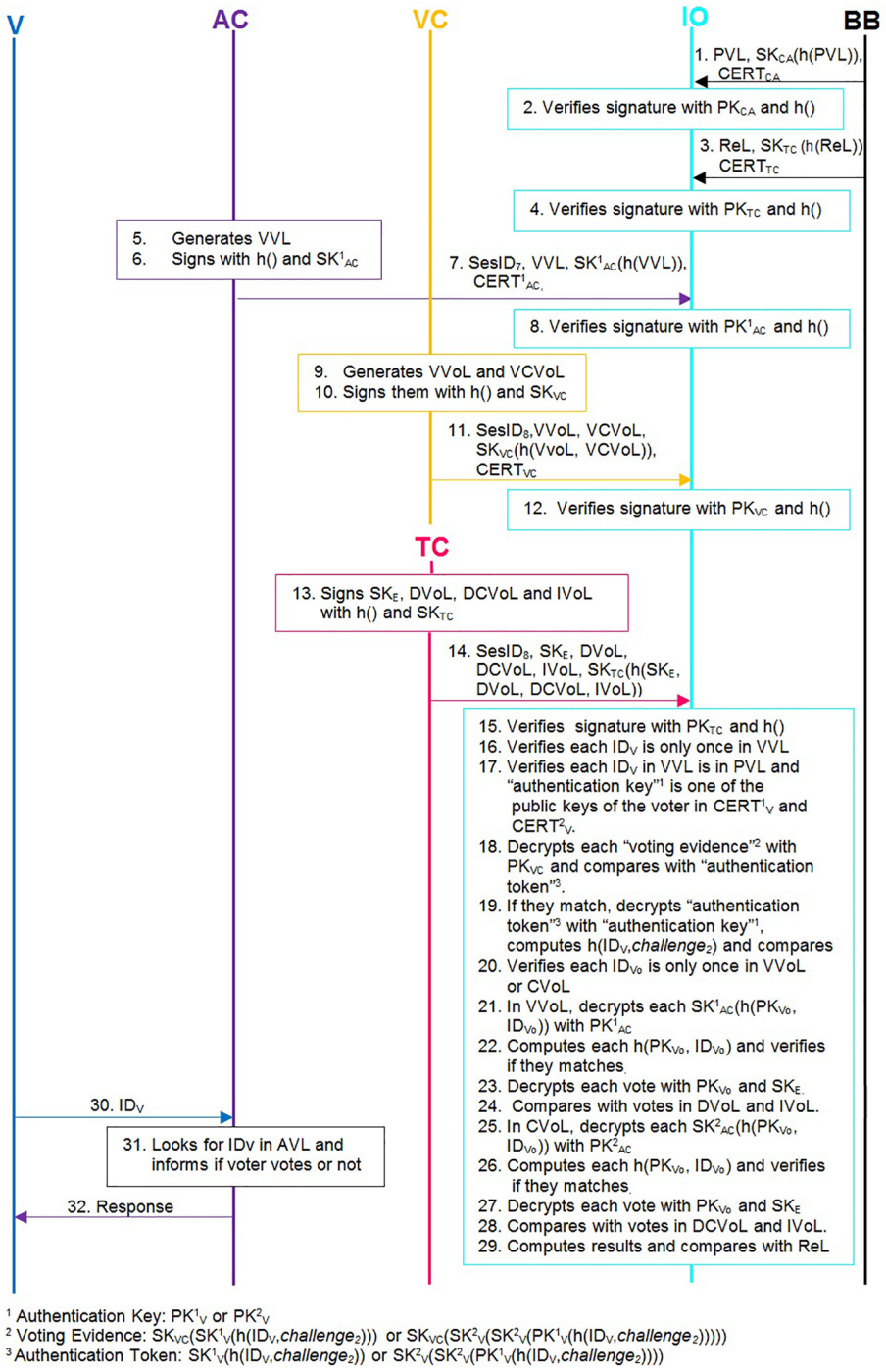

Fig. 9. Verification Phase.

Table 2
Security features of e-voting protocols.

Feature	Li, Hwang and Lai	Meng	EVIV	I-Voting	SIVP
eligibility				X	X
democracy	X			X	X
privacy	X	X	X	X	X
verifiability	X	X	X		X
accuracy	X		X	X	X
fairness		X		X	X
robustness					X
receipt-freeness		X			X
coercion-resistant	X	X			X

Table 3
Number of cryptographic operations per phase (N = 36.783.940, v = 19.636.714, k = 9, n = 16).

Protocol	Phase 1	Phase 2	Phase 3	Phase 4
MENG	**Preparation**	**Registration**		**Voting**
Formula	5 + n + 2N	3N + 7nN + 2n		v
# Operations	73.567.901	4.230.153.132		19.636.714
LI, HWANG, LAI		**Registering**	**Authentication**	**Voting**
Formula		1 + N	28v	15v
# Operations		36.783.941	549.827.992	294.550.710
I-VOTING		**Setup**		**Voting**
Formula		10 + 6N		27v
# Operations		220.703.650		530.191.278
EVIV	**Voter Enrolment**	**Election Registration**		**Vote Casting**
Formula	1 + 3N	7 + 24N + 5kN + 2n		3 + 22v + 5kv + 7n
# Operations	110.351.821	2.538.091.899		1.315.659.953
SIVP	**Announcement**	**Registration**	**Authentication**	**Voting**
Formula	4	4 + 24N	21v	4 + 22v
# Operations	4	882.814.564	412.370.994	432.007.712
PROTOCOL	**PHASE 5**	**PHASE 6**	**PHASE 7**	**TOTAL**
MENG		**Talling**		
Formula		$2 + n^2v + 6nv + 3n$		$6 + 6n + 5\ N+7nN + n^2v + 6nv + v$
# Operations		6.912.123.378		11.235.481.125
LI, HWANG, LAI		**Counting**		
Formula		v		1 + N + 44v
# Operations		19.636.714		900.799.357
I-VOTING	**Revocation**	**Tabulation**		
Formula	4 + 2v	1 + v		15 + 6 N + 30v
# Operations	39.273.432	19.636.715		809.805.075
EVIV		**Public Verification and Vote Counting**		
Formula		16 + 2kN + 2v + 2kv		27 + 27 N + 7kN + 9n + 24v + 7kv
# Operations		1.054.845.216		5.018.948.889
SIVP		**Counting**	**Verification**	
Formula		21 + 4v	24 + 7v	57 + 24 N + 54v
# Operations		78.546.877	137.457.022	1.943.197.173

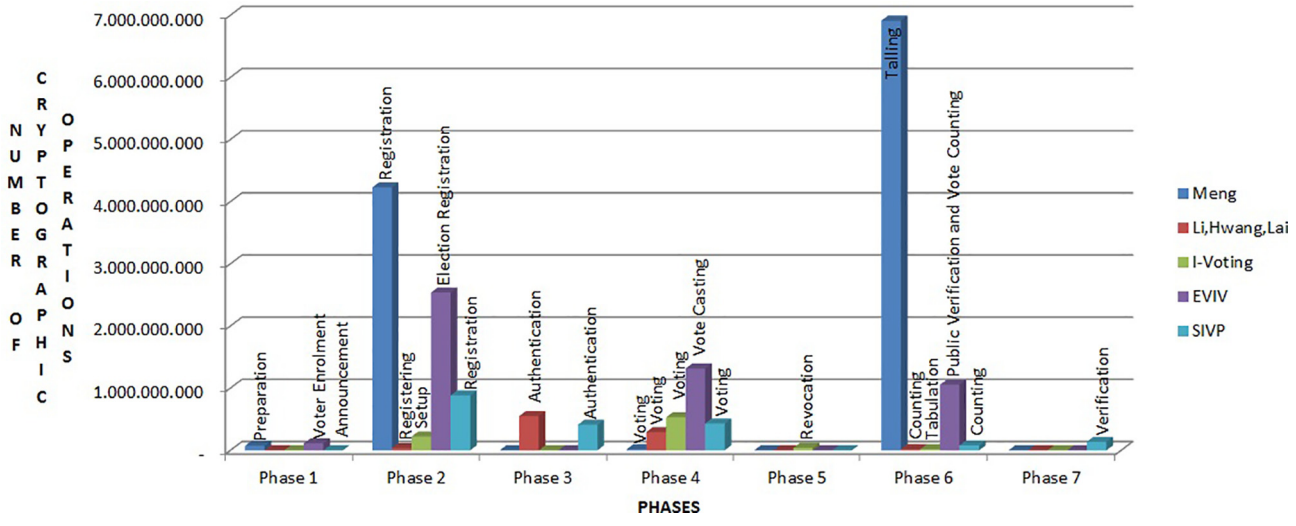

Fig. 10. Number of Cryptographic Operations per Phase ((N = 36.783.940, v = 19.636.714, k = 9, n = 16).

Since the number and the name of the phases of these protocols does not match, we group those with similar names, obtaining 7 phases. Thus, we can see that the common phases of the protocols are: registration, voting and counting. Only SIVP has a verification phase, although EVIV puts together the counting and verification phases. However, the addition of counting and verification phases operations of SIVP is 216.003.899, which is much lower than the number of operations of public verification and vote counting phase of EVIV.

Therefore, we can see that Meng and EVIV are the costliest protocols at cryptographic level and SIVP has the third place. However, the number of cryptographic operations was not excessively increased in SIVP despite the addition of more security features. Comparing SIVP with Li, Hwang, Lai and I-Voting protocols, the largest increase in the number of cryptographic operations occurs in the registration phase, since 45,43% of the cryptographic operations of SIVP are carried out during this phase, but this phase lasts two months before the election so the computational load can be distributed during this period of time. Also, in each phase, the computational load is distributed between 2 or more entities, so the number of cryptographic operations is not so high per entity. If we compare authentication phase of SIVP and Li, Hwang, Lai, our protocol carries out a less number of cryptographic operations. In voting phase, our protocol carries out a less number of cryptographic operations than I-Voting but a greater number of operations compared with Li, Hwang, Lai protocol. In counting phase,

our protocol carries out a greater number of operations than I-Voting and Li, Hwang, Lai protocols.

In addition, it is good that the number of cryptographic operations of SIVP does not depend on the number of candidates or the number of political parties, since Colombia currently has 16 political parties and 1114 candidates were registered in 2018 Colombian Senate elections.

6. Conclusions

SIVP is a new voting protocol for national electoral processes in Colombia based on blind signatures and public key cryptography. This protocol has six phases that are carried out sequentially: announcement, registration, authentication, voting, counting and verification. In addition, this includes 10 entities: voter (V), certification authority (CA), registration authority (RA), authentication center (AC), voting center (VC), tally center (TC), electoral authority (EA), trustees (Tr), independent organizations (IO) and bulletin board (BB).

SIVP stands out from the other analyzed protocols because it includes 9 security features (eligibility, democracy, privacy, verifiability, accuracy, fairness, robustness, receipt-freeness and coercion-resistant) while the others only offer 3 or 5. Furthermore, despite increasing the level of security, the number of cryptographic operations of SIVP does not increase excessively, occupying the third place among the analyzed protocols.

Since in Colombia, the authentication, voting and counting phases must be carried out the same day, we want to use ECC (Elliptic Curve Cryptography) (Certicom Research, 2009) to reduce the cost of cryptographic operations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Pontificia Universidad Javeriana – Bogotá (Colombia).

Appendix A. Technologies used

A.1 Blind signature

According to (Asghar, 2012), a blind signature scheme is a protocol for obtaining a signature σ on a message m from the signer S, such that S does not learn anything about σ and m. The basic layout of such a protocol is as follows: user U generates a secret random number r, embeds it into m to obtain m', the masked/blinded message, using function B (see Eq. (A.1)).

$$m\prime = \mathrm{B}(m,r) \tag{A.1}$$

Then, U sends m' to S. S has its key pair (SK,PK), where SK is the secret key and PK is the public key. S generates a signature σ' on m' (see Eq. (A.2)) and returns it to U.

$$\sigma' = SK(m') \tag{A.2}$$

U then removes the random blinding factor using function B^{-1} to obtain σ, the signature on m (see Eq. (A.3)).

$$\sigma = B^{-1}(\sigma', r) \tag{A.3}$$

Finally, U can verifies whether σ is a valid signature on m with respect to public key PK (see Eq. (A.4)).

$$PK(\sigma) = \mathrm{m} \tag{A.4}$$

A.2 Public key cryptography

It is also known as asymmetric cryptography since it involves the use of two keys: one is public (known to all) and the other is private (known only to its owner).

Diffie and Hellman (1976) postulated the conditions of a public key system:

1. It is computationally easy for user B to generate a key pair: public key $\mathrm{PK_B}$ and private key $\mathrm{SK_B}$
2. It is computationally easy for a sender A, knowing $\mathrm{PK_B}$ and the message to be encrypted m, generate the encrypted text C, applying the asymmetric encryption algorithm (see Eq. (A.5)).

$$C = PK_{\mathrm{B}}(m) \tag{A.5}$$

3. It is computationally easy for receiver B to decrypt C using its private key $\mathrm{SK_B}$ and the asymmetric decryption algorithm to retrieve the original message m (see Eq. (A.6)).

$$m = SK_{\mathrm{B}}(C) = SK_{\mathrm{B}}(PK_{\mathrm{B}}(m)) \tag{A.6}$$

4. It is computationally impossible for an opponent who knows $\mathrm{PK_B}$, to determine the private key $\mathrm{SK_B}$.
5. It is computationally impossible for an opponent who knows $\mathrm{PK_B}$ and C, to retrieve the original message m.

Examples of public key algorithms are: RSA (Rivest et al., 1978), DSS (NIST, 2000) and DH (Diffie and Hellman, 1976).

A.3 Security token

A security token is a portable device, such as an USB token, that stores some sort of personal information used to authenticate a person's identity and to grant access to a service or resource (Majaski, 2020).

A.4 Secret sharing

In a secret sharing scheme a dealer distributes shares of a secret among a set of n parties (Beimel, 2011). *"Informally, an n-party FSS (Function Secret Sharing) scheme splits a function "f" into "n" functions: $f_1;\ldots;f_n$, such that $f = f_1 + \ldots + f_n$"* (Luo et al., 2020).

Also, there is a collection A of subsets of parties called the access structure. In threshold secret-sharing schemes, all subsets whose size is bigger than some threshold (t), can reconstruct the secret, where $1 \leq \mathrm{t} \leq \mathrm{n}$ is an integer (Beimel, 2011).

A.5 Facial recognition

"*A facial recognition system uses biometrics to map facial features from a photograph or video and recognize a human face.*

The basic steps are:

1. *A picture of your face is captured from a photo or video.*
2. *Facial recognition software reads the geometry of your face. The software identifies facial landmarks — one system identifies 68 of them — that are key to distinguishing your face. The result: your facial signature.*

3. *Your facial signature — a mathematical formula — is compared to a database of known faces.*
4. *A determination is made. Your faceprint may match with an image in a facial recognition system database*" (Symanovich, 2019).

References

Acar, A., Aksu, H., Uluagac, A. S., Conti, M., 2017, October 06. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. Retrieved November 15, 2019, from <https://arxiv.org/abs/1704.03578> (accessed 11 November 2017).

Asghar, N., 2012. A Survey on Blind Digital Signatures. University of Waterloo.

Beimel, A., 2011. Secret-Sharing Schemes: A Survey. Lect. Notes Comput. Sci. 6639, 11–46.

Certicom Research, 2009. Standards for Efficient Cryptography SEC 1: Elliptic Curve Cryptography Version 2.0. Certicom Corp..

Collins English Dictionary. (s.a.). Definition of 'e-voting'. Retrieved October 03, 2019, from <https://www.collinsdictionary.com/dictionary/english/e-voting>.

Colombia.com, 2018, May 27. Resultados Elecciones Presidenciales 2018. Retrieved November 05, 2020, from <https://www.colombia.com/elecciones/2018/resultados/presidente.aspx?C=P1>.

Diffie, W., Hellman, M., 1976. New Directions in Cryptography. IEEE Trans. Inform. Theory 22 (6), 644–654.

Heiberg, S., Laud, P., Willemson, J., 2012. The Application of E-voting for Estonian Parliamentary Elections of 2011. Lect. Notes Comput. Sci. 7187, 208–223.

Housley, R., Polk, W., Ford, W., Solo, D., 2002. RFC3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF Network Working Group.

ITU-T, 2000. Recommendation X.509: Information Processing Systems - Open Systems Interconnection - The Directory : Authentication Framework (Technical Corrigendum). International Telecommunication Union.

Joaquim, R., Ferreira, P., Ribeiro, C., 2013. EVIV: An End-to-End Verifiable Internet Voting System. Comput. Secur. 32, 170–191.

Li, C.-T., Hwang, M.-S., Lai, Y.-C., 2009. A Verifiable Electronic Voting Scheme over the Internet. In: 6th International Conference on Information Technology: New Generations, USA. pp. 449–454.

Línea Democracia y Gobernabilidad, 2018, May. Cómo Opera la Corrupción Electoral en Colombia. Retrieved July 16, 2020, from Pares, Fundación Paz y Reconciliación: <https://pares.com.co/2018/05/15/como-opera-la-corrupcion-electoral-en-colombia/>.

Luo, J., Zang, L.F., Lin, F., Lin, C., 2020. Efficient Threshold Function Secret Sharing with Information-Theoretic Security. IEEE Access 8, 6523–6532.

Majaski, C., 2020, June 28. Security Token Definition. Retrieved November 2, 2020, from Investopedia: <https://www.investopedia.com/terms/s/security-token.asp>.

Meng, B., 2007. An Internet Voting Protocol with Receipt-Free and Coercion-Resistant. In: IEEE 7th Int. Conf. on Computer and Information Technology, Japan. IEEE, pp. 721–726.

NIST. 2000. Digital Signature Standard (DSS). FIPS PUB 186.

Pereira, O., Rivest, R.L., 2017. In: Marked Mix-Nets. Workshop on Advances in Secure Electronic Voting, Malta, pp. 1–17.

Rivest, R.L., Shamir, A., Adleman, L., 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21 (2), 120–126.

Sampigethaya, K., Poovendran, R., 2006. A Framework and Taxonomy for Comparison of Electronic Voting Schemes. Comput. Secur. 25 (2), 137–153.

Satizábal, C., Páez, R., 2018. Internet Voting Protocols: An Analysis of the Cryptographic Operations per Phase. Int. J. Commun. Netw. Inf. Secur. (IJCNIS) 10 (2), 305–322.

Symanovich, S., 2019, February 8. How Does Facial Recognition Work? Retrieved November 4, 2020, from Norton: <https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html>.

Tubella i Casadevall, I., Vilaseca i Requena, J., 2005. Sociedad del Conocimiento, Cómo Cambia el Mundo ante Nuestros Ojos. Editorial UOC, Barcelona.