An ideal functionality for End-to-End Verifiable Elections

Aggelos Kiayias*

June 7, 2014

Abstract

We define the problem of realizing End-to-End Verifiable elections (E2E) in a simulation-based sense.

1 The ideal functionality expressing E2E verifiable elections

We introduce an ideal functionality capturing the operation of end-to-end verifiable election systems. The ideal functionality called $\mathcal{F}_{\mathsf{e2e}}$ recognizes and interacts with the election authority EA, the set of eligible voters $V_1, \ldots, V_n$ and the auditor AU. These are "ideal-world" entities. Note that a "real-world" implementation of $\mathcal{F}_{\mathsf{e2e}}$ may involve more parties that will enable the implementation to realize the ideal functionality.

The ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ accepts a number of commands from the election authority EA, the voters and the auditors. At the same time it informs the (ideal world) adversary of certain actions that take place and also is influenced by the adversary to perform certain actions. The ideal functionality keeps track of which parties are corrupted and may act according to their corruption status.

There are two parameters for the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$: (i) a function $f : (X \cup \{\perp\})^n \to E$ that defines the election function where X defines the set of all possible ways to vote for a single voter and E is the set of all possible election results. The notation X^n defines all possible strings of length n over the alphabet X. The symbol $\perp$ stands for "undefined." The election function f is invariant with respect to $\perp$, i.e., $f(\perp, x) = f(x)$ for all x and so on. (ii) a relation Q that defines the level of sensitivity to manipulation that $\mathcal{F}_{\mathsf{e2e}}$ may permit. In particular for two possible election results T, T' we say that $Q(T, T')$ holds if and only if T' is sufficiently close to T. For the most strict version of $\mathcal{F}_{\mathsf{e2e}}$ one will define Q to be the equality relation over E (the reader may consider only this case in a first reading).

The ideal functionality $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$ captures the following set of security characteristics:

- Provided the EA is not corrupted, the adversary is incapable of extracting the choices of the voters.

*National and Kapodistrian University of Athens, aggelos@di.uoa.gr.

- Provided the EA is not corrupted, all votes are recorded and tallied according to the election function $f(\cdot)$.
- Even in the case that EA is corrupted, a set of well defined votes are assigned to the voters of the election (however such votes may deviate from the original voters' intent). Note that the votes cannot be manipulated when the EA is honest.
- Even in case that EA is corrupted, the functionality returns consistently the same tally result to all parties that request it. Furthermore, the functionality always tests the reported tally according to the predicate Q and reports the outcome, hence any substantial (according to Q) deviation from the recorded tally will be detectable by all honest parties.
- The functionality preserves the voter intent, and in case of vote manipulation, the voter or an auditor can use the unique receipts provided in the completion of ballot-casting to test whether the original voter intent was manipulated by a corrupt EA. Any party may use those receipts and hence verification is "delegatable."

Consider now a protocol π that is implementing syntactically the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ (i.e., has the same I/O characteristics as $\mathcal{F}_{\mathsf{e2e}}$). Following standard notation and terminology we have the following :

Definition 1.1 *Let f be an election function and Q a predicate over the election results. The protocol π implements $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$ provided that for all adversaries $\mathcal{A}$, there is a simulator $\mathcal{S}$ so that for all environments $\mathcal{Z}$ it holds that*

$$\mathsf{Exec}_{\pi,\mathcal{A},\mathcal{Z}} \approx \mathsf{Exec}^{\mathcal{F}_{\mathsf{e2e}}^{f,Q}}_{\mathcal{S},\mathcal{Z}}$$

Note the protocols that will be considered in practice may utilize other simpler ideal functionalities. In such case, the protocol π implements $\mathcal{F}_{\mathsf{e2e}}$ conditional on the existence and availability of these other functionalities. Such functionalities include "authenticated channels", a "bulletin board" etc.

Party corruption. As stated, the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ enables the adversary $\mathcal{A}$ to corrupt parties by issuing special $(\texttt{Corrupt}, P)$ messages. Given such a message the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ will divulge to the adversary the complete I/O transcript from the interface between $\mathcal{F}_{\mathsf{e2e}}$ and P. We distinguish between static and adaptive corruptions. In the case of static corruptions all the messages $(\texttt{Corrupt}, P)$ should be delivered at the onset of the execution while for adaptive corruptions they are delivered at any time. For brevity we do not include explicitly the actions taken for `Corrupt` messages in the description of the functionality.

In the real-world the corruption of an entity expresses the action that is taken by the adversary that results in the complete control of the entity's computing environment. A corrupted voter, specifically, looses privacy completely and the adversary may take any action on her behalf. Specifically, if corruption happens prior to ballot-casting the adversary may vote on her behalf, while if corruption happens after ballot-casting the adversary will learn her choice. On the other hand, if the adversary corrupts the EA, it may try to manipulate some of the voter's ballots however only

up to the extend that is permitted by $\mathcal{F}_{\mathsf{e2e}}$ (no matter how many parties are corrupted $\mathcal{F}_{\mathsf{e2e}}$ always has the "upper hand").

Functionality $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$

The functionality recognizes and interacts with the following parties: (i) the election authority EA, (ii) the eligible voters $\mathcal{V} = \{V_1, \ldots, V_n\}$, (iii) the auditor AU. (iv) the adversary $\mathcal{A}$. It is parameterized by the relation Q over E and the election function $f : (X \cup \{\perp\})^n \to E$.

- Upon receiving an input $(\texttt{Create}, sid, B)$ from the EA, record the tuple so that sid is the election identifier and B is a string defining the ballot of the election. Send $(\texttt{Create}, sid, B)$ to the adversary $\mathcal{A}$.
- Upon receiving an input $(\texttt{Deliver}, sid)$ from EA, deliver (B, s_i) to each voter V_i where s_i is some voter-specific information that is provided by the adversary $\mathcal{A}$.[a]
- Upon receiving an input $(\texttt{Vote}, sid, a)$ from V_i, select a unique identifier vid and record (vid, a) provided $a \in X$. If EA is honest, notify the adversary $\mathcal{A}$ with message $(\texttt{Vote}, sid, vid)$ while if EA is corrupted send $(\texttt{Vote}, sid, vid, V_i, a)$ to $\mathcal{A}$.[b]
- Upon receiving $(\texttt{RecordVote}, sid, vid, b)$ from $\mathcal{A}$ verify that a symbol (vid, a) has been recorded before and then record the triple (V_i, a, b) provided that (i) V_i is a voter that has not been assigned any vote previously[c], (ii) the value a is a valid choice consistent with the ballot description B. (iii) the value b is unique. Finally return $(\texttt{Receipt}, b)$ to V_i.
- Upon receiving an input $(\texttt{Tally}, sid)$ from the EA, collect all recorded inputs $\{(V_j, a_j, b_j)\}_{j \in \tilde{\mathcal{V}}}$ where $\tilde{\mathcal{V}}$ is the set of voters that voted successfully and set $a_j = \perp$ for all $j \notin \tilde{\mathcal{V}}$. Compute $T = f(\langle a_1, \ldots, a_n \rangle)$ and return $(\texttt{Tally}, T)$ to $\mathcal{A}$.
- Upon receiving $(\texttt{RecordTally}, sid, \mathcal{M}, \hat{T})$ from $\mathcal{A}$, where $\mathcal{M}$ can be parsed as a polynomial-size circuit, set $\langle a'_1, \ldots, a'_n \rangle = \mathcal{M}(a_1, \ldots, a_n)$ and if $\exists j : (a'_j \neq a_j)$ and EA is honest then ignore the message. In any other case, the functionality records $(\texttt{Result}, T', \hat{T})$ and $\langle a'_1, \ldots, a'_n \rangle$ where T' is the election result calculated as $T' = f(\langle a'_1, \ldots, a'_n \rangle)$.
- Upon receiving $(\texttt{ReadTally}, sid)$ from any party, return $(\texttt{Result}, \hat{T}, Q(\hat{T}, T'))$.
- Upon receiving $(\texttt{Audit}, b)$ from from any party, recover the triple (V_j, a_j, b_j) such that $b_j = b$ and return 1 if and only if $(a'_j = a_j)$.

[a] In some systems, voters may request this information actively and hence $\mathcal{F}_{\mathsf{e2e}}$ will be passive and will not deliver the ballots. In such case the adversary will adaptively provide the s_i values.

[b] In some systems the voter identity V_i is leaked to the adversary during ballot-casting.

[c] In some systems the voter is allowed to change his/her mind and hence vote multiple times.

2 Claims regarding the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$

Claim 1 *Assuming the* EA *is not corrupted, the ideal functionality* $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$ *leaks no information about how honest voters vote, except for the information that is revealed from the partial tally of the votes of the honest voters (according to* f*).*

Claim 2 *The adversary may delay the recording of an honest voter's ballot, however when it is recorded the voter obtains a receipt that enables her to verify that her vote has been properly recorded and tallied.*

Claim 3 *The receipts the voters obtain after the vote is recorded are unique and assuming the* EA *is honest they are independent of the way the voters have voted and hence they can safely be passed around to e.g., a third party auditor* AU*.*

Claim 4 *When the* EA *is corrupted it is possible for the adversary to manipulate all the votes (via computational manipulation* $\mathcal{M}(a_1, \ldots, a_n) = (a'_1, \ldots, a'_n)$*) and even provide an incorrect tally* $\hat{T}$*; nevertheless, the ideal functionality ensures that honest parties are notified about whether the reported tally* $\hat{T}$ *and the recorded tally* T' *satisfy the relation* Q*, i.e., it returns* $Q(T', \hat{T})$*.*

3 Security notions not captured by the ideal functionality

We (intentionally) left out from this rendering of the end-to-end functionality $\mathcal{F}_{\mathsf{e2e}}$ a number of security aspects.

- Denial of service attacks. The ideal functionality as written enables the adversary to deny voters from completing the ballot-casting protocol and prevent the tally from becoming available. From a definitional point of view, expressing such level of security is feasible by assuming certain qualities of the underlying communication and message passing mechanisms that are employed in the implementation. One way to extend the functionality to capture such a setting is to oblige the adversary to deliver the (`RecordVote`) and (`RecordTally`) messages by certain deadlines. In order to do this formally, a notion of time will have to be introduced in the model. This may be achieved by introducing a global clock functionality.
- Coercion via corrupting voters. Even though $\mathcal{F}_{\mathsf{e2e}}$ does not permit coercion via the receipts it provides, the adversary may still achieve coercion by corrupting a voter (e.g., hacking into the voter's PC). If this happens after the ballot-casting protocol, $\mathcal{F}_{\mathsf{e2e}}$ reveals the choice of the voter and hence the voter can be vulnerable to coercion. Addressing this in the model is feasible by further restricting the information that is divulged when voter corruption takes place. Various intermediate levels of corruption may be considered e.g., is the voter capable of erasing some information? rewriting some information? and so on.
- Sybil attacks. The set of voters $V_1, \ldots, V_n$ is predetermined and integrated into the functionality $\mathcal{F}_{\mathsf{e2e}}$. Hence, the adversary cannot manipulate the list of voters. It follows that $\mathcal{F}_{\mathsf{e2e}}$ is applicable to the setting where the list of voters is predetermined, assumed to be public and the adversary may not tamper with it.