Is the JCJ voting system really coercion-resistant?

Véronique Cortier, Pierrick Gaudry, Quentin Yang

Université de Lorraine, Inria, CNRS

Abstract. Coercion-resistance is a security property of electronic voting, often considered as a must-have for high-stake elections. The JCJ voting scheme, proposed in 2005, is still the reference when designing a coercion-resistant protocol. We highlight a weakness in JCJ that is also present in all the systems following its general structure. It comes from the procedure that precedes the tally, where the trustees remove the ballots that should not be counted. This phase leaks more information than necessary, leading to potential threats for the coerced voters. Fixing this leads to the notion of *cleansing-hiding*, that we apply to form a variant of JCJ that we call CHide. This is a shorter version of [5].

1 Introduction

Internet voting allows to take part into an election without being physically present. It is used for politically-binding elections in several countries. For such contexts, coercion is an important threat which occurs when an attacker forces a voter to vote in a specific way, using a threat or a reward. It is known to exist in traditional elections, but an electronic voting solution which is not designed to tackle it could allow the attacker to coerce a larger number of voters, or to gain a more convincing evidence that the coerced voters actually obeyed.

A famous protocol designed to counter coercion was proposed in 2005 [6], along with a formalization of the notion which allows to give security arguments. It is called the JCJ protocol and remains the reference on coercion-resistance. We unveil that a phase (that we name the *cleansing phase*) of JCJ leaks some information that can be exploited by the coercer. We provide an example where this allows to fully break coercion-resistance. This highlights that, in general, the attacker has a non-negligible advantage by exploiting the leakage. All the variants and improvements on JCJ that we know of are also affected.

We propose a modification of JCJ, that we call CHide, and that is not subject to this weakness. The key modification is the introduction of a *cleansing hiding* procedure, that replaces the original leaky phase. As a consequence, in CHide, the adversary can only learn minimal information from the cleansing phase. Of course, each step comes with a zero-knowledge proof (ZKP) that the expected operation has been performed, so that anyone can check that the result of the election is correct.

2 Unveiling a Shortcoming in JCJ

We present a vulnerability in the JCJ scheme and discuss its impact.

2.1 Leakage in case of revoting

For a verifiable voting system which uses a public board, it seems unavoidable to leak the total number n_r of received ballots. The number n_v of valid ballots is also leaked unless more sophisticated tally methods are used [2, 7]. However, JCJ leaks more information in case of revoting, namely n_r the number ot revotes and even the complete distribution of revotes per credential. This leakage occurs during the *cleansing phase*, where duplicated and unauthorized credentials are removed, and can be exploited by a coercer to detect when a coerced voter disobeys. Indeed, there is no reason to assume that revoting is independent from the choice of candidate, as it is often due to voters changing their mind between candidates, for instance after a late announcements in the press.

Attacking coercion-resistance. We consider an extreme case, with two candidates such that voters voting for A do not revote while voters voting for B always revote once. Let r_A (resp. r_B) the number of votes for A (resp. B). Due to the distribution of voting behaviors that we consider, the number of revotes corresponds to the number of votes for B sent by the honest voters.

Assume now that Alice wants to vote for B but is instructed by her coercer to vote for A (or abstain): if Alice obeys, the coercer observes $r_B = n_r$ and if she disobeys and casts one ballot for B, the coercer observes that $r_B = n_r + 1$. Hence the coercer detects when Alice disobeys, which breaks coercion-resistance.

One could argue that Alice should follow a different strategy and cast one ballot (resp. two) when she votes for A (resp. B). In this case, a similar attack is possible when she wants to vote for A but is instructed to vote for B.

Discussion. The distribution considered above is very contrived. But as soon as the distribution of revotes is not independent from the distribution of votes per candidate, the coercer will learn some information, and hence detect when a voter disobeys with some non negligible advantage.

2.2 More noise is needed

A known issue of JCJ is that fake ballots should be randomly added, in order to hide to a coercer that the ballot cast under coercion has been removed. In JCJ, this “noise” comes from honest voters sending ballots with an invalid credential, but this source alone may not be sufficient. A natural approach is to have the authorities add a random number of dummies, as proposed in [9] (to mitigate a leakage during the tally). This noise made of fake ballots should however be calibrated carefully since the computation overhead of additional ballots is important. In a context where revoting is a well spread behavior, it could be judicious to rely on revoting, at least partially, as an additional source of noise. This is however not possible in JCJ since the two sources of noise can be distinguished.

3 CHide: A Cleansing-Hiding Variant of JCJ

We propose a modification of JCJ, where the cleansing phase is replaced by an MPC protocol that does not leak any extra information.

3.1 Cyrptographic primitives

ElGamal encryption scheme. We use the ElGamal encryption scheme on elliptic curves, which is convenient for its efficiency and homomorphic property. If (g, h) is the public key, we encrypt m by computing $(g^r, g^m h^r)$ with some random r. To decrypt, we need m to be taken in a small list of valid messages. An important special case is $\{0, 1\}$, when the MPC primitives we mention below can be applied. For a general message m, we use the bit-wise encryption of m, which is the list of the encryptions of the bits of a binary encoding of m.

Logical operations on encrypted bits. There are verifiable MPC protocols that allow to jointly perform logical operations on encrypted bits, without revealing the cleartexts to anyone. The main building block we use is the `CGate` protocol [8], that allows to compute an encryption of a logical and (a conjunction) of the encrypted bits given in input. Combining this with the homomorphic property of ElGamal encryption, we designed various protocols for all the logical operations on bits, and ultimately for realizing any function; see [4] for a more extensive description of the protocols we use. In CHide, we especially use the `Eq` (equality test) and the `Or` (disjunction) protocols that work on encrypted bits. The `Eq` protocol is extended to bit-wise encrypted data, by computing the conjunction of all the equality tests on encrypted bits.

3.2 Description of the CHide protocol

Setup phase. In the setup phase of CHide, the voters receive a credential. A bitwise encryption of the credential is published in the public board.

Voting phase. During this phase, the voters encrypt their vote as well as each bit of their credential. They also prove the knowledge of the plaintexts and that all encryptions are linked.

Cleansing and tallying phase. Once the voting phase is finished, the election trustees get the list of ballots published on the board. They run an MPC procedure on them, which allows to add an encrypted bit of validity to each ballot. Afterwards, the ballots are shuffled and the validity bit is decrypted, so that only the number of total and invalid ballots is revealed.

Efficiency considerations. In terms of computational and communication costs, the CHide system is slightly less efficient, but still in the same ballpark as JCJ. The encrypted credentials are now formed by κ ciphertexts instead of a single one, where κ is the security parameter. This factor is probably affordable by the authorities whose task is highly parallel. For the voters, the computational load increases but the total cost for realistic parameters is around a thousand exponentiations, which should be a matter of seconds with a standard implementation in Javascript running within a modern browser.

For the talliers, the cleansing phase is more complex than the one in JCJ, but still requires a number of exponentiations that grows quadratically with the number of ballots received on the board. The main difference is that due to the MPC tools, the number of communication rounds between them is no longer constant, but becomes logarithmic in the number of ballots.

4 Discussion

We conclude by discussing two other coercion-resistance protocols, which also have their own leakages.

We start with the AFT scheme presented in [1]. Its main feature is that it has a linear time complexity for the cleansing and tallying phase. While it uses different cryptographic primitives from JCJ, it has a similar structure. Assuming that the cryptography is perfect, we remark that both the number of duplicated and unauthorized credentials are revealed during the protocol, just as in JCJ. In addition, it is possible to deduce, by observing the board, the complete distribution of revote per credential. In JCJ, this information is only available during the tally, when it is no longer possible to submit a ballot. Hence, in the AFT scheme, the adversary may exploit this information to submit ballots in a specific way. Consequently, it provides a coercion-resistance level which is very similar to JCJ, but slightly (but strictly) weaker.

Another interesting example is Civitas [3], a scheme considered as equivalent to JCJ, but that actually leaks more information. First, it provides the same leakage as the AFT protocol: the number of revotes for each ballot can be directly deduced from the board. Furthermore, in order to mitigate the (still quadratic) cost of the cleansing, it proposes to group voters by blocks: each credential is publicly assigned to one block, and the voter indicates their block in clear when casting their ballot. Compared to JCJ, the adversary still learns how many revotes each ballot has and how many invalid ballots there is, but also has access to this information block by block, which leads to a stricly weaker security.

References

1. R. Araújo, S. Foulle, and J. Traoré. A practical and secure coercion-resistant scheme for remote elections. In *Frontiers of Electronic Voting*. IBFI, 2007.
2. J. Benaloh, T. Moran, L. Naish, K. Ramchen, and V. Teague. Shuffle-sum: coercion-resistant verifiable tallying for STV voting. *IEEE Trans. Inf. Forensics Secur.*, 2009.
3. M. Clarkson, S. Chong, and A. Myers. Civitas: Toward a Secure Voting System. In *S&P'08*. IEEE Computer Society, 2008.
4. V. Cortier, P. Gaudry, and Q. Yang. A toolbox for verifiable tally-hiding e-voting systems. Cryptology ePrint Archive, Report 2021/491, 2021. `https://ia.cr/2021/491`.
5. V. Cortier, P. Gaudry, and Q. Yang. Is the JCJ voting system really coercion-resistant? Cryptology ePrint Archive, Paper 2022/430, 2022. `https://eprint.iacr.org/2022/430`.
6. A. Juels, D. Catalano, and M. Jakobsson. Coercion-Resistant Electronic Elections. In *ACM Workshop on Privacy in the Electronic Society (WPES'05)*. ACM, 2005.
7. R. Küsters, J. Liedtke, J. Müller, D. Rausch, and A. Vogt. Ordinos: A Verifiable Tally-Hiding E-Voting System. In *EuroS&P'20*. IEEE Computer Society, 2020.
8. B. Schoenmakers and P. Tuyls. Practical Two-Party Computation Based on the Conditional Gate. In *Advances in Cryptology (ASIACRYPT'04)*. Springer, 2004.
9. O. Spycher, R. E. Koenig, R. Haenni, and M. Schläpfer. A New Approach towards Coercion-Resistant Remote E-Voting in Linear Time. In *15th International Conference on Financial Cryptography and Data Security (FC'11)*. Springer, 2011.