Surveying global verifiability

Ben Smyth

Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg, Luxembourg

Abstract

We explore global verifiability; discovering that voting systems vulnerable to attack can be proven to satisfy that security notion, whereas many secure systems cannot. We conclude that current definitions are unsuitable for the analysis of voting systems, fuelling the exploration for a suitable definition.

Keywords: Voting systems; global verifiability; provable security.

1. Introduction

Electronic voting systems for large-scale public elections place extensive trust in software and hardware. Unfortunately, instead of being trustworthy, many are vulnerable to attacks that could unduly influence election outcomes [2, 3, 4]: Trusting voting systems is unwise; proving systems can detect undue influence is essential. Küsters et al. propose *global verifiability* to determine whether a voting system is vulnerable to undue influence [5, 6, 7, 8]. Global verifiability must be instantiated with a *goal*, which is a property required to hold in system executions. For instance, goals γ_ℓ [6, §5.2] and γ [7, §6.2] are introduced as follows:

> γ_ℓ contains all runs for which there exist choices of the dishonest voters (where a choice is either to abstain or to vote for one of the candidates) such that the result obtained together with the choices made by the honest voters in this run differs only by ℓ votes from the published result (i.e. the result that can be computed from the...ballots on the bulletin board)

A preliminary version of this manuscript appeared in work with Frink & Clarkson [1].

A brief introduction to elections and voting systems

Elections are decision-making procedures to select representatives in accordance with voter choices. Voters traditionally mark choices on paper ballots and deposit them in ballot boxes, which are subsequently opened to reveal choice distributions, from which selected representatives can be determined, e.g., representatives with the highest frequency (for first-past-the-post elections). Electronic voting systems are similar: Voters encapsulate choices in digital ballots and record those ballots on bulletin boards, which are subsequently tallied to reveal choice distributions. Traditional paper-based voting systems are reliant on observers to ensure no undue influence. No such observation is possible for electronic voting systems. Instead, tallying should provide sufficient evidence for auditors to check that no undue influence has occurred.

Hence, with respect to goal γ_ℓ, global verifiability should enable the determination of whether voting systems can guarantee that honest voters' choices will be included in announced results. Goal γ should enable a similar determination.

> γ is satisfied in a run if the published result exactly reflects the actual votes of the honest voters in this run and votes of dishonest voters are distributed in some way on the candidates, possibly in a different way than how the dishonest voters actually voted

These informally stated goals are appealing, but they do not constitute rigorous mathematical definitions. As Kiayias et al. note, "[global verifiability] has the disadvantage that the set γ remains undetermined and thus the level of verifiability that is offered by the definition hinges on the proper definition of γ which may not be simple" [9, p. 476]. Küsters et al. have since updated their technical report to propose a formal goal [8, §5.2].

Contribution and structure. We explore global verifiability when instantiated with the formal goal proposed by Küsters et al. (Section 2) and discover that voting systems vulnerable to attacks can be proven secure (Section 3), whereas many secure systems cannot (Section 4). More precisely, we show that incorrect tallying cannot be detected when coins used to construct some ballots are leaked, permitting tallies that exclude or replace some choices to

go unnoticed. Moreover, we identify a class of secure voting systems that cannot be proven secure. Finally, we present a brief conclusion (Section 5).

2. Global Verifiability

Küsters et al. [5] propose a definition, called Protocols, to describe any kind of protocol, not just electronic voting protocols. Their definition is independent of any particular computational model, assuming the model provides a notion of processes. These processes must be able to perform internal computation and communicate with each other, and must define a family of probability distributions over runs, indexed by a security parameter.

2.1. Protocols

We consider the following simplified definition of Protocols.

Definition 1 (Protocol)**.** *A* Protocol *is a tuple of sets of processes* $\Pi_1, \ldots, \Pi_n$ *and processes* $\hat{\pi}_1, \ldots, \hat{\pi}_n$*, such that each process in* $\hat{\pi}_1, \ldots, \hat{\pi}_n$ *has a special output channel which no process can input on, and* $\Pi_i = \{\hat{\pi}_1\}$ *or* $\Pi_i = \Pi(\hat{\pi}_i)$ *for all* $1 \leq i \leq n$*, where* $\Pi(\pi)$ *denotes the set of all processes with input and output channels that coincide with those used by process* π*, excluding* π*'s special output channel.*

Processes $\hat{\pi}_1, \ldots, \hat{\pi}_n$ capture protocol participants, and sets of processes $\Pi_1, \ldots, \Pi_n$ capture adversarial behavior, in particular, if $\Pi_i = \{\hat{\pi}_i\}$, then an adversary following the protocol is captured. Otherwise, an adversary controlling the channels in process $\hat{\pi}_i$ is captured.

An *instance of Protocol* $(\Pi_1, \ldots, \Pi_n, \hat{\pi}_1, \ldots, \hat{\pi}_n)$ is the composition of processes $\pi_1, \ldots, \pi_n$, where $\pi_i \in \Pi_i$. Process π_i is *honest* in such an instance, if $\hat{\pi}_i = \pi_i$. Each instance of a Protocol defines a set of *runs*. We say an instance of a Protocol produces a run, if the run belongs to that set. A process is honest in a run produced by an instance of a Protocol, if the process is honest in the instance.

Comparison with the original definition. Definition 1 simplifies the original definition [5, §2] as follows. First, we omit agents, since they are only used to refer to a process's owner. Secondly, we omit the finite set of channels used by agents and we omit functions to compute the channels of a particular agent, because these sets can be derived from processes. Thirdly, we

restrict Protocols to some processes $\hat{\pi}_1, \ldots, \hat{\pi}_n$, whereas the original definition considers sets of processes $\hat{\Pi}_1, \ldots, \hat{\Pi}_n$. Finally, we require a stronger assumption on the sets of processes: we require $\Pi_i = \{\hat{\pi}_1\}$ or $\Pi_i = \Pi(\hat{\pi}_i)$, whereas the original definition requires $\Pi_i \subseteq \Pi(\hat{\pi}_i)$. These simplifications narrow the original definition.

Beyond simplifications, our definition modifies the original in two ways. First, we forbid the sets of processes from using special channels. This restriction does not appear in the original, but it is necessary to ensure that global verifiability is satisfiable by interesting protocols: A Protocol is *not* globally verifiable (with respect to a goal), if the Protocol produces a run that does not achieve the goal, but is nevertheless accepted. Given that acceptance is captured by outputting on special channels and the original definition permits the adversary to output on such channels, global verifiability is unsatisfiable for interesting protocols. Insisting that $\Pi(\pi)$ excludes π's special output channel suffices to overcome this problem.

The second modification permits channels to be shared between processes. That is, we drop the implicit assumption that communication is authenticated, and we permit broadcast channels, which is necessary to ensure a realistic adversary model. By comparison, the original definition prohibits communication between a process in Π_i and a process in Π_j, when process $\hat{\pi}_i$ cannot input (respectively output) on a public channel that process $\hat{\pi}_j$ can output (respectively input) on. Consequently, the original definition of global verifiability cannot detect some attacks. For instance, given a Protocol $P = (..., \hat{\pi})$, let $\mathsf{Accept}(P) = (..., \hat{\pi}')$ such that process $\hat{\pi}'$ awaits input on a channel that is not used by any other process in P and if such an input is received, then the process outputs on $\hat{\pi}$'s special channel, otherwise, the process executes $\hat{\pi}$. Hence, $\mathsf{Accept}(P)$ accepts all runs that input on the public channel introduced by Accept. Thus, $\mathsf{Accept}(P)$ should not satisfy any definition of verifiability. Yet, the adversary model prohibits input on the channel introduced by Accept, therefore, Protocols P and $\mathsf{Accept}(P)$ are identical from the adversary's perspective. It follows that: given a Protocol $P = (..., \hat{\pi})$ and goal γ of P, such that γ is globally verifiable by $\hat{\pi}$, it holds that γ is globally verifiable by $\mathsf{Accept}(P)$. This problem can be overcome by assuming a single, shared broadcast channel between all processes.

Our simplifications narrow the original definition to ease understanding and our two modifications ensure that global verifiability is satisfiable by interesting protocols and that a realistic adversary model is captured.

2.2. Security definition

A *goal* of a Protocol is a subset of the sets of runs produced by instances of the Protocol. Processes can accept runs by outputting on their special channels. Global verifiability is intended to ensure that processes only accept runs when the goal has been achieved in those runs. We consider the following simplified definition of global verifiability.

Definition 2 (Global verifiability)**.** *Given a Protocol P, goal γ of P, and process $\hat{\pi}$ of P, we say γ is* globally verifiable by $\hat{\pi}$*, if for all instances Λ of P parameterized by k, there exists a negligible function μ such that for all security parameters k and (efficient, i.e., polynomial time) runs r of Λ that include an output on $\hat{\pi}$'s special channel, we have $r \notin \gamma$, with probability less than or equal to $\mu(k)$.*

Our simplified definition refines the original definition by incorporating our simplified syntax and considering a tighter security bound. Moreover, we require that runs are efficient. (This is necessary to ensure that global verifiability is satisfiable by interesting protocols.) Finally, we omit the definition's notion of Completeness for brevity.

2.3. Goal γ_{GV} by Küsters et al.

We consider a simplified case of a goal proposed by Küsters et al. [8, §5.2].

Definition 3. *Suppose r is a run of some instance of a Protocol. Let n_h be the number of honest voters in r and $\beta_1, \ldots, \beta_{n_h}$ be the choices of honest voters in r. Let n_d be the number of dishonest voters in r. We say that* we are satisfied with *r, if a tally is published in r and that tally contains $n_d + n_h$ choices including $\beta_1, \ldots, \beta_{n_h}$.*

Given a Protocol, we define γ_{GV} as the following set of runs: for all instances Λ of the Protocol and for each run r produced by Λ, we include r in γ_{GV}, if we are satisfied with r.

Our simplified definition is a special case of the original: Set γ_{GV} contains runs in which no choices of honest voters may be excluded from the tally. (We remark that Küsters et al. only define when to be satisfied with run r and do not define γ_{GV} as a set. Nonetheless, we believe our definition captures their intent.) Hence, goal γ_{GV} is a more formal presentation of goal γ_l for $l = 0$.

3. Proving vulnerable systems secure

We show that incorrect tallying cannot be detected when coins used to construct some ballots are leaked, permitting tallies that exclude or replace some choices to go unnoticed. Indeed, from a Protocol with ballots that do not leak coins, we design two Protocols that cannot detect incorrect tallies when coins are leaked:

Replace choices. Let Replace be derived from a Protocol by modifying the auditor's process as follows: The process checks whether a tally and a bulletin board appear in a run, and the bulletin board is a set $\{b_1, \ldots, b_m, (\beta_1, \beta'_1, r_1), \ldots, (\beta_\ell, \beta'_\ell, r_\ell)\}$ such that $b_1, \ldots, b_\ell$ are ballots for choices $\beta_1, \ldots, \beta_\ell$, constructed using coins $r_1, \ldots, r_\ell$. If so, the process runs the original auditor's process after replacing choices $\beta'_1, \ldots, \beta'_\ell$ with $\beta_1, \ldots, \beta_\ell$ in the tally (provided as input to that original process), otherwise, the process runs the original auditor's process (without modifying the tally). Finally, the process outputs on its special channel if the original auditor process outputs on its.

Intuitively, Protocol Replace defines an auditor process that checks whether coins have been leaked and permits acceptance of invalid tallies if they have. More precisely, if the underlying auditor process would accept a tally, then the new auditor process will accept that tally after replacing choices $\beta_1, \ldots, \beta_\ell$ (cast in ballots $b_1, \ldots, b_\ell$) with choices $\beta'_1, \ldots, \beta'_\ell$.

The following Protocol is a variant of the former, whereby choices are dropped, rather than replaced.

Drop choices. Let Drop be a variant of Replace that adds choices $\beta_1, \ldots, \beta_\ell$ to the tally provided as input to the original auditor process (hence, those choices are dropped from the tally of the modified auditor process).

Global verifiability fails to detect vulnerabilities in the above Protocols when instantiated with the goal by Küsters et al. (Theorems 1 & 2):

Theorem 1. *Suppose a Protocol does not leak coins used to construct ballots and γ_{GV} is globally verifiable by the Protocol's auditor process, we have γ_{GV} is globally verifiable by the modified auditor process defined by Protocol* Replace.

Proof. Suppose (to the contrary) that γ_{GV} is *not* globally verifiable by the modified auditor process, hence, there exists an instance of Protocol Replace such that for all negligible functions μ there exists a security parameter k and

run $r \notin \gamma_{GV}$ of the instance that includes an output on the special channel belonging the modified auditor process, with probability greater than $\mu(k)$. By definition of goal γ_{GV}, either no tally is published in run r or the run publishes a tally that does not contain $n_d + n_h$ choices including $\beta_1, \ldots, \beta_{n_h}$, where n_h is the number of honest voters, $\beta_1, \ldots, \beta_{n_h}$ are the choices of honest voters, and n_d is the number of dishonest voters.

Both the original and modified Protocols are equivalent, unless the bulletin board is a set $\{b_1, \ldots, b_m, (\alpha_1, \alpha'_1, r_1), \ldots, (\alpha_\ell, \alpha'_\ell, r_\ell)\}$ such that $b_1, \ldots, b_\ell$ are ballots for choices $\alpha_1, \ldots, \alpha_\ell$, constructed using coins $r_1, \ldots, r_\ell$. We proceed by consideration of that case. (The other case is uninteresting: The Protocols are equivalent, which permits an immediate conclusion by contradiction.) By definition of the modified auditor process, the original auditor process must accept a variant of run r wherein the tally is updated by replacing choices $\alpha'_1, \ldots, \alpha'_\ell$ with $\alpha_1, \ldots, \alpha_\ell$. Since γ_{GV} is globally verifiable by the original auditor process, that tally contains $n_d + n_h$ choices including the choices of honest voters, namely, $\beta_1, \ldots, \beta_{n_h}$. By comparison, the tally in run r does not. Since the tallies contain the same number of choices, there exists an honest choice $\beta \in \{\beta_1, \ldots, \beta_{n_h}\}$ which is replaced by a distinct choice $\alpha \in \{\alpha'_1, \ldots, \alpha'_\ell\}$. Suppose that honest choice belongs to a process representing an honest voter. Since the original Protocol does not leak coins used to construct ballots, it follows that coins by the honest voter cannot appear on bulletin board, thereby deriving a contradiction and concluding our proof. □

Theorem 2. *Suppose a Protocol does not leak coins used to construct ballots and γ_{GV} is globally verifiable by the Protocol's auditor process, we have γ_{GV} is globally verifiable by the modified auditor process defined by Protocol* Drop.

A proof of our theorem can be constructed on the basis that coins used to construct ballots are not leaked. That idea has already been demonstrated in our proof of Theorem 1, so we omit a formal proof.

Vulnerabilities are missed because (honest voters') coins cannot be leaked, even when the software, hardware, voter, etc. that selected those coins has the ability to leak them. One could argue that our analysis is at fault. A proponent can claim that detection requires analysts to explicitly model leaks. We must object. Disregarding choices encapsulated in well-formed ballots is never tolerable. A definition that does not detect omissions is unacceptable; undue influence in election outcomes must *always* be detectable. Ultimately, a voting system branded "verifiable," having satisfied a suitable definition,

must always enable detection of undue influence, otherwise we invite cheating.

Beyond exclusion and replacement of honest voters' choices, spurious bulletin board entries can also be used to swing tallies.

Swing. Let Swing be a variant of Replace wherein choices are only replaced in the tally when they will cause a swing in the election outcome.

Intuitively, Protocol Swing defines an auditor process that accepts swung outcomes, permitting undue influence. More precisely, if the underlying auditor process would accept a tally and the corresponding election outcome can be swung, then the new auditor process will accept a swing. For instance, the auditor will accept tallies that swing outcomes in the adversary's favour. For example, suppose two honest voters favour one choice, the other favours another, as does the adversary. Further suppose the adversary tolerates the two honest voters' choice. When the adversary represents one voter, an election should result in a tie. Yet, the auditor will accept tallies favouring the adversary's tolerated choice, when the adversary includes a spurious bulletin board entry comprising their primary choice, their tolerated choice, and the coins used to construct the ballot encapsulating their primary choice. (The adversary is free to reveal their coins and the attack does not depend upon honest voters' coins leaking.) Global verifiability fails to detect such influence when instantiated with the goal by Küesters et al., because their definition permits bulletin board entries of dishonest voters to be interpreted for multiple candidates.

Global verifiability similarly fails to detect exclusions, replacements, and swings when instantiated with the goal by Cortier et al. [10]. We omit recalling further details, because the ideas remain the same.

Our results cast doubt over the security of voting systems proven to satisfy definitions of global verifiability, and establishing their security is a possible direction for future research.

4. Unsatisfiable by secure voting systems

In essence, the formal goal by Küsters et al. is satisfied in a run if choices $\beta_1, \ldots, \beta_{n_h}$ of honest voters are included in the tally and the tally contains $n_h + n_d$ choices, where n_d is the number of dishonest voters. We found that many voting systems do not satisfy global verifiability with this goal,

because the goal requires: 1) participation of all voters, 2) ballot posting to always succeed, and 3) bulletin boards not to drop, inject, nor modify ballots. The first and second requirements define availability properties, which an adversary can disrupt. The third can be disrupted by an adversary that controls the bulletin board. Thus, there exist runs of many voting systems that cannot satisfy the goal by Küsters et al., and we formally show that a non-participating (honest or dishonest) voter violates the goal.

Theorem 3. *Suppose an instance of a Protocol produces a run wherein*

- *an empty bulletin board and an empty tally are announced,*
- *an auditor accepts the empty tally and bulletin board by outputting on its special channel, and*
- *there exists one voter,*

we have γ_{GV} is not global verifiable by the auditor.

The aforementioned run captures scenarios in which a voter does not participate, ballot posting fails, or the bulletin board drops a ballot. Given that secure voting systems need only detect such scenarios, rather than preclude them, many well-known, secure voting systems cannot be proven to satisfy global verifiability, including (patched) Helios [11, 12, 13, 14, 1] and the system by Juels, Catalano & Jakobsson [15]. Detection of such scenarios, rather than preclusion, may seem counter-intuitive, because attacks should surely be prevented. Yet, prevention is not always possible. For example, an adversary can refuse to announce the result or simply announce an incorrect result. Such behaviour can be detected, but not prevented.

Proof. The auditor rightly accepts, since the empty tally correctly results from tallying the empty bulletin board. However, goal γ_{GV} teaches us to expect a tally containing all honest voters' choices and a choice for each dishonest voter. Given that the empty tally contains no choices, we have γ_{GV} is unsatisfied in r, hence, $r \notin \gamma_{GV}$, but, nonetheless, the auditor outputs on its special channel, concluding our proof. □

Cortier et al. [10, §10.2] propose a variant of the goal by Küsters et al. Their goal is informally claimed to permit some honest voters' choices to be dropped from the tally, which would intuitively address problems associated with the requirement that bulletin boards do not drop, inject, nor modify

ballots. However, this claim is not supported by their formally stated goal, because that goal requires the tally to include $n_h + n_d$ choices, where n_h, respectively n_d, is the number of honest, respectively dishonest, voters. Thus, the goals by Cortier et al. and Küsters et al. have similar drawbacks. We omit recalling further details, because the ideas remain the same.

5. Conclusion

Global verifiability is proven to be satisfied by voting systems vulnerable to attack, whereas many secure systems cannot satisfy global verifiability, including (patched) Helios and the system by Juels, Catalano & Jakobsson. Global verifiability has not been adequately formalised. Use of current definitions must cease; the exploration for a suitable definition must begin.

Acknowledgements. I am grateful to Michael R. Clarkson, Steven Frink and anonymous reviewers for insightful commentary and discussions that have helped improve this manuscript, and to the Luxembourg National Research Fund (FNR) for financial support, under the FNR-INTER-VoteVerif project (10415467).

References

[1] B. Smyth, S. Frink, M. R. Clarkson, Election Verifiability: Cryptographic Definitions and an Analysis of Helios, Helios-C, and JCJ (2019).

[2] T. Kohno, A. Stubblefield, A. D. Rubin, D. S. Wallach, Analysis of an Electronic Voting System, in: S&P'04: 25th Security and Privacy Symposium, IEEE Computer Society, 2004, pp. 27–40.

[3] S. Wolchok, E. Wustrow, J. A. Halderman, H. K. Prasad, A. Kankipati, S. K. Sakhamuri, V. Yagati, R. Gonggrijp, Security Analysis of India's Electronic Voting Machines, in: CCS'10: 17th ACM Conference on Computer and Communications Security, ACM Press, 2010, pp. 1–14.

[4] D. W. Jones, B. Simons, Broken Ballots: Will Your Vote Count?, Vol. 204 of CSLI Lecture Notes, Center for the Study of Language and Information, Stanford University, 2012.

[5] R. Küsters, T. Truderung, A. Vogt, Accountability: Definition and relationship to verifiability, in: CCS'10: 17th ACM Conference on Computer and Communications Security, ACM Press, 2010, pp. 526–535.

[6] R. Küsters, T. Truderung, A. Vogt, Verifiability, Privacy, and Coercion-Resistance: New Insights from a Case Study, in: S&P'11: 32nd IEEE Symposium on Security and Privacy, IEEE Computer Society, 2011, pp. 538–553.

[7] R. Küsters, T. Truderung, A. Vogt, Clash Attacks on the Verifiability of E-Voting Systems, in: S&P'12: 33rd IEEE Symposium on Security and Privacy, IEEE Computer Society, 2012, pp. 395–409.

[8] R. Küsters, T. Truderung, A. Vogt, Accountability: Definition and relationship to verifiability, Cryptology ePrint Archive, Report 2010/236 (version 20150202:163211) (2015).

[9] A. Kiayias, T. Zacharias, B. Zhang, End-to-end verifiable elections in the standard model, in: EUROCRYPT'15: 34th International Conference on the Theory and Applications of Cryptographic Techniques, Vol. 9057 of LNCS, Springer, 2015, pp. 468–498.

[10] V. Cortier, D. Galindo, R. Küsters, J. Müller, T. Truderung, SoK: Verifiability Notions for E-Voting Protocols, in: S&P'16: 37th IEEE Symposium on Security and Privacy, IEEE Computer Society, 2016, pp. 779–798.

[11] B. Adida, O. Marneffe, O. Pereira, J. Quisquater, Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios, in: EVT/WOTE'09: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, USENIX Association, 2009.

[12] V. Cortier, B. Smyth, Attacking and fixing Helios: An analysis of ballot secrecy, Journal of Computer Security 21 (1) (2013) 89–148.

[13] D. Bernhard, O. Pereira, B. Warinschi, How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios, in: ASIACRYPT'12: 18th International Conference on the Theory and Application of Cryptology and Information Security, Vol. 7658 of LNCS, Springer, 2012, pp. 626–643.

[14] N. Chang-Fong, A. Essex, The Cloudier Side of Cryptographic End-to-end Verifiable Voting: A Security Analysis of Helios, in: ACSAC'16: 32nd Annual Conference on Computer Security Applications, ACM Press, 2016, pp. 324–335.

[15] A. Juels, D. Catalano, M. Jakobsson, Coercion-Resistant Electronic Elections, in: D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. Ryan (Eds.), Towards Trustworthy Elections: New Directions in Electronic Voting, Vol. 6000 of LNCS, Springer, 2010, pp. 37–63.