I-FOO: An Enhancement to FOO Protocol

Nafiseh Mohamadi Shakiba
Information Security and E-voting Research Laboratory, Department of Electronic and Computer Science, Shahed University
Tehran, Iran

Mohamad-Ali Doostari
Information Security and E-voting Research Laboratory, Department of Electronic and Computer Science, Shahed University
Tehran, Iran

Shahin Norouzi
Information Security and E-voting Research Laboratory, Department of Electronic and Computer Science, Shahed University
Tehran, Iran

ABSTRACT
Various e-voting protocols have been proposed by researchers, while a few numbers of them are suitable to be implemented and utilized as an Internet-voting schema. Among these protocols, the FOO **Error! Reference source not found.** protocol has been implemented and utilized for several factors, due to simplicity and collusion resistance perspective. However, challenges of bribery, coercion, and unfairness which are dominant in the field of Internet-voting in this protocol have not been fully analyzed or predicted. It has only been assumed that the election itself can prevent them. This paper will analyze ways, which can enhance security features and eliminate defects of the FOO protocol under a new and secure protocol named I-FOO, which serves to be applicable in unsecure public environments such as the Internet. I-FOO protocol is able to protect the security of voting principles through modifications in encryption schemas, databases and information flow. Based on our proposed schema, voters' ability to change their vote in a given time can greatly decrease the possibility of bribery, and can prevent the Tallier from gaining early access to the election results. This protocol ultimately has managed to greatly satisfy fairness, bribery resistance, collusion resistance, and multiple casting while preserving democracy.

General Terms
Security, e-voting protocols.

Keywords
E-voting, Internet-voting, Bribery resistance, Fairness, Collusion resistance, Blind signature.

1. INTRODUCTION
The need for speed, accuracy and quality of delivered service has convinced public and private managers and decision makers to turn to new technologies for rapidly growing e-services, such as e-Government, e-Commerce, e-City, e-Voting, and etc. Although the use of eServices in an organization can provide good quality in businesses, it can also result in security problems, which must be addressed. E-voting is by far the most challenging, yet intriguing eService across countries of the world.

The rest of this paper is organized as below; Section 2 mentions the security requirements of any Internet-voting protocol. In section 3, a general model for election is described. In section 4, basic protocols used in Internet-voting protocols is mentioned and besides an extended version of blind signature protocol is proposed with a different definition which best suits to I-FOO protocol security requirements. In section 5, FOO protocol is completely investigated accompany with its advantages and disadvantages. Section 6 proposes the I-FOO protocol and phases in detail. Section 7 evaluates the I-FOO goals and compares the I-FOO with FOO and other variations of FOO protocol. Our conclusion is presented in last section.

2. INTERNET-VOTING GOALS
Internet-voting protocols offer many interesting features, however, their electronic nature, pose many security concerns that need to be addressed to guarantee the validity of election. Despite the fact that specific standards for meeting all security requirements of e-voting have not been met, many proposals in the field of e-voting **Error! Reference source not found.** emphasize the following requirements:

- ***Anonymity*****:** It is not possible for anyone to link a vote to a voter.

-***Democracy and Eligibility*****:** Only authorized voters and only one valid vote per voter is counted.

- ***Accuracy*****:** Only valid votes are counted. Altering, deleting and adding votes are not permitted.

- ***Resistance to Collusion*****:** Collusion in election is a critical matter, and in this research, we have investigated it from two novel perspectives:

- Masquerade and impersonation: electoral officers (especially those who are responsible for identification and registration) can collude to vote instead of eligible but absent voters.
- Breaching voter's anonymity: an ideal Internet-voting protocol accomplishes identification and registration processes in two completely separate phases, and shares voter's information between the voter and the other parties. According to FOO **Error! Reference source not found.** the Tallier knows a voter's vote, but does not know the voter's identity. In a different case, the Validator knows a voter's real identity, but cannot see his/her vote. The link between a vote and the voter is just in the voter's hand. Therefore, in the case of Validator and Tallier collusion, the voter should have the main say.

- ***Verifiability*****:** Verifiability can be investigated from two perspectives: individual verifiability and public verifiability. In individual verifiability, every voter can individually verify the integrity and accuracy of the tally process. In public verifiability, all citizens can verify the accuracy of the counting process.

- ***Convenience in implementation*****:** Convenience is an open problem related to protocols, which employ complicated

security mechanisms and tools. A practical way to popularize such protocols is to supply a trouble-free interfacing procedure, which requires minimum users' interaction. This method can greatly reduce voter confusion that may result in faulty procedures during the voting phases.

- ***Resumability*:** Systems should permit voters to resume voting processes from any interrupted point up to election deadlines. When the electoral officers are limited to two entities, the voter can simply understand what to do as far as he/she encounters an interruption. However, when the e-voting protocol requires more than two entities, in case of an interruption, a state diagram is required to guide the voter what to do and where to resume.

- ***Bribery Resistance and Uncoercibility*:** System should not permit voters to convince their voting behavior for the briber or coercer. In Internet-voting environments, since voters can vote from any terminal without an electoral officer's supervision, coercion and bribery are serious threats. In order to avoid coercion and bribery, different solutions are proposed. For example, in e-voting context, voters are isolated in voting kiosks, and in **Error! Reference source not found.** the Receipt-Free solution is proposed.

Likewise, we propose a new solution, which enables the voter to deceive the briber and the coercer. Based on this solution, a voter can cast a vote as many times as he/she wishes. Every time he/she cast a new vote, a new entry is added to a public bulletin board. The public bulletin board displays the vote accompany with Vote ID which is uniquely produced per vote. In this scheme, no constant information is accompanied with the vote; therefore, voter tractability based on the public bulletin board display is not possible.

- ***Fairness*:** Prior to the end of an election, voters and electoral officers cannot presume partial tabulation of the counting phase.

- ***Vote & Go*:** There is no need for voter participation in the counting phase.

- ***Mobility*:** Voters can cast their votes from any location.

- ***Communication security*:** all interactions between voter and other electoral entities are protected against man-in-the-middle and eavesdropping attacks.

- ***Robustness*:** System must be resistant against defect and failure.

3. ELECTION GENERAL MODEL

Internet-voting protocols commonly follow a general model depicted in **Error! Reference source not found.**, which depends on varying factors organized by the election and electoral officers; the manner in which they hold the elections may vary. This model presents five distinct phases:

- ***Announcement and*** *Acknowledgment*: In this phase, required protocols, list of candidates, protocol's security parameters and volume of activities are determined.

- ***Registration:*** In the registration stage, the authorities determine who is eligible to vote, and maintain proper lists of registered voters.

- ***Validation:*** When the election begins, administrators validate the credentials of registered voters.

- ***Voting and Collection:*** In this stage, the votes are collected before the final stage of the tally.

- ***Tallying:*** This final stage counts, verifies, and publishes the accumulated votes.

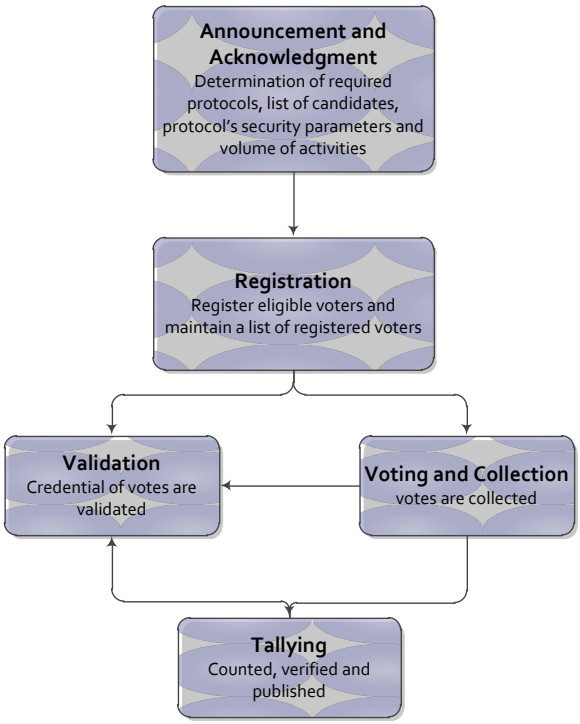

Figure 1: A general model for a voting schema (Sampigethaya and Poovendran 2006)

4. BASIC PROTOCOLS USED IN INTERNET-VOTING

The majority of Internet-voting protocols are based on cryptographic basic protocols including:

- PKI **Error! Reference source not found.**
- Mixnet **Error! Reference source not found.**
- Blind Signature **Error! Reference source not found.**
- Homomorphism cryptography model **Error! Reference source not found.**

The idea behind blind signature was first introduced by **Error! Reference source not found.**. Blind signature is a method used in verifying messages without revealing its content to the signer. In the I-FOO protocol, blind signature is used to satisfy the below scenario:

" ***A*** sends a blind message to ***B***, ***B*** blindly signs the message, adds a statement along with the message, and sends it to ***A***. This enables ***A*** to unblind and extract the message, without the ability to change the added part."

Below is a description of blind signature and the extended version.

4.1 The blind signature schema

Definition: Suppose *A* has a message *m* that she wishes to have it signed by *B* in order to send it for *C*; and she does not want *B* to learn anything about *m*. Let *(n,e)* be *B*'s public key and *(n,d)* be his private key. *A* generates a random value *r* such that $gcd(r, n) = 1$ and sends $m' = m \times r^e \pmod n$ to *B*. The value *m* is "blinded" by the random value *r*; hence,

B can derive no useful information from it. *B* returns the signed value of m'. If *A* divides signed m' received from *B* to *r*, she gains the *m* signature from *B*. The aforementioned steps for executing this protocol are shown in **Error! Reference source not found.**:

Table 1: The blind signature Schema

Mathematics	Symbolic
$A: m' = m \times r^e \pmod n$ *where e, d , n are B's PKI module* $B: s' = (m')^d \pmod n$ *where r is random number, gcd(r, n) =1* $A: s = s' \times r^{-1} \pmod n$ $=(m')^d \times r^{-1} \pmod n$ $=(m \times r^e)^d \times r^{-1} \pmod n$ $=m^d \times r^{ed} \times r^{-1} \pmod n$ *where e×d=1* $=m^d \times r \times r^{-1} \pmod n$ $=m^d \pmod n$	$A: m' = Blind(m)$ $A \rightarrow B: m'$ $B: s' = SB(m')$ $B \rightarrow A: s'$ $A: s = s'r^{-1} = SB(m)$ $A \rightarrow C : s , m$

4.2 The extended blind signature schema

Definition: Suppose *A* has a message *m* that she wishes to have it signed by *B* to send it for *C*, and she does not want *B* to learn anything about *m*. *B* also wants to send message *K* to *C* via *A*, and he doesn't want *A* to change *K*. Let *(n,e)* be *B*'s public key and *(n,d)* be his private key. *A* generates a random value *r* such that *gcd(r, n) = 1* and sends $m' = m \times r^e \pmod n$ to *B*. The value *m* is "blinded" by the random value *r*; hence *B* can derive no useful information from it. *B* returns the signed value of m'. *B* multiples m' in *K* and signs it and sends it to *A*. Alongside this message he will also send *k* signature to *A*. Since *B* doesn't know the *r* value, he gains no useful information from m'. *A* receives $m' \times k$ signature, and by dividing this value to *r* she can retrieve the signature over $m \times k$. Now, A sends $m \times k$ signature, plain m and plain *k* to *C*. *C* adds m in *k*, and by verifying the signature over $m \times k$ he can verify the message's integrity. So *B* has delivered a message to *C*, without allowing *A* to change it, and without having a direct interaction with *C*. **Error! Reference source not found.** shows the aforementioned steps for this protocol.

5. FOO PROTOCOL

As shown in Error! Reference source not found., the FOO protocol consists of three main entities:

- Voter
- Validator
- Tallier

Similar to a majority of e-voting protocols, voters are registered before the election starts; they receive a key pair, specifically designed for use in the election.

Table 2: The extended blind signature Schema

Mathematics	Symbolic
$A: m' = m \times r^e \pmod n$ *where e, d , n are B's PKI module* $B: s' = (m')^d \times k^d \pmod n$ *where r is random number, gcd(r, n) =1* $A: s = s' \times r^{-1} \pmod n$ $= (m')^d \times k^d \times r^{-1} \pmod n$ $= (m \times r^e)^d \times k^d \times r^{-1} \pmod n$ $= m^d \times r^{ed} \times k^d \times r^{-1} \pmod n$ *where e×d=1* $= m^d \times k^d \pmod n$ $= (m \times k)^d \pmod n$	$A: m' = Blind(m)$ $A \rightarrow B: m'$ $B: s' = SB(m')S_B(k)$ $B \rightarrow A: s', k$ $A: s = s'r^{-1}$ $= S_B(m \times k)$ $A \rightarrow C: s, m, k$

The private key is known to the voter alone, and must be kept confidential until the end of the election. During election, the voter encrypts his/her vote with his/her voting public key, and blinds it. Then he/she signs it with his/her voting private key, and sends it to Validator for verification. The Validator verifies the signature and confirms that the voter is among the registered list of voters. If the voter has been registered before and the signature is valid, the Validator signs the vote and sends it back to voter. The voter unblinds the vote, and forwards it to the Tallier. The Tallier verifies the signature (Validator's signature must be presented on the vote), and presents the vote on the public bulletin board. Once the voter traces his/her vote on the bulletin board, and is at peace regarding vote integrity, he/she ends up his/her voting procedure by handing the private key to the Tallier. Now, the Tallier can decrypt the vote and count it. After the completion of the election, the Tallier will publish the encrypted votes accompany with voters' private keys and tally results.

Many protocols have been designed and implemented to satisfy the aspects and principals of an actual e-voting process, including the EVOX **Error! Reference source not found.**, EVOX-MA **Error! Reference source not found.**, REVS **Error! Reference source not found.** and FOO protocols. Each of these protocols are comprised of a specified set of elements with defined relations; but they all seek to provide an environment with the utmost authentic security principals of voting in an insecure Internet environment. Among these, the FOO protocol is one, which can not only maintain the simplicity of implementation for its' optimized number of involved parties and lack of complicated interfaces, but can also meet almost all the security principles of a typical e-voting protocol. Nonetheless, despite the many advantages of FOO, especially its collusion resistance perspective, this protocol suffers from two fundamental/vital disadvantages:

- FOO is an e-voting protocol that forces voter presence at voting stations during the election, which is inconvenient for the voter.
- This protocol has not solved the problems for coercion and bribery.
- FOO cannot guarantee fairness, since the electoral entities can gain partial tabulation of tally results before election final if the voters present their private keys during collection phase.

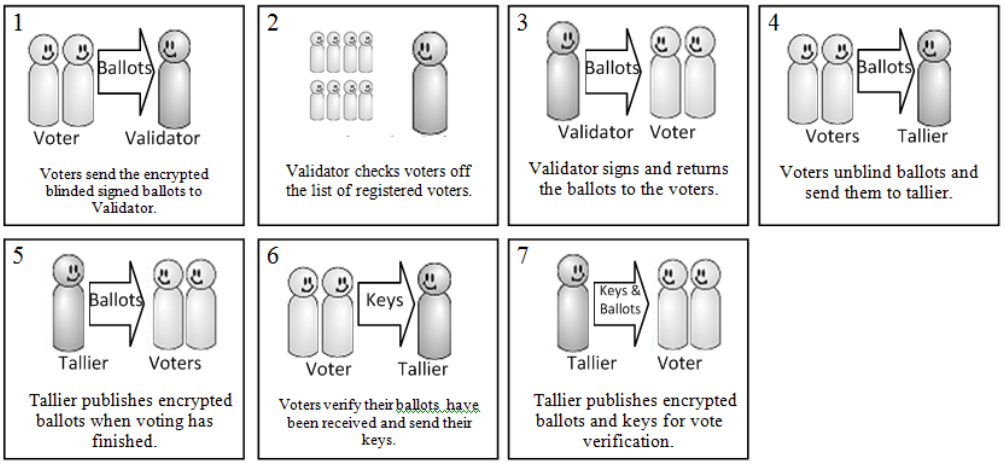

Figure 2: The FOO Protocol Error! Reference source not found.

The aforementioned disadvantages have convinced authorities not to utilize FOO for national Internet-voting. This paper will propose new solutions for FOO's disadvantages under a new secure and applicable protocol named I-FOO. In comparison to FOO, the I-FOO protocol is different in areas including encryption schemas, databases and information flow. This new protocol not only inherits the advantages of FOO, but to a great extent also offers new security principals/features, such as fairness, multiple casting and bribery resistance.

6. I-FOO PROTOCOL

I-FOO protocol has been fundamentally designed according to the FOO protocol, with certain modifications in the registration and voting phases. The applied changes mainly affect areas of data fields, databases and information flow. There are four parties in the I-FOO protocol listed below:

- Registration Authority
- Voter
- Validator
- Tallier

Notations:

R: Registration Authority
V: Validator
T: Tallier
A: Voter
NID: Citizen National Identity Number
vote
sn: A random number (unique per vote)
BB: Public Bulletin Board
***Blind*[m]**:blind signature of message *m*
$E_x(m)$: Encryption of message *m* with public Key *x*
$S_x^{-1}\{m\}$: Signature of message *m* with private key x^{-1}
vc: Number of times a voter participates in the election
r, r^{-1}: RA's public and private key pair
v, v^{-1}: Validator's public and private key pair
t, t^{-1}: Tallier's public and private key pair
a, a^{-1}: Voter's public and private key pair (unique per voter)
e, e^{-1}: Election's public and private key pair

Id_a^t: Registered Vote Identifier (ID) produced by Tallier (unique per vote)

In this protocol, the election's private key is protected inside a Hardware Security Module. This module protects the key in an offline mode, in a secure and controlled environment until the end of the election. While, this paper does not focus on ISMS related concepts such as monitoring and protecting the KMS environment's physical security and access control mechanisms, but for purposes of clarification, it is noteworthy to say that gaining access to KMS can be made possible only with the presence of a predefined and trusted number of electoral authorities and candidates. The phases are described below.

6.1 Registration Phase

In this phase, steps are followed as below:

1. A←→ R: Authentication transactions in order to ensure citizen's authenticity
2. A→R: *NID*, *Blind*(*a*)
3. R→A: $S_r^{-1}\{Blind(a)\}$

   A : Unblinds $S_r^{-1}\{Blind(a)\}$to retrieve $S_r^{-1}\{a\}$

Under the registration phase, after ensuring the voter's authenticity (for example based on fingerprint matching) in order to detect voter forgery, a voting-specific key pair

(a,a^{-1}) is generated for the voter. He/she then blinds the public key and presents the key, along with his/her *NID* to the Registration Authority. The Registration Authority identifies the voter through the received *NID*, blindly signs the voter's public key value, and then sends it back to the voter. The voter then unblinds the message to retrieve his/her public key, which is signed by the Registration Authority. In fact $S_{r^{-1}}\{a\}$ is a factor to testify voter eligibility. Since Registration Authority has received the blinded public key, it cannot extract the voter's public key to authorize the voter; as the only one with knowledge of the link between voter identity information (*NID*) and his/her key pair is the voter. Voters are registered once, and in case of suspecting a sign request with a repetitive *NID*, the Registration Authority will not respond.

In this protocol we assume that in *step 1*, some interactions between the voter and Registration Authority have been accomplished to ensure that the person who has sent his/her *NID* in *step 2*, once in the past has been successfully and strongly authenticated. Therefore, masquerade, impersonation and NID forgery in registration phase is not feasible. This strong authentication procedure can be based on voters' national/voting Smart card, fingerprint and biometric template matching or any other alternative solution.

After the completion of this phase, the Registration Authority detaches from the protocol's system, and ignores all other voter requests. The Registration Authority's database and logs will also remain hidden from other entities. Since Registration Authority is an authority responsible for not disclosing citizen's identity and personal information, we assume that it is in fact a trusted party responsible for secure sensitive information gathering, processing and storage.

6.2 Voting phase

Steps below illustrate the voting phase:

$Bvt = Blind\,[E_e(E_a(vote))]$

$vt = E_e(E_a(vote))$

4. A→V: $E_v(S_{a^{-1}}\{\text{Bvt}\},\ S_{r^{-1}}\{\text{a}\})$

5. V→A: $E_a\ (S_{v^{-1}}\{Bvt \times vc\}^1,\ vc)$
   A: *Unblinds* $S_{v^{-1}}\{Bvt \times vc\}$ *to retrieve* $S_{v^{-1}}\{vt \times vc\}$

6. A→T: $E_t(S_{v^{-1}}\{\text{vt}\times\text{vc}\}^*, \text{vt}, \text{vc}, \text{sn}, \text{OldId}_a^t, \text{Oldsn})$

7. T→A: Id_a^t

8. T→BB: $E_e\ (E_a(vote)), \text{Id}_a^t$

9. A→T: $E_t(a^{-1}, \text{Id}_a^t)$

During the voting phase:

- Initially, the voter double encrypts his/her vote with his/her own public key, a, and the election's public key, e, as:
  $E_e(E_a(vote))$
- Second he/she blinds the message as:
  $Blind\ [E_e(E_a(vote))]$
- Finally he/she signs the message as

$$S_{a^{-1}}\{Blind\ [E_e(E_a(vote))]\}$$

Along with the signed public key received from Registration Authority, $S_{a^{-1}}\{a\}$, he/she sends the whole message to the Validator. The Validator stores voter's public key accompany with *vc* into its database as shown in Table 3. In the first interaction between a voter and Validator, *vc* is set to zero. Lateran, every time this voter (recognized by his/her key, a) sends a new vote, the Validator increments *vc* once.

The Validator verifies $S_{r^{-1}}\{a\}$ and send a select query to its database to extract the *vc* with the help of voter's public key, and appends it to the signed, blinded and encrypted vote (the appending method is described in section 4-2 based on extended blind signature schema). The Validator then signs the result with v^{-1} and sends $S_{v^{-1}}$ {Blind[E_e (E_a (*vote*))]×*vc*} back to the voter. In addition, the Validator sends an update query to its database in order to increments the voter's *vc* to *vc+1*.

Since *vc* is assigned to the blinded and signed *vote* by the Validator, the voter has no control for modification. Otherwise, if the voter was responsible for delivering the *vc* to the Tallier, apart from the fact that memorizing the number of participation would be difficult for voters, a malicious voter could present *vc*=0 every time he casts a new vote. In this situation, since the Tallier does not receive any signed-unchangeable information from Validator to recognize a particular voter, he could not detect that all these votes belong to one voter. Therefore, he would consider and count all the votes received from one malicious voter as votes from different voters. This scenario could breach democracy. Based on our solution in which the *vc* is signed by the Validator, in case of *vc* modification Validator's signature will be unverifiable.

When the voter receives information from the Validator, he/she unblinds the vote, to retrieve a signed *vote* by the Validator, even though the Validator is clueless of its content. In the next step, the voter generates a random number, *sn*, and sends it to the Tallier, along with the signed *vote* (by the Validator), and the *vc*. The *vc* is in fact a criteria to help the Tallier to understand the voter is revoting or not. In this stage, if the voter is revoting, in addition to *sn* and signed *vote* by the Validator, he/she should send to the Tallier his/her previously received vote's unique ID from the Tallier, *Old*Id_a^t, accompany with the previous random number (*Oldsn*). The Tallier's stores field depicted in Table 4 into its database per voter. This information allows the Tallier to invalidate the previous vote and insert the new vote. The Tallier verifies the signature of the vote, registers the *sn*, and sends back a unique ID (Id_a^t) to the vote.

[1]the description is presented in section 3-2-2 based on extended blind signature schema.

Since Id_a^t is not encrypted, an eavesdropper can simply receive it to masquerade the voter and send a new vote to invalidate voter's previous vote. In order to prevent such attack, *Oldsn* is required which is only known to the voter and Tallier.

The Tallier can detect repetitive votes based on the *vc* parameter, which has been added to the vote by the Validator (the voter is not entitled to modify this parameter). If the voter doesn't intercalate information about his/her previous vote in his/her new vote, or add wrong information, the Tallier will not accept the new *vote* and retain the previous vote. Since, only the voter and the Tallier are aware of the random numbers, when someone sends a valid *sn*and Id_a^t, Tallier is sure that this person is in fact the voter.

After receiving *vote*, the Tallier presents theId_a^t, the encrypted vote by the voter, and the $E_e(E_a(vote))$ on the public bulletin board. The voter can trace his/her vote on the bulletin board, and upon verification$of\ E_e(E_a(vote))$he/she can present his/her voting private key $a^{-1}$to the Tallier. The Tallier cannot decrypt votes until the end of the election, because they are encrypted by the election public key either which will not be revealed until the election deadline.

It is important to consider the following principals in this phase:

- As mentioned before, in this phase if a voter wishes to change his/her vote (or simply revote), his/her previous vote will be neglected, but not totally omitted. The Information about the previous vote on the bulletin board will not be modified, instead only the new vote is added to the board. Therefore, no one except the voter and the Validator are aware of the new vote. In this way, a voter can easily deceive a briber or a coercer and convince them with fake votes.
- In this phase, exchanged information is totally encrypted with receiver's public key to protect it from eavesdropping.

The Tallier multiplies *vc* into *vt* so to confirm its information integrity. It then compares the results of the signed *vt*×*vc*, which were received from the Validator. If the verification succeeds, that means that *vc* and *vt* are trusted. If the voter or an attacker modifies *vc* or *vt*, the signature over these values will be unverifiable and the Validator will be aware about such alteration.

6.3 Tallying/counting phase

In this phase below steps are considered:

10. KMS->T: e^{-1}
11. T->BB: $E_e(E_a(vote))$, E_a (vote), vote, e^{-1}, a^{-1}, Id_a^t, sn

Once the election comes to an end, the election private key, e^{-1}, is retrieved from KMS and presented to the Tallier. The Tallier has received a voter's private key, a^{-1}, in the preceding phase, and is now able to decrypt the vote, which have been encrypted with voter's public key and the election's public key, and count them. The Tallier counts the final vote per each voter in its accumulation. After completion of tally phase, the public bulletin board is cleared out to show tally results along with all the information related to votes, including vote ID provided by the Tallier, the random number generated by the voter, the election's private key, the voter's private key specific for e-voting, the encrypted vote and the vote itself as shown in Table 5.

The sequence diagram of I-FOO protocol is depicted in Figure 3.

7. An evaluation for I-FOO protocol

Anonymity: In this protocol, the Registration Authority knows the voter's personal (registration) information, but since he receives the blinded form of voter's public key as a blinded signed message, he is unaware of the key's actual value. In this approach, the Validator confirms and validates the blinded vote based on the Registration Authority's signature over voter's public key; the Validator is able to validate that an eligible voter is voting regardless of knowing the voter's true identity and his/her vote. The Tallier also trusts the voter to receive his/her vote because of Validator's signature on the vote; he has no idea about neither the vote nor the voter's identity. Under these circumstances, collusion among Registration Authority, Validator, and Tallier cannot reveal the link between a vote and a voter's true identity. Only the voter can prove which vote belongs to him/her (since the blinding factor in his/her hand). In addition, the voter's identification based on IP addresses is an anonymity-related issue that can be prevented by employing public proxy servers as a mediator between voters and all other authorities.

***Collusion resistance*:** Is investigated under different perspectives:

- *Anonymity Collusion*: As mentioned above, in I-FOO a segmented trust-relationship is established between authorities and the voter to protect voter anonymity.
- *Masquerade Collusion*:
    - Since, in this protocol, the voter is the only entity who knows the mapping and link between delivered information to the Validator, with those delivered to the Tallier, Validator and Tallier collusion cannot result in voter masquerade (impersonation of voters who have received a signature from the Validator but have not voted yet). It's only the voter who can produce a valid statement including $S_{v^{-1}}\{vt{\times}vc\}$, *vt, vc* based on extended blind signature protocol described in section 4.2 in which the comparison of$S_{v^{-1}}\{vt{\times}vc\}$with *vt* and *vc* is valid.
    - In I-FOO protocol, contrary to FOO protocol, the Validator is unaware of the list of eligible voters, therefore cannot introduce key pairs and votes in place of eligible, but absent voters. List of eligible voters accompany with their blinded public key values is stored into Registration Authority's database while the Validator receives the voter's signed public value.

- As described in the registration phase of this protocol, our assumption is that the Registration Authority is a trusted entity who has access to the voter's identity information, and is responsible for protecting them from disclosure. In I-FOO, if the Registration Authority abuses his right, he cannot breach vote secrecy (it is only the voter who can prove his/her vote). Although, he can collude with Validator in order to masquerade on behalf of eligible, but absent voters.

Table 3: Mandatory parameters for the Validator's database

entry	Voter's public key	Participation counter
i	a_i	vc_i

Table 4: Mandatory parameters for Tallier's database

entry	Voter's private key	Vote	Participation counter	Encrypted Vote	Voter-produced random number	Tallier-produced Registered Vote Id
i	a^{-1}	$vote$	vc	$E_e(E_a(vote))$	sn	Id_a^t

Table 5: Mandatory parameters for Public bulletin Board

entry	Voter's private key	Vote	Votes' decryption key (election's private key)	Encrypted Vote	Tallier-produced Vote ID
i	a^{-1}	$vote$	e^{-1}	$E_e(E_a(vote))$	Id_a^t

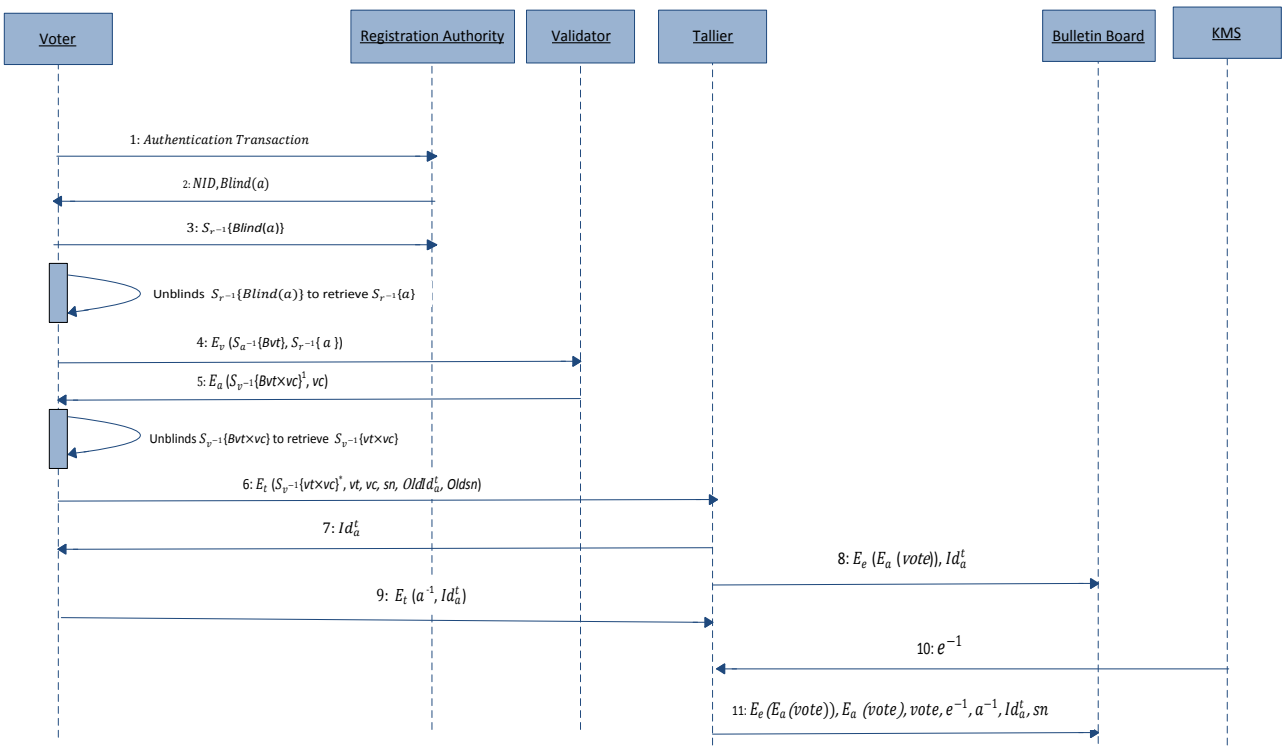

Figure3: Sequence Diagram of I-FOO protocol

***Verifiability*:** The proposed schema is designed as to enable individual and public verifiability through the aid of the information presented on bulletin board. Before the election deadline (a^{-1}, Id_a^t, $vc, E_e(E_a(vote))$, sn) are displayed on

the bulletin board. Therefore, a voter can be ensured that his/her vote has been correctly collected and received by the Tallier. When the votes are decrypted the bulletin board is refreshed to show (a^{-1}, Id_a^t, e^{-1}, $vote, E_e(E_a(vote))$). Since the Encrypted vote $E_e(E_a(vote))$, the votes' decryption keye^{-1}, voter's private key a^{-1} and the decrypted vote are all presented in the bulletin board in addition that each voter can accomplish individual verifiability, everyone can download the public bulletin board's content before and after the election deadline to verify the tally accuracy.

***Convenience in implementation*:** I-FOO entities are the same as FOO, and can be easily implemented. No special device or communication channel is required either at client or server side.

***Resumability*:** In I-FOO, since the voter interacts with only two entities during the election (Validator and Tallier) he/she can easily resume voting operations from any interrupted stage.

***Bribery Resistant*:** Because in Internet-voting protocols, voters can vote from any remote location, there is no definite solution against bribery and coercion. The voter can easily agree to sell bribers his/her vote. I-FOO has been trying to eliminate the bribery incentive with the aid of the possibility of multiple voting opportunities. Based on this approach, voters can cast their votes for as many times as they wish, and the briber cannot trace them based on the information presented on the public bulletin board (E_e (E_a(*vote*)) and Id_a^t), leaving no incentive for bribery.

Upon receiving a new *vote* from a voter, no consistent information related to him/her is displayed on the bulletin board. In fact, the voter's previous information per each received vote presented on the public bulletin board is not changed or deleted and also encrypted value of the new vote with its identifier which are changed per vote are added. Therefore, voter's electoral behavior tracking is not possible. The briber is unaware of the changes and so a voter can deceive and convince him with fake (not his/her last) vote and voting information. Only when the bulletin board is completed after the tally phase, the voter can prove his/her vote based on this information. Although, a briber can postpone payment to the end of the election, once votes have been counted, and if the briber's prospected candidate does not win the election, the briber may not pay the bribed voters any longer. Consequently, because of the voter's ability to cast their votes multiple times, voters and bribers cannot trust each other. This solution could dramatically decrease vote selling.

***Coercion*:** In Internet-voting protocols, because controls cannot be enforced on remote terminals, coercion is an open problem. In I-FOO while there is no perfect solution to prevent coercion, there is at least this possibility given to the voter to vote again freely and correct his/her previous forced vote (of course in the case of coercer absence).

***Fairness*:** In this protocol, two mechanisms are considered to prevent the Tallier from gaining access to result before election deadline:

- Since a voter can change his/her vote any time before the election final, the Tallier has to wait till the election final to decrypt and count the votes. Based on this schema tally results are not predictable even though the Tallier decrypts the votes.
- Furthermore, in I-FOO votes are encrypted by both election and voter's public keys. Based on this schema, even if the voter presents his/her voting private key before the election deadline (same as the FOO), there is no way to decrypt and count votes as the election's private key is not available before the deadline. The election's private key is kept inside a KMS(Key Management System) in a completely secure environment. The KMS access control and protection methods are basically employed based on standard ISMS controls.

***Vote & Go*:** In I-FOO the voter can present his/her voting private key as soon as he/she has voted and verified the vote on the public bulletin board. Vote tally is possible only after the election deadline.

***Mobility*:** I-FOO is an Internet-voting protocol, which contains cryptographic methods, which are trusted and can be easily employed in Internet computing environments. Based on this protocol, voters can easily, and without the need for any special equipment connect to the Internet and vote.

***Communication security*:** All sensitive information that are exchanged in I-FOO except Id_a^t are encrypted with the receiver's public key; only the receiver can decrypt messages with its private key. Channel eavesdropping, therefore cannot provide confidential information for attackers. The encryption and signed information in this protocol are quite secure that a change within the information and communication by the attacker will quickly be identified and disqualified by the receiver.

***Robustness*:** Similar to the FOO protocol, in I-FOO, failure within the entities can result in chaos within the election. To solve this problem, in addition to implementing the protocol itself, we can increase the redundancy of the Validator and Tallier servers, so in the case of server unavailability, we can resume the election and maintain individual servers.

***Accuracy*:** Is met within two mechanisms:

- To strongly identify voters prior to the election, different authentication mechanisms especially smart card-based authentication or biometric verification can be employed. In I-FOO, the Registration Authority is responsible for registering voters based on a popular and secure identification method.

Table 6: A comparison on FOO-based e-voting protocols

Issue	Our scheme	FOO scheme	Sensus	SEAS
Anonymity	Yes (Since the voter is the only entity who knows the mapping of delivered information to Validator and Registration Authority)	Yes (Anonymous channel is used)	Yes (Anonymous channel is used)	Yes (Anonymous channel is used)
Resistance to Collusion	Yes (Since only the voter knows the mapping of delivered information to Validator and Registration Authority, voters participation is also required for anonymity and masquerade collusion. Masquerade is also prevented because no list is in Validator's hand to detect absent voters)	NO (Since registration phase implementation is not mentioned, Validator may masquerade. also entities may collude to breach voter's anonymity)	NO (An entity involved in the protocol is able to masquerade. Furthermore entities may collude to breach voter's anonymity)	It is assumed that entity in charge of the Registration phase is trusted and will not collude to breach anonymity or to masquerade
Verifiability	Individual/public verifiability	individual verifiability	individual verifiability	Individual/public verifiability
Convenience in implementation	Yes	Yes	Yes	Yes
Resumability	Yes	Yes	Yes	Yes
Bribery resistance	Multiple casting ability of our schema highly Prevent both bribery and coercion	NO	NO (because of the receipt delivered to the voter, voter can prove what he voted to the briber)	NO (because of the receipt delivered to the voter, voter can prove what he voted to the briber)
Fairness	Yes	NO (depends on voters participation)	NO	NO
Vote and Go	Yes	NO	Yes	Yes
Mobility	Yes	Yes	Yes	Yes
Communication security	Yes (all communications are encrypted with receiver's public key)	NO	Yes (all communications are encrypted with receiver's public key)	Yes (all communications are encrypted with receiver's public key)
Robustness	Based on implementation	Based on implementation	Based on implementation	Based on implementation

Accuracy	Yes	Yes	Yes	Yes

- Accuracy and integrity during an election is provided by the Public key infrastructure (PKI) concepts and mechanisms.

Table 6 shows a comparison of our scheme with other variations of FOO protocol namely FOO, Sensus **Error! Reference source not found.** and SEAS **Error! Reference source not found.**.

8. Conclusion:

This paper has proposed and analyzed the I-FOO protocol; a variant of the well- known and famous FOO protocol. Our proposed protocol, in addition to retaining the excellent features of FOO, it has provided solutions to fix two major challenges of Internet-voting protocols: possibility of bribery, and possibility of unfairness. These are satisfied based on modification done at Validator and Tallier's databases, providing multiple voting opportunities to voters, and some variations in the field of information cryptography. In order to satisfy multiple casting without breaching security concerns of the protocol, a new version for blind signature, which can facilitate multiple voting abilities, is also proposed.

9. REFERENCES

[1] Fujioka, A., T. Okamoto, et al. (1993). A practical secret voting scheme for large scale elections, Springer.

[2] Chen, Y. Y., J. K. Jan, et al. (2004). "The design of a secure anonymous Internet voting system." Computers & Security 23(4): 330-337.

[3] Baiardi, F., A. Falleni, et al. (2005). "SEAS, a secure e-voting protocol: design and implementation." Computers & Security 24(8): 642-652.

[4] Fan, C. I. and W. Z. Sun (2008). "An efficient multi-receipt mechanism for uncoercible anonymous electronic voting." Mathematical and Computer Modelling 48(9-10): 1611-1627.

[5] Spycher, O ,.R. Koenig, et al. (2012). "A new approach towards coercion-resistant remote e-voting in linear time." Financial Cryptography and Data Security: 182-189.

[6] Kremer, S., M. Ryan, et al. (2011). "Election verifiability in electronic voting protocols." Computer Security–ESORICS 2010: 389-404.

[7] Benaloh, J. and D. Tuinstra (1994). Receipt-free secret-ballot elections, ACM.

[8] Chen, G., C. Wu, et al. (2008). "A New Receipt-Free Voting Scheme Based on Linkable Ring Signature for Designated Verifiers (PDF".(

[9] Chen, X., Q. Wu, et al. (2011). "New receipt-free voting scheme using double-trapdoor commitment." Information Sciences 181(8): 1493-1502.

[10] Philip, A. A. and S. A. Simon (2011). "A Receipt-free Multi-Authority E-Voting System." International Journal of Computer Applications 30(6).

[11] Sampigethaya, K. and R. Poovendran (2006). "A framework and taxonomy for comparison of electronic voting schemes." Computers & Security 25(2): 137-153.

[12] Lee, Y., S. Han, et al. (2009). Anonymous Authentication System Using Group Signature. Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on, IEEE.

[13] Chaum, D. L. (1981). "Untraceable electronic mail, return addresses, and digital pseudonyms ". Communications of the ACM 24(2): 84-90.

[14] Juels, A., D. Catalano, et al. (2005). Coercion-resistant electronic elections. Proceedings of the 2005 ACM workshop on Privacy in the electronic society, ACM.

[15] Chaum, D. (1982). Blind signatures for untraceable payments. Advances in Cryptology: Proceedings of Crypto.

[16] Benaloh, J. (1987). Verifiable secret-ballot elections, PhD thesis, Yale University.

[17] Adida, B., O. De Marneffe, et al. (2009). Electing a university president using open-audit voting: analysis of real-world use of Helios. Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections, USENIX Association.

[18] Herschberg, M. A. (1997). Secure electronic voting over the world wide web, Massachusetts Institute of Technology.

[19] DuRette, B. W. (1999). "Multiple administrators for electronic voting." Bachelor thesis, Massachusetts Institute of Technology, Boston, USA.

[20] Joaquim, R., A. Zúquete, et al. (2003). "REVS–a robust electronic voting system." IADIS International Journal of WWW/Internet 1(2): 47–63.

[21] Rosner, I. M. and G. Rosner (2002" .(Electronic Voting Protocols and Schemes." The Hebrew University of Jerusalem, Israel.

[22] Cranor, L. F. and R. K. Cytron (1997). Sensus: A security-conscious electronic polling system for the internet. System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on, IEEE.

[23] Baiardi, F., A. Falleni, et al. (2005). "SEAS, a secure e-voting protocol: design and implementation." Computers & Security 24(8): 642-652.