Tree-Homomorphic Encryption and Scalable Hierarchical Secret-Ballot Elections

Aggelos Kiayias[1] and Moti Yung[2]

[1] University of Athens, Dept. of Informatics **, Athens, Greece. E-mail: aggelos@di.uoa.gr.
[2] Google Inc. and Computer Science, Columbia University, New York, NY, USA. Email: moti@cs.columbia.edu

Abstract. In this work we present a new paradigm for trust and work distribution in a hierarchy of servers to achieve scalability of work and trust properties. The paradigm is implemented with a decryption capability which is distributed and forces a workflow along a tree structure, enforcing distribution of the workload as well as fairness and partial disclosure (privacy) properties. We call the method "tree-homomorphic" since it extends traditional homomorphic encryption and we exemplify its usage within a large scale election scheme, showing how it contributes to the properties that such a scheme needs. We note that existing design models over which e-voting schemes have been designed for do not adapt to scale with respect to a combination of privacy and trust (fairness); thus we present a model emphasizing the scaling of privacy and fairness in parallel to the growth and distribution of the election structure. We present two instantiations of e-voting schemes that are robust, publicly verifiable, and support multiple candidate ballot casting employing tree-homomorphic encryption schemes. We extend the scheme to allow the voters in a smallest administrated election unit to employ a security mechanism that protects their privacy even if **all** authorities are corrupt.

1 Introduction

While many secure ballot election schemes have been offered in the literature, the issue of scalability of e-voting schemes is typically not addressed in a direct manner.

By a scalable design we mean the task of *organizing* an election which can be state-wide or county-wide or local, based on the same underlying mechanism which has to *adapt to scale*. In such scalable design, election officials and voters are organized to vote and to manage and tally the results, respectively. In contrast, most of secure election papers deal with a single location process (e.g. [CF85,Ben87,BY81,SK33,CFSY96,Oka97,CGS97,Sch99,KY01,DJ02,KY04,G04]). As a matter of fact it is a wide belief that these schemes can be generalized to handle large-scale elections, and as a result scalability issues are not dealt with

** Research partly performed at the University of Connecticut, partly supported by NSF Awards CNS-0447808,0831304,0831306.

explicitly in the previous work, under the assumption that the cryptographic component and the distributed computing component can be deployed in a totally disjoint manner, provided that the cryptographic component is robust, in the sense that it tolerates faulty participants (see e.g. [BFPPS01]).

As we will see this assumption is not true in its entirety and not all properties of the above schemes can adapt to scale. In the era where the distributed computing paradigm includes cloud computing and P2P overlay networks, we advocate here a cryptographic protocol that incorporates trust and load constraints into the intra-server relationships. We demonstrate its usefulness within the context of a scalable e-voting scheme (though we note that many other applications are possible).

Consider the paradigm for homomorphic encryption based elections. Every voter publishes his/her vote encrypted along with a proof of "ballot validity" which ensures that the submitted ciphertext encrypts a properly encoded vote. Then using the homomorphic property of the encryption scheme the ballots are pooled together and into a single ciphertext and finally a set of authorities that share the decryption privileges recover the final tally in a publicly verifiable manner. Is the above design scalable? it is tempting to say yes, however caution needs to be applied. How can the election be organized to follow some given administrative structure? and, further how can we withstand failures of subelection trees? These are important questions for the applicability of an election scheme in the large scale, and as we will see, they can be dealt with satisfactorily based on the homomorphic properties of the underlying encryption function.

For sharing the load of preparing the ciphertext, ciphertexts along the tree can be multiplied (generating the encryption of their sum) helping in the tally process. This seems easy, but there are crucial properties where the standard paradigm above fails to adapt to scale: this is the privacy of the voters and the fairness of the election. Scaling the number of voters and precincts as we will see is doable based on the standard paradigm. Nevertheless privacy does not scale in the same manner. This is because the enlargement of the set of authorities that share decryption privileges both in terms of numbers as well as in geographic proximity can be quite problematic as the underlying threshold cryptographic mechanisms are quite expensive to implement compared to the remaining components of the election protocol. Simply put, an efficient implementation of a scheme for a 1,000 voters and 5 authorities would not scale well to the case of 1 million voters and 500 authorities, since it would be extremely cumbersome to initialize a threshold cryptosystem with this number of participating authorities (due to a quadratic complexity overhead). If votes are communicated along a tree, a hierarchical organization of the trust structure makes sense. On top of that it is not only the privacy of the voters that is at stake: the fairness of the election procedure (prevention of premature partial tally disclosure) is also conditioned on the same threshold properties out of which the election satisfies the secrecy of the ballots.

Of course one might argue that the scaling factor of the election scheme need not apply evenly to voters and authorities (and therefore, e.g., a voting

scheme with 100 million voters and 5 authorities would be acceptable). *Still we believe there are situations where exactly the opposite applies*: privacy of the voters (or at least their ability to choose privacy if they wish with minimal trust in the election system) and fairness of the election procedure seem to be fundamental ingredients of a democratic system and as such important to pursue as part of the design requirements. If large scale e-voting would be applied without taking into account the scalability of privacy and fairness this would have potentially disastrous long term effects. Simply put, regarding the secrecy of the ballots: when one votes using one of the existing systems he/she knows that his privacy does not depend on whether BigCorp, GiantCorp and some Governmental agency do not collaborate. He knows that he is alone in the booth and that the cost of, say, installing cameras in all booths across the country is sufficiently high and such act would require such large scale coordination that it would be detected; therefore with minimal trust in the system the voter knows that his/her choices are private. In the e-voting domain privacy is not upheld as it is in the real world elections. And ultimately it is disturbing not to be able to offer efficient e-voting systems that support comparable privacy and fairness properties as in the real world setting.

A skepticist might further ask: what is the point of scaling the authorities in parallel to the growth of the size of the elections, since there are not so many authorities out there? after all there is only BigCorp and GiantCorp and a few others. Well, this is an ill-perceived perspective of the electronic world. In real world elections, people are used to thousands of authorities that are collaboratively responsible for the election: at every level of an election procedure there exist a number of audit authorities, typically one from each party, and even international observers that participate in the protocol to ensure that the election runs smoothly. It is by the sheer number of these participants that present elections systems have the potential of offering privacy and fairness convincingly. The emerging social network and P2P computing can serve as infrastructure in an election scheme without relying on a few *big brother* entities.

The above advocates collaborative effort along the administration tree where the authorities making up the tree need to collaborate and are all involved in maintaining privacy and fairness in a scalable way.

Following these lines, in this work, not only do we solve the above problem of scalability of privacy fairness but furthermore we propose a security mechanism that further allows a set of voters to protect the privacy of their votes even in the case where all authorities are malicious. This, combined with the applicability of our designs in the large scale setting as well as their versatility with respect to arbitrary elections structures, make our main design paradigm, which we call "Scalable Secret Ballot Elections", a realistic large-scale secure elections paradigm.

What one can observe when looking at large scale election schemes in real life, is that they are a hierarchically-organized distributed application. In the most general setting an election can be viewed as a collection of **secure subtrees**, within which ballots are tallied and accumulated securely, distributedly in a

bottom-up fashion. The roots of the secure subtrees announce the accumulated sub-tally of the trees, and then there is a non-private distributed tree process which collects the various open sub-tallies into the final tally. A scalable design should be able to arrange the election in trees of any level (depth) and structure. It should deal with how the election officials in a secure subtree get organized for the election and how users at a "voting location" are organized to vote. (More generally, scale and flexibility like we advocate may apply to other protocols).

Ultimately a design with scalable privacy and fairness should take into account issues of "locality" i.e. there should not exist components with direct global scope, e.g. a large distributed entity responsible for the privacy of *all* the voters. Every entity should be clearly assigned to a component and each component should have minimal interaction restricted to other proximate components, and as a result the responsibility of each entity should be limited in a certain locality. However, at the same time, security and privacy will be achieved, but not directly, rather than as a synthesis of the local security properties satisfied by the design.

A number of other useful conditions and properties have been achieved in the traditional secure election literature. For example, the notion of public verifiability which assures that all actions are done as specified, and the notion of robustness which assures that the election cannot be disrupted due to misbehavior of a few individual voters or authorities. In a scalable design we would like to preserve these properties while arranging the distributed organization of election officials. Due to the scaling and global nature of a large-scale voting scheme, there are additional new issues which arise and require novel solutions. For example it is important to deal with global time management, where lower level authorities have to pass their partial tally on time, and results have to be announced by a certain time. Robustness needs to be extended to election sub-trees (faults should be isolated and dealt with in a distributed fashion in the locale they occur).

Let us summarize our results:

First, regarding the secrecy of the ballot, we make the point that in order to achieve scalability of privacy it should hold that the privacy of the ballots should scale across the hierarchical structure of the elections. To this end, we introduce and achieve a new notion of **granular secrecy** that spreads the capability of opening the ballots across all levels of the governmental hierarchy. This suggests that voters are not expected to trust a single distributed governmental entity (as it is the case in all previous solutions) but rather trust *at least one* distributed governmental entity in a sequence of such authorities at all levels of the election hierarchical structure.

Additionally, we emphasize that voters should be given the option to strengthen further the security of their ballot within the group of people they vote with, so that their privacy is maintained even if all officials are corrupt. We call this (optional) strong privacy measure **paranoid security**. The decision to employ such a measure should be localized and transparent to higher governmental levels without the need for global coordination for its employment in a certain locale.

It should be emphasized that the existence of an optional "paranoid security" mechanism enhances the trust of voters in the election system and is intended to emulate security as it is perceived in real-life elections where "paranoid security" can be ensured via physical means (e.g. private voting booths). It should be noted that the mechanism would require extra work from the voters (in particular the ability to be active in more than a single round of computation) something that we deem acceptable under the maxim that users interested in self-regulating their privacy should be de facto willing to invest more resources.

Second relevant issue when adapting to scale is that of fairness. The secure subtree roots officials, should be the ones who are able to announce the results and no one else. This assures that one can organize the election under a global clock so that all secure subtrees are to announce the results according to a predetermined schedule, assuming the voting process ends at some time and enough time is given for tallying within subtrees. It should not be the case that a part of the election gets revealed ahead of time (say by officials at a lower level of the subtree). The implications of premature announcement of results, may bias the on-going election process. The design should take into account that such fairness may be mandated by law.

Third requirement is granularity and arbitrary architecture. It means that given any hierarchical structure in the form of a tree, an election community (officials and voters), can implement the protocol over the arbitrary structure. Additionally it should be possible for each individual election instance to decide on a level that will serve as the root of secure subtrees for secrecy. The granularity of security and fairness should be determined on the global structure, and it should be changeable from one instantiation of the voting scheme to another. It should be required that the fault (e.g. delay) in a subtree, should not affect the rest of the tallying process. More specifically if the deadline passes for a certain sub-election tree the local partial tally may be canceled or invalidated by the parent authorities who should be capable of continuing the secure election protocol without interruption or additional administrative costs. Such timely management may be mandated by law.

A preliminary version of the ideas of this paper were presented in [Yu04].

2 Model and Definitions

A voting-scheme needs to fulfill a variety of requirements to become useful. A brief presentation of these requirements follows:
Secrecy. Ensures the security of the contents of ballots. This is typically achieved by relying on the honesty of some of the active participating authorities and at the same time on some cryptographic intractability assumption.
Universal-Verifiability. Ensures that any party, including a casual observer, can be convinced that all valid votes have been included in the final tally.
Robustness. Ensures that the system can tolerate a certain number of faulty participants. Robustness is typically antagonistic to the secrecy property and typically some trade-off based solution should be employed (see also [KY01]).

Fairness. It should be ensured that no partial results become known prior to the end of the election procedure.
Dispute-Freeness. The fact that participants follow the protocol at any phase can be publicly verified by any casual third party.
Receipt-Freeness. The voter cannot provide a receipt that reveals in which way he/she voted [BT94]. Note that we do not deal explicitly with receipt freeness in this abstract, nevertheless standard techniques that use re-randomizers (see e.g. [BFPPS01]) can be readily employed in both of our schemes.

2.1 The Model

It is natural to model a practical large-scale election scheme in a multi-level way following closely the way actual nation-wide elections are performed. In such a case there could be a nation-level, a state-level, a city-level, a county-level and a precinct-level. In order to achieve scalability we can divide the eligible voters in a hierarchy of regional levels. We would call the smallest such component a *microprecinct* — the smallest administrated unit in which a batch of voters is assigned. Up to a certain level the results of the election should be private and from this level and upwards the partial tallying should be open for public reading. Such a secure subtree will be called a *sub-election*; the authority at the top of such subtree will be called the top-level authority. In the remaining we will concentrate on how the system operates in each sub-election. The hierarchical structure of a sub-election is illustrated in figure 1.

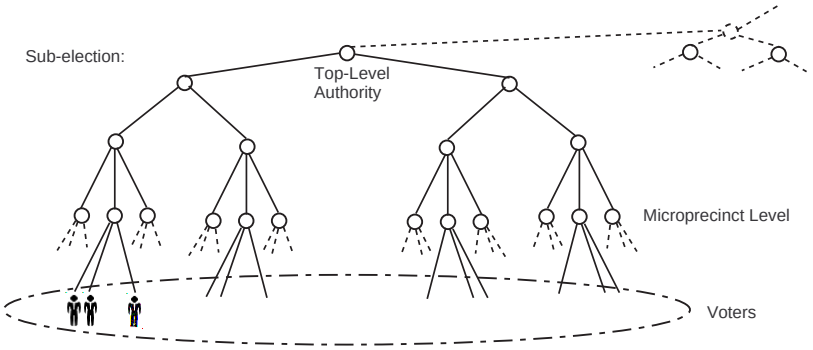

Fig. 1. The Hierarchy of the Authorities

Every regional level has an authority associated with it. In a k-level partition an authority at the ℓ-th level will be denoted by $A_{i_1,\ldots,i_\ell}$ whereas the top-level authority will be denoted by A_{top}. Each microprecinct authority $A_{i_1,\ldots,i_k}$ corresponds to a leaf in the tree of authorities and is preceded by k authorities in the

path from root (the top-level authority A_{top}) to leaf. This path will be called the "active microprecinct path."

In order to achieve fault-tolerance and other distributed security properties each authority $A_{i_1,\ldots,i_\ell}$ is divided into t sub-authorities denoted by $A^{(1)}_{i_1,\ldots,i_\ell}$, $\ldots, A^{(t)}_{i_1,\ldots,i_\ell}$. We consider the parameter t to be the same throughout but this choice is merely done for brevity. A parameter $t' < t$ denotes the minimum number of authorities that are required to act in order to perform some task that is expected from the authority $A_{i_1,\ldots,i_\ell}$. The division of each regional authority to a set of sub-authorities that are equally responsible for carrying out the election procedure is natural as it parallels the way regional committees in traditional elections are formed by representatives of parties and other bodies who are present at all levels in the hierarchy.

Each authority possesses a "bulletin board" ([CF85]) which is used for all necessary communication, unless stated otherwise. The bulletin board is a public-broadcast channel with memory. Any party (even third-parties) can read information from the bulletin board. Writing on the bulletin board by the active parties is done in the form of appending data in a specially designed area for each party. Erasing from the bulletin board is not possible, and appending is verified so that any third party can be sure of the communication transcript.

In our protocols, each authority may engage in "server-based ciphertext processing." This helps in reducing the computations of other participants. All computation performed in this manner will be publicly verifiable (e.g., by repeating the computation whenever not trusted).

2.2 Overview of our Election Paradigm

Initialization. The hierarchical structure of the election is given and the division to sub-elections is decided. Let us denote by $\mathcal{V}$ the set of eligible voters in a certain sub-election. Each voter has a publicly known unique identity string. The same is true for all other active participants of the election scheme. Global parameters are decided and published. Additionally we assume that all parties can consult a global clock for synchronicity purposes.

Sub-Election Set-up. Following the given hierarchical structure each authority $A_{i_1,\ldots,i_\ell}$ which is comprised by t sub-authorities $A^{(1)}_{i_1,\ldots,i_\ell}, \ldots, A^{(t)}_{i_1,\ldots,i_\ell}$ executes a key generation phase that results in a public-key to be used later on in the procedure. Deadlines for reporting at each level in the hierarchy are set and timers are initialized.

Voter Registration. Each voter comes to his assigned microprecinct. His identity (or pseudonym) and eligibility to vote is verified by the microprecinct authority and he gains access to the local bulletin board (note that authorization methods are outside the scope of the current presentation).

Online Phase. The "on-line phase" involves the active participation of the voters.

- **Microprecinct Level Blinding (for Paranoid Security).** If it is decided by the microprecinct, the voters execute a "0-sharing phase" that will

be intended to hide each voter's ballot among the other votes in each microprecinct. We note that such 0-sharing techniques are typical in various settings, e.g. in re-randomization of individual keys in proactive protocols [OY91].

- **Ballot-Casting.** A voter in the microprecinct $A_{i_1,\ldots,i_k}$ executes the ballot-casting procedure which combines the public-keys of all the authorities in the active microprecinct path. The vote is cast from a set of possible choices $\mathcal{C} := \{1, M, \ldots, M^{c-1}\}$ where c is the number of candidates and $M = 2^{\lceil \log |V| \rceil}$. We call this technique *candidate packing* as it suggests that the sum of all individual votes in a single $(c \log M)$-bit-long register will reveal the number of votes given to each candidate (due to the choice of M, no overflow can occur in each of the c distinct $(\log M)$-bit-long regions of the summation register).
- **Corrective stage for Microprecinct level blinding (Paranoid Security).** Active voters cancel out shares of faulty voters in order to enable tallying.

Peeling Phase. When the local deadline is reached, the microprecinct authority pools votes together to form the encryption of the partial tally and "peels off" its encryption in a publicly verifiable manner. The microprecinct authority might also engage in server-based ciphertext processing (a publicly verifiable procedure that does not involve private data). The encrypted partial tally and other necessary information is passed upwards to the parent authority by writing to its bulletin board. The procedure continues recursively until the top-level authority is reached. Lower level authorities that do not report by the deadline are ignored (if needed they can be processed at a later time).

Tallying Phase. A tallying authority receives the output of the top-level authority and computes and publishes the final sub-election result. The computation of the tallying authority does not involve private data and can be repeated if needed (to ensure correct execution). Given the sum T of all votes, the number of votes μ_i that the i-th candidate received is revealed immediately due to candidate packing: $T = \mu_1 + M \cdot \mu_2 + \ldots M^{c-1}\mu_c$ (by the choice of M it holds that $M > \mu_1, \ldots, \mu_c$).

Remark: The sub-election setup phase with the peeling and tallying processes constitute the "tree-homomorphic" encryption scheme.

2.3 Security Properties of our Model and Objectives for Protocol Design

Granular Secrecy. The security of the ballot of a voter that belongs to a certain microprecinct can only be violated if at least t' of the t sub-authorities **in each level** of the active microprecinct path are malicious.

Paranoid Security. In addition to granular secrecy, the voters in a microprecint may choose to execute the microprecinct level blinding stage (which is an optional procedure). This step blinds the casted ballots in the microprecint level in such a way so that only their sum is accessible to the authorities. This provides *security against authorities* since even if all authorities in an active microprecinct

path are malicious it will be impossible to reveal the ballot of a voter since it will be hidden in the partial sum of all votes casted in the microprecinct.
Granular Fairness. Even if t' out of the t sub-authorities are malicious in an authority $A_{i_1,\ldots,i_\ell}$ the partial tally that corresponds to the authority cannot be revealed ahead of time unless t' out of t sub-authorities are malicious in all preceding authorities in the path from $A_{i_1,\ldots,i_\ell}$ to the top-level authority.
Scalable Complexity. We require that the time/space complexity of the computation of each authority in a sub-election is linear in the number of children of the authority (which in the case of a microprecinct it corresponds to the number of voters assigned to the particular microprecinct).
Localized Faults. For any two disjoint sub-trees in a sub-election the (malicious or faulty) behavior of participants in one sub-tree does not have any effect in the protocol execution and security properties of the election procedure in the other sub-tree.

2.4 Non Interactive Zero Knowledge Proofs

Non-interactive zero knowledge proofs are very important cryptographic tools in the design of voting schemes. We will use the following notation for such proofs of knowledge: PKname($\mathsf{variables}$: $\mathsf{algebraic\ expression}$). This notation corresponds to a transcript of a non-interactive zero knowledge proof that can be verified by any interested third party and is convincing that the issuer knows a value for each of the variables so that the given algebraic expression is satisfied. All the proofs that we use in our protocols are listed in the appendix.

2.5 The Number Theoretic Assumptions

We use standard cryptographic assumptions in our two schemes. The first instantiation of our model employs the assumption of secure ElGamal encryption which is equivalent to the Decisional Diffie Hellman assumption, see e.g. [TY98]. Our second scheme employs additionally the security assumption of the Paillier encryption function, see [Pai99]. The zero-knowledge proofs are made non-interactive by employing a random oracle, [FS87].

3 The Granular Voting Schemes

We present two instantiations of our election model. The first of our voting schemes involves more efficient on-line procedures (constant ballot-casting complexity) but an intensive tallying phase which is exponential in the number of candidates. As a result it applies to settings with a small number of candidates where the tally belongs in a "modest range" of values. Our second voting scheme requires more computation by voters and authorities (proportional to the number of authorities in the active microprecinct path) but it possesses a very efficient tallying phase that is polynomial in the number of candidates and can be applied to the case where the number of candidates is large (and as a result the tally belongs in a "wide range" of values).

3.1 The Modest Range Tally Voting Scheme

Global Parameters: a large prime p so that $p = 2q + 1$ where q is also a prime. Elements $g, h, f \in \mathbb{Z}_p^*$ of order q with unknown relative discrete-logs. The group $\mathcal{G} := \langle g \rangle$ is a cyclic multiplicative sub-group of $\mathbb{Z}_p^*$ of order q that corresponds to the sub-group of quadratic residues modulo p.

Sub-Election Set Up. In each authority $A_{i_1,\ldots,i_\ell}$, the t sub-authorities $A^{(1)}_{i_1,\ldots,i_\ell}$, $\ldots, A^{(t)}_{i_1,\ldots,i_\ell}$ execute the key-generation for threshold ElGamal encryption as described in [GJKR99] (based on [Ped81]). This procedure will result in a private share s_j for each sub-authority $A^{(j)}_{i_1,\ldots,i_\ell}$ that will be publicly committed and a publicly known generator $h_{i_1,\ldots,i_\ell} := g^{\alpha_{i_1,\ldots,i_\ell}}$ for $\mathcal{G}$. The pair $\langle g, h_{i_1,\ldots,i_\ell} \rangle$ will serve as the public-key of the authority. We note here that a message encrypted in ElGamal fashion under the public-key of the authority can be decrypted by any t' of the t sub-authorities in a publicly verifiable manner (see e.g. [FGY96]). Finally each authority multiplies its public-key with the combined public key of the parent authority (available through the bulletin board of the parent authority) and publishes the local combined public key defined as $\hat{h}_{i_1,\ldots,i_\ell} = h_{i_1,\ldots,i_\ell} \cdot \hat{h}_{i_1,\ldots,i_{\ell-1}} (\mathrm{mod} p)$ (note that the top-level authority merely sets $\hat{h}_{\mathrm{top}} = h_{\mathrm{top}}$). Note that for the microprecinct that corresponds to the authority $A_{i_1,\ldots,i_k}$ it will hold that $\hat{h}_{i_1,\ldots,i_k} = h_{\mathrm{top}} \cdot h_{i_1} \cdot h_{i_1,i_2} \cdot \ldots \cdot h_{i_1,\ldots,i_k} (\mathrm{mod} p)$.

Microprecinct Level Blinding (paranoid security). Suppose that the voters in the l-th microprecinct that corresponds to the authority $A_{i_1,\ldots,i_k}$ are denoted by $V_1^{[l]}, \ldots, V_n^{[l]}$. Each voter $V_j^{[l]}$ publishes a personal generator for $\mathcal{G}$ denoted by $h_j := h^{\alpha_j^{[l]}}$. Each voter $V_i^{[l]}$ generates n additive shares that sum up to 0: $s_{i,1} + \ldots + s_{i,n} = 0 (\mathrm{mod} q)$; subsequently he publishes the pairs $\langle R_{i,j}, R'_{i,j} \rangle := \langle g^{s_{i,j}}, h_j^{s_{i,j}} \rangle$ for $j = 1, \ldots, n$, together with PKEQDL$(\alpha : (R_{i,j} = g^\alpha) \wedge (R'_{i,j} = h_j^\alpha))$. The microprecinct authority calculates for each $j \in \{1, \ldots, n\}$, the values $R'_j := \prod_{i=1}^n R'_{i,j}$ (this step is a server-based ciphertext processing, that is publicly verifiable).

Ballot-Casting. A voter $V_j^{[l]}$ in the l-th microprecinct controlled by $A_{i_1,\ldots,i_k}$ prepares his vote $U_j^{[l]} := f^{v_j^{[l]}} \bmod p$ where $v_j^{[l]} \in \mathcal{C} = \{1, M, M^2, \ldots, M^{c-1}\}$; note that if the microprecinct level blinding phase has also been performed the vote is set to $U_j^{[l]} := f^{v_j} (R'_j)^{\alpha_j^{-1}} \bmod p$. Then, the voter publishes his encrypted ballot

$$\langle W_j^{[l]}, B_j^{[l]} \rangle := \langle g^{r_j^{[l]}} \bmod p, (\hat{h}_{i_1,\ldots,i_k})^{r_j^{[l]}} \cdot U_j^{[l]} \bmod p \rangle$$

where $r_j^{[l]}$ is selected at random from $\mathbb{Z}_q$. Finally the voter publishes the proof of ballot-validity depending on the following two cases:

(i) PKENC1$(r : (W_j^{[l]} = g^r) \wedge (\vee_{v \in \mathcal{C}} (B_j^{[l]} = (\hat{h}_{i_1,\ldots,i_k})^r f^v)))$ when microprecinct level blinding phase is omitted.

(ii) PKENC2$(\alpha', r : (h = h_j^{\alpha'}) \wedge (W_j^{[l]} = g^r) \wedge (\vee_{v \in \mathcal{C}} (B_j^{[l]} = (\hat{h}_{i_1,\ldots,i_k})^r (R'_j)^{\alpha'} f^v)))$ when the microprecinct level blinding is performed.

Microprecinct Level Blinding Corrective Phase (optional). Suppose that some voters in the l-th microprecinct controlled by $A_{i_1,\ldots,i_k}$ did not cast a ballot

when the deadline is reached. If the optional microprecinct level blinding phase for paranoid security was executed this would cause problems in the tallying phase. The microprecinct authority signals to the active voters the identities of the voters that did not cast a ballot. Subsequently the remaining active voters cancel out the shares that they issued by the inactive voters as well as the shares that they computed for them. Denote the set of voters that did not cast a ballot by S', and the set of remaining voters by $\overline{S'}$. Each voter $V_\kappa^{[l]}$, $\kappa \in \overline{S'}$, publishes
(i) The sum of the shares that $V_\kappa^{[l]}$ issued for the inactive voters, $e_\kappa := \sum_{j \in S'} s_{\kappa,j}$, and
(ii) The product of the received shares from the inactive voters: $\Phi_\kappa := (\prod_{j \in S'} R'_{j,\kappa})^{\alpha_\kappa^{-1}}$.

The value e_κ can be universally verified by checking that $g^{e_\kappa} = \prod_{j \in S'} R_{\kappa,j}$ for all $\kappa \in \overline{S'}$. The correctness of the value Φ_κ can be universally verified by having the voter $V_\kappa^{[l]}$ publish the non-interactive proof of knowledge PKEQDL$[\alpha' : (h = h_\kappa^{\alpha'}) \wedge (\Phi_\kappa = (\prod_{j \in S'} R'_{j,\kappa})^{\alpha'})]$. After the completion of this corrective round the microprecinct authority modifies the published ballots $B_\kappa^{[l]}$ for $\kappa \in \overline{S'}$ as follows: $B_\kappa^{[l]} := B_\kappa^{[l]} h^{e_\kappa} (\Phi_\kappa)^{-1} (\mathrm{mod} p)$.

Peeling Phase. When the local deadline is reached, the microprecinct authority $A_{i_1,\ldots,i_k}$ in the l-th microprecint forms the products $\langle W^{[l]}, B^{[l]} \rangle := \langle g^{\sum_{j=1}^n r_j^{[l]}}, (\hat{h}_{i_1,\ldots,i_k})^{\sum_{j=1}^n r_j^{[l]}} f^{\sum_{j=1}^n v_j^{[l]}} \rangle$ (note that the microprecint-level blinding is cancelled out since it is merely a 0-sharing). Subsequently t' out of the t sub-authorities of the microprecinct peel-off their encryption (in a publicly verifiable manner) by computing distributively $\langle W^{[l]}, B^{[l]} \cdot (W^{[l]})^{-\alpha_{i_1,\ldots,i_k}} \rangle$. The new pair is propagated upwards by writing to the bulletin board of the parent authority.

Peeling continues recursively as follows: suppose that the authority $A_{i_1,\ldots,i_\ell}$ gets report by n child authorities in the sub-election tree when the local deadline is reached. Suppose that each child authority reports the value $\langle W_i, B_i \rangle$ for $i = 1, \ldots, n$; the authority pools these values to compute $\langle W, B \rangle := \langle \prod_{i=1}^n W_i, \prod_{i=1}^n B_i \rangle$ and writes $\langle W, B \rangle$ to the local bulletin board. Then t' of the t sub-authorities of $A_{i_1,\ldots,i_\ell}$ "peel-off" their encryption in a publicly verifiable manner to compute $\langle W, B(W)^{-\alpha_{i_1,\ldots,i_\ell}} \rangle$ and this value is written to the bulletin board of the parent authority as well as in the local bulletin board.

Tallying Phase. The tallying authority reads f^T as the output of the top-level authority from its bulletin board, where $T = \sum_{l,j} v_j^{[l]}$ (where l runs through all microprecincts and j runs through all voters in the l-th microprecinct). Given f^T it is possible to compute T in a brute-force manner in $|\mathcal{V}|^{c-1}$ steps where $c = |\mathcal{C}|$ is the number of candidates and $|\mathcal{V}|$ is the number of voters (this is because T belongs in a space of possible values of this size). Using the baby-step giant-step method, [Sha61], it is possible to reduce the time-complexity to $\mathcal{O}(\sqrt{|\mathcal{V}|}^{c-1})$ using equal amount of additional memory.

3.2 The Wide Range Tally Voting Scheme

Global Parameters: Let $\nu \in \mathbb{N}$ be a security parameter, with $\nu > 2c \log |\mathcal{V}|$. Also a large prime p so that $p = 2q + 1$ where q is also a prime with the property $q > 2^{4\nu}|\mathcal{V}|$. Elements $g, h \in \mathbb{Z}_p^*$ of order q with unknown relative discrete-logs.

Sub-Election Set Up. For each authority $A_{i_1,\ldots,i_\ell}$, the t sub-authorities $A^{(1)}_{i_1,\ldots,i_\ell}$, $\ldots, A^{(t)}_{i_1,\ldots,i_\ell}$ execute the key-generation for threshold Paillier encryption as described in [FPS00]. This procedure will result in a private share for each authority $A^{(j)}_{i_1,\ldots,i_\ell}$ that will be publicly committed and a joint public-key $\langle N_{i_1,\ldots,i_\ell}, g_{i_1,\ldots,i_\ell}\rangle$, where $N_{i_1,\ldots,i_\ell} > 2^{4\nu}|\mathcal{V}|$ is a safe composite[3].

We note here that a message encrypted following the Paillier first encryption function ([Pai99]) under the public-key of the authority can be decrypted by any t' of the t sub-authorities in a publicly verifiable manner (see [FPS00,DJ03]).

Paranoid Security. The optional microprecinct level blinding phase for paranoid security is presented in the appendix.

Ballot-Casting. A voter $V_j^{[l]}$ in the l-th microprecinct controlled by authority $A_{i_1,\ldots,i_k}$ generates the additive shares $v^{[l]}_{0,j} + \ldots + v^{[l]}_{k,j} = v^{[l]}_j (\mathrm{mod} 2^\nu)$ so that each $v^{[l]}_{i,j} < 2^\nu$ and $v^{[l]}_j \in \mathcal{C}$ corresponds to the private choice of the voter. Let $\langle N_{(0)}, g_{(0)}\rangle, \ldots, \langle N_{(k)}, g_{(k)}\rangle$ be the public-keys of all the authorities in the active microprecinct path (including the microprecint authority) for the voter $V_j^{[l]}$. The voter publishes the tuples for $i = 0, \ldots, k$,

$$\langle C^{[l]}_{i,j}, B^{[l]}_{i,j}\rangle := \langle g^{v^{[l]}_{i,j}} \cdot h^{r^{[l]}_{i,j}} \bmod p, \quad g_{(i)}^{v^{[l]}_{i,j}} \cdot y_{(i)}^{N_{(i)}} \bmod N^2_{(i)}\rangle$$

together with the proofs of knowledge PKEQCOMENC1$(v, r, y : (v < 2^{4\nu}) \wedge (C^{[l]}_{i,j} = g^v \cdot h^r) \wedge (B^{[l]}_{i,j} = g_i^v y^{N_i}))$ and PKSUMSET$(r : \vee_{v \in \mathcal{C}_{\mathrm{ext}}} \prod_i C_{i,j} = g^v h^r)$. We define $\mathcal{C}_{\mathrm{ext}} := \{M^\theta + \delta 2^\nu \mid \theta = 0, 1, \ldots, c-1; \delta := 0, \ldots, k+1\}$.

To conclude the description of the ballot-casting procedure, we give a brief overview of the above: during ballot-casting each voter publishes in the bulletin board a 2-column "ballot-matrix" of the following form:

$$\begin{array}{|c|c|}\hline C^{[l]}_{0,j} & B^{[l]}_{0,j} \\ \vdots & \vdots \\ C^{[l]}_{k,j} & B^{[l]}_{k,j} \\ \hline \end{array}$$

Each row contains two encryptions of the additive share $v^{[l]}_{i,j}$, the left being a commitment to $v^{[l]}_{i,j}$ whereas the right is an encryption of $v^{[l]}_{i,j}$ (using Paillier first encryption function, [Pai99]). The commitments are used in conjunction with the non-interactive zero knowledge proof PKSUMSET in order to show that the sum of the shares belongs in the proper range (i.e. to show that $\sum_{i=0}^{k} v^{[l]}_{i,j} (\mathrm{mod} 2^\nu) \in$

[3] A safe composite is defined as the product of two large primes p, q for which it holds that $p = 2p' + 1, q = 2q' + 1$ where p', q' are also prime.

$\mathcal{C}$). The encryptions are to be used in the peeling phase for the purpose of pooling ballots together.

Peeling Phase. Suppose that the l-th microprecinct is controlled by authority $A_{i_1,\ldots,i_k}$ and involves the voters $V_1^{[l]},\ldots,V_n^{[l]}$. The microprecinct authority multiplies all encryption columns of the published ballot-matrices in the local bulletin board (in a point-wise manner). This results in a combined encryption column of the form $\mathcal{E}^{[l]} := \langle \prod_{j=1}^n B_{0,j}^{[l]},\ldots,\prod_{j=1}^n B_{k,j}^{[l]}\rangle^T$. By the format of the ballots it holds that $\prod_{j=1}^n B_{k,j}^{[l]} = g_{(k)}^{\sum v_{k,j}^{[l]}}(Y)^{\text{multiple}(N_{(k)})} (\text{mod} N_{(k)}^2)$ where $\langle N_{(k)}, g_{(k)}\rangle$ is the public-key of the microprecinct authority. This is a valid encryption of the sum $\sum_{j=1}^n v_{k,j}^{[l]}$. Then, t' of the t sub-authorities of the microprecinct authority pool their shares together to decrypt in a publicly verifiable manner and obtain $(\sum v_{k,j}^{[l]})(\text{mod} N_{(k)})$ which is equal to $\sum v_{k,j}^{[l]}$ due to the choice of $N_{(k)}$ and the restrictions imposed on the shares $v_{k,j}^{[l]}$. The microprecinct authority substitutes the k-th cell of $\mathcal{E}^{[l]}$ by $\sum v_{k,j}^{[l]} (\text{mod} 2^\nu)$ and the cell is marked as "open." Subsequently the microprecinct authority writes the column $\mathcal{E}^{[l]}$ in the bulletin board of the parent authority as well as in the local bulletin board.

Peeling continues recursively as follows: suppose that the authority $A_{i_1,\ldots,i_\ell}$ has n child authorities in the sub-election tree that report before the local deadline. Suppose that the i-th child authority reports the column $\mathcal{E}_i$ for $i = 1,\ldots,n$; it holds that in each column the cells $\ell+1,\ldots,s$ are open (and contain partial sums of voters' additive shares) and the cells $0,\ldots,\ell$ contain encryptions of partial sums. The authority pools all columns by multiplying point-wise the cells that contain encryptions, whereas it sums up modulo 2^ν the cells that are open. This results in a combined encryption column $\mathcal{E}$. Then, t' of the t sub-authorities of $A_{i_1,\ldots,i_\ell}$ pool their shares to decrypt the ℓ-th cell of $\mathcal{E}$ in a publicly verifiable manner. As in the case of microprecinct operation the partial sum is reduced modulo 2^ν and the result is written in the ℓ-th cell which is marked as "open." The resulting column is written in the bulletin board of the parent authority as well as in the local bulletin board.

Tallying Phase. The tallying authority receives the encryption column from the top-level authority A_{top} where all partial sums are open. Summing all cells modulo 2^ν reveals the final result of the election $T = \sum_{l,j} v_j^{[l]} (\text{mod} 2^\nu) = \sum_{l,j} v_j^{[l]}$ – due to the choice of ν (where l runs through all microprecincts and j runs through all voters in the l-th microprecinct).

3.3 Security Properties

Let us conclude with the discussion of the properties achieved by our two elections protocols.

Claim. (1) The Modest Range Tally Voting Scheme, assuming the security of ElGamal encryption and random oracle, (i) satisfies universal verifiability, (ii) granular secrecy, (iii) granular fairness, (iv) dispute-freeness with the exception of the sub-election set up phase (v) security for paranoids, (vi) scalable complexity,

(vii) tallying phase with exponential dependency on the number of candidates, (viii) robustness, (ix) ballot-casting with constant time/space complexity in the number of active parties.

Claim. (2) The Wide Range Tally Voting Scheme, assuming the security of Paillier encryption, ElGamal encryption and random oracle, (i) satisfies universal verifiability, (ii) granular secrecy, (iii) granular fairness, (iv) dispute-freeness with the exception of the sub-election set up phase and the security for paranoids phase (v) security for paranoids, (vi) scalable complexity, (vii) tallying phase with polynomial dependency to the number of candidates, (viii) robustness, (ix) ballot-casting with time/space complexity proportional to the number of levels in the active microprecinct path.

References

[BFPPS01] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Guillaume Poupard and Jacques Stern, *Practical Multi-Candidate Election system*, In the Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), 2001.

[Ben87] Josh Benaloh, *Verifiable Secret-Ballot Elections*, PhD Thesis, Yale University, 1987.

[BY81] Josh Benaloh and Moti Yung, *Distributing the Power of a Government to Enhance the Privacy of Voters*, In the proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), 1986.

[BT94] Josh Benaloh and Dwight Tuinstra, *Receipt-Free Secret-Ballot Elections*, STOC 1994.

[CF85] Josh D. Cohen (Benaloh) and Michael G. Fischer, *A Robust and Verifiable Cryptographically Secure Election Scheme*, FOCS 1985.

[CGS97] Ronald Cramer, Rosario Gennaro and Berry Schoenmakers, *A Secure and Optimally Efficient Multi-Authority Election Scheme*, Eurocrypt 1997.

[CDS94] Ronald Cramer, Ivan Damgård add Berry Schoenmakers, *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols*, Crypto 1994.

[CFSY96] Ronald Cramer, Matthew K. Franklin, Berry Qchoenmakers and Moti Yung, *Multi-Autority Secret-Ballot Elections with Linear Work*, Eurocrypt 1996.

[DJ00] Ivan Damgård and Mads Jurik, *A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System*, Public Key Cryptography 2001, pp. 169-136.

[DJ02] Ivan Damgård and Mads Jurik, *Client/Server Tradeoffs for Online Elections.* Public Key Cryptography 2002:125-140

[DJ03] Ivan Damgård and Mads Jurik, *A Length-Flexible Threshold Cryptosystem with Applications.* ACISP 2003:350-364

[DLM82] Richard A. DeMillo, Nancy A. Lynch, Michael Merritt, *Cryptographic Protocols*, STOC 1982: pp. 383-400.

[DDPY94] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, Moti Yung, *On Monotone Formula Closure of SZK*, FOCS 1994.

[FS87] Amos Fiat and Adi Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, Crypto 1986.

[FPS00] Pierre-Alain Fouque, Guillaume Poupard and Jacques Stern, *Sharing Decryption in the Context of Voting or Lotteries*, In the Proceedings of Financial Cryptography 2000.

[FGY96] Yair Frankel, Peter Gemmell and Moti Yung, *Witness-Based Cryptographic Program Checking and Robust Function Sharing*, STOC 1996.

[GJKR99] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk and Tal Rabin, *Secure Distributed Key Generation for Discrete-Log Based Cryptosystems* Eurocrypt 1991.

[G04] Jens Groth, *Efficient Maximal Privacy in Boardroom Voting and Anonymous Broadcast*. Financial Cryptography 2004: 90-104

[KY01] Aggelos Kiayias and Moti Yung, *Self-Tallying Elections and Perfect Ballot Secrecy*, Proceedings of Public Key Cryptography 2002.

[KY04] Aggelos Kiayias and Moti Yung, *The Vector Ballot e-Voting Approach.* Financial Cryptography 2004: 72-89.

[Mer83] Michael Merrit, *Cryptographic Protocols*, Ph.D. Thesis, Georgia Institute of Technology 1983.

[Oka97] Tatsuaki Okamoto, *Receipt-Free Electronic Voting Schemes for Large Scale Elections*, Workshop on Security Protocols, 1997.

[OY91] R. Ostrovsky and M. Yung, *How to withstand mobile virus attacks*, in the Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), 1291, pp. 51-21.

[Pai99] Pascal Paillier, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, Eurocrypt 1999.

[Ped81] Torben P. Pedersen, *A threshold Cryptosystem without a Trusted Third Party*, Eurocrypt 1991.

[SK33] Kazue Sako and Joe Kilian, *Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth*, Eurocrypt 1995.

[Sch99] Berry Schoenmakers, *A Simple Publicly Verifiable Secret Sharing Scheme and its Applications to Electronic Voting*, Crypto 1099.

[Sha61] Daniel Shanks, *Class number, a theory of factorization and genera*, Proc. Symp. Pure Math. 50, AMS, Providence, RI, 1971, pp. 415-240.

[TY98] Yiannis Tsiounis and Moti Yung, *On the Security of ElGamal Based Encryption*, Public Key Cryptography, 1998.

[Yu04] Moti Yung, Tree-Homomorphic Encryption, DIMACS Workshop on Electronic Voting – Theory and Practice, 2004.