# Conversation with Grok about Feasibility and Risk Level of Encrypted Copy Attack

Other sources:

Since *grok*'s answer is pretty long, here are *Google Gemini* and *Microsoft Copilot* shorter answers to the same question:
(remember we just seek confirmation that our judgment is right, we don't let them do the job for us)

**Gemini (12/7/2025)**

"*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*"
https://anonymous.4open.science/r/SoK_Estonia_IVXV_EVoteID-C2E4/Google_Gemini_encrypted_attack.pdf

**Copilot (8/7/2025)**

*"The statement is **correct in its technical reasoning and threat modeling**. It reflects a nuanced understanding of how coercion and vote buying could scale in remote settings—especially when adversaries leverage cryptographic primitives and client-side control. The IVXV system, while robust in many respects, does not currently offer **strong coercion-resistance guarantees** against such advanced client-side attacks."*

They used 3 references, the 3rd of which we didn't know before; the 2022 PhD now in many places in my paper referenced as [32].

---

Outline:

On 7/7/2025, we asked grok ….

**-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* $\boldsymbol{(g^r, y^r.v)}$ *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* $\boldsymbol{(g^{r+r\_new}, y^{r+r\_new}.v)}$."

**The final conclusion** (reached in 3 steps)

*The statement is* ***correct*** *and reinforced by your clarifications and Willemson's article. IVXV's lack of mandatory voting application authentication enables a*

*malicious VA' to facilitate covert vote buying/coercion, as demonstrated by the 2023 observer's test and the Pereira attack.*
*-The attack's* *__feasibility__* *is heightened by the low verification rate, reliance on voter diligence, and the absence of system-enforced client authentication, making "massive" manipulation plausible against sophisticated adversaries.*
*-__Unlike poll station__ e-voting, where voters know their choice, IVXV's remote nature allows votes to be cast for unknown candidates, supporting the statement's claim.*
*-Willemson's defense underestimates __the threat of large-scale adversaries in high-stakes elections__, and the lack of "easy solutions" like signature verification exacerbates the risk. Implementing your suggested measures (e.g., digital signatures, checksum integration) would significantly mitigate this vulnerability, aligning with recommendations from OSCE/ODIHR () and academic analyses (,).*

*For further details, see the IVXV documentation (https://valimised.ee) or Willemson's article (https://arxiv.org/html/2502.06385v2).*

**__Full Details:__**

**-Grok**
To assess the correctness of the statement about the Estonian i-voting system (IVXV) potentially enabling "massive vote buying/coercion" that is not possible in poll station e-voting, we need to break down the claim, evaluate it against the design and security features of IVXV, and consider the specific scenario described involving a malicious voting application (VA'). The statement raises concerns about vulnerabilities to coercion and vote buying, particularly through a hypothetical attack involving a malicious client application. Let's analyze this systematically.Understanding the IVXV System and Its Anti-Coercion MeasuresThe Estonian IVXV i-voting system, operational since 2005, is designed to provide secure, verifiable, and coercion-resistant online voting. Key features relevant to this discussion include:

- __Multiple Voting__: To mitigate coercion and vote buying, IVXV allows voters to cast multiple votes during the advance voting period, with only the last vote counted. Additionally, a vote cast in person at a polling station overrides any online votes. This mechanism aims to allow voters to change their vote if coerced, reducing the effectiveness of coercion or vote buying.
- __End-to-End Verifiability__: Voters can verify their vote within 30 minutes of casting it using a separate device (e.g., a smartphone app scanning a QR code). This verification ensures that the vote recorded matches the voter's intent, though only about 4-5.3% of voters use this feature.

- Encryption and Anonymity: IVXV uses ElGamal homomorphic encryption to encrypt votes, which are digitally signed by the voter (akin to a double-envelope system). Mix-nets shuffle votes to ensure anonymity before decryption, and non-interactive zero-knowledge (NIZK) proofs are used to verify correct decryption without revealing voter identities.
- Voter Authentication: Votes are cast using a national ID card or mobile ID, requiring a PIN for authentication. This ensures only eligible voters can vote, though the security of the client device is critical.
- Auditability and Transparency: The system includes audit mechanisms, such as vote registration services and decryption proof verifiers, to detect tampering. The source code is publicly available for review, and independent audits have been conducted.

Analyzing the Statement's Scenario; The statement describes a hypothetical attack where an adversary controls "n" voters' devices via a malicious voting application (VA') that uses "execution attests" (likely referring to trusted execution environments or attestation mechanisms) to ensure a specific malicious code is executed. The adversary could:

- Conduct an auction among corrupted candidates to buy votes.
- Work for a foreign entity to manipulate votes.
- Send an encrypted vote (using ElGamal encryption: $(g^r , y^r.v)$ to VA', which voters copy unknowingly.
- Add fresh randomness to each vote $(g^{r+r\_new} , y^{r+r\_new}.v)$ to avoid detection of repeated ballots.

The claim suggests this enables "massive vote buying/coercion" not possible in poll station e-voting, as voters could be forced to cast votes they are unaware of, and the anonymity of ballots prevents detection of such manipulation. Evaluating the Scenario Let's assess the feasibility and correctness of this attack in the context of IVXV's design:

- Malicious Voting Application (VA'):
    - The scenario assumes the adversary compromises the voter's device to run a malicious voting application (VA') that casts a predetermined vote without the voter's knowledge. This is a plausible threat, as IVXV relies on the voter's device (e.g., a desktop PC) being secure. If malware infects the device, it could intercept the voter's authentication (ID card or mobile ID) and cast a vote.
    - Research has shown vulnerabilities in IVXV's client-side security. For example, a 2014 analysis by Springall et al. demonstrated that a compromised client device could allow attackers to alter votes or undermine election integrity undetected.

- The "***execution attests***" mentioned likely refer to mechanisms ensuring the malicious code runs as intended, possibly leveraging trusted execution environments (e.g., Intel SGX). However, IVXV's current client application does not use such attestation, so this would require a sophisticated attack to replace the official client with a malicious one.

- <u>Vote Buying/Coercion:</u>
  - <u>Vote Buying</u>: The statement suggests an adversary could orchestrate vote buying by controlling votes via VA'. In traditional vote buying, a voter is bribed to vote a certain way and may need to prove their vote. IVXV's multiple-voting feature and in-person override reduce the reliability of vote buying, as voters can change their vote later. However, if VA' casts votes without the voter's knowledge, the voter might not realize the need to revote, enabling undetected vote buying.
  - <u>Coercion:</u> Coercion involves forcing a voter to vote a certain way, often under observation. IVXV's design assumes voters can revote privately later, but if VA' operates covertly, the voter may not know their vote was manipulated, undermining the anti-coercion mechanism. This is a valid concern, as remote voting lacks the controlled environment of a polling station, where coercion is harder to conceal.

- <u>Encrypted Vote Copying:</u>
  - The statement describes an adversary sending an encrypted vote $(g^r , y^r.v)$ (standard ElGamal encryption, where ( g ) is a generator, ( r ) is a random value, ( y ) is the public key, and ( v ) is the vote) to VA', which the compromised device casts. This implies the voter's device signs and submits a precomputed ciphertext without the voter's input.
  - IVXV's use of digital signatures requires the voter's private key (via ID card or mobile ID) to sign the vote. If VA' compromises the device and accesses the private key, it could indeed sign and submit a malicious vote. This is a known vulnerability in systems reliant on client-side security.

- <u>Adding Randomness to Avoid Detection:</u>
  - The statement claims VA' could add fresh randomness ***r_new*** to each vote to avoid detection of identical ballots. In ElGamal encryption, re-randomizing a ciphertext $(g^r, y^r.v) \rightarrow (g^{r+r\_new} , y^{r+r\_new}.v)$ is possible without decrypting it, preserving the vote ( v ). This would make it difficult to detect repeated votes by looking for identical ciphertexts, as each vote would appear unique.
  - However, IVXV's mix-net and decryption process would still count these votes correctly, as the randomness does not alter the plaintext vote. The

issue lies not in detection of duplicates but in the fact that the voter is unaware of the manipulation.

- Comparison to Poll Station E-Voting:
    - The statement claims this attack is “never possible” in poll station e-voting. In poll station e-voting (e.g., using direct recording electronic machines), voters interact with a controlled device in a secure environment, reducing the risk of a malicious application. Physical oversight and procedural controls (e.g., poll workers, observers) make it harder for an adversary to install malware or manipulate votes en masse without detection.
    - In contrast, IVXV’s remote voting occurs on voters’ personal devices, which are outside the election authority’s control and vulnerable to malware. This makes the described attack more feasible in i-voting than in poll station e-voting, supporting the statement’s claim.
- Scale of the Attack (“Massive”):
    - The term “massive” implies a large-scale attack affecting many voters. For this to occur, the adversary would need to compromise a significant number of devices, which is challenging but not impossible (e.g., through widespread malware distribution). The 2014 analysis by Springall et al. noted that state-sponsored actors or insiders could target election servers or client devices, potentially enabling large-scale manipulation.
    - The DARKDAOs reference suggests a decentralized, coordinated attack (akin to a decentralized autonomous organization). While theoretically possible, this would require significant resources and coordination, likely beyond typical vote-buying schemes but plausible for a nation-state actor.

## Correctness of the Statement

The statement is partially correct but requires nuance:

- Correct Aspects:
    - Client-Side Vulnerability: The scenario of a malicious voting application (VA') controlling a voter’s device is a realistic threat. IVXV’s reliance on the security of personal devices makes it vulnerable to malware that could cast votes without the voter’s knowledge.
    - Coercion and Vote Buying: The multiple-voting feature mitigates traditional coercion and vote buying, but a covert malicious application could bypass this by casting votes without the voter’s awareness, undermining the anti-coercion mechanism.

    - Randomized Votes: Adding fresh randomness to ElGamal ciphertexts is feasible and would make it difficult to detect identical votes, enhancing the attack's stealth.
    - Comparison to Poll Stations: The attack is indeed more feasible in remote i-voting than in poll station e-voting, where physical oversight and controlled devices reduce such risks.
- Incorrect or Exaggerated Aspects:
    - "***Never Possible***" in Poll Station E-Voting: While less likely, poll station e-voting systems are not immune to manipulation (e.g., compromised voting machines or insider attacks). The statement overstates the absolute security of poll station systems.
    - "***Massive***" Scale: The term "massive" is vague. While the attack is theoretically scalable, compromising a large number of devices requires significant resources, likely limiting it to sophisticated actors (e.g., nation-states). Smaller-scale vote buying or coercion is more plausible but still serious.
    - ***Detection Challenges***: The statement implies that detecting such an attack is impossible due to anonymized ballots. While true that re-randomized votes are hard to detect as duplicates, IVXV's audit mechanisms (e.g., vote registration service, NIZK proofs) could detect discrepancies if votes are added or removed, though not necessarily if they are replaced covertly.

Mitigations and Counterarguments

IVXV includes features to mitigate such risks, though they are not foolproof:

- Verification: The 30-minute verification window allows voters to check their vote, but low adoption (4-5.3%) limits its effectiveness. Malware could also recast a vote after the verification window, as noted in some analyses.
- Multiple Voting: This reduces the impact of coercion, but only if voters are aware of the manipulation and can revote privately.
- Audits and Transparency: The public availability of the source code and independent audits (e.g., decryption proof verifiers) enhance trust and detectability of some attacks, though client-side vulnerabilities remain a weak point.
- Device Security: The reliance on national ID cards and PINs is robust, but compromised devices could bypass these protections.

## Conclusion

The statement is largely correct in highlighting a vulnerability in IVXV that could enable vote buying or coercion via a malicious voting application, particularly in ways not feasible in poll station e-voting due to the lack of physical oversight. The described attack—using a compromised client to cast precomputed votes with added randomness—is technically plausible and aligns with known concerns about client-side security. However, the claim of "massive" manipulation may exaggerate the practicality for widespread attacks, and poll station e-voting is not entirely immune to manipulation. IVXV's design includes mitigations like multiple voting and verification, but these rely on voter awareness and secure devices, which the scenario circumvents. The statement underscores a critical challenge in remote i-voting: ensuring the integrity of the voter's device in an uncontrolled environment. For further details on IVXV's security, you can review the official documentation at https://valimised.ee or academic analyses like those on ResearchGate.

## Authors Replying back

1-By "***infeasible in poll stations***", I mean to force or buy the vote without letting the voter know who he or she is voting to. In poll station voting you buy or coerce by telling the voter to vote say for X; here it is possible that even the middleman who negotiates the vote buying or coercion doesn't know X (kept a secret till the results appear).

2-By "***massive***" I mean finding a way that a lot of voters get deceived in downloading the malicious voting application VA' (the statement is in the context that they ought to have a way to authenticate their voting application; if they want to allow people to have the freedom to use their own VA' then the code must be scanned before the election and have a registered signature to be used)

## Grok:

Thank you for clarifying the specific meanings of "infeasible in poll stations" and "massive" in the context of the statement about the Estonian IVXV i-voting system. These clarifications help refine the analysis of the statement's correctness regarding the potential for vote buying or coercion via a malicious voting application (VA'). Let's address the statement again, incorporating your clarifications, and evaluate its correctness in light of the IVXV system's design and the described attack scenario. Clarifications Provided

- ***"Infeasible in poll stations":*** You specify that the statement refers to a scenario where vote buying or coercion occurs without the voter (or even the intermediary facilitating the transaction) knowing who the vote is cast for. In poll station voting, vote buying/coercion typically involves instructing the voter to vote for a specific candidate (e.g., "vote for X"), and the voter is aware of the choice. In contrast, the i-voting attack allows the adversary to cast a vote for a candidate (X) that remains secret until results are published, even from the voter or intermediary.

- "***Massive***": You define "massive" as deceiving many voters into downloading a malicious voting application (VA'). The statement assumes voters should authenticate their voting application, and if they are allowed to use their own VA', the code must be scanned and registered with a signature before the election to prevent malicious software.

**Revised Analysis of the Statement**

The statement claims that IVXV's i-voting system could enable "massive vote buying/coercion" not possible in poll station e-voting, where voters are forced to cast votes they don't know via a malicious voting application (VA'). The adversary could use execution attestation (e.g., DARKDAOs-like mechanisms) to ensure malicious code runs, send precomputed encrypted votes $(g^r, y^r.v)$ to VA', and add fresh randomness $(g^{r+r\_new}, y^{r+r\_new}.v)$ to avoid detection of repeated ballots. Let's evaluate this with the clarifications.

## 1. Feasibility of the Attack in IVXV

- Malicious Voting Application (VA'):
    - The scenario assumes voters are deceived into downloading a malicious VA' that casts votes without their knowledge. This is plausible, as IVXV relies on voters using their personal devices (e.g., PCs) to run the official client software. If attackers distribute malware disguised as a legitimate voting application (e.g., via phishing or fake websites), voters could unknowingly install VA'.
    - The clarification about authenticating VA' suggests that IVXV should require code scanning and registered signatures for any voting application. In practice, IVXV provides an official client application, and voters are not permitted to use arbitrary third-party software. However, if an attacker replaces the official client with a malicious version (e.g., through a supply chain attack or social engineering), the device could execute VA' without proper authentication.
    - The "execution attests" (likely referring to trusted execution environments like Intel SGX or attestation mechanisms) imply the adversary ensures VA' runs specific malicious code. This is a sophisticated attack but feasible if the voter's device is compromised, as IVXV does not currently enforce client-side attestation.
- Encrypted Vote Copying:
    - The statement describes the adversary sending an ElGamal-encrypted vote $(g^r, y^r.v)$ to VA', which the compromised device signs and submits. Since IVXV uses ElGamal encryption and digital signatures via national ID cards or mobile ID, a compromised device with access to the voter's private key could sign and cast a precomputed vote without the voter's knowledge.

- Adding fresh randomness ***($g^{r+r\_new}$ , $y^{r+r\_new}.v$)*** is technically valid, as ElGamal ciphertexts can be re-randomized without altering the plaintext vote. This makes it difficult to detect identical votes, enhancing the attack's stealth.

- Voter and Intermediary Unawareness:
    - Your clarification emphasizes that neither the voter nor the intermediary (e.g., a middleman facilitating vote buying) knows the candidate (X) being voted for. This is possible in the described attack: VA' could encrypt a vote for a candidate chosen by the adversary, and the encrypted ciphertext ***($g^r$, $y^r.v$)*** reveals nothing about the vote until decryption by election officials. This secrecy is a unique feature of i-voting, as the voter interacts with a digital system that can hide the vote's content.
    - In IVXV, voters typically select their choice via the client interface, but a malicious VA' could display a fake interface (showing a different candidate) while submitting the adversary's precomputed vote. The voter, unaware of the manipulation, signs the vote using their ID card or mobile ID, believing they voted for their intended candidate.

## 2. Comparison to Poll Station E-Voting

- Poll Station E-Voting:
    - In poll station e-voting (e.g., using direct recording electronic machines), voters interact with a controlled device in a supervised environment. Vote buying or coercion typically requires instructing the voter to select a specific candidate (e.g., "vote for X"), and the voter sees the choice on the screen or paper receipt (if verifiable).
    - Your clarification highlights that in poll stations, it's infeasible to force a vote without the voter knowing who they're voting for. This is correct: the physical environment (poll workers, observers) and direct interaction with the voting machine make it nearly impossible for a malicious application to covertly cast a vote for an unknown candidate. Even if a voter is coerced, they are aware of the candidate they select, and any tampering (e.g., compromised machine) would require physical access, which is harder to scale and conceal.
    - Paper-based poll station voting further ensures voter awareness, as voters mark their choice on a visible ballot. Electronic systems with voter-verified paper audit trails (VVPAT) also allow voters to confirm their choice, making covert vote manipulation difficult.

- IVXV I-Voting:

- In contrast, IVXV’s remote voting occurs on personal devices, which are vulnerable to malware. A malicious VA' could display a fake interface, cast a vote for an unknown candidate, and sign it using the voter’s credentials, all without the voter or intermediary knowing the candidate’s identity until results are published. This aligns with your clarification and supports the statement’s claim that such an attack is unique to i-voting.
- IVXV’s verification mechanism (using a QR code and a separate device within 30 minutes) could detect this if voters verify their vote, but only 4-5.3% of voters do so, per studies. Malware could also recast a vote after verification, as noted in analyses like Springall et al. (2014).

## 3. “Massive” Scale of Deception

- Your clarification defines “massive” as deceiving many voters into downloading VA'. This is a critical point, as the attack’s scale depends on the adversary’s ability to distribute malicious software widely.
- Feasibility:
    - Deceiving voters into downloading VA' is plausible through social engineering (e.g., phishing emails, fake election websites, or app store attacks). For example, attackers could mimic Estonia’s official election website (valimised.ee) or distribute malware via email campaigns targeting Estonian voters.
    - Estonia’s high digital literacy and widespread use of national ID cards (used by 98% of citizens) make phishing a viable vector, as voters may trust official-looking digital communications. A 2014 study noted vulnerabilities in IVXV’s server infrastructure to state-sponsored attacks, suggesting client-side attacks are also feasible.
    - However, IVXV’s official client is distributed through trusted channels, and voters are instructed to use only the official software. The statement’s suggestion that voters might “have the freedom to use their own VA'” is not accurate for IVXV, as it mandates the official client. Still, a compromised official client (via supply chain attack) or a convincing fake could achieve the same effect.
- Authentication of VA':
    - You note that VA' should be authenticated via code scanning and registered signatures. IVXV does not currently allow third-party voting applications, and the official client is open-source and auditable. However, there’s no mandatory client-side code verification by voters, and a sophisticated attacker could distribute a malicious version that mimics the official client.

    - Implementing code scanning and signature registration (e.g., via cryptographic signatures verified by the election authority) would mitigate this risk but is not currently part of IVXV's process. This supports your implied critique that stronger client authentication is needed to prevent such attacks.
- Scale:
    - Achieving "massive" deception requires compromising many devices, which is challenging but not impossible. For example, a nation-state actor could launch a widespread malware campaign, as seen in past cyberattacks (e.g., Estonia's 2007 DDoS attacks). The DARKDAOs reference suggests a decentralized, coordinated effort, which could scale through botnets or automated distribution.
    - The actual number of affected voters would depend on the attack's sophistication and the adversary's resources. Even a few thousand compromised votes could be significant in Estonia's elections (e.g., 247,232 i-votes in 2019), but "massive" implies a larger scale, which is less likely without noticeable detection.

## 4. Mitigations in IVXV

IVXV includes features to mitigate coercion and vote buying, but they are less effective against this attack:

- Multiple Voting: Voters can cast multiple online votes or vote in person to override online votes. However, if VA' casts a vote covertly, the voter may not know to revote, rendering this mitigation ineffective.
- Verification: The 30-minute verification window allows voters to check their vote using a separate device, but low adoption (4-5.3%) and the possibility of malware recasting votes post-verification limit its impact.
- Auditability: IVXV's use of mix-nets, NIZK proofs, and public source code allows auditing, but these focus on server-side integrity, not client-side attacks. Detecting widespread use of VA' would require additional monitoring (e.g., analyzing client software distribution).
- Client Security: IVXV relies on voters' devices being secure. Your suggestion of code scanning and registered signatures would strengthen this, but it's not currently implemented.

## **5. Correctness of the Statement**

With your clarifications, the statement is correct in its core claims:

- Unique Vulnerability in I-Voting: The ability to cast votes without the voter or intermediary knowing the candidate is indeed unique to i-voting. In poll station e-voting or paper voting, the voter sees their choice (via screen, paper, or VVPAT), making it infeasible to hide the candidate's identity. IVXV's remote nature and reliance on personal devices enable a malicious VA' to covertly cast votes for an unknown candidate, as described.

- Malicious VA' and Encryption: The technical mechanism (using precomputed ElGamal ciphertexts with added randomness) is accurate and feasible, allowing the adversary to maintain secrecy and avoid detection of repeated votes.
- Massive Scale: Deceiving many voters into downloading VA' is plausible, especially with sophisticated social engineering or supply chain attacks. Your point about authenticating VA' highlights a gap in IVXV's current design, as it lacks mandatory client-side code verification. While IVXV restricts voters to the official client, a compromised or fake client could enable the attack.
- Comparison to Poll Stations: The statement correctly asserts that this attack is not feasible in poll stations, where physical oversight and controlled devices ensure voters know their choice.

However, the statement has minor exaggerations or assumptions:

- "Massive" Feasibility: While possible, achieving a "massive" scale requires significant resources and coordination, likely limiting it to state-sponsored actors. Smaller-scale attacks are more realistic but still concerning.
- Voter Freedom to Use VA': The statement's context implies voters might use their own VA', but IVXV mandates the official client. The attack remains feasible if the official client is compromised or replaced, but this assumption slightly misaligns with IVXV's design.

## Conclusion

The statement is correct in asserting that IVXV's i-voting system enables a form of vote buying/coercion not feasible in poll station e-voting, where voters (and intermediaries) are unaware of the candidate voted for until results are published. The described attack—using a malicious VA' to cast precomputed, re-randomized ElGamal-encrypted votes—is technically sound and exploits IVXV's reliance on voter device security. The "massive" scale is plausible through widespread deception (e.g., phishing or fake clients), though challenging without significant resources. Your suggestion of authenticating VA' via code scanning and signatures would mitigate this risk but is not currently implemented. In contrast, poll station e-voting's controlled environment prevents such covert manipulation, supporting the statement's comparison. For further details, see IVXV documentation at https://valimised.ee or analyses like Springall et al. (2014) on ResearchGate.

## Authors Replying back

*To clarify something currently IVXV does not authenticate the voting application (does not require its digital signature when interacting with it while at the voter device). It did happened in 2023 that a voter voted using his own voting application (an observer to test the possibility), and the risk was mentioned in OSCE/ODIHR 2023 report. In his recent article on Feb 2025, https://arxiv.org/html/2502.06385v2, Jan Willemson (works in Cybernetics, the company behind IVXV)while talking about ways to defend the Olivier Pereira attack*

" All the following assumptions need to hold:

- the voter is successfully directed to an unofficial distribution channel,
- the voter does not verify authenticity of the voting application, and
- the voter does not report suspicious crash of the application. For assumption 1, it is easy to set up a lookalike website, but it will very probably be noticed and reported. For assumption 2, we need to take into account that under Windows and macOS the voting application is signed with a developer key and the OS verifies the signature before running it. It is possible for an attacker to register as a developer, but this will leave more traces. Linux users are supposed to verify the checksum of the application themselves, but on the other hand, Linux users are more likely to do it. For assumption 3, we note that in order to have a significant effect, the attacker needs to manipulate many votes. However, the probability that no crashes will be reported decreases exponentially fast in the number of crashes (see below for the computations)." 1- This implies the IVXV system still doesn't authenticate the voting application and depends on users to do so through auxiliary ways (they did not even provide an Electrum wallet like check for it. 2- In country's political elections (or in European Parliament elections) you don't build your safety assumptions based on having only "small scale/budget adversaries". 3-The point is they should or maybe they must do something about it and there are many easy solutions to implement."

## Grok:

Thank you for the additional clarifications and the reference to Jan Willemson's February 2025 article [(https://arxiv.org/html/2502.06385v2)](https://arxiv.org/html/2502.06385v2). Your points highlight critical concerns about the Estonian IVXV i-voting system's lack of mandatory authentication for the voting application, reliance on voter behavior for security, and the inadequacy of assuming only small-scale adversaries in high-stakes elections.

Let's address these points and re-evaluate the original statement's correctness in light of this new information, focusing on the clarified context and the specific issues raised.Recap of the Original Statement

The statement claims that IVXV could enable "massive vote buying/coercion" not possible in poll station e-voting, where voters are forced to cast votes they don't know via a malicious voting application (VA'). The adversary uses execution attestation (e.g., DARKDAOs-like mechanisms) to ensure malicious code runs, sends precomputed ElGamal-encrypted votes ***($g^r$, $y^r.v$)*** to VA', and adds randomness ***($g^{r+r\_new}$ , $y^{r+r\_new}.v$)*** to avoid detection.

Your clarifications specify:

- "Infeasible in poll stations": The attack involves casting votes without the voter or intermediary knowing the candidate until results are published, which is not feasible in poll stations where voters see their choice.
- "Massive": Deceiving many voters into downloading VA' via social engineering or unofficial channels.
- Authentication Context: IVXV does not currently require digital signature authentication of the voting application on the voter's device, as confirmed by the 2023 incident and Willemson's article. Voters are expected to verify the application's authenticity themselves (e.g., via OS checks or checksums), and the system's security relies on these auxiliary measures.

Addressing Your Points, Your three points highlight specific issues with IVXV's approach to voting application security and the assumptions in Willemson's defense against the Pereira attack (a malicious VA' faking a crash to deceive voters, as described in and). Let's analyze each point in the context of the statement and IVXV's design.

<u>1. Lack of Mandatory Voting Application Authentication</u>

- Current State: You note that IVXV does not authenticate the voting application (i.e., does not require a digital signature verified by the election system when the application interacts with the voter's device). This is supported by the 2023 incident where an election observer used their own voting application to test the system, as mentioned in the OSCE/ODIHR 2023 report (). Willemson's article () confirms that IVXV relies on voters to verify the application's authenticity through auxiliary methods:
    - Windows/macOS: The operating systems verify the application's developer signature, but this is a standard OS security feature, not an IVXV-specific requirement enforced by the election system.
    - Linux: Voters are expected to manually verify the application's checksum, which assumes technical expertise and diligence.
    - No centralized mechanism (e.g., an Electrum wallet-like signature check integrated into the voting process) ensures the application's authenticity before it interacts with the voter's credentials or submits votes.
- <u>Implications for the Statement:</u> The lack of mandatory authentication increases the risk of voters downloading a malicious VA' from an unofficial channel, as described in the statement. The 2023 incident demonstrates this vulnerability, as

an observer successfully used an unauthorized application, highlighting that IVXV does not block such applications. This supports the statement's claim that a malicious VA' could be used to cast votes covertly, enabling vote buying/coercion without voter awareness.

- Willemson's Defense: Willemson argues that voters must be directed to an unofficial distribution channel and fail to verify the application's authenticity (). However, as you point out, relying on voters to perform these checks is a weak security assumption, especially since:
    - Many voters lack the technical knowledge to verify checksums (Linux) or understand developer signatures (Windows/macOS).
    - The absence of an IVXV-enforced signature check during the voting process means a malicious VA' could operate undetected if it mimics the official client's interface.
    - The 2023 OSCE/ODIHR report likely noted this risk (exact details unavailable, but implied by the observer's test), reinforcing that the system's design allows unauthorized applications to interact with the voting infrastructure.

## 2. Inappropriate Assumption of Small-Scale Adversaries

- Context:  Willemson's defense against the Pereira attack assumes that a malicious VA' would require a large number of crashes to manipulate many votes, and the probability of these crashes going unreported decreases exponentially (). This assumes a small-scale or budget-limited adversary who cannot orchestrate a widespread, stealthy attack. You argue that in high-stakes elections (e.g., Estonian parliamentary or European Parliament elections), security assumptions should account for sophisticated adversaries, such as state-sponsored actors.
- Analysis:
    - High-Stakes Elections: Estonian parliamentary elections (e.g., 2023, with 51% i-votes) and European Parliament elections (e.g., 2024) are critical democratic processes with national and international implications. Sophisticated adversaries (e.g., nation-states) could invest significant resources to compromise devices en masse, as seen in historical cyberattacks like Estonia's 2007 DDoS attacks (). Assuming only small-scale adversaries is inadequate, as state actors could:
        - Distribute malware via advanced phishing or supply chain attacks to deploy VA' widely.
        - Use social engineering to direct voters to fake websites, as Willemson acknowledges is "easy" to set up ().

    - Suppress crash reports by designing VA' to minimize noticeable disruptions or by targeting non-verifying voters (94% of voters don't verify, per).
- Pereira Attack: The Pereira attack (,) involves a malicious VA' faking a crash to generate two votes, deceiving the voter into believing their intended vote was cast while submitting a different one. Willemson argues that frequent crashes would be reported, but a sophisticated adversary could:
    - Design VA' to fake crashes sparingly or only for targeted voters.
    - Exploit the low verification rate (4-5.3%) to reduce detection risk.
    - Use execution attestation (as in the statement's DARKDAOs scenario) to ensure malicious code runs reliably, increasing the attack's scalability.
- Implications for the Statement: The statement's reference to "massive" vote buying/coercion aligns with the potential for a large-scale attack by a sophisticated adversary. Willemson's assumption that crashes would be detected underestimates the capabilities of well-funded actors, supporting your critique and the statement's concern about widespread manipulation via VA'.

### 3. Need for Easy Solutions

- Current Gaps: You argue that IVXV should (or must) implement solutions to authenticate the voting application, and many "easy solutions" exist. The 2023 incident and OSCE/ODIHR report highlight the risk of unauthorized applications, and Willemson's reliance on OS-level checks or voter diligence () indicates a lack of robust, system-enforced authentication.
- Possible Solutions:
    - Digital Signature Verification: Require the voting application to present a cryptographic signature verified by the IVXV server before it can interact with the voting infrastructure. This could mirror Electrum wallet's signature verification, where the client software is signed by a trusted authority, and the system rejects unsigned or tampered applications.
    - Checksum Integration: Embed a checksum verification step in the voting process, prompting voters to confirm the application's integrity before voting. This could be automated (e.g., the IVXV server provides the expected checksum) to reduce user burden.
    - Official Distribution Channels: Restrict voting to applications downloaded from a verified source (e.g., valimised.ee) with server-side checks to reject

votes from unauthorized clients. This could involve a token-based authentication mechanism tied to the official client.

- Execution Attestation Defense: To counter the DARKDAOs-like attestation in the statement, IVXV could use trusted execution environments (e.g., Intel SGX) to verify that the official client is running unmodified code, though this adds complexity and hardware dependencies.

- Feasibility: These solutions are technically feasible and align with standard security practices in other domains (e.g., software updates, cryptocurrency wallets). For example, digital signature verification is widely used in software distribution (e.g., Microsoft's Authenticode, Apple's Gatekeeper). Implementing them in IVXV would require updates to the client-server protocol but is not prohibitively complex, supporting your claim of "easy solutions."

- **Implications for the Statement:** The lack of these measures strengthens the statement's claim that IVXV is vulnerable to a malicious VA' enabling covert vote manipulation. The 2023 observer's test () and the Pereira attack () demonstrate that such vulnerabilities are exploitable, and the absence of mandatory authentication exacerbates the risk of massive vote buying/coercion.

## Re-Evaluating the Statement's Correctness

With your clarifications and the new information, the statement remains correct and is further strengthened:

- Unique Vulnerability: The ability to cast votes without the voter or intermediary knowing the candidate (via a malicious VA' submitting precomputed ElGamal ciphertexts) is unique to i-voting and infeasible in poll station e-voting, where voters see their choice (e.g., on a screen or paper). The Pereira attack () confirms this by exploiting IVXV's revoting feature to deceive voters, and the lack of VA' authentication amplifies this risk.

- Massive Scale: The statement's scenario of deceiving many voters into downloading VA' is plausible, especially given the 2023 incident where an observer used an unauthorized application (). Sophisticated adversaries could scale this via phishing, fake websites, or supply chain attacks, as Willemson acknowledges (). The low verification rate (94% non-verifying voters) and lack of system-enforced authentication make large-scale deception feasible.

- Technical Accuracy: The use of ElGamal encryption with added randomness $(g^{r+r\_new}, y^{r+r\_new}.v)$ to avoid detection is correct, as it preserves vote secrecy while enabling covert manipulation. The DARKDAOs-like attestation mechanism is a realistic threat, as adversaries could ensure malicious code runs reliably on compromised devices.

- Comparison to Poll Stations: The statement's claim that this attack is "never possible" in poll station e-voting holds, as the controlled environment ensures voters know their choice. Even in compromised e-voting machines, physical oversight and voter-verified paper trails (VVPAT) prevent covert vote manipulation without detection.

## Critiques and Nuances

- Willemson's Assumptions: Willemson's defense () relies on voters noticing unofficial channels, verifying application authenticity, and reporting crashes. These assumptions are flawed for high-stakes elections:
  - Unofficial Channels: Fake websites are easy to create and can deceive non-technical voters, as Willemson admits (). The 2023 incident shows voters can use unauthorized applications, undetected by IVXV.
  - Verification: Most voters (94%) don't verify their votes (), and fewer are likely to verify application checksums, especially on Linux. OS-level signature checks (Windows/macOS) are insufficient, as attackers can register as developers ().
  - Crash Reporting: Sophisticated adversaries could minimize crashes or target non-verifying voters, reducing detection risk. The exponential decrease in detection probability assumes uniform crash reporting, which may not hold for targeted attacks.
- **High-Stakes Adversaries:** Your point that security assumptions should not rely on small-scale adversaries is critical. Estonia's history of cyberattacks (e.g., 2007 DDoS) and the geopolitical significance of its elections (, EKRE results flipped by i-votes) necessitate defenses against state-sponsored actors capable of large-scale, stealthy attacks.
- Mitigation Gaps: IVXV's mitigations (multiple voting, 30-minute verification) are ineffective if voters are unaware of VA' manipulation. The 2023 OSCE/ODIHR report () and academic analyses (,) highlight ongoing vulnerabilities, including insider attacks and verification weaknesses, supporting the statement's concerns.

Recommendations

Your point about "easy solutions" is well-taken. IVXV could address the VA' vulnerability by:

- Mandatory Signature Verification: Require the voting application to present a digital signature verified by the IVXV server before accepting votes. This could use public-key infrastructure (PKI) similar to Estonia's e-ID system.

- Automated Checksum Validation: Integrate checksum verification into the voting client, prompting users to confirm the application's integrity against a server-provided hash.
- Secure Distribution: Enforce downloads from valimised.ee with server-side checks to reject votes from unauthorized clients. A token-based handshake could ensure client legitimacy.
- Improved Verification: Increase voter awareness of the 30-minute verification process (e.g., via mandatory prompts) and extend verification to detect re-voting by malicious applications, addressing the Pereira attack.
- Audit Enhancements: Implement independent client-side auditing tools, as suggested by election observers (), to detect unauthorized applications during elections.

These measures would align with Estonia's robust e-ID infrastructure () and reduce reliance on voter diligence, addressing the risks highlighted in the statement and your critiques.

## Conclusion

The statement is correct and reinforced by your clarifications and Willemson's article. IVXV's lack of mandatory voting application authentication enables a malicious VA' to facilitate covert vote buying/coercion, as demonstrated by the 2023 observer's test and the Pereira attack. The attack's feasibility is heightened by the low verification rate, reliance on voter diligence, and the absence of system-enforced client authentication, making "massive" manipulation plausible against sophisticated adversaries. Unlike poll station e-voting, where voters know their choice, IVXV's remote nature allows votes to be cast for unknown candidates, supporting the statement's claim. Willemson's defense underestimates the threat of large-scale adversaries in high-stakes elections, and the lack of "easy solutions" like signature verification exacerbates the risk. Implementing your suggested measures (e.g., digital signatures, checksum integration) would significantly mitigate this vulnerability, aligning with recommendations from OSCE/ODIHR () and academic analyses (,). For further details, see the IVXV documentation (https://valimised.ee) or Willemson's article (https://arxiv.org/html/2502.06385v2).