Breaking Bad: The Subtle Challenges of Insider Threats

Elizabeth A. Quaglia[1] and Peter Y A Ryan[2]

[1] Royal Holloway, University of London
[2] University of Luxembourg and
iTrust Abstractions Lab Luxembourg

Abstract. Usable security research has long emphasised that users are not the enemy: many security failures arise from poor system design rather than user incompetence. However, while rejecting user-blaming is essential, security analysis must also account for the possibility that some users may act with malicious intent. In cryptography, this is formalised through the notion of the insider attacker, a participant with legitimate access, elevated privileges and trusted status and therefore greater power than an external adversary. In this paper, we argue that insider threats remain insufficiently examined in the context of electronic voting (e-voting). Through a series of case studies, we analyse three insider scenarios: passive non-engagement that undermines security guarantees, active malicious behaviour capable of disputing an election's legitimacy, and coercion of insiders by external actors. We show how these cases expose subtle design vulnerabilities that are often overlooked when threat models focus primarily on outsiders. We conclude that securing e-voting systems requires careful design that addresses insider risk while preserving public trust, as legitimacy is as crucial as technical correctness in democratic processes.

1 Introduction

A foundational insight in usable security, articulated most prominently in Angela Sasse's work [1], is that users are not the enemy. Security failures often arise not from user incompetence, but from poorly designed systems that misalign incentives, overload cognition, or impose unrealistic behavioural demands. Blaming users obscures structural weaknesses and discourages better design. There are circumstances, however, in which users may act with malicious intent and deliberately attempt to subvert or sabotage a system. To ensure meaningful protection for honest users, such possibilities must also be accounted for in the security design.

The idea that the user could be the intentional enemy is conceptualised in information security with the notion of an inside attacker, or *insider*, and many security properties and models have been developed in order to capture this kind of attacker, who is indeed more powerful than an outsider as it has access to additional data, has various privileges and enjoys trusted status.

In this paper we argue that when considering the concepts of users and enemies of a system, it is important not to overlook the subtle threats posed by an insider, and we offer a series of case studies in the context of e-voting that highlights precisely the risks when this occurs.

In particular, we offer insights into how the insider can a) by simply not engaging lead to fundamental security properties not being met[3], b) be malicious and bring the whole election into dispute, c) be coerced and therefore controlled by a malicious entity (either outsider or another insider).

We shall see how, in each of these scenarios, careful design is required to deliver security without eroding trust. A loss of confidence would undermine the system as a whole, leading honest participants to lose faith in the process, a risk that recent political developments have underscored all too clearly.

Another insight of this work is to remark that, as we shall see, it is good to enable users/voters to contribute to the security, i.e., passively by monitoring the system but also actively by contributing entropy etc., but this must be done with great care. To this end, we propose the following guidelines:

- Keep the HCI as simple as possible, ideally linear.
- Avoid the system security being dependent on user behaviour, and consider if it is possible to avoid user involvement at all, i.e., to base the security on other sources of trust.
- Good accountability/dispute resolution is essential to counter potential user abuses.

The first is illustrated in Section 2.1 by the case of Benaloh challenges: complexity and user disengagement can give rise to issues and attacks. The second is illustrated by the reliance on individual checks in "conventional" E2E V systems: if insufficient voters perform the "individual" audits the election cannot be deemed to be verified. The outsourcing individual checks case study in Section 2.4 shows how to avoid such reliance for E2E V systems. The third is illustrated again by Benaloh challenges, but also the case studies in Sections 2.2 and 2.3 (Caveat Coercitor and the tracker-based systems), and the discussion around coercion resistance (Section 3).

2 With users like this, who needs enemies?

Enabling users to contribute to the system security is an excellent goal, especially for voting systems, but we must bear in mind that not all users will be competent and trustworthy all the time. For many security and safety-critical systems there

[3] This case refers to a user that does not care, rather than being actively malicious.

are contexts in which the goals of some users do not align with those of the system. In particular, for secure voting systems some voters may be motivated to try to undermine the goals of the system: to run a free and fair election and deliver a credible outcome. One might suppose at first glance that voters should share this goal, and indeed most will [4], but some may have other ideas. For example, some may fear, based perhaps on polls, that the outcome will not be to their liking and so will seek to disrupt or discredit the election. We have seen this with Trump's claims of a "stolen election" [17], but also many other contested elections around the world. Some might not like the election system and so seek to discredit it.

The above are threats to the system integrity, but voters may also be motivated to undermine other properties, for example vote privacy. They might seek to violate the privacy of other voters in order to coerce them, or even their own vote privacy in order to be able to sell their vote.

Below we will examine more closely ways that voters might be motivated and have the capability to undermine the goals of a voting system, in particular an End-to-end verifiable (E2E V) system.

In response to concerns about the trustworthiness of computer-based voting systems, security researchers and cryptographers have been exploring the notion of E2E V systems. The goal is avoid dependence on the correctness of system implementation, in particular the code running on the devices and servers. This is not a fanciful or theoretical concern: in the US in particular many voting "solutions" were deployed with proprietary code that was not available for expert, independent analysis. When access to the code of several voting systems was ordered by the Secretary of State for California for the Top to Bottom Report it was found to be riddled with bad security practice, e.g., hardwired crypto keys etc., [5].

The E2E V approach is one of runtime monitoring rather than system verification: as the system executes an election it generates sufficient evidence to prove or disprove the correctness of the outcome. An E2E V should have the property that if any (legitimately cast) vote is corrupted from the point of casting to the point at which it is counted then this will be detectable on examination of the evidence laid down during execution.

The standard approach to E2E V is for the vote to be encrypted, for this encryption to be posted to the BB and for these encryptions to be later processed to extract the outcome while preserving vote privacy [5]. Voters are encouraged to check the correctness of the encryption of their vote and the presence of their encrypted ballot in the BB. Giving voters the opportunity to catch any error or cheating in the casting and recording of their vote is laudable, making them the basis for trust, but in practice it has drawbacks:

– Most voters are disinclined to perform the checks and prefer to just vote and go.

[4] One would hope the majority otherwise the very notion of democracy is undermined.

[5] Vote privacy is typically achieved by using either anonymising mixes or homomorphic tabulation.

- The motivation for these checks is not so easy for voters and stakeholders to understand.
- Asking voters to perform such checks complicates the voting ceremony and so impacts usability and acceptability.
- If an insufficient number of voters perform the checks with sufficient diligence, then the election cannot be deemed to have been verified.

Many E2E V schemes have been proposed over the years (a comprehensive survey can be found at [8]), and have mainly focused on providing so-called "individual" and "universal" verifiability. The former, cast-as-intended and recorded-as-cast, are checks that must be performed by the voter and concerns their vote. The last concerns the tallying, i.e., the set of votes, and can be performed by any observer and typically involves auditing evidence posted on the Bulletin Board (BB).

The distinction is important for what follows: universal checks are mathematical, e.g., computations, digital signature or zero-knowledge proof verifications, and as such they can be double, triple...checked by anyone. Furthermore, assuming the immutability of the BB, these checks can be performed at any time. This means that any challenge can readily and publicly be examined and dismissed if fake. Contrast this with individual checks, which typically can only be performed by the voter whose vote it concerns, and often must be performed in a specific time interval.

First of all, the voter could decide *not* to perform these checks. However sometimes the system's verifiability relies on these checks to happen, leading to some tensions (cf. discussion in Case Study III and IV in Sections 2.3 and 2.4). There is also the case in which the voter might claim to have detected an error and for many systems it is hard to determine if it is a real system fault or voter who is mistaken or lying. Such issues are referred to as *dispute resolution* and systems which enable users to monitor the system's behaviour are particularly prone to them, as we illustrate in Case Study I in Section 2.1. Finally, a voter could be forced into making specific choices by a malicious party, leading to difficult to navigate situations as explored in Case Study II in 2.2.

2.1 Case Study I: Benaloh Challenges

Benaloh challenges [2] illustrate the issue of dispute resolution very clearly. They seek to assure the voter that their vote is correctly encrypted, cast-as-intended assurance, by allowing the voter to audit the encryption of their vote. However, auditing an encryption undermines its secrecy, so it is usually thought that an audited ballot cannot subsequently be cast. This leads to the adoption of a cut-and-choose (C&C) style protocols, which can be either in parallel or sequential.

Benaloh challenges are a sequential form of C&C: The voter enters their selection into a device that commits to an encryption, typically in printed form. The voter now gets a choice to either audit or cast. If they chose to audit the encryption is opened, for example by revealing the randomisation, allowing the

voter to check that the revealed plaintext agrees with their selection and complain if these do not match. After a ballot audit a new encryption is committed and again the voter has the choice: audit or cast, and so on until the voter is convinced that the device is honest and they opt to cast the ballot. The idea is that the encryption device should not be able to predict when the voter will audit and so stands a chance of being caught if it tries to cheat in a particular round. If the voter does audit and cry foul, the difficulty is that only the voter should know what they input and consequently it is hard to determine if it is the system or the voter at fault. If a significant number of voters make false challenges then the integrity of the system is called into question, even if it has in fact behaved impeccably.

A possible counter to the above issue is for the voter to *commit* to a vote, by say filling out a paper ballot, and inputting the vote via a scanner. The commitment can then be used to support a challenge. However, done naively, this undermines a notion of privacy known as *receipt freeness* [12], which prevents attacks such as vote buying. A possible counter is for the device to retain the ballot and if the voter opts to cast it is fed into a ballot box, if the voter opts to audit it is released back to the voter. But now we hit a second order dispute scenario: the device is cheating and the voter presses the audit button but the device casts the vote, claiming that the voter pressed "cast". In [3] a counter to this last issue is proposed by introducing a further layer of dispute resolution by requiring the voter to commit, in secret sealed in an envelope, to their cast/audit choice.

This layering of dispute resolution challenges illustrates the complexity and subtlety of the voter as an insider threat, particularly when their goal is to undermine the election by casting doubt on its verifiability.

2.2 Case Study II: Caveat Coercitor

Caveat Coercitor [7] is designed to make a system coercion evident rather than (fully) coercion resistant. The justification for such a weakening of the usual property is that it is difficult to achieve full verifiability, usability and coercion-resistance simultaneously, and so it seems necessary to weaken one of these. Verifiability and usability seem to be non-negotiable, so the compromise falls on coercion-resistance.

The scheme is based on the JCJ approach [10] of providing voters with deniable credentials, but with the twist that if two ballots are cast for different candidates with the same valid credential both are nullified, and this is flagged on the BB. This makes the extent of coercion publicly visible, and the authorities could announce in advance that if coercion exceeds a given threshold then the election will be nullified. The idea is to discourage attempts to coerce voters. It also has the pleasing feature that the coercion evasion strategy for the voter is simply to cast their vote as normal, no additional steps needed. It even detects so-called "silent coercion" where the attacker has somehow got hold of the voter's credential and casts a vote using it. In such a situation the voter may

not even be aware that their vote is nullified but the occurrence of an instance of manipulation is nonetheless flagged by the protocol.

The upshot is that a coerced voter may in effect loose their right to vote, possibly without even knowing it. Full coercion resistance requires that a voter be able to cast their intended vote without the coercer being able to detect this, i.e., while appearing to comply.

We will skip the details here, the interested reader can consult the citation for more details. The key point here is that the protocol, while having desirable features, opens up possibilities for hostile voters to attack the system: it is easy to fake being coerced and very difficult to distinguish fake from real without undermining vote privacy.

2.3 Cast Study III: Tracker-based E2EV

To date, aside from some one-off trials, there has been little uptake of E2E V systems. This prompts the question: why? The above points doubtless contribute. Furthermore, election officials presumably do not take kindly to any implication that they might not be completely competent or honest [6].

All of this suggests the need to rethink the approach and an alternative avenue is to use private trackers. Each voter gets a unique, private tracker. After tabulation the votes are posted alongside the associated tracker allowing each voter to check that their vote appears correctly, in plaintext, in the tally. This has the appeal of being very transparent and intuitive, but done naively this will of course introduce coercion threats: the coercer or vote-buyer demands that the voter reveal their tracker. Selene and Hyperion, [16], [6], introduce coercion mitigation mechanisms. Intuitively, voters do not learn their tracker until after the tally has been posted, giving a coerced voter the opportunity to identify a tracker that points to the coercer's choice and claim it as their own.

A set of trackers are first posted to the BB and each element of the set is ElGamal encrypted. The resulting set of encrypted trackers $\{\tau_j\}_{PK_{EA}}$ is then put through anonymising, verified re-encryption mixes and then paired off with voter IDs. As long as the original set of trackers are pairwise distinct the shuffles guarantee that each voter will get a unique tracker, and the voter/tracker link is secret.

Tracker notification is over a private channel and takes the form of a secondary key which, combined with a trapdoor key known only to the voter reveals the tracker. To evade a determined coercer, who demands not just the tracker but also the two keys, the voter can with their trapdoor key compute an alternative secondary key that will reveal the appropriate fake tracker. Such a computation is however infeasible without the trapdoor key, so an attacker cannot mislead the voter as to their real tracker. Thus, if the secondary key

[6] Even though this is not actually the point.The point is rather that, were someone to have the audacity to accuse them, an E2E V system would provide them with the evidence to refute any such accusations.

reveals a valid tracker, an element of the set posted during the setup phase, the voter can be confident that this is indeed their assigned tracker.

This gives voters the ability to check that their vote is correctly tallied as cast in one intuitive step. Voters thus no longer need to be able to identify their encrypted ballot on the BB and so we do not need a mechanism for the voter to identify a re-encryption of their ballot.

This is appealing but has at least two down-sides:

- It is susceptible to disputes: voters claim that the vote besides their tracker is not what they cast.
- The checks occur late in the process, after the tally has been posted, so recovery can be tricky and may undermine vote privacy.

Designing a tracker-based scheme with good dispute resolution while retaining the direct and intuitive nature of the checks remains an open challenge.

2.4 Case Study IV: Outsourcing the Voter Checks

It is generally thought that cast-as-intended (CaI) checks must be performed by the voter. At first glance, given that only the voter should know what vote they input to the encryption, this seems plausible. In fact, as argued in [15], protocols can be designed such that CaI checks can be performed by independent observers and not confined to the voter.

In schemes where the votes are encrypted on-the-fly, e.g., the voter inputs the vote to an encryption device or server, then indeed it seems inescapable that the voter must be involved in the audit. However, schemes exist in which the ballots are pre-prepared, for example, Prêt à Voter [14]. Here a ballot comprises an encryption of each voting option, and the voter simply selects the appropriate encryption [7]. Ballot auditing now reduces to verifying the well-formedness of the pre-prepared ballots and this is independent of the voter or vote and can be performed by anyone.

The in-person scheme in [15] takes this further and combines this observation with some additional primitives and mechanisms. Using Signatures over Randomisable Ciphertexts (SoCR), [4], each ballot pack can take the form of SoCRs, denoted SignM over trivial encryptions w.r.t the Election Authority public key PK_{EA} of each of the options, each printed on a separate slip of paper:

$$\begin{aligned}
&\mathsf{SignM}(sk_j, \{\text{Candidate}_1 \| \text{ElecID}; r = 0\}_{PK_{EA}}),\\
&\mathsf{SignM}(sk_j, \{\text{Candidate}_2 \| \text{ElecID}; r = 0\}_{PK_{EA}}),\\
&\quad\vdots\\
&\mathsf{SignM}(sk_j, \{\text{Candidate}_n \| \text{ElecID}; r = 0\}_{PK_{EA}}),
\end{aligned}$$

ElecID is an election ID ensuring that the ballots cannot be reused across elections, and the random coin r is trivial in each encryption. Each ballot pack

[7] More exactly, in Prêt à Voter each ballot shows an independently randomised permutation of the options and what is encrypted in the permutation.

has its own unique signing key, sk_j. Using trivial encryptions means that the voter sees the voting options in plaintext, resulting in greater transparency and usability.

Skipping the details, the voter registers, takes a ballot pack at random and proceeds to the booth. In the booth the voter presents the signed encryption of candidate of choice to a scanner attached to the device on the booth. The booth device has limited connectivity, just a scanner for input and a printer for output and limited capabilities: it can re-encrypt a ciphertext and transform the SoRC accordingly. The device prints off a copy of the transformed encryption and signature. The voter drops all the original slips from the ballot pack into an audit box and takes printed output back to the registration desk. The signature is checked by observers and officials and if valid the ballot is posted to the BB.

This gives a scheme where there is no dependence on voters to perform ballot audits. All the voter needs to do is present the correct signed encryption to the booth device, and only the one. For cast-as-intended assurance we do need to assume that the SoRC signatures are not leaked, that auditors are honest and competent and that the voter presents only the one SoRC to the booth device. These assumptions seem justifiable and a reasonable trade-off to achieve greater usability and understandability and to remove dependence on voters performing the checks.

Cast as intended assurance follows from the well-formedness of the pre-prepared ballot packs, i.e., that they are syntactically correct with valid signatures over the correct candidates etc. Ballot packs can be independently audited before, during and after the election. The contents of the audit boxes will also be audited ensuring that all ballots including cast ballots are subject to audit-no cut-and-choose necessary.

In this scheme we are not reliant on sufficient voters performing the CaI checks; an election run with this scheme will be verified even if no voters perform their checks. We are placing trust in the auditors but this is pretty robust: any ballot can be audited by any observer and audits are clear cut, essentially a signature validity check, so any disputes are readily resolved.

A companion paper to [15] proposes techniques to outsource the recorded-as-cast checks, [11]. Given that E2E V ballots are always encrypted in some way, it is safe from a privacy perspective to keep additional copies of cast ballots. As long as these are well curated they can be used by observers to check consistency with what is posted to the BB. Taken together with the the CaI techniques sketched above we see that all "individual" checks can be outsourced.

A possible attack on E2E V schemes is for coercers to persuade voters that they can deduce votes from encrypted ballots, [13]. In a well-designed and implemented scheme such a claim is false, but if some voters fall for it, or even just fear that it might be true, they may still be coerced. A possible counter is to take the outsourcing of RaC checks to the extreme: do not provide voters with a copy of their ballot, so that they cannot perform the check for their ballot, and rely fully on observers. This runs counter to the usual philosophy of enabling users/voters to contribute to the security but does mitigate such psychological

attacks. Whether on balance it is better to go to such extremes is debatable and probably depends on the threat context etc. This does however illustrate that there may be situations in which it is both possible and preferable to make the security mechanisms entirely transparent to the users.

3 Dispute Resolution vs Coercion-Resistance

By this point, it should be apparent that designing a secure voting system requires reconciling fundamentally conflicting requirements.

Two properties that are particularly in tension are dispute resolution and coercion resistance, especially in relation to cast-as-intended verification. This tension is most acute in online voting schemes. On the one hand, we require strong evidence of what occurred in order to resolve disputes; on the other, voters must retain the ability to equivocate about how they voted in order to preserve coercion resistance.

One way to understand this challenge is to observe that a voter must be able to ensure that their device cast the intended vote even though the device itself may be compromised and attempt to alter the ballot undetectably. In effect, the voter must be able to "coerce" their own device to behave honestly, and yet at the same time, we must prevent an external coercer from coercing the voter. We therefore have two co-existing, similar scenarios: a voter seeking to control a potentially malicious device, and a coercer seeking control over the voter. The players, interfaces, and incentives overlap significantly. The design challenge lies in enabling coercion in the first setting while preventing it in the second.

4 Conclusions

Insider threats are not merely abstract categories within formal security definitions; they expose subtle and often underappreciated vulnerabilities in real-world systems. E2E-V voting systems provide a compelling example of this.

While involving voters in verification is a laudable and democratically appealing design choice, the model can fail when participation in the checks is low. Malicious insiders may exploit the auditing mechanisms to cast doubt on legitimate outcomes, and fabricated claims of irregularities can be difficult to refute convincingly. At the same time, strengthening dispute resolution procedures may conflict with advanced privacy guarantees such as receipt-freeness and coercion resistance. These tensions highlight the need to rethink current approaches to insider modelling, verifiability, and user engagement.

Future work must also examine how well these properties are understood by voters in practice (undertaking a scoping study similar to [9]). Ultimately, secure e-voting must deliver not only accurate results, but outcomes that are collectively trusted and accepted by the electorate (and the candidates).

Acknowledgments. Elizabeth A. Quaglia is an Emmy Noether Fellow and would like to thank the London Mathematical Society.

References

1. Adams, A., Sasse, M.A.: Users are not the enemy. Commun. ACM **42**(12), 40–46 (1999). https://doi.org/10.1145/322796.322806, https://doi.org/10.1145/322796.322806
2. Benaloh, J.: Simple verifiable elections. In: 2006 Electronic Voting Technology Workshop, EVT '06. USENIX Association (2006)
3. Benaloh, J., Naehrig, M., Pereira, O.: The DROP protocol: Dispute resolution via observation in public for verifiable, in-person voting. Cryptology ePrint Archive, Paper 2025/929 (2025), https://eprint.iacr.org/2025/929
4. Blazy, O., Fuchsbauer, G., Pointcheval, D., Vergnaud, D.: Signatures on Randomizable Ciphertexts. In: Gennaro, R. (ed.) LNCS - Lecture Notes in Computer Science. LNCS - Lecture Notes in Computer Science, vol. 6571, pp. 403–422. Springer, Taormina, Italy (Mar 2011). https://doi.org/10.1007/978-3-642-19379-8_25, https://inria.hal.science/inria-00542643
5. D Wagner et al.: Top to bottom review (2025), https://www.sos.ca.gov/elections/ovsta/frequently-requested-information/top-bottom-review
6. Damodaran, A., Rastikian, S., Rønne, P.B., Ryan, P.Y.A.: Hyperion: Transparent end-to-end verifiable voting with coercion mitigation. In: ESORICS 2025. p. 124–143. Springer-Verlag, Berlin, Heidelberg (2025)
7. Grewal, G.S., Ryan, M.D., Bursuc, S., Ryan, P.Y.: Caveat coercitor: Coercion-evidence in electronic voting. In: 2013 IEEE Symposium on Security and Privacy. pp. 367–381 (2013). https://doi.org/10.1109/SP.2013.32
8. Hao, F., Ryan, P.Y.A.: Real-world electronic voting: Design, analysis and deployment. CRC Press (2016)
9. Herbert, F., Becker, S., Schaewitz, L., Hielscher, J., Kowalewski, M., Sasse, M.A., Acar, Y., Dürmuth, M.: A world full of privacy and security (mis)conceptions? findings of a representative survey in 12 countries. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). pp. 582:1–582:23. ACM (2023). https://doi.org/10.1145/3544548.3581410
10. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Towards Trustworthy Elections, pp. 37–63. Springer (2010)
11. P Y A Ryan: Towards universally verified recorded as cast. To appear in Advances in Verifiable Voting 2026 (2026)
12. Ryan, M., Delaune, S., Kremer, S.: Coercion-Resistance and Receipt-Freeness in Electronic Voting . In: 19th IEEE Computer Security Foundations Workshop. pp. 28–42. IEEE Computer Society (2006). https://doi.org/10.1109/CSFW.2006.8
13. Ryan, P.Y.A.: Verifiable Encrypted Paper Audit Trails. Newcastle University, CS Tech Reports 1024 (2006)
14. Ryan, P.Y.A., Bismark, D., Heather, J., Schneider, S., Xia, Z.: Prêt à voter: a voter-verifiable voting system. IEEE Transactions on Information Forensics and Security **4**(4), 662–673 (2009). https://doi.org/10.1109/TIFS.2009.2033233
15. Ryan, P.Y.A., Roenne, P.B., Arriaga1, A., Pereira, O.: End-to-end verifiable elections with casting of publicly audited, cleartext ballots. In: E-Vote-ID 2025, LNI Proceedings (to appear) (2025)
16. Ryan, P.Y.A., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: International Conference on Financial Cryptography and Data Security. pp. 176–192. Springer (2016)
17. Wikipedia contributors: Trump fake electors plot. https://en.wikipedia.org/wiki/Trump_fake_electors_plot (2025), accessed: 14 February 2026