Conversation with Gemini

Authors: on 12/7/2025

-Can you judge the correctness of this statement about the Estonian i-voting system IVXV?:
"*Could lead to a kind of massive vote buying/coercion that is never possible even in poll station e-voting; voters could be forced to copy a vote they do not even know. Imagine a scenario where an adversary controls "n" voters with a malicious VA' as in DARKDAOs (using execution attests to assure a certain code is executed, which is the malicious voting application VA' in this scenario); the adversary either perform an auction between corrupted candidates, or even worse works for another country, then sends an encrypted choice $(g^r, y^r.v)$ to VA' to copy. If you think you can search the anonymous ballots for suspected repetitions, VA' could add a fresh randomness to each vote $(g^{r+r\_new}, y^{r+r\_new}.v)$.*"
-Is the original statement accurate?
-Is the attack really feasible on large-scale?
-Do smart contracts facilitate generating and sending the encrypted vote online whether auctioned between competitors or from another adversarial country?

Gemini

Here's my plan to tackle that topic. Let me know if you need to make changes.

(1) Research the fundamental challenges and vulnerabilities associated with vote buying and coercion in various e-voting system architectures, including remote and in-person models.
(2) Investigate the concept of malicious voting applications (VA') and explore how decentralized autonomous organizations (DAOs) or similar structures could be exploited to facilitate large-scale vote manipulation, such as in 'DARKDAOs'.
(3) Analyze the role of execution attestation in securing e-voting systems and how an adversary might subvert or misuse it to ensure the execution of a malicious voting application (VA') without detection.
(4) Examine cryptographic protocols and system designs aimed at preventing or detecting vote buying and coercion in anonymous and encrypted e-voting schemes.
(5) Evaluate the impact of adding fresh randomness to encrypted votes on the ability to detect coerced or repeated votes, and its implications for the overall auditability and integrity of the election.
(6) Compare the susceptibility of different e-voting models (e.g., internet voting, blockchain-based voting, kiosk voting) to the specific type of attack scenario described, focusing on the client-side vulnerabilities.
(7) Identify proposed technical and procedural countermeasures, including cryptographic techniques and secure software development practices, designed to mitigate the risks of malicious client-side applications and subverted attestation in e-voting.
(8) Synthesize the findings to provide a comprehensive analysis of the feasibility and potential impact of the described vote buying/coercion scenario, along with a summary of effective prevention and detection strategies.

<u>Answer:</u>

You've outlined a highly sophisticated and concerning attack scenario, which indeed combines several advanced techniques to subvert e-voting integrity. Let's break down each part of your question.

1 Is the original statement accurate?

Yes, the core elements of your statement accurately describe plausible and significant threats to modern e-voting systems, particularly those involving remote or client-side interactions.

1. **"Could lead to a kind of massive vote buying/coercion that is never possible even in poll station e-voting; voters could be forced to copy a vote they do not even know."**
   o **Vote Buying/Coercion:** The primary defense against vote buying and coercion in e-voting is "receipt-freeness," which ensures a voter cannot prove how they voted to a third party. "Coercion-resistance" is a stronger property that aims to make it infeasible for an adversary to determine if a coerced voter complied, even if they are forced to abstain or vote randomly. Remote voting in uncontrolled environments (like personal devices) inherently presents a higher risk of fraud, coercion, and vote-selling compared to in-person voting, where poll workers can provide oversight. So, while some forms of coercion might be possible in physical polling stations, the scale and ease of such attacks are significantly amplified in remote e-voting.
   o **Forced to copy a vote they do not even know:** This is a direct consequence of malicious client-side applications. Malware on a voter's device can indeed alter ballot selections and display a fake user interface (UI) to the voter, making them believe their intended vote was cast, while the actual ballot transmitted contains the attacker's manipulated choices. This manipulation occurs before the vote leaves the voter's device, making it difficult for the election system to detect.
2. **"Imagine a scenario where an adversary controls "n" voters with a malicious VA' as in DARKDAOs (using execution attests to assure a certain code is executed, which is the malicious voting application VA' in this scenario); the adversary either perform an auction between corrupted candidates, or even worse works for another country, then sends an encrypted choice (gr , yr.v) to VA' to copy."**
   o **Adversary controls voters with a malicious VA':** This is a well-documented threat. Malicious applications can gain control over a voter's device, alter votes, steal authentication data, and impersonate users. Such malware can be distributed massively, for instance, through exploiting security flaws or infected emails.
   o **"as in DARKDAOs":** While "DARKDAO" specifically refers to a historical crypto hack involving a malicious child DAO, the underlying concept of a compromised decentralized entity or a malicious dApp is relevant. Malicious dApps can disguise themselves as legitimate services to commit fraud, execute unauthorized transactions, and steal private keys. Malware has been used to compromise hardware wallets of DAO contributors, leading to malicious transaction injection.

- o **"using execution attests to assure a certain code is executed, which is the malicious voting application VA' in this scenario":** This points to a critical vulnerability in the misuse or subversion of attestation. Trusted Execution Environments (TEEs) and attestation are designed to verify the integrity of code and data. However, TEEs can be paradoxically repurposed to facilitate vote selling through "identity lease" services, where the TEE acts as a "simulated voter" to verify that a vote was cast, thereby undermining traditional coercion-resistance mechanisms. Furthermore, attestation itself is not infallible; attackers can intercept and spoof attestation verdicts locally, reuse valid tokens from healthy devices, or even forge verdicts using leaked Original Equipment Manufacturer (OEM) keys. Vulnerabilities in the Trusted Computing Base (TCB) or side-channel attacks can also compromise the integrity guarantees of attestation. Thus, an adversary could potentially exploit or bypass attestation mechanisms to make their malicious VA' appear legitimate, or to verify that a coerced vote was indeed cast.
- o **"adversary either perform an auction between corrupted candidates, or even worse works for another country, then sends an encrypted choice (gr , yr.v) to VA' to copy":** Smart contracts can be used to implement auction logic. While direct "auctions between corrupted candidates" for votes in a general election are not a commonly documented smart contract use case, the underlying mechanisms for illicit payments and vote manipulation via smart contracts are plausible. State-sponsored actors are a significant threat to election systems , and have been involved in leveraging blockchain technology for illicit financial activities. The concept of sending an encrypted choice to be copied by the VA' is consistent with the use of homomorphic encryption, which allows computations on encrypted data, such as tallying votes, without decrypting individual ballots.
3. **"If you think you can search the anonymous ballots for suspected repetitions, VA' could add a fresh randomness to each vote $(g^{r+r\_new}, y^{r+r\_new}.v)$."**
   - o This is accurate. Cryptographic schemes used in e-voting, particularly homomorphic encryption, rely on adding randomness to encrypted votes to ensure privacy and prevent linking a vote to a voter. However, this property can be exploited. With fully homomorphic encryption and its "bootstrapping" operations, an attacker can re-randomize duplicated ballots "beyond anything one can detect," making it extremely difficult to identify copied or crafted ballots through traditional auditing methods. This is a known challenge in balancing privacy and auditability in e-voting systems.

2 Is the attack really feasible on large-scale?

Yes, the attack described is highly feasible on a large scale, particularly against remote e-voting systems that lack robust client-side security and comprehensive end-to-end verifiability.

- **Pervasive Client-Side Malware:** Malware can be distributed massively and automatically, for example, by exploiting security flaws in operating systems or by sending infected emails. Internet voting inherently introduces vulnerabilities that can be exploited by adversaries. A single vulnerability in an online voting solution can lead to

large-scale manipulations of votes. Studies on mobile voting applications like Voatz have shown that attackers with root access to a device can alter or expose a user's vote, and that the system's network protocol can leak vote details.

- **Scalability of Coercion/Vote Buying:** The "identity lease" concept, which leverages TEEs, demonstrates a mechanism for large-scale, deniable vote selling by allowing a trusted environment (the TEE) to simulate a voter and verify that a vote was cast. This makes traditional coercion-resistance less effective.
- **Attestation Subversion at Scale:** The widespread issue of leaked OEM keys for forging attestation verdicts and the challenges in keeping Trusted Computing Bases (TCBs) up-to-date mean that an adversary could potentially bypass attestation checks across a significant number of devices, making their malicious VA' appear legitimate.
- **Hiding Fraud with Randomness:** The ability of malicious applications to add fresh randomness to encrypted votes, especially when homomorphic encryption is used, makes it extremely difficult to detect duplicate or manipulated votes through auditing. This inherent challenge in auditing randomized encrypted ballots contributes to the feasibility of large-scale, undetectable fraud. While Risk-Limiting Audits (RLAs) are designed to detect miscounts, they typically rely on voter-verifiable paper records, which may not be present in purely electronic systems.
- **Overall Risk:** The consensus among election security experts is that online voting, due to its inherent risks, could lead to "undetectable, nation-scale election failures".

3 Do smart contracts facilitate generating and sending the encrypted vote online whether auctioned between competitors or from another adversarial country?

Yes, smart contracts can indeed facilitate various aspects of such a sophisticated attack, including the generation and transmission of encrypted votes, and potentially the coordination of illicit payments.

- **Smart Contracts for Voting:** Smart contracts are fundamental to blockchain-based voting systems, managing processes like voter authentication, vote recording, and tallying. They can define governance rules and automate decisions.
- **Generating and Sending Encrypted Votes:** Blockchain-based voting systems often integrate cryptographic techniques like homomorphic encryption with smart contracts to ensure vote privacy and integrity. Votes are typically encrypted before being cast and recorded on the blockchain. Smart contracts can handle these encrypted transactions.
- **Auctioning Votes/Illicit Payments:** Smart contracts can be programmed to implement auction logic. While the research doesn't explicitly detail "auctions between corrupted candidates" for votes in a general election, smart contracts can enforce financial incentives for voter participation, such as requiring deposits that are refunded upon casting a vote. This mechanism could theoretically be adapted for illicit vote-buying schemes, where the smart contract facilitates the payment upon "verification" of a coerced vote (e.g., through a subverted attestation mechanism). Smart contracts are designed to execute automatically once conditions are met, making them suitable for coordinating such illicit transactions without human intermediaries.
- **Coordination by Adversarial Countries:** State-sponsored actors have demonstrated capabilities in leveraging blockchain technology for illicit financial activities, including

stealing virtual currency using fraudulently obtained identities. While direct evidence of smart contracts being used by adversarial countries for large-scale vote manipulation in e-voting is not explicitly detailed, the ability of smart contracts to automate complex, transparent, and immutable transactions, combined with state-level interest in cyber warfare and election interference, makes them a plausible tool for coordinating and executing such schemes. The transparent and immutable nature of blockchain, while beneficial for legitimate uses, could also be leveraged by adversaries to ensure "proof" of illicit transactions if not properly secured.

In conclusion, your scenario outlines a highly sophisticated and technically informed attack that leverages known vulnerabilities and properties of e-voting systems, client-side security, TEEs, and blockchain technology. The feasibility of such an attack on a large scale is a significant concern, underscoring the complex challenges in securing modern elections.

Sources and related content

What to Do If You Connect to a Malicious dApp: Step-by-step Guide | Tangem Blog

tangem.com

Auction Smart Contracts: A Deep Dive | by Lea Lobanov | Coinmonks | Medium

medium.com

How does malware spread? Top 5 ways malware gets into your network - Emsisoft

www.emsisoft.com

Going from bad to worse: from Internet voting to blockchain voting | Journal of Cybersecurity

academic.oup.com

Secure Voting Website Using Ethereum and Smart Contracts - MDPI

www.mdpi.com