Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa

How to prove the validity of a complex ballot encryption to the voter and the public☆

Rui Joaquim

University of Luxembourg, Interdisciplinary Centre for Security, Reliability and Trust, 6, rue Richard Coudenhove-Kalergi, L-1359, Luxembourg

ARTICLE INFO

Article history:
Available online 10 June 2014

Keywords:
Verifiable vote encryption
Complex ballots
Electronic voting
Cast-as-intended verification
Verification codes

ABSTRACT

One crucial aspect of any verifiable electronic voting system that uses encryption is the proof that the vote encryption is well-formed, i.e. the proof that the vote encryption encrypts a valid vote accordingly to the race specification. It makes no sense accepting an encrypted vote if, at the end of the election, the vote cannot be included in the tally because it is badly formed.

Proving the validity of a complex vote encryption, without revealing the vote, is a hard problem. This paper first contribution addresses exactly that problem and provides a set of new constructions to create a vote encryption and the corresponding public proof of validity for several types of complex ballots ($[k_{min},k_{max}]$-out-of-n approval, weighted and ranked ballots). The second contribution is a technique that allows to create a single, constant size, verification code for a ballot containing one or several races of any mix of the race types considered. With this single verification code the voter can verify that her vote was casted-as-intended.

Moreover, our constructions can be tuned for either mix net or homomorphic tallying and support both types of tallying in the same multi-race ballot.

1. Introduction

Elections are essential for a democratic society as they are the basis of our democracies. Therefore, in order to allow free voting to everyone, it is critical that the election system ensures the correctness of the elections' results while preserving the voter's privacy.

An election system should not only provide proofs that the votes are counted correctly but also that they capture the intention of the voters, preserving the anonymity of the votes. While the techniques to provide an electronic verifiable tally are well established, the same cannot be said about the techniques to prove that a vote encryption performed by an untrusted machine encrypts the voter's vote intention.

This paper describes a set of new constructions inspired on the MarkPledge family of voter verifiable vote encryption protocols (Andrew Neff, 2004; Adida and Neff, 2009; Joaquim and Ribeiro, 2012). Despite significant differences at the technical details, at a high level, a MarkPledge voter verification protocol is a slightly modified version of a fairly straightforward zero-knowledge proof, in which the voter chooses the challenge and performs a simple string comparison to verify that her vote was encrypted correctly. All equations necessary for soundness are publicly verifiable, thus can be verified by any interested party, including the voter, using an independent machine.

☆ This work was supported by the Fonds National de la Recherche, Luxembourg (INTER/SNF/11/11).
E-mail address: rui.joaquim@uni.lu.
http://dx.doi.org/10.1016/j.jisa.2014.04.004
2214-2126/

To our knowledge, our constructions are the first to offer a highly sound voter verification mechanism for complex ballots with a constant size vote verification code. The use of a single verification code requires the voter to have access to a trusted piece of software/hardware to help her compute the verification code from the public election data and hers secret selections. We also show how to create a more traditional code voting receipt (with one verification code per candidate) from our constructions. This receipt allows the voter to verify the correct vote encryption without the need of a trusted device, although, like other vote confirmation techniques that use one verification code for each vote selection, it rapidly becomes unusable with the increase of the ballot complexity.

In this work we only address the problem of creating a voter verifiable vote encryption and prove it to be honest-verifier zero-knowledge. We do not propose any full vote protocol, although the adaptation to the coercion resistant protocol in (Adida and Neff, 2006) or the simplified vote protocol described in (Andrew Neff, 2004) is straightforward.

The new constructions proposed in this paper are very flexibly and allow to support in an uniform way several types of complex ballots ($[k_{min},k_{max}]$-out-of-n, weighted, ranked and multi-race ballots). It can be tuned for either mix net or homomorphic tallying, allowing even the use of both types of tallying on different races in a single multi-race ballot. This flexibility can be very useful in anonymous surveys where some answers must be correlated.

One crucial aspect of our vote encryption verification constructions, and of electronic voting in general, is the proof that the vote encryption is well-formed, i.e. the proof that the vote encryption encrypts a valid vote accordingly to the race specification. It makes no sense in verifying a vote encryption if, at the end of the election, the vote cannot be included in the tally because it is badly formed.

We solve the vote encryption well-formness verification problem by: i) creating the vote encryption from a verifiable shuffle of a set of encryptions of known messages; and ii) whenever necessary, using additional zero-knowledge proofs of compliance to the vote specification. To our knowledge this approach is new and completely different from the previous ones that impose a certain mathematical structure to the vote plaintext construction, e.g. (Groth et al., 2005).

The remainder of the paper is organized as follows: the next section presents the related work. Then, Section 3 gives the background necessary for our constructions. Section 4 details the constructions for mix net tallying and Section 5 describe the constructions for homomorphic tallying. Section 6 shows how to extract a MarkPledge style voter verifiable receipt from our complex ballot encryption constructions and Section 7 presents the conclusions.

2. Related work

In 2004, with the work of Chaum (2004) and Andrew Neff (2004) a new paradigm in electronic voting research has emerged: End-to-End (E2E) voting systems. The goal of E2E voting systems is to develop voting systems with both voter cast-as-intended and universal counted-as-cast verifications.

Chaum (2004) addresses the voter cast-as-intended verification using visual cryptography (Naor and Shamir, 1995). His proposal uses special printers that print a human readable vote encryption on two overlapped transparent sheets.

Later the Prêt-à-Voter (Chaum et al., 2005; Ryan et al., 2009; Prêt-à-Voter web site, 2014) and the Punchscan (Chaum, 2011; Popoveniuc and Hosp, 2010) systems simplified the original Chaum's setup using pre-printed ballots. Adida and Rivest proposed the Scratch-and-Vote system (Adida and Rivest, 2006), based on Prêt-à-Voter, which uses scratch strips to allow off-line ballot verification. In 2007, Moran and Naor (Moran and Naor, 2007) proposed an everlasting private[1] system based on Punchscan. The E2E ideas proposed in Punchscan served also as inspiration for the development of the optical scanner based E2E verifiable voting system Scantegrity (Chaum et al., 2008b), which was improved in Scantegrity II (Chaum et al., 2008a) with the incorporation of vote confirmation codes. All these protocols, have pre-printed ballots, which allow for a ballot auditing process before the election to minimize the risk of using bogus ballots.

The ideas of the above described poll station E2E systems where later adapted to the Internet voting scenario in the Pretty Good Democracy (PGD) (Ryan and Teague, 2009) and the Scratch, Click and Vote (SCV) (Mirosław Kutyłowski and Filip Zagórski, 2010) voting systems. PGD achieves E2E verifiability by enhancing a code voting protocol inspired by some ideas used in the Scantegrity II and Prêt-à-Voter systems. PGD was later enhanced to support expressive voting schemes in which the voter lists the candidates in order of preference (Heather et al., 2010). SCV uses the voter cast-as-intended verification ideas of Punchscan, Prêt-à-Voter and ThreeBallot.[2]

Neff's proposal (Andrew Neff, 2004) (also known as MarkPledge) uses a quite different approach. It codifies a verification code for each candidate into a set of 1-out-of-2 cut and choose proofs of encryption. The verification codes are then computed from each set of encryptions and a vote receipt with a random looking verification code for each candidate is created. This technique achieves a soundness of $1/2^{\alpha}$ for a verification code with a length of α bits. Neff's proposal main disadvantages are: i) the high computational costs; and ii) the complex vote protocol, which forces the voter to perform a complex challenge-response style protocol with the voting machine, at the voting booth. The usability issues were addressed in Adida and Neff (2006), Joaquim et al. (2013) and the efficiency issues in Adida and Neff (2009), Joaquim and Ribeiro (2012). Moran and Naor presented a MarkPledge like system with "everlasting privacy" (Moran and Naor, 2006) by replacing the vote encryptions with bit commitments.

A completely different voter verification approach was proposed by Benaloh (Benaloh, 2006; Benaloh, 2007). Benaloh's proposal separates the vote encryption from the vote casting process. There is a vote preparation machine that encrypts the vote but does not cast it, instead it delivers the vote encryption to

[1] Moran and Naor define that a system has "everlasting" privacy if a computational unbounded adversary gains no information about specific votes from observing the protocol's output.

[2] The ThreeBallot system is a paper based voter verifiable voting system proposed by Rivest (Rivest, 2006; Rivest and Smith, 2007) which "does not" use any cryptography.