Receipt-Free K-out-of-L Voting Based on ElGamal Encryption*

Martin Hirt

ETH Zurich, Department of Computer Science,
hirt@inf.ethz.ch

Abstract. We present a K-out-of-L voting scheme, i.e., a voting scheme that allows every voter to vote for (up to) K candidates from a set of L candidates. The scheme is receipt-free, which means that even a malicious voter cannot prove to anybody how he voted. Furthermore, the scheme can be based on any semantically secure homomorphic encryption scheme, in particular also on the modified ElGamal encryption scheme which does not allow for efficient decryption of arbitrary large messages (but is more efficient than Paillier's encryption scheme).
We note that in contrast to the standard setting with receipts, in a receipt-free setting a K-out-of-L voting scheme cannot be derived directly from a yes/no voting scheme.
Finally, we show that the voting protocol of Lee and Kim is not receipt-free, opposed to what is claimed in the paper.

1 Introduction

1.1 Problem Summary

The goal of an electronic voting protocol is to compute the sum of the votes of all entitled voters. In the simplest case, every voter can cast one of two possible votes (Yes/No-votes). More generally, every voter may vote for any K candidates out of a list of L candidates (K-out-of-L voting schemes). A secure voting protocol must (at least) satisfy the following fundamental properties:

- Eligibility. Only entitled voters are able to submit a vote (respectively, the votes of unauthorized voters are not counted), and they are able to submit only one single vote.
- Correctness. The tally that pops up at the end of the vote is the correct sum of all valid votes; invalid votes have no influence to the tally.
- Universal verifiability. Anyone can verify that the published tally is correct.
- Secrecy. It is infeasible to find out which voter has submitted which vote. Secrecy should also be satisfied for partial information on votes, as well as for relation between votes of several voters.

* A preliminary version of this text can be found in [Hir01, Chapter 5].

D. Chaum et al. (Eds.): Towards Trustworthy Elections, LNCS 6000, pp. 64–82, 2010.

– Receipt-freeness. The voter cannot obtain a receipt proving the vote he has cast.

The receipt-free property is required to prevent voters from selling their votes. Its importance is disputed within the voting community, as the problem of vote-selling can be seen marginal. However, in a classical voting scheme, the absence of vote-buying can never be demonstrated; even long after a vote, rumors about a vote-buying server cannot be resolved, This is in contrast to the correctness of the result, which can be proven at the end of the vote (using universal verifiability). Hence, to our mind, limited correctness (with universal verifiability) might be acceptable; limited receipt-freeness is not.

Receipt-freeness is not achievable without taking some additional assumption on the communication model (e.g., untappable channels, voting booth) and/or the trust model (e.g., trusted hardware tokens): Evidently, if the vote-buyer can read all communication channels, then the voter's initial randomness, secret-keys etc. are a verifiable receipt for the submitted vote (the vote-buyer can simulate the voter's behavior by using the correct voting program, and compare the communication with the effective communication seen on the channels). This receipt can even be made zero-knowledge for the vote-buyer by using standard techniques (the voter proves knowledge of a secret key matching his public key, some randomness, such that when applying the voting program, the effective communication is produced).

1.2 Contributions

We propose a construction for receipt-free voting protocols based on homomorphic encryption with the following advantages over previous voting protocols:

– Genericalness. The construction as well as the security proofs are generic in the underlying encryption scheme, and can equally be instantiated with Paillier's scheme [Pai99] or with the modified ElGamal scheme [ElG84, CGS97]. Note that the latter is significantly more efficient with comparable security (Paillier requires a bigger field for the same level of security than ElGamal).
– Generality. The new protocol supports K-out-of-L elections for arbitrary K and L. In contrast to most (even non-receipt-free) voting protocols in the literature, we do not have to adjust the security parameter of the underlying encryption scheme when L is large. Furthermore, this is the first receipt-free scheme supporting arbitrary large K without exponential complexity (the complexity of the new scheme is linear in L and independent of K).
– Efficiency. The proposed voting scheme is more efficient than any receipt-free voting scheme in the literature. For K-out-of-L voting (for any K), it requires only three times more communication than the most efficient 1-out-of-L scheme which is *not receipt-free* [CGS97].

Note that the apparent idea for constructing K-out-of-L voting protocols, namely running L parallel instances of a 1-out-of-2 protocol and have each

voter prove that at most K instances contain a 1-vote [BY86], cannot generically be applied in a receipt-free model: For example, in the protocols of [SK95, HS00, BFP$^+$01], the voter does not know the randomness used for encrypting his own vote, and hence he cannot prove any statement on the submitted vote(s).

The new protocol is constructed along the lines of the protocol of [CGS97]: A set of N authorities jointly set-up a secret-key/public-key pair, where the secret-key is shared among the authorities. Every voter then encrypts his vote under the public-key, the authorities compute the sum of all submitted votes with using the homomorphic property of the encryption function, and jointly decrypt (and prove) the tally by applying techniques from threshold cryptography. Receipt-freeness is achieved by techniques similar to those of [LK00, BFP$^+$01]: Every voter must have his encrypted vote re-randomized by a *randomizer*. This randomizer can be a designated authority, or a piece of hardware given to the voter. The randomizer acts as an "observer" [CP92] establishing receipt-freeness, but cannot violate the secrecy or the correctness of the vote. More precisely, the randomizer does not learn the vote, hence cannot violate the privacy of the protocol. Furthermore, the randomizer must prove to the voter that the new encryption is really a re-randomization of the original encryption, hence he cannot cannot violate the correctness of the protocol. However, a malicious randomizer could help a voter to sell his vote.

The security of the protocol is specified with respect to a fixed parameter t: The correctness of the computed tally is guaranteed as long as at least t authorities remain honest during the whole protocol execution, and the secrecy of each vote is guaranteed as long as no t authorities maliciously collaborate with each other. Vote-buying is disabled under the assumption that the randomizer does not collaborate with the vote-buyer, and that the vote-buyer cannot tap the communication between the voter and the randomizer. Therefore, we require that the vote-buyer cannot tap the channels between the voters and the randomizer. We stress that these additional assumptions are required solely for the receipt-freeness of the scheme; even when the randomizer cooperates with the adversary and/or the adversary can tap the channels between the randomizer and the voter, still our voting scheme provides all security properties of non-receipt-free voting schemes. Hence, receipt-freeness is provided as a strict add-on.

Finally, we analyze the security of the protocol of [LK00] and show that it is not receipt-free, in contrast to what is claimed in the paper.

1.3 Previous Work and Comparison

Secret-ballot voting protocols were first proposed by Chaum [Cha81], based on the idea of a mix-net. Cohen (Benaloh) and Fischer [CF85] and Benaloh [Ben87] suggested a voting protocol based on a homomorphic encryption function. The first voting schemes based on blind signatures and anonymous channels were proposed by Chaum [Cha89] and Fujioka, Okamoto, and Ohta [FOO92]. Later, many schemes based on these approaches were published [BY86, Ive91, PIK93, Sak94, SK94, CFSY96, CGS97].

The concept of receipt-freeness was first introduced by Benaloh and Tuinstra [BT94], where also a first receipt-free voting protocol based on homomorphic encryption is proposed. However, their main protocol turned out to be not receipt-free [HS00]. Another receipt-free voting protocol was proposed in [NR94], but as this scheme bases on generic cryptographic tools (like general zero-knowledge proofs) it is very inefficient. We mention that using incoercible multi-party computation [CG96] does not suffice to achieve receipt-freeness: Voters who *want* to sell their vote can use committed random bits in the set-up phase, and can then later prove their vote based on this randomness. In the sequel, we briefly compare our scheme with the most prominent receipt-free voting schemes in the literature.

Sako/Kilian [SK95]. This voting scheme is based on a mix-net channel. The scheme suffers under similar disadvantages as other mix-net voting protocols: it requires a high communication complexity in the mix (especially the cut-and-choose proofs), and the tallying process cannot be performed incrementally (the whole mixing load must be performed after the last vote has been cast). Furthermore, this scheme is vulnerable to the so-called randomization attack [Sch99]: The coercer can force a voter to vote randomly by instructing him which encrypted ballot to take from the generated list. In this scheme, receipt-freeness is assumed under the assumption of physically untappable channel. If an adversary could tap these channels, then not only he could violate the receipt-free property, but also the secrecy property. However, this drawback can be fixed.

Okamoto [Oka96, Oka97]. This scheme uses the blind-signature approach. It requires each voter to be active in three rounds, which is a significant disadvantage in practice. Receipt-freeness is achieved under the (rather demanding) assumption of untappable anonymous channels. An adversary who can violate this assumption can break both receipt-freeness and secrecy of the scheme. It seems unclear how to get rid of this drawback.

Hirt/Sako [HS00]. This protocol uses homomorphic encryption for tallying and a small mix-net for vote generation. This approach awards higher efficiency than the previous approaches. However, also this protocol is vulnerable to the randomization attack [Sch99]. Also this scheme relies on the assumption of untappable channels, and also in this scheme, tapping these channels violates the secrecy of the votes (can be fixed). Furthermore, this protocol implements only 1-out-of-L elections for small L (the computational complexity of decrypting the tally is exponential in L).

Lee/Kim [LK00], Baudron *et al* [BFP$^+$01]. Recently, [LK00] introduced the idea of using a randomizer for achieving receipt-freeness. However, their protocol is insecure (cf. Appendix A). Independently, [BFP$^+$01] proposed a receipt-free voting protocol based on randomizers, using Paillier encryption [Pai99] for secrecy and general diverted proofs [OO89] for receipt-freeness. Paillier encryption makes the scheme less efficient than schemes based on modified ElGamal (like ours): for achieving the same level of security, Paillier requires a bigger security parameter than ElGamal. Further-

more, using general diverted proofs might also yield a high bit complexity; this is not analyzed in the paper. Finally, the protocol is limited to 1-out-of-L votes (in contrast to K-out-of-L votes), and for large L, the security parameter of the underlying encryption scheme must be increased, slowing down all computations.

2 Preliminaries

2.1 Σ-Proofs

A Σ-proof is a three-move special honest-verifier zero-knowledge proof of knowledge. This notion originates from the notion of a Σ-protocol, as introduced by Cramer [Cra96]. We call a Σ-proof *linear* if the verifier's test predicate is linear, i.e., the sum of two accepting conversation is accepting as well. Several Σ-proofs can easily be combined to a new Σ-proof proving knowledge of all (AND-combination) or either (OR-combination) of the witnesses. For the AND-combination, the protocols are run in parallel, but the verifier requests the same challenge for all parallel instances. For the OR-combination, again the verifier requests only one challenge, but the prover is allowed to split this challenge into one sub-challenge for each instance, where the sub-challenges must add up to the challenge. This allows the prover to run the simulator for all but one instance. Note that both the AND- and the OR-combination preserves linearity. Any Σ-proof can be made non-interactive by applying the Fiat-Shamir heuristics [FS86]. Details and formal definitions of Σ-proofs are omitted due to space restrictions.

2.2 Identification Scheme

For voter identification, we assume an identification scheme where the identification protocol can be written as a linear Σ-proof. One can easily verify that Schnorr's identification scheme [Sch91] satisfies this requirement. A voter's secret key is denoted by z_v, the corresponding public key by $Z_v = g^{z_v}$ for an an appropriate generator g. Furthermore, in a model providing receipt-freeness, it is essential that each voter knows his own secret key, and this should be ensured by the underlying public-key infrastructure. A protocol for ensuring knowledge of the secret-key for Schnorr's identification scheme is given in [HS00].

2.3 Designated-Verifier Proofs

We will also make use of so-called designated-verifier proofs. A designated-verifier proof is a proof which is convincing for one particular (designated) verifier, but completely useless when transferred from this designated verifier to any other entity. The notion of designated-verifier proofs was introduced in [JSI96]. The key idea of designated-verifier proofs is to prove knowledge of either the witness in question, or of the secret key of the designated verifier. Formally, the proof will be constructed as the OR-combination of the proof in question and a proof of knowledge of the designated verifier's secret-key.

3 The Encryption Function

We first state the requirements on the encryption function, and then show that the two classical homomorphic encryption functions, namely modified ElGamal and Paillier, satisfy the requirements. For space limitations, the full descriptions have been deleted from this extended abstract.

3.1 Requirements

We consider a semantically-secure probabilistic public-key encryption function $E_Z : \mathbb{V} \times \mathbb{R} \to \mathbb{E}, (v, \alpha) \mapsto e$, where Z denotes the public key, $\mathbb{V}$ denotes a set of votes, $\mathbb{R}$ denotes the set of random strings, and $\mathbb{E}$ denotes the set of encryptions. We write E instead of E_Z for shorthand. The decryption function is $D_z : \mathbb{E} \to \mathbb{V}, e \mapsto v$, where z denotes the secret key. Again, we write D instead of D_z. Note that the computational complexity of the decryption function D_z may be polynomial in the decrypted cleartext v. For arbitrary large v, decryption is not required to be feasible.

We assume that E is a group homomorphism, i.e., $E(v_1, \alpha_1) \oplus E(v_2, \alpha_2) = E(v_1 + v_2, \alpha_1 \boxplus \alpha_2)$ for the corresponding group operations $+$ in $\mathbb{V}$, $\boxplus$ in $\mathbb{R}$, and $\oplus$ in $\mathbb{E}$, respectively. Note that the group operation in $\mathbb{V}$ must be modular addition, but the operations in the other groups can be arbitrary.

Furthermore, we require E to be *q-invertible* for a given $q \in \mathbb{Z}$ meaning that for every encryption e, the decryption v and the randomness α of qe can be efficiently computed, i.e., the function $D_q : e \mapsto (v_q, \alpha_q)$ such that $qe = E(v_q, \alpha_q)$ is efficient (given Z). Additionally, we require that there is a number $u \leq q$, large enough that $1/u$ is considered negligible, with the property that all integers smaller than u are co-prime with q, i.e., $\forall u' < u : \gcd(u', q) = 1$. This property will be used in the knowledge extractors of the Σ-proofs.[1] Note that v_q must be 0 due to the semantic security of E and the group structure of $\mathbb{V}$. This notion of q-invertibility is inspired by the notion of q-one-way group-homomorphism of Cramer [Cra96, CD98].

Finally, we require the existence of verifiable distributed protocols for key generation and for decryption. Note that every encryption scheme can be turned into a threshold variant by applying techniques of general multi-party computations, but such an approach would be rather inefficient.

3.2 Modified ElGamal Encryption

The ElGamal encryption function [ElG84], modified according to [CGS97], enhanced with a threshold setup protocol and a threshold group decryption [Ped91], satisfies all above properties. When used over a finite field G with $|G| = q$ prime, then the encryption function is q-invertible, and we set $u = q$.

[1] More generally, it would be sufficient to assume that for a given large u, there exists an efficiently computable and invertible bijection from $\mathbb{Z}_u$ onto a subset of $\mathbb{Z}_q$, where each element in this subset is co-prime with q.

What should still be mentioned here is that computational complexity for decryption is linear in the size of the cleartext. However, in the context of this work, this issue will not be a problem.

3.3 Paillier Encryption

Also the probabilistic encryption function of Paillier [Pai99], enhanced by threshold setup and decryption [FPS00, DJ01], satisfies all required properties. For an RSA modulus n, this encryption function is n-invertible, and let u be a large prime which is guaranteed to be smaller than the smaller prime factor of n (e.g., we let n be the product of two secret 512-bit integers, and let u be a fixed 511-bit prime).

4 Re-encrypting and Proving Re-encryptions

A random re-encryption e' of a given encryption $e = E(v, \alpha)$ is an encryption with the same vote v, but a new (independently chosen) randomness α'. Such a re-encryption can be computed by adding a random encryption of 0 to e. Formally, a witness $\xi \in_R \mathbb{R}$ is chosen at random, and $e' = e \oplus E(0, \xi)$, i.e.,

$$e' = R(e, \xi) = e \oplus E(0, \xi).$$

Due to the homomorphic property of E, the randomness in e' is uniformly distributed over $\mathbb{R}$ for a uniformly chosen $\xi \in_R \mathbb{R}$.

Proving that a given e' is indeed a re-encryption of e can easily be done by proving that $e' \ominus e$ is an encryption of 0. We present a simple linear Σ-proof for proving knowledge of a witness α such that $e = E(0, \alpha)$ for any given encryption e. The challenge for the protocol is uniformly selected from $\mathbb{Z}_u$, and the soundness of the protocol is proven under the assumption that E is q-invertible and that $\forall u' < u : \gcd(u', q) = 1$.

Prover		**Verifier**
knows e, α, s.t. $e = E(0, \alpha)$		knows e
$\alpha' \in_R \mathbb{R}$, $e' = E(0, \alpha')$	$\xrightarrow{e'}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_u$
$\beta = c\alpha \boxplus \alpha'$	$\xrightarrow{\beta}$	$E(0, \beta) \stackrel{?}{=} ce \oplus e'$

Completeness of the protocol is obvious by inspection. We next show that the protocol satisfies special soundness, by showing that if for any e' the prover can

reply to two different challenges $c_1 \neq c_2$, then he can compute a witness α with $e = E(0, \alpha)$. So assume that for two different challenges c_1 and c_2, the prover can answer with β_1 and β_2, respectively, such that both conversations (e', c_1, β_1) and (e', c_2, β_2) are accepting, i.e., $E(0, \beta_1) = c_1 e \oplus e'$ and $E(0, \beta_2) = c_2 e \oplus e'$, and hence $E\big(0, \beta_1 \boxminus \beta_2\big) = (c_1 - c_2)e$. Without loss of generality assume that $c_1 > c_2$, hence $0 < c_1 - c_2 < u$, and $\gcd(c_1 - c_2, q) = 1$. Hence we can apply the extended Euclidean algorithm to find two integers a and b such that $a(c_1 - c_2) + bq = 1$. Then, using the q-invertibility of the encryption function we compute α_q such that $qe = E(0, \alpha_q)$. This results in

$$\begin{aligned} e &= \big(a(c_1 - c_2) + bq\big)e = a(c_1 - c_2)e \oplus bqe \\ &= aE\big(0, \beta_1 \boxminus \beta_2\big) \oplus bE(0, \alpha_q) = E\big(0, a(\beta_1 \boxminus \beta_2) \boxplus b\alpha_q\big). \end{aligned}$$

This concludes that indeed e encrypts 0 with witness $\alpha = a(\beta_1 \boxminus \beta_2) \boxplus b\alpha_q$.

We now show that the protocol is special honest-verifier zero-knowledge by constructing a simulator. The simulator is constructed as follows: For any given $c \in \mathbb{Z}_u$, we select β from $\mathbb{R}$ at random, and set $e' = E(0, \beta) \ominus ce$. Obviously, the probability distribution of β is the same as the distribution of a real conversation in which α is chosen uniformly distributed (for the same challenge c).

It is important to note that the simulator can also be applied for an encryption e which does not encrypt 0, and the simulated conversation is computationally indistinguishable from a conversation where e encrypts 0 (an efficient distinguisher of these conversations would contradict the semantic security of the encryption function). This indistinguishability is important when several re-encryption proofs are OR-combined.

5 Non-Receipt-Free Voting Protocol

In this section we present a very simple K-out-of-L voting protocol which is *not* receipt free. The protocol is similar to the voting protocol of [CGS97], but due to a different ballot encoding it allows for votes with $K \geq 2$ and provides a substantially better computation complexity for $L > 2$. The protocol will be used as basis for the receipt-free protocol in the next section.

5.1 Model

We consider a model with N authorities $A_1, \ldots, A_N$ and M voters. Communication takes place by means of a bulletin board which is publicly readable, and which every participant can write to (into his own section), but nobody can delete from. The bulletin board can be considered as an authenticated public channel with memory. A threshold t denotes the number of authorities that is required for decrypting the tally, and which also is able to annihilate the secrecy of any vote.

5.2 Ballots

A ballot consists of a vector of votes, $\vec{v} = (v_1, \ldots, v_L)$, where v_i is the vote for the i-th candidate. In a K-out-of-L election, a ballot is valid if and only if each vote v_i is either 0 or 1, and the votes on the ballot sum up to K. If voters should be allowed to vote for less than K candidates, then this is modeled as K-out-of-$(L+K)$ election, where the latter K candidates represent "abstain" and will not be tallied.

As simple notation, we write $E(\vec{v}, \vec{\alpha})$ for L-vectors $\vec{v} = (v_1, \ldots, v_L)$ and $\vec{\alpha} = (\alpha_1, \ldots, \alpha_L)$, meaning the component-wise application of the encryption function, i.e., $E(\vec{v}, \vec{\alpha}) = \big(E(v_1, \alpha_1), \ldots, E(v_L, \alpha_L)\big)$. Analogously, we defined $R(\vec{e}, \vec{\xi})$, $\vec{v}_1 + \vec{v}_2$, $\vec{\alpha}_1 \boxplus \vec{\alpha}_2$, and $\vec{e}_1 \oplus \vec{e}_2$.

5.3 Set-up

In the set-up phase, the authorities jointly generate a uniformly distributed secret key and the corresponding public key for the encryption scheme, where the secret key is shared among the authorities, and the public key is publicly known. A protocol for (verifiable) generating a sharing of a randomly chosen secret key and a public key is a requirement on the encryption function.

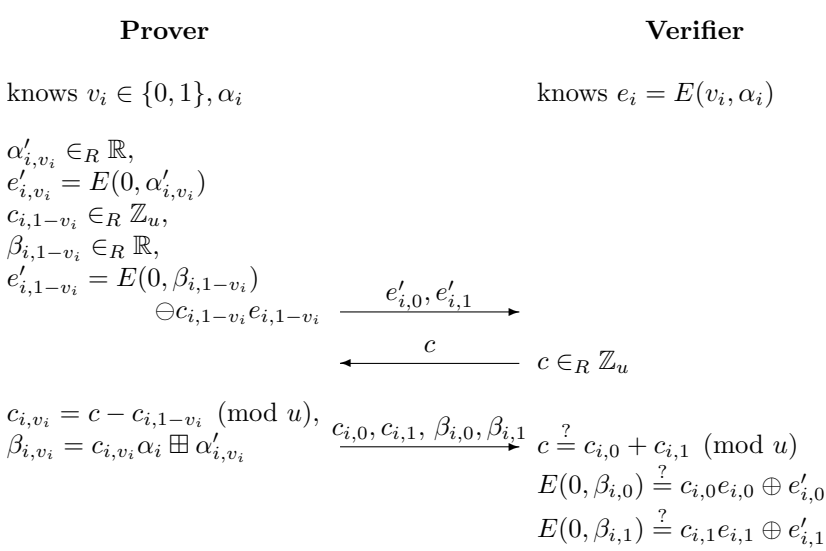
5.4 Casting a Ballot

A ballot is cast as follows: The voter constructs a random encryption $\vec{e} = E(\vec{v}, \vec{\alpha})$ for his vote vector $\vec{v}$ and randomness $\vec{\alpha} \in_R \mathbb{R}^L$, and posts it onto the bulletin board. Furthermore, the voter posts a proof of validity. A ballot $\vec{v} = (v_1, \ldots, v_L)$ is valid if and only if $v_i \in \{0, 1\}$ for $i = 1, \ldots, L$ and $\sum v_i = K$. In the following we construct a (finally non-interactive) validity proof for the encrypted ballot $\vec{e} = (e_1, \ldots, e_L)$.

The validity proof is constructed as the AND-combination of a Σ-proof for each $i = 1, \ldots, L$, each stating that e_i is an encryption of either 0 or 1, and a Σ-proof stating that $e_1 \oplus \ldots \oplus e_L$ is an encryption of K. The proofs that e_i is an encryption of either 0 or 1 is constructed as an OR-combination of a proof stating that e_i encrypts 0 and a proof stating that e_i encrypts 1.

For easier notation, we write $e_{i,0} = e_i$ and $e_{i,1} = e_i \ominus E(1, 0)$, that is, e_{i,v_i} is an encryption of 0 with randomness α_i. Furthermore, we write $e_\Sigma = (e_1 \oplus \ldots \oplus e_L)$, $\alpha_\Sigma = \alpha_1 \boxplus \ldots \boxplus \alpha_L$, and $e_{\Sigma,K} = e_\Sigma \ominus E(K, 0)$. A ballot is valid exactly if for each i, either $e_{i,0}$ or $e_{i,1}$ encrypts 0, and $e_{\Sigma,K}$ encrypts 0. This proof can be constructed straight-forward as AND-combination of OR-combinations of proofs that a given encryption contain 0 (Section 4).

The following protocol is a OR-combined Σ-proof of knowledge of a witness α_i such that $e_{i,0} = E(0, \alpha_i)$ OR $e_{i,1} = E(0, \alpha_i)$. In the protocol for proving $e_{i,1-v_i}$, the prover applies the simulator.



The finally validity proof is the AND-combination (i.e., parallel execution, but same challenge for all instances) of the above protocol for $i = 1, \ldots, L$ plus a Σ-proof that $e_{\Sigma,K}$ encrypts 0. A (short) non-interactive proof is then the vector $[c, c_{1,0}, \ldots, c_{L,0}, \beta_{1,0}, \ldots, \beta_{L,0}, \beta_{1,1}, \ldots, \beta_{L,1}, \beta_\Sigma]$ satisfying

$$\begin{aligned} c \stackrel{?}{=} H\Big(& E(0,\beta_{1,0}) \ominus c_{1,0} e_{1,0} \Big\| \ldots \Big\| E(0,\beta_{L,0}) \ominus c_{L,0} e_{L,0} \Big\| \\ & E(0,\beta_{1,1}) \ominus (c - c_{1,0}) e_{1,1} \Big\| \ldots \Big\| E(0,\beta_{L,1}) \ominus (c - c_{L,0}) e_{L,1} \Big\| \\ & E(0,\beta_\Sigma) \ominus c e_{\Sigma,K} \Big). \end{aligned}$$

The proof takes $3L + 2$ field elements.

5.5 Tallying

Tallying is performed for each candidate separately: For candidate i, the i-th components of each valid ballot are summed up (using the homomorphic property of the encryption function) and decrypted (using the verifiable decryption protocol of the encryption function). Note that it is known in advance that the decrypted tally will be in the range $(0, M)$; hence, decryption is efficient also for the modified ElGamal scheme.

5.6 Security Analysis

The privacy of the proposed protocol is guaranteed under the assumption that no t authorities maliciously pool their information, plus the assumption that the encryption function is semantically secure. The tally is correct if at least t authorities honestly participate in the tally decryption, plus the assumption that no verifier can cast an invalid ballot. The probability that an invalid ballot passes the validity proof is negligible if $1/u$ is negligible. The scheme is not receipt-free.

5.7 Efficiency Analysis and Comparison

We analyze the communication efficiency of this voting protocol for a K-out-of-L vote. The number of bits used to store one group element is denoted by B.

We ignore the costs for initialization and decryption of the final tally — they are independent of the number M of voters. It remains to count the costs for casting and proving votes. In order to cast his vote, every voter sends his encrypted ballot (LB bits) together with the validity proof ($(3L+2)B$ bits) onto the bulletin board. In total, $(4L+2)MB$ bits are posted to the bulletin board.

As comparison, in [CGS97] a ballot takes only B bits, but the proof takes $2LB$ bits. This gives a total of $(2L+1)MB$ bits. However, this scheme only allows for $K=1$ (for larger K the communication complexity would grow exponentially), and its decryption function is computationally inefficient for large L.

6 Receipt-Free Voting Protocol based on Randomizers

In this section, the voting protocol of Section 5 is enhanced to be receipt-free. Therefore, the procedure for casting a vote must be modified.

The protocol relies on special authority called *randomizer*, who re-randomizes encrypted ballots of the voters. More precisely, each voter constructs an encrypted ballot containing his vote and secretly sends it to the randomizer. The randomizer re-encrypts this ballot and posts it to the bulletin board. Furthermore, the randomizer proves to the voter (in designated-verifier manner) that indeed the new encrypted ballot contains the same vote, and the voter and the randomizer jointly generate a proof of validity for this new ballot.

In the following, we briefly discuss the new model, then formally describe the new protocol for casting a ballot.

6.1 Model

In addition to the model of Section 5, we assume a special authority called *randomizer*. Collaboration of the randomizer with a vote-buyer or coercer cannot be tolerated. The randomizer does not learn the vote of any voter, nor can he interfere with the correctness of the tally, but he can reject to re-encrypt the ballot of any voter and thereby prevent this voter from participating the vote. Therefore, several randomizers can be used.

We assume that the communication channels between the voter and the randomizer are untappable for the vote-buyer. The privacy of these channel must be physical, in such a way that even the recipient cannot prove to the vote-buyer what was received from the channel (of course, the recipient can record all received data, but he must not be able to *prove* that he received a particular string). The untappable channels need not to be authenticated.

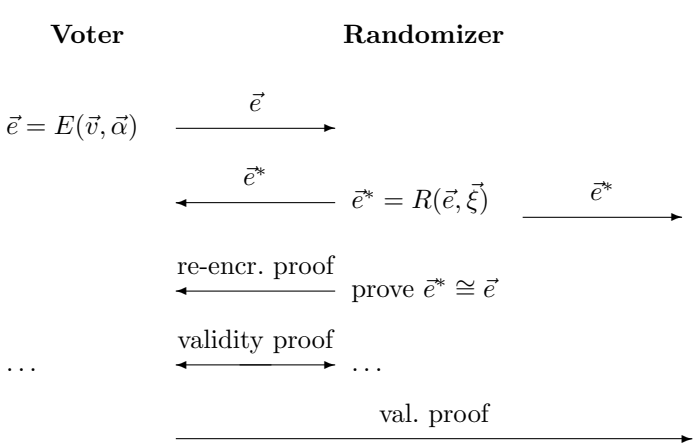
Furthermore, to each voter a secret key and a public key is associated, where the public key must be publicly known and the secret key must be kept private. We stress that in order to achieve receipt-freeness it must be guaranteed that

each voter knows the secret key corresponding to his known public key (but the voter is allowed to reveal the secret-key to the coercer). A protocol ensuring so is given in [HS00].

Note that all above requirements are uniquely relevant for receipt-freeness. If they are not met, then the proposed voting scheme still achieves all security requirements but receipt-freeness.

6.2 Casting a Ballot

A ballot is cast as follows: The voter constructs a random encryption $\vec{e} = E(\vec{v}, \vec{\alpha})$ of his vote vector $\vec{v}$ with randomness $\vec{\alpha} \in_R \mathbb{R}^L$, and sends it through the untappable channel to the randomizer. The randomizer then computes random re-encryption $\vec{e}^* = R(\vec{e}, \vec{\xi})$ of $\vec{e}$, and proves to the voter in designated-verifier manner that indeed $\vec{e}^*$ is a re-encryption of $\vec{e}$. Then, the voter and the randomizer jointly generate a validity proof for $\vec{e}^*$, without the randomizer learning anything about the vote vector $\vec{v}$, and without the voter learning anything about the re-encryption witness $\vec{\xi}$. Finally, the randomizer posts the validity proof to the bulletin board, and the voter posts the re-encrypted ballot $\vec{e}^*$.



Designated-verifier re-encryption proof. The purpose of this proof is to have the randomizer prove to the voter that the new encryption $\vec{e}^*$ is indeed a re-encryption of $\vec{e}$. However, this proof must be non-transferable, such that the verifier cannot convince someone else that $\vec{e}^*$ is a re-encryption of $\vec{e}$. This is achieved by a designated-verifier proof (cf. Section 2.3): The randomizer proves knowledge of either a re-randomization witness $\vec{\xi}$ with $\vec{e}^* = R(\vec{e}, \vec{\xi})$, or of the voter's secret key. Obviously, this proof is convincing for the voter, but completely useless when transferred from the voter to a third party.

The proof is constructed as an OR-combination of the Σ-proof that the encryption $\vec{e}^* \ominus \vec{e}$ contains the vote $\vec{0}$ (which again is an AND-combination that $E(0, \xi_i) = e_i^* \ominus e_i$ for $i = 1, \ldots, L$), and the Σ-proof of the identification scheme.

The resulting proof will require $L+1$ encryptions in the first message, then one challenge, and in the final message, 2 sub-challenges plus $2L+1$ randoms. Non-interactively, the proof can be made in $2L+3$ field elements.

We show the proof for Schnorr's identification scheme. We denote the voter's secret key with z_v and the public key with $Z_v = g^{z_v}$. For shorthand, we set $\vec{e}^{\leftarrow} = \vec{e}^* \ominus \vec{e}$. The following protocol is a Σ-proof of knowledge of either the voter's secret key z_v satisfying $g^{z_v} = Z_v$, OR a witness $\vec{\xi}$ satisfying $\vec{e}^{\leftarrow} = E(\vec{0}, \vec{\xi})$.

Randomizer		**Voter**
knows $\vec{e}^{\leftarrow}, \vec{\xi}$		knows $\vec{e}^{\leftarrow} = E(\vec{0}, \vec{\xi})$
$\vec{\alpha}' \in_R \mathbb{R}^L,\ \vec{e}' = E(\vec{0}, \vec{\alpha}')$ $c_2 \in_R \mathbb{Z}_u,\ s_2 \in_R \mathbb{Z}_q,$ $t_2 = g^{s_2} Z_v^{-c_2}$	$\xrightarrow{\vec{e}', t_2}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_u$
$c_1 = c - c_2 \pmod u$ $\vec{\beta} = c_1 \vec{\xi} \boxplus \vec{\alpha}'$	$\xrightarrow{c_1, c_2, \vec{\beta}, s_2}$	$c \stackrel{?}{=} c_1 + c_2 \pmod u$ $E(\vec{0}, \vec{\beta}) \stackrel{?}{=} c_1 \vec{e}^{\leftarrow} \oplus \vec{e}'$ $g^{s_2} \stackrel{?}{=} Z_v^{c_2} \cdot t_2$

A non-interactive version of the proof is the vector $[c_1, c_2, \vec{\beta}, s_2]$ satisfying the equation

$$c_1 + c_2 \stackrel{?}{=} H\Big(E(0,\beta_1) \ominus c_1 e_1^- \Big\| \ldots \Big\| E(0,\beta_L) \ominus c_1 e_L^- \Big\| g^{s_2} Z_v^{-c_2}\Big).$$

This proof takes $L+3$ field elements.

Validity proof. The validity proof is a non-interactive proof that the randomized encryption $\vec{e}^*$ contains a valid vote, i.e., each e_i is an encryption of either 0 or 1, and in total, there are exactly K encryptions of 1. Neither the voter (who does not know the re-encryption witness $\vec{\xi}$) nor the randomizer (who does not know the ballot $\vec{v}$) can generate the proof on their own, hence they need to generate the proof interactively. The generation of the proof proceeds in two steps: First, the voter and the randomizer engage in an interactive protocol which gives to the randomizer a uniformly selected random non-interactive validity proof for $\vec{e}$ (a so-called diverted proof [OO89]). Then, the randomizer adjusts this proof into a validity proof for $\vec{e}^*$.

Generating a diverted validity proof for $\vec{e}$. We first observe that validity proofs are linear Σ-protocols; hence, the sum of two accepting validity proofs (for the same vote $\vec{e}$) is again an accepting validity proof for $\vec{e}$. A diverted version of the validity proof can hence be generated as the sum of the normal validity proof (from Section 5.4) and a uniformly random validity proof for $\vec{e}$, generated by the

simulator. More precisely, a diverted proof for $\vec{e}$ is generated as follows: First, the randomizer used the simulator to generate a random validity proof for $\vec{e}$ with challenge 0 (here we use that the Σ-proof is special zero-knowledge). Then, the voter and the randomizer engage in an interactive validity proof for $\vec{e}$. The diverted proof is then the sum of these two proofs.

More precisely, the randomizer selects random "displacements" $c'_{i,0} \in_R \mathbb{Z}_u$, $c'_{i,1} = -c'_{i,0}$, and $\beta'_{i,0}, \beta'_{i,1} \in_R \mathbb{R}$ for $i = 1, \ldots, L$. The displacements are chosen such that $c'_{i,1} + c'_{i,0} = 0$ for all i, i.e., the sum of the new sub-challenges will not change. Upon reception of the first message $(e'_{i,0}, e'_{i,1})$ of the interactive Σ-proof, the randomizer computes the first "message" of the non-interactive diverted proof as

$$e''_{i,0} = e'_{i,0} \oplus E(0, \beta'_{i,0}) \ominus c'_{i,0} e_{i,0}, \qquad e''_{i,1} = e'_{i,1} \oplus E(0, \beta'_{i,1}) \ominus c'_{i,1} e_{i,1}$$

and asks as challenge $c = H(e''_{i,0}, e''_{i,1})$. When receiving the third message $(c_{i,0}, c_{i,1}, \beta_{i,0}, \beta_{i,1})$, the randomizer computes the third "message" $(c''_{i,0}, c''_{i,1}, \beta''_{i,0}, \beta''_{i,1})$ of the non-interactive diverted proof as

$$c''_{i,0} = c_{i,0} \boxplus c'_{i,0}, \quad c''_{i,1} = c_{i,1} \boxplus c'_{i,1}, \quad \beta''_{i,0} = \beta_{i,0} \boxplus \beta'_{i,0}, \quad \beta''_{i,1} = \beta_{i,1} \boxplus \beta'_{i,1}.$$

One can easily verify that the diverted conversation $\big((e''_{i,0}, e''_{i,1}),\ c,\ (c''_{i,0}, c''_{i,1}, \beta''_{i,0}, \beta''_{i,1})\big)$ is accepting for e_i (due to the linearity of the validity proof). Note that in the interactive validity proof, L such proofs are run in parallel with the same challenge (AND-combination). The above diversion is then applied on each parallel instance independently. Furthermore, as the original interactive proof is *honest-verifier* zero-knowledge only, one must ensure that the challenge of the randomizer is chosen at random. This is achieved by having the randomizer not only send c to the voter, but instead all $e''_{i,j}$, such that the voter can apply the hash function himself. Obviously, then the voter knows that the challenge is selected at random under the random oracle assumption.

Adjusting the diverted validity proof to $\vec{e}^*$. With the above protocol, the randomizer can construct a diverted non-interactive validity proof for $\vec{e}$. It remains to convert this proof into a validity proof for $\vec{e}^*$. So consider the following diverted validity proof for $\vec{e}$: $[c, c''_{1,0}, \ldots, c''_{L,0}, \beta''_{1,0}, \ldots, \beta''_{L,0}, \beta''_{1,1}, \ldots, \beta''_{L,1}, \beta''_{\Sigma}]$. Then one can easily verify that the following vector is a validity proof for the re-encrypted ballot $\vec{e}^* = \vec{e} \oplus E(0, \vec{\xi})$:

$$[c, c''_{1,0}, \ldots, c''_{L,0}, \quad \beta''_{\Sigma} \boxplus (\xi_1 \boxplus \ldots \boxplus \xi_L), \\ \beta''_{1,0} \boxplus c''_{1,0}\xi_1, \ldots, \beta''_{L,0} \boxplus c''_{L,0}\xi_L, \quad \beta''_{1,1} \boxplus c''_{1,1}\xi_1, \ldots, \beta''_{L,1} \boxplus c''_{L,1}\xi_L].$$

6.3 Security Analysis (of the vote-casting protocol)

The vote-casting protocol must satisfy two requirements: First, the randomizer must not learn the vote. Second, the voter must not be able to proof any correspondence between the original ballot $\vec{e}$ and the re-encrypted ballot $\vec{e}^*$.

In order to show that the randomizer does not learn the voters vote, we only need to analyze the protocol for generating the diverted proof. This protocol is an interactive honest-verifier zero-knowledge proof of knowledge, which gives no information to the verifier (the randomizer) when the challenge is chosen honestly at random. Due to the modification that the voter applies the hash function by himself, it is clear that under the random oracle assumption the challenge is random, hence the protocol is zero-knowledge, and the randomizer learns nothing about the vote.

Secondly, in order to show that the protocol is receipt-free, we make use of two observations: First, the generated diverted validity proof is uniformly chosen among all validity proofs for $\vec{e}^*$, and second, the randomizer does not give any information beyond the diverted proof to the voter. The second observation can be verified by inspecting the protocol, but the first observation needs some more explanations: The diverted validity proof is the sum of the interactive proof as executed with the voter (and hence known to the voter), and a simulated proof which is selected completely uniformly among all accepting proofs (except for the challenge, which is random in the random oracle model). Hence the diverted proof is random and statistically unlinked to the interactive protocol that the voter is involved in. From the voter's viewpoint, the validity proof for $\vec{e}^*$ is uniformly random and independent from all his own information.

Once more we stress that even a malicious randomizer cannot interfere with the secrecy or the correctness of the voting protocol. He only receives an *encrypted* ballot, and he must prove to the voter that the new ballot is a re-encryption of the original ballot.

6.4 Efficiency Analysis and Comparison

We consider K-out-of-L voting, and denote the number of bits per group element with B. As usual, we ignore the costs for initialization and decryption of the final tally.

In order to cast his vote, every voter sends the ballot to the randomizer LB bits, who sends a re-encryption and a re-encryption proof to the voter ($LB + (L+3)B$ bits). Then, the voter and the randomizer run the interactive validity protocol ($(6L+3)B$ bits), and the voter posts the randomized ballot (LB bits) and the randomizer posts the non-interactive proof to the bulletin board ($(3L+2)B$ bits). This gives a total of $(9L+6)MB$ bits sent through the untappable channels, and $(4L+2)MB$ bits sent to the bulletin board.

In comparison, the 1-out-of-L voting protocol of [HS00] with N authorities and M voters requires $4LMNB$ bits sent through the untappable channels and $2L^2MNB$ bits posted to the bulletin board. For $K \geq 2$, this protocol has exponential communication complexity. Furthermore, the protocol has exponential computation complexity in L, and is hence applicable only for very small L.

Finally, we compare the proposed protocol with the 1-out-of-L voting protocol of [BFP+01]. The exact communication complexity of their protocol cannot be determined, as they do not provide a concrete diverted proof. As a rough estimate, the protocol communicates $18LMB$ bits over the untappable channels.

The size of their validity proof stored on the bulletin board is (according to their analysis) $(9L + 11)MB$. Furthermore, as they are restricted to Paillier encryption, they require a larger B than our scheme with ElGamal encryption for the same security level. Furthermore, they must require $B \geq L \log_2 M$ (a message must have enough bits for the tally of each of the L candidates), which for large L might require increasing B. Also their scheme cannot be used for $K \geq 2$; the size of the validity proof would grow exponentially.

6.5 Hardware Randomizer

In the proposed scheme, the randomizer essentially does not need to communicate with the bulletin board or the authorities (he can send the diverted validity proof signed to the voter, who then casts it on the bulletin board — a vote on the bulletin board is accept only if it is signed by the randomizer). This allows for a hardware-based receipt-free voting scheme: Every voter receives a personalized randomization-token, which performs the randomization of the vote, and generates a signed diverted validity proof for the randomized vote. Note that this randomization device acts as an "observer" [CP92]: It does not learn the vote, nor can it falsify it. Even when the vote authorities would distribute bad randomization tokens to the voter, still the privacy and the correctness of the vote would be guaranteed (but not the receipt-freeness). However, the device could reject to provide a proper validity proof; but in this case, the voter could demonstrate other people that his token is broken, and could get a new one.

7 Conclusions

We have presented a generic receipt-free voting scheme, which is secure with *any* homomorphic encryption scheme satisfying the required properties. There is no need to adapt the protocol and proofs to the encryption function, as is necessary for most voting schemes in the literature.

The resulting voting scheme is more efficient than any other receipt-free voting scheme. For K-out-of-L votes and N authorities, the communication complexity per voter is linear in L and independent of K and N. No other scheme in the literature has these properties.

For 1-out-of-L votes, the storage complexity on the bulletin board is the same that of the most efficient voting protocol which is not receipt-free [CGS97]. However, due to the communication with the randomizer, our communication complexity is about 3 times higher.

To the best of our knowledge, the presented scheme is the first scheme which can be based on ElGamal encryption without having a computation complexity growing exponentially in K. There are schemes with efficient computation also for large K, but they base on Paillier encryption [FPS00, DJ01]. Such schemes rely one stronger cryptographic assumptions and require larger security parameters, resulting in bigger constants in the computation and communication complexities.

References

[Ben87] J. C. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, Dec. 1987.

[BFP+01] O. Baudron, P.-A. Fouque, D. Pointcheval, G. Poupard, and J. Stern. Practical multi-candidate election system. In *Proc. 20th ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.

[BT94] J. C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proc. 26th ACM Symposium on the Theory of Computing (STOC)*, pp. 544–553. ACM, 1994.

[BY86] J. C. Benaloh and M. Yung. Distributing the power of a government to enhance the privacy of voters. In *Proc. 5th ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 52–62, Aug. 1986.

[CD98] R. Cramer and I. Damgård. Zero-knowledge for finite field arithmetic. Or: Can zero-knowledge be for free? In *Advances in Cryptology — CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pp. 424–441, 1998.

[CF85] J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographically secure election scheme. In *Proc. 26th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 372–382. IEEE, 1985.

[CFSY96] R. Cramer, M. K. Franklin, B. Schoenmakers, and M. Yung. Multi-authority secret-ballot elections with linear work. In *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pp. 72–83. IACR, Springer-Verlag, May 1996.

[CG96] R. Canetti and R. Gennaro. Incoercible multiparty computation. In *Proc. 37th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 504–513, 1996.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Advances in Cryptology — EUROCRYPT '97*, Lecture Notes in Computer Science, 1997.

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cha89] D. Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In *Advances in Cryptology — EUROCRYPT '89*, volume 434 of *Lecture Notes in Computer Science*, pp. 177–182, 1989.

[CP92] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Advances in Cryptology — CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, 1992.

[Cra96] R. Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and Univ. of Amsterdam, Nov. 1996.

[DJ01] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *International Workshop on Practice and Theory in Public Key Cryptography (PKC) 2001*, 2001.

[ElG84] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology — CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, 1984.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *Advances in Cryptology — AUSCRYPT '92*, pp. 244–251, 1992.

[FPS00] P.-A. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography 2000*, Lecture Notes in Computer Science, 2000.

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pp. 186–194, 1986.

[Hir01] Martin Hirt. *Multi-Party Computation: Efficient Protocols, General Adversaries, and Voting*. PhD thesis, ETH Zurich, 2001. Reprint as vol. 3 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-747-2, Hartung-Gorre Verlag, Konstanz, 2001.

[HS00] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pp. 539–556, 2000.

[Ive91] K. R. Iversen. A cryptographic scheme for computerized general elections. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pp. 405–419. IACR, Springer-Verlag, 1991.

[JSI96] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pp. 143–154, 1996.

[LK00] B. Lee and K. Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. In *Japan-Korea Joint Workshop on Information Security and Cryptology (JW-ISC2000)*, pp. 101–108, 2000.

[NR94] V. Niemi and A. Renvall. How to prevent buying of votes in computer elections. In *Advances in Cryptology — ASIACRYPT '94*, volume 917 of *Lecture Notes in Computer Science*, pp. 164–170, 1994.

[Oka96] T. Okamoto. An electronic voting scheme. In *Proc. of IFIP '96, Advanced IT Tools*, pp. 21–30. Chapman & Hall, 1996.

[Oka97] T. Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Proc. of Workshop on Security Protocols '97*, volume 1361 of *Lecture Notes in Computer Science*, pp. 25–35, 1997.

[OO89] T. Okamoto and K. Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. In *Advances in Cryptology — EUROCRYPT '89*, volume 434 of *Lecture Notes in Computer Science*, pp. 134–148, 1989.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pp. 223–238, 1999.

[Ped91] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pp. 522–526, 1991.

[PIK93] C. Park, K. Itoh, and K. Kurosawa. Efficient anonymous channel and all/nothing election scheme. In *Advances in Cryptology — EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pp. 248–259, 1993.

[Sak94] K. Sako. Electronic voting schemes allowing open objection to the tally. *Transactions of IEICE*, E77-A(1), Jan. 1994.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 1991.

[Sch99] B. Schoenmakers, 1999. Personal communication.
[SK94] K. Sako and J. Kilian. Secure voting using partially compatible homomorphisms. In *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pp. 411–424. IACR, Springer-Verlag, 1994.
[SK95] K. Sako and J. Kilian. Receipt-free mix-type voting scheme – A practical solution to the implementation of a voting booth. In *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pp. 393–403. Springer-Verlag, 1995.

A Analysis of the Lee-Kim Protocol

In this section, we show that the protocol of Kim and Lee [LK00] is not receipt-free, opposed to what is claimed in the paper.

A.1 Key Ideas of [LK00]

The protocol of [LK00] is based on the assumption of an *honest verifier* who ensures the validity of all cast votes. Each voter sends an encryption e of his vote to this honest verifier and proves its validity. Then, the honest verifier sends a random encryption e' of 0 to the voter and proves (with a three-move honest-verifier zero-knowledge protocol) that indeed e' is an encryption of 0. The final ballot of the voter is $e^* = e + e'$, which obviously contains the same vote as e, but different randomness. All communication between the voter and the randomizer must take place over an untappable channel.

Note that in this protocol a malicious "honest verifier" can help a voter to cast an invalid vote and thereby falsify the outcome of the whole vote. In our opinion, such a protocol in which the correctness of the tally relies on the trustworthiness of a single entity is questionable.

A.2 How to Construct a Receipt

The voter can easily construct a receipt: In the protocol where the honest verifier proves to the voter that indeed e' is an encryption of 0, the voter can choose the challenge as the output of a hash function applied to the message in the first move. This makes the transcript of the protocol a non-interactive proof (according to Fiat-Shamir heuristics) that e' is an encryption of 0. Hence, the values e', e, the witness of e, and this proof are a receipt of the cast vote e^*.