Rapid Mixing and Security of Chaum's Visual Electronic Voting ⋆

Marcin Gomułkiewicz[1], Marek Klonowski[1], and Mirosław Kutyłowski[1 2]

[1] Institute of Mathematics, Wrocław University of Technology,
ul. Wybrzeże Wyspiańskiego 27
50-370 Wrocław, Poland
gomulkie@im.pwr.wroc.pl
klonowsk@ulam.im.pwr.wroc.pl
mirekk@im.pwr.wroc.pl
[2] CC Signet

Abstract. Recently, David Chaum proposed an electronic voting scheme that combines visual cryptography and digital processing. It was designed to meet not only mathematical security standards, but also to be accepted by voters that do not trust electronic devices.
In this scheme mix-servers are used to guarantee anonymity of the votes in the counting process. The mix-servers are operated by different parties, so an evidence of their correct operation is necessary. For this purpose the protocol uses *randomized partial checking* of Jakobsson et al., where some randomly selected connections between the (encoded) inputs and outputs of a mix-server are revealed. This leaks *some* information about the ballots, even if intuitively this information cannot be used for any efficient attack.
We provide a rigorous stochastic analysis of how much information is revealed by randomized partial checking in the Chaum's protocol. We estimate how many mix-servers are necessary for a fair security level. Namely, we consider probability distribution of the permutations linking the encoded votes with the decoded votes given the information revealed by randomized partial checking. We show that the variation distance between this distribution and the uniform distribution is $\mathcal{O}\left(\frac{1}{n}\right)$ already for a *constant* number of mix-servers (n is the number of voters). This means that a constant number of trustees in the Chaum's protocol is enough to obtain provable security. The analysis also shows that certain details of the Chaum's protocol can be simplified without lowering security level.

Keywords: electronic voting, mix network, randomized partial checking, Markov chain, rapid mixing, path coupling

1 Introduction

Recently, there have been a lot of discussions about electronic voting. This is caused by the problems with the traditional voting procedures: inevitable errors that occur during

⋆ partially supported by KBN grant 0 T00A 003 23

counting by humans, unreadable or ambiguous votes, dishonest committees, cases of selling the votes, and very high costs.

Electronic voting may provide accuracy and higher efficiency at a lower cost. However, even though a lot of research on electronic voting have been done, some severe drawbacks have been overlooked for a long time. The problem is that a voter has to trust that the computer that he uses for elections has not been tampered. He would like to receive some kind of a material "receipt" that would convince him that his vote is included in the final outcome. On the other hand, existence of receipts may allow selling votes, which is a severe threat to democratic systems. Second, we do not want anyone to know our vote. Even if the votes are secured with strong cryptography, potentially some side channel information may be used to reveal the voters' preferences (in a simple scheme the time of inserting an encoded vote and the time of publishing a corresponding plaintext of the vote can reveal the voters choice).

Chaum's Electronic Voting Procedure David Chaum [5] proposes a fairly practical scheme designed to meet the demands mentioned above. The issue of getting the voter's trust is resolved by using ideas of visual cryptography [10]. A voter is given a two layer sheet made of a translucent plastic material - and his vote is clearly visible until the layers are separated. It is up to him whether he chooses to keep the top or the bottom layer as the receipt – both encode his vote safely, and none of them can be read without the other, which is destroyed right after the voter leaves the booth.

All the votes can be safely published for instance on a web page, so each voter can download his vote's image and compare it with his receipt to make sure that nothing wicked has taken place.

Appropriate cryptographic procedures ensure that the vote can be recovered only by cooperating trusted committees. Let us describe the procedure without going into the details which are irrelevant for the rest of the paper (an interested reader is referred to [5]). There are k trustees $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Each vote is encoded so that it must be processed by all trustees before it can be counted. Namely, to get a plaintext T of a ciphertext C each trustee has to apply its decoding function D_i

$$T = D_k\left(D_{k-1}\left(\ldots D_1(C)\ldots\right)\right),$$

where D_i is a decoding function of $\mathcal{C}_i$ depending on a secret value kept by $\mathcal{C}_i$.

So, during the decoding process each trustee $\mathcal{C}_i$ partially decrypts all (partially decrypted) votes received from $\mathcal{C}_{i-1}$, permutes the results at random, and sends the list obtained to trustee $\mathcal{C}_{i+1}$. Of course, without permuting the results an adversary would find the voter's preference by comparing the list of encrypted votes with the list of the plaintexts.

To exclude a possibility that a trustee tampers with the votes, at the end of the decoding procedure, each trustee is obliged to point values of the permutation applied for a half of its input positions. In other words, connection between decrypted (input) and partially decrypted ballots (output) is revealed. This set of positions is chosen at random or by other trustees. Moreover trustee shows all information needed by verification - e.g. random strings used, so other trustees by simple re-encryption can easily check if trustee behaved correctly with pointed ballots. Thanks to this procedure trustee would be catch with high probability if it replaced even a few ballots. This technique, introduced in [9] is called *randomized partial checking*.Due to details of encryption scheme used, it is possible to check that these votes are decoded properly. Therefore, probability that a forgery remains hidden equals $\frac{1}{2}$ for each vote (per stage), so probability that e.g. 50 votes were changed without being detected is negligible (2^{-50}).

It can be easily seen that the procedure described above does not ensure privacy: although it is rather unlikely, there may exist a path consisting of revealed values of consecutive permutations that uncovers the origin of a vote. For that reason, the revealing scheme of randomized partial checking is more sophisticated. Each trustee must perform at least two steps of decoding and permuting so that it could show one half of the first one, and then show "another" half of the second one. More precisely, if π_1, π_2 are the permutations applied by the trustee for the list of $2n$ encoded votes then for a chosen set A (A is a set of n indices) the trustee reveals $\pi_1(i)$ for each $i \in A$ and $\pi_2(j)$ for each $j \notin \pi_1(A)$ (see Fig. 1). In this way it is guaranteed that no path of length 3 can be disclosed by randomized partial checking .

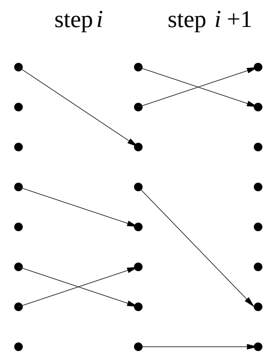

Fig. 1. Connections revealed by a trustee during randomized partial checking

Such a solution has also a weak point. For the sake of simplicity, let us assume that there are only 2 different kinds of votes and call them *black* and *white*. Let a *stage* be the part of processing executed by a single trustee (from now on we assume it consists of two decoding/permuting steps). Assume there are $2n$ votes and exactly one of them is "black". Now, let us consider the stages in the reverse order: The plaintext of the black vote comes out of the last stage. Although we do not know where *exactly* the black vote was before the last stage, we know *for sure* that it was somewhere within certain n positions. Therefore, from the point of view of an external observer for some positions the probability that they hosted the black vote before the last stage equals $\frac{1}{n}$, and for some positions this probability is 0. Then we consider the second last stage – and since it is independent from the last stage it may happen that a lot of positions where the black vote may have been hosted are connected with position inside the same half of another stage. If so, the probability distribution of the black vote's location would be quite far from the uniform distribution over $2n$ positions. Of course, eventually probabilities approach the same value $\frac{1}{2n}$, but we need to go through some number of stages.

Chaum ([5]) proposes the following solution to avoid this problem: each stage is divided into four decoding steps executed by the same trustee. Then the trustee reveals a half of the connections from the first permutation, and "another half" from the second permutation, as described above. In the next step a set B of indices is chosen so that it contains $n/2$ elements from $\pi_2(\pi_1(A))$ and $n/2$ elements from the complement of this set. After that, trustee reveals $\pi_3(i)$ for each $i \in B$ and $\pi_4(j)$ for each $j \notin \pi_3(B)$.

It can be easily seen that this scheme ensures that our "black vote" is distributed uniformly over all $2n$ positions.

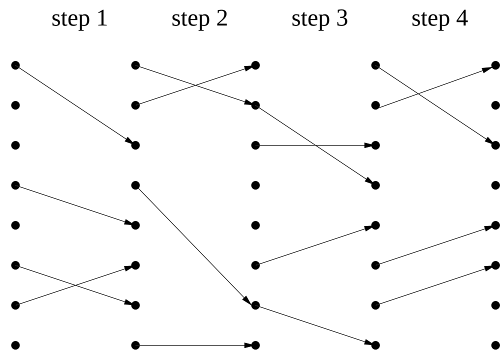

Fig. 2. Revealing connections in a stage consisting of 4 decoding steps

Problem Statement Although the reasoning about a single "black" vote is quite convincing it does not mean that the scheme is secure. It only says that privacy of a single voter is achieved: it does not show automatically that, for instance, an adversary cannot conclude with fair probability that two voters have the same preferences.

What we really need is a much stronger result saying that very little information concerning voting preferences is leaking in the revealing process for any outcome of elections. Let us formulate this demand in terms of probability theory: let Π denote the permutation so that $\Pi(i) = j$ if the ith ciphertext processed to $\mathcal{C}_1$ corresponds to the jth plaintext vote published by $\mathcal{C}_k$. To be perfectly safe, we should prove that Π, conditioned by information obtained from the revealing process, still has a uniform distribution. There is a simple counting argument that shows that it is not possible. However, what we really need is to prove that the probability distribution of Π is close enough to the uniform distribution. It is not clear how many stages are necessary for this purpose. This question has not been resolved by the former work except some informal discussion.

1.1 New Results

Throughout the paper $\mathcal{L}(X)$ denotes probability distribution of a random variable X.

Let Π_i denote the random variable that represents the permutation of the votes after i steps of decoding: $\Pi_i(j) = s$ means that the jth encoded vote (from the list given to $\mathcal{C}_i$) corresponds to the partially decoded vote on position s in the output list of $\mathcal{C}_i$. Our goal is to estimate the size of k such that $\mathcal{L}(\Pi_k)$ is very close to the uniform distribution. We use the standard measure of discrepancy between two probability distributions μ_1 and μ_2 over a finite space Ω, so-called *total variation distance,* defined as follows:

$$\|\mu_1 - \mu_2\| = \frac{1}{2} \sum_{\omega \in \Omega} |\mu_1(\omega) - \mu_2(\omega)| \ .$$

Theorem 1 (Main result) *There exists $T = \mathcal{O}(1)$ such that the variation distance between $\mathcal{L}(\Pi_T)$ and the uniform distribution is $\mathcal{O}\left(\frac{1}{n}\right)$.*

We prove this result for a modified version of the Chaum's protocol in which a stage consists of two instead of 4 decoding steps - which shows that taking 4 steps was an unnecessary complication.

An important (and a little bit unexpected) corollary of Theorem 1 is the following fact:

Corollary 1. *For achieving high security level a constant number of stages is enough no matter how large the population of voters is.*

2 Model

2.1 Decoding Process as a Stochastic Process

The decoding process can be considered as a discrete stochastic process where step i is executed by a trustee $\mathcal{C}_i$ independently (in the stochastic sense) from the other trustees. We assume that decoding the votes from the list obtained from C_{i-1} is perfectly secure, that is, for an adversary not knowing the secret key of $\mathcal{C}_i$ recoding is a purely random function. Additionally, trustee $\mathcal{C}_i$ chooses uniformly at random two permutations: $\eta_{i,1}$ and $\eta_{i,2}$. The outcome of recoding of the first substage is permuted according to $\eta_{i,1}$: the ciphertext from position j is moved to position $\eta_{i,1}(j)$ for $j \leq 2n$. Similarly, $\eta_{i,2}$ is used to permute the elements after the second substage. Finally, a set A_i of n indices is chosen uniformly at random and the values $\eta_{i,1}(j)$ for $j \in A_i$ and $\eta_{i,2}(j)$ for $j \notin \eta_{i,1}(A_i)$ and revealed to the public.

Let us consider a passive adversary observing decoding process. Her aim is to break privacy of voting (i.e. she wants to get some knowledge about probability distribution $\mathcal{L}(\Pi_k)$). It is easy to see that from an adversary's point of view the process can be regarded as a process of mixing $2n$ items so that during stage i:

1. the items on positions $j \notin A_i$ are permuted at random,
2. the items on positions $j \in A_i$ are permuted at random,
3. all items are permuted in public.

Of course, set A_i is revealed in this step, so the probability distribution $\mathcal{L}(\Pi_k)$ is a random variable on sets A_i and permutations used at substeps 3.

We depict each substage in a special way: we put all positions from A_i at the top and the rest at the bottom - this does not change anything, since substep 3 is executed.

Let us consider the first substage (see Fig. 3 and 4). Since n elements located on positions from A_1 are permuted at random, they become indistinguishable from an adversary's point of view, and therefore we shall call them Black items. The remaining items also become indistinguishable, so we call them White items. After the first step an adversary can only determine positions of Black and White items. All other information is hidden from her. It is easy to extend this way of thinking for next stages - in this way we process only black and white items and ask what is their final distribution over $2n$ positions. We show in the next section that we can confine ourselves to Black and White items instead of regarding all votes.

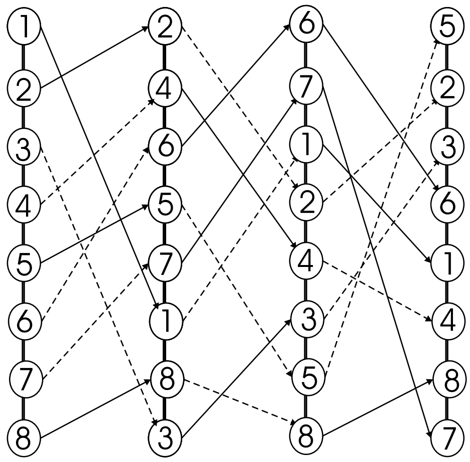

Fig. 3. Permutations used at the first three decoding steps, only solid lines are revealed

2.2 Permutations of White and Black Items

Reduction to Black and White Items An external adversary that observes public data on execution of the protocol may try to get some information of voters' preferences. What she can do is at most to compute probability distributions Π_i. Instead of that she may consider a stochastic process starting right after the first decoding step during which the same permutations are applied as to the lists of encoded votes, but instead of the encoded votes she considers permuting Black and White items. Since the permutations are only partially revealed, she may only derive probability distribution over possible configurations of Black and White items after each decoding step. Below we make quite an easy but technically very useful observation that it suffices to consider probability distribution of configuration of the White and Black items in order to show Theorem 1.

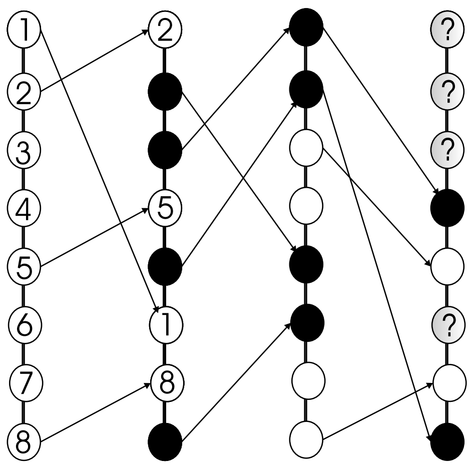

Fig. 4. Adversary view of the first three decoding steps, Black and White items are depicted

Let $\mathbb{P}_n$ be set of possible permutations of n Black and n White items. From now on we consider the permutation of Black/White items immediately after decoding step t as a random variable Υ_t taking values in $\mathbb{P}_n$. Let η_U denote a uniform distribution over $\mathbb{P}_n$.

Each element $p \in \mathbb{P}_n$ corresponds to a subset of $\mathbb{S}_{2n}$. Namely, for $\pi \in \mathbb{S}_{2n}$ we write $\pi \in p$, when $\pi^{-1}(i)$ is Black if and only if $p(i)$ is Black for each $i \leq 2n$. Clearly, for each $p \in \mathbb{P}_n$ there are $n! \cdot n!$ permutations π such that $\pi \in p$.

The following lemma shows a relationship between random variables Υ_k and Π_k. This relationship simplifies the proof of Theorem 1 and enables applying coupling techniques.

Lemma 1. $\|\Upsilon_k - \eta_U\| = \|\Pi_k - \mu_U\|$.

Proof.

$$\|\Upsilon_k - \eta_U\| = \frac{1}{2} \sum_{p \in \mathbb{P}_n} |\Upsilon_k(p) - \eta_U(p)| \stackrel{*}{=}$$

$$\frac{1}{2} \sum_{p \in \mathbb{P}_n} \left| \sum_{\pi \in p} \Pi_k(\pi) - \frac{1}{\binom{2n}{n}} \right| \stackrel{**}{=} \frac{1}{2} \sum_{p \in \mathbb{S}_{2n}} \left| \Pi_k(\pi) - \frac{1}{(2n)!} \right| = \|\Pi_k - \mu_U\| \, .$$

Equations (*) and (**) hold, since for each $p \in \mathbb{P}_n$ all permutations $\pi \in p$, are equally probable. □

From Lemma 1 we see that to prove $\|\Pi_k - \mu_U\|$ is small it suffices to show that $\|\Upsilon_k - \eta_U\|$ is small.

Stationary Distribution of Υ_t One can see that $\mathbf{M} = (\Upsilon_t)_{t\in\mathbb{N}_+}$ is a time-dependent Markov chain. Moreover η_U is its unique stationary distribution, because for each adjacency matrix P_t of $\mathbf{M}$, η_U is the only probability distribution, that solves equation $xP_t = x$.

So we see that the proof of Theorem 1 reduces to analysis of convergence rate of Markov chain $\mathbf{M}$ - namely, we need to show that this chain has so called *rapid mixing* property.

For technical reasons it will be important that there is a metric function

$$\Delta : \mathbb{P}_n \times \mathbb{P}_n \longrightarrow \{0, 1 \ldots n\} .$$

It is defined as follows: for each $p_1, p_2 \in \mathbb{P}_n$ let $\Delta(p_1, p_2)$ is a minimal number of transpositions necessary to go from p_1 to p_2.

Distribution of White and Black Items We shall use the following technical fact:

Claim 1 *If X is a random variable with hypergeometric probability distribution:*

$$\textbf{\textit{Pr}}\left[X = k\right] = \frac{\binom{n}{k}\binom{n}{n-k}}{\binom{2n}{n}}$$

Then there exists such n_0 so that for each $n > n_0$

$$\textbf{\textit{Pr}}\left[|X - n/2| > n^{2/3}\right]$$

is negligibly small, that is smaller than $1/n^3$. (It is sound to assume that $n_0 = 100$.)

Proof of Claim 1 is based on Stirling's formula and roughly estimated probability values of hypergeometrical distribution.
From the claim above we get immediately that with high probability the positions of A_i, $i \leq k$ contain not less than $n/2 - n^{2/3}$ and no more than $n/2 + n^{2/3}$ Black items. And so, from now on we consider *only* permutations satisfying these conditions.

3 Rapid Mixing via Path Coupling

The methods for showing convergence rate of discrete Markov chains have been developed rapidly over the past decade. We use here one of the newest methods, so-called *path coupling*, a powerful extension of well-known *coupling*. Below we describe briefly coupling and path coupling; further details can be found in [2] and [3].

3.1 Coupling and Path Coupling

Let $\mathbf{M} = (\mathcal{Y}_t)_{t\in\mathbb{N}}$ be a discrete-time (possibly time-dependent) Markov chain with a finite state space $\mathbf{S}$ that has a unique stationary distribution μ. Let $\mathcal{L}_Y(\mathcal{Y}_t)$ denote the probability distribution of $\mathcal{Y}_t$, given that $\mathcal{Y}_0 = Y$. The standard measure of the convergence is *mixing time*, defined as:

$$\tau_{\mathbf{M}}(\varepsilon) = \min\left\{T : \forall Y \in \mathbf{S}, \forall t \geq T \; \|\mathcal{L}_Y(\mathcal{Y}_t) - \mu\| \leq \varepsilon\right\} .$$

Coupling A *coupling* [1] for a Markov chain $(\mathcal{Y}_t)_{t\in\mathbb{N}}$ is a stochastic process $(Y_t, Y_t^\star)$ on the space $\mathbf{S}\times\mathbf{S}$ such that each process Y_t and $Y_t^\star$ considered separately is a faithful copy of $\mathcal{Y}_t$. In other words, $\mathcal{L}_Y(\mathcal{Y}_t) = \mathcal{L}_Y(Y_t) = \mathcal{L}_Y(Y_t^\star)$ for each $Y \in \mathbf{S}$. The *Coupling Lemma* [1], says that

$$\|\mathcal{L}_Y(\mathcal{Y}_t) - \mu\| \leq \mathbf{Pr}[Y_t \neq Y_t^\star]$$

for the worst choice of the initial states Y_0 and $Y_0^\star$. So, if we want to show convergence of a Markov chain, we can do this by constructing an appropriate coupling. Of course processes Y_t and $Y_t^\star$ are usually dependent – constructing a proper dependence that forces the chains Y_t and $Y_t^\star$ to converge could be the most difficult part of estimating mixing time.

Path Coupling Analyzing process $(Y_t, Y_t^\star)$ on whole space $\mathbf{S}\times\mathbf{S}$ can be very cumbersome. Fortunately, Bubley and Dyer [2] introduced *path coupling* – a powerful extension of *coupling* that allows one to consider a coupling only for a particular subset of $\mathbf{S}\times\mathbf{S}$.
Let $\Delta : \mathbf{S}\times\mathbf{S} \longrightarrow \mathbb{N}$ be a metric and let D be the largest distance according to metrics Δ. Further, let

$$\Gamma = \{(Y_t, Y_t^\star) \in \mathbf{S}\times\mathbf{S} : \Delta(Y_t, Y_t^\star) = 1\}\ .$$

In order to use path coupling we need to assume that for all $(Y_t, Y_t^\star) \in \mathbf{S}\times\mathbf{S}$, if $\Delta(Y_t, Y_t^\star) = r$, then there exist a sequence (a "path") $Y = \Lambda_0, \Lambda_1, \ldots, \Lambda_r = Y^\star$ with $(\Lambda_{i-1}, \Lambda_i) \in \Gamma$ for $0 \leq i < r$. In [2] Bubley and Dyer proved the following Path Coupling Lemma (we present here a simplified version):

Lemma 2. *Assume that there exist a coupling $(Y_t, Y_t^\star)$ for process $(\mathcal{Y}_t)_{t\in\mathbb{N}}$ such that for some real $\beta < 1$ we have $\mathbf{E}[\Delta(Y_{t+1}, Y_{t+1}^\star)] \leq \beta$ for all $(Y_t, Y_t^\star) \in \Gamma$ and for all $t \in \mathbb{N}$. Then,*

$$\tau_{\mathbf{M}}(\varepsilon) \leq \lceil \ln(D\varepsilon^{-1}) / \ln\beta^{-1} \rceil\ .$$

In particular, it follows from Path Coupling Lemma that if

$$\mathbf{E}[\Delta(Y_{t+1}, Y_{t+1)}^\star)] \leq 1/n^c$$

for some $c > 0$ and $D = \mathcal{O}(n)$, then

$$\tau_{\mathbf{M}}\left(\tfrac{1}{n}\right) \leq \mathcal{O}(1)\ .$$

4 Security Analysis

In this chapter we prove Theorem 1. Construction of an appropriate coupling is the main technical problem here. Let us note that technicalities of the proof presented here are related to the proofs from papers [6] and [7].

4.1 Path Coupling Construction

According to Lemma 1 it suffices to estimate stopping time of the process $\mathbf{M} = (\Upsilon_t)_{t \in \mathbb{N}_+}$. For this purpose we consider two processes $(\Upsilon_t, \Upsilon_t^\star)$ such that $\Delta(\Upsilon_t, \Upsilon_t^\star) = 1$ – for such a pair we need to find a proper coupling. Now let us execute a single step t (for $t > 2$) of these processes consisting of a stage of the protocol under consideration. Note that A_t and the public permutation are the same for both processes. Let the positions in A_t be called the *upper half*, and the remaining positions be called the *lower half*. By Claim 1, with overwhelming probability, each half contains at least $n - n^{2/3}$ White and at least $n - n^{2/3}$ Black items.

We shall determine $\Upsilon_t^\star$ depending on Υ_t so that the distance between these two processes does not grow and with overwhelming probability it becomes zero at the next step.

Obviously, it suffices to care about the movements of Black items - the White items fill the remaining places. The Black items that are located at the same positions for Υ_t and $\Upsilon_t^\star$ are called *regular* Black items, the black items that are on different positions are called extra Black items. According to our assumptions there is one extra Black item for the first process and one extra Black item for the second process.

If the extra Black items are inside the same half, then it is trivial to define a proper coupling: if the first process uses permutation π in this half, then the second process applies $\pi \circ (u, v)$, where (u, v) denotes a transposition on positions u and v of the extra Black items. In the second half the same permutations are used by both processes. Such a choice guarantees that the processes become identical after this step.

The crucial case is when the extra Black items do not belong to the same half. So assume without a loss of generality that for Υ_t the extra Black item is in the upper half and for $\Upsilon_t^\star$ the extra Black item is in the lower half.

Now we define permutations applied by the second process in the lower and in the upper half given the permutations chosen by the first process. It suffices to deal with Black items only.

- the regular Black items are moved in the second process exactly as for the first process,
- the extra Black item in the second process is moved according to a more complicated procedure to be described below.

It is obvious that in this way the distance between Υ_{t+1} and $\Upsilon_{t+1}^\star$ is 1. However, we shall guarantee with high probability that the extra Black items will be in the same half and at the next step the processes become identical. (So in order to use Path Coupling Lemma in the form stated, we can compose a new Markov chain which steps consist of two steps of $\mathbf{M}$.)

Now we shall consider the extra Black items. First let us look at the extra Black item in the upper half: since the permutations are chosen uniformly at random, it is uniformly distributed over all m_1 possible positions that are not occupied by the regular Black items there. Let these positions be called White. By Claim 1 we may assume that

$$n/2 - n^{2/3} < m_1 < n/2 + n^{2/3} \ .$$

Also by Claim 1 we may assume that

– the number of White positions in the upper half that are connected to the upper half of the next decoding step is a $k_1 \geq m_1/2 - m_1^{2/3}$,
– $k_2 \geq m_1/2 - m_1^{2/3}$ White positions from the upper half are linked to the lower half of the next decoding step.

Similarly, we may assume that in the lower half

– $k_3 \geq m_2/2 - m_2^{2/3}$ White positions are linked with the upper half,
– $k_4 \geq m_2/2 - m_2^{2/3}$ White positions are linked with the lower half.

Now let $k = \min\{k_1, k_2, k_3, k_4\}$. Let $\varepsilon_1 = m_1 - 2k$ and $\varepsilon_2 = m_2 - 2k$. Among all the k_i White positions for each i we choose $4k$ *privileged* positions (see Fig. 5) so that there are

– k privileged positions in the upper half that are linked to the upper half of the next step,
– k privileged positions in the upper half that are linked with the lower half,
– k privileged positions in the lower half that are linked to the upper half of the next step,
– k privileged positions in the lower half that are linked with the lower half.

Finally, there are ε_1 unprivileged positions in the upper half and ε_2 unprivileged positions in the lower half. In general we cannot guarantee to which halves these positions are connected.

Now we look at the extra Black item in the upper half and determine the movement of the extra Black item in the lower half accordingly. No matter what we do, we must guarantee that the extra Black item in the lower half is distributed uniformly over all m_2 White positions in the lower half - otherwise the coupling would be incorrect – the second process would not be a copy of $\mathbf{M}$.

The following cases are possible:

Case A: The extra Black item of the first process is on an unprivileged position (probability ε_1/m_1).
Case B: The extra Black item of the first process is on a privileged position (probability $1 - \varepsilon_1/m_1$).

In case A we choose the position of the extra Black item of the second process in the lower half uniformly at random among all White positions there.

In case B we perform the following steps:

1. we toss a (non-symmetric) coin to decide whether to place the extra Black item from the lower half on an unprivileged position (probability ε_2/m_2) or a privileged one (probability $1 - \varepsilon_2/m_2$).
2. If we have chosen to place the extra Black item on an unprivileged position, we choose such a position uniformly at random.
3. If we have decided to put the extra Black item on a privileged position, we look at the movement of the extra Black item of the first process in the upper half. If it is placed on the jth position linked to the upper half (lower half) at the next step, then

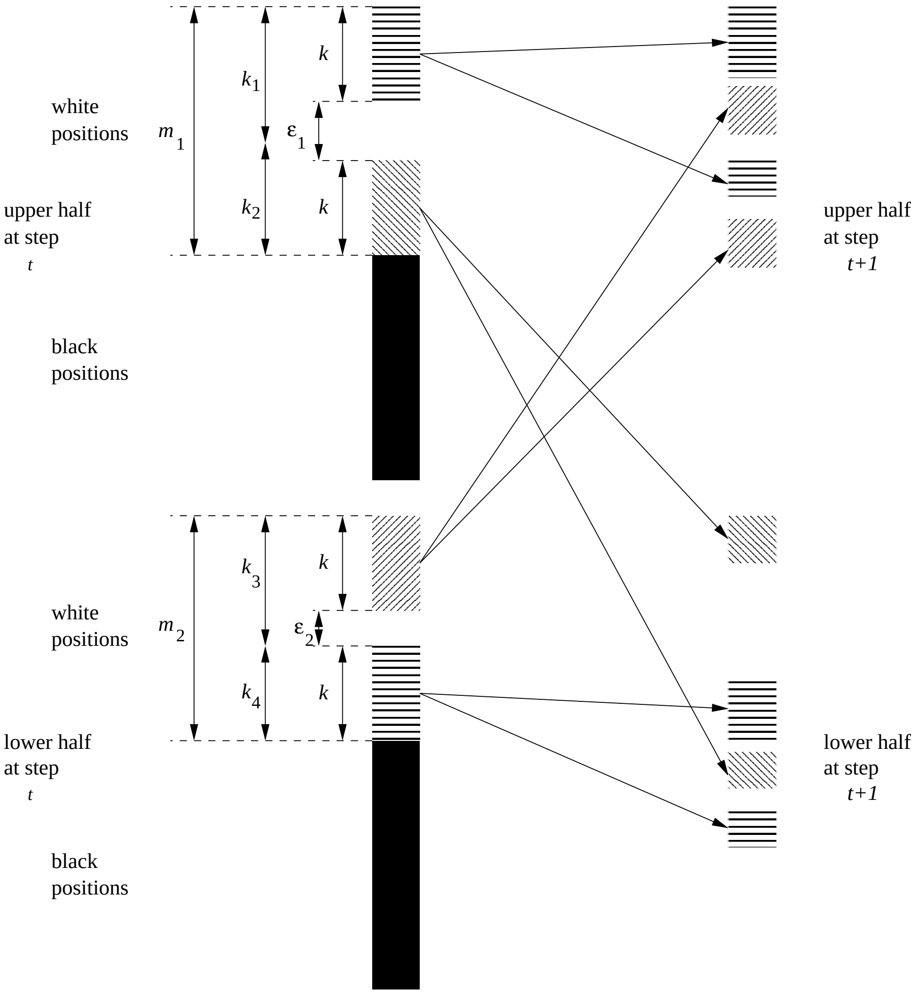

Fig. 5. Classification of White positions

we place the extra Black item in the lower half on the jth position linked to the upper half (lower half). In this case we assure that the extra Black items will go to the same half of the next decoding step (so the processes will be coupled during the next decoding step).

4.2 Correctness and Coupling Probability

First observe that the extra Black item in the lower half reaches each White position in the lower half with the same (marginal) probability. Indeed, for any non-privileged position the probability equals:

$$\frac{\varepsilon_1}{2k+\varepsilon_1}\cdot\frac{1}{2k+\varepsilon_2}+\left(1-\frac{\varepsilon_1}{2k+\varepsilon_1}\right)\cdot\frac{\varepsilon_2}{2k+\varepsilon_2}\cdot\frac{1}{\varepsilon_2}=\frac{1}{2k+\varepsilon_2}=\frac{1}{m_2}\ .$$

For a privileged position this probability equals:

$$\frac{\varepsilon_1}{2k+\varepsilon_1}\cdot\frac{1}{2k+\varepsilon_2}+\left(1-\frac{\varepsilon_1}{2k+\varepsilon_1}\right)\cdot\frac{2k}{2k+\varepsilon_2}\cdot\frac{1}{2k}=\frac{1}{2k+\varepsilon_2}=\frac{1}{m_2}\ .$$

So the coupling is correct.

With probability

$$p\geq\frac{2k}{2k+\varepsilon_1}\cdot\frac{2k}{2k+\varepsilon_2}=\left(1-\frac{\varepsilon_1}{2k+\varepsilon_1}\right)\cdot\left(1-\frac{\varepsilon_2}{2k+\varepsilon_2}\right)$$

during one decoding step the extra Black items are placed so that they are in the same half (and the processes get coupled in the next step for sure). Since $\varepsilon_i\leq(2n)^{2/3}$, one can easily show that $p\geq 1-16\sqrt[3]{4}\frac{1}{\sqrt[3]{n}}$. Then

$$E\left(\Delta(\Upsilon_{t+2},\Upsilon^\star_{t+2})\right)\leq 16\sqrt[3]{4}\frac{1}{\sqrt[3]{n}}=\beta\ .$$

Thus, according to Path Coupling Lemma

$$\tau_{\mathbf{M}}(\varepsilon)\leq\left\lceil\ln(D\varepsilon^{-1})/\ln\beta^{-1}\right\rceil$$

$$\tau_{\mathbf{M}}\left(\tfrac{1}{n}\right)\leq\left\lceil 2\ln n/\tfrac{1}{3}\ln n-\ln 4\right\rceil=O(1)\ .$$

This concludes the proof of Theorem 1.

5 Conclusions

We provide a rigorous proof that using mix-networks for Chaum's electronic elections meet high level demands on privacy: the connection between the plaintext votes and their ciphertexts remains almost purely random. This is a strong argument for using such a scenario in practice, provided that all technical problems (special printers and so) are solved. Furthermore, without loosing *privacy* we can divide whole mix-cascade into rounds including two mixing steps, instead of grouping into batches of four (as it was originally proposed in [5]). In this way we reduce the decoding complexity.

Even if the results are stated in general terms with the number of voters denoted by n, the analysis works as well for small values of n. So for practical applications concrete values may be derived easily. They may be used to choose the optimal number of trustees for a given security level and the number of voters.

Of course, such a security analysis may be applied to many mix networks as well. The convergence rate depends very much on that how large fraction of all ciphertexts goes through single mixes.

References

1. Aldous, D.: Random Walks of Finite Groups and Rapidly Mixing Markov Chains. In: Azéma, J., Yor, M. (eds.): Séminare de Probabilités XVII 1981/82. Lecture Notes in Mathematics, Vol. 986. Springer-Verlang, Berlin (1983) 243-297
2. Bubley, B., Dyer, M.: Path Coupling: A Technique for Proving Rapid Mixing in Markov Chains. In: Proceedings of the 38th Symposium on Foundations of Computer Science. Miami Beach, FL, 19-22 October 1997. IEEE Computer Society Press, Los Alamitos, CA 223-231
3. Bubley, R., Dyer, M.: Faster Random Generation of Linear Extensions. In: Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms. San Francisco, CA, 25-27 January 1998. SIAM, Philadelphia, PA 355-363
4. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: Communications of the ACM. (1981) 24(2):84-88
5. Chaum, D.: Secret-Ballot Receipts and Transparent Integrity. Better and Less-costly Electronic Voting at Polling Places. http://www.vreceipt.com/article.pdf
6. Czumaj, A., Kutyłowski, M.: Delayed Path Coupling and Generating Random Permutations. In: Random Structures and Algorithms. (2000) 17(3-4): 238-259
7. Czumaj, A., Kanarek, P., Kutyłowski, M., Loryś K.: Switching Networks for Generating Random Permutations. In: Switching Networks: Recent Advances. Kluwer Academic Publishers, (2001)
8. Czumaj, A., Kanarek, P., Kutyłowski, M., Loryś, K.: Delayed Path Coupling and Generating Random Permutations via Distributed Stochastic Processes. 10th Annual Symposium on Discrete Algorithms, ACM-SIAM, New York- Philadelphia, (1999) 271-280
9. Jakobsson, M., Juels, A., Rivest, R.R.:
   Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. http://www.rsasecurity.com/rsalabs/staff/bios/mjakobsson/rpcmix/rpcmix.pdf
10. Naor, M., Shamir, A.: Visual Cryptography. In: De Santis, A. (ed): Advances in Cryptology – Eurocrypt '94. Lecture Notes in Computer Science, Vol. 950. Springer-Verlag, Berlin (1995) 1-12