

Committal Deniable Proofs and Electronic Campaign Finance

Matt Franklin¹ * and Tomas Sander²

¹ Department of Computer Science
University of California, Davis
One Shields Avenue
Davis, CA 95616-8562, USA
franklin@cs.ucdavis.edu

² InterTrust Technologies
STAR Lab,
4750 Patrick Henry Drive
Santa Clara, CA 95054, USA
sander@intertrust.com

Abstract. In a recent Stanford Law Review article, Ayres and Bulow [1] propose a radical anonymity-based solution to disrupt the “market” for monetary influence in political campaigns. To realize their proposal, we propose new cryptographic protocols for committal deniable proofs and deniable payment schemes.

“[T]here is little reason to doubt that sometimes large contributions will work actual corruption of our political system, and no reason to question the existence of a corresponding suspicion among voters.”

– U.S. Supreme Court Justice David Souter, Nixon v. Shrink Missouri Government PAC, Jan 24, 2000.

“[Spiritually] lower than this is one who gives to the poor in a way that the giver does not know to whom he is giving and the poor person does not know who he took from. Lower than this is where the giver knows who he is giving to and the poor does not know who he is receiving from. Lower than this is where the poor knows who he is receiving from but the giver does not.” – Maimonides, Laws of Gifts to the Poor 10:7-14, 12th Century.

1 Introduction

The success of political candidates in U.S. elections depends critically on the amount of money they can spend on their campaign. Candidates may thus become vulnerable to influence buying by wealthy citizens, corporations, or Political Action Committees (i.e., groups that are able to raise and bundle significant

* Work on this paper was done while author was at Xerox PARC.

amounts). Influence buying can range from simply buying time with the candidate, to buying the opportunity to express opinions on particular political issues, to outright quid pro quo corruption where political positions are traded for donations. Candidates may also extort donations from potential donors, by threatening them with punitive treatment or indifference. The potential for political corruption has led to regular attempts to reform the system of campaign finance. Mainstream proposals include mandated disclosure of campaign donations (to expose suspicious correlations between the candidate’s positions and the donors’ interests) and limits on the amount of donations.

Ayres and Bulow [1] propose a more radical approach to disrupt the “market” for monetary influence. Any donor can contribute any amount to any candidate’s campaign, but must not be able to prove to the candidate that he made a donation. Since a true influence buyer has no more credibility than a fake influence claimer, potential influence buyers have no incentive to actually make a contribution. Furthermore a candidate who tries to extort donations (or “launder” funds through phony donors) has no way to verify that the extorted party in fact followed his blackmailing. We refer the reader directly to [1] for a more detailed discussion of “mandated donor anonymity” and its consequences, constitutionality, and political feasibility.

To implement their proposal, Ayres and Bulow offer only a trusted third party design called the “Blind Trust.” All donations are made through the Blind Trust, which has a policy of never revealing the identity of the donors. This complete reliance on a trusted third party is unsatisfying. Moreover, if donations to the blind trust are made by check or other traditional payment mechanisms, then external paper trails and bank records could later be used by the donor to prove to a candidate that a certain donation was made.

We view this as a cryptographic problem. In this paper, we will introduce the problem of deniable electronic payment mechanisms, and show how they can be applied to the mandated donor anonymity problem. Our proposal is practical, and improves on the original Blind Trust of Ayres and Bulow in several critical aspects.

Most importantly our protocol achieves verifiability: although donations are deniable it can still be publicly verified that no donations were withheld from the candidate. To achieve this we introduce the notion of commital deniable proofs and show that every statement can be proved in a commital deniable way. Commital deniable proofs allow a player to prove knowledge of certain decommitals that satisfy a predicate, without revealing which commitments he is using. Later, given witnesses to *any* decommitals that satisfy the predicate, he can later claim that these were the ones used to produce the proof. This fairly general construction seems to be applicable in many situations in which provability and deniability should be combined. Our protocol builds on ideas of Cramer and Damgard [12] for efficient zero-knowledge proofs, which in turn builds on ideas of Cramer, Damgard, and Schoenmakers [13]. In essence, we show that these earlier protocols have our new property of commital deniability.

We also elaborate on the differences of anonymity and deniability for payment schemes and show how the basic Chaumian ecash system [11] can be made deniable. Although the deniability properties of this variant of Chaumian ecash are not strong enough to give a solution to the campaign financing problem it still offers stronger privacy protecting guarantees than “solely” anonymous ecash as it allows payers to convincingly lie about how they spent their electronic money under coercion, which may be of independent interest.

1.1 Organization of the Paper

In Section 2 we describe related work. In Section 3 we state the requirements and assumptions for the donation protocol. In Section 4, we give definitions and protocols for commital deniable proofs. In Section 5, we show how to use commital deniable proofs in a deniable payment scheme. In Section 6, we elaborate on the similarities and differences of anonymity and deniability for payment schemes. Conclusions are given in Section 7.

2 Related Work

The previous discussion motivates the design of a payment mechanism that protects against the “adversarial” behavior of

1. A donor who wishes to prove to a candidate that he made donation. This influence buying donor is an example of a self-coercing adversary. If a protocol protects against this attack, we say that it is “receipt-free” for the donor.
2. A candidate who tries to extort contributions from a donor, or tries to extract information from the trust. If a protocol protects against this attack, we say that it is “incoercible” for the donor or trust.

Our deniable payment protocol will be receipt-free for the donor, and incoercible for the trust. There will be other requirements as well, e.g., public verifiability for the trust. This is all discussed in more detail in Section 3.

A scheme for “deniable encryption” was introduced by Canetti et al. [7]. Consider encryption to be a one-round, one-message protocol from the sender to the receiver. Consider deniability with respect to the message that was encrypted, i.e., a coercer who saw the ciphertext wants to know what plaintext was actually encrypted and sent. They use the term “sender-deniability” (or “receiver-” or “sender-and-receiver-”) for what we are calling incoercibility. They give a solution to these problems which require that the size of the ciphertext is linear in $1/\delta$, where δ is the probability that the adversary can distinguish the real plaintext-plus-coinflips of the sender from the phony plaintext-plus-coinflips. Canetti et al. also consider a setting they call “flexibly deniable encryption”. In this setting the sender chooses the fake message already before encryption. They give a good solution for this case. Intuitively, our notion of commital deniability can be said to lie somewhere between deniability and flexible-deniability. Canetti et al. do not consider receipt-freeness.

Building on the primitive of deniable encryption Canetti and Gennaro [9] show that any function can be securely evaluated in an incoercible way, i.e. the parties can lie about their inputs to the secure function evaluation under external coercion.

Even if one does not care about efficiency the problem with applying these incoercible protocols to the campaign donation problem is that they are not receipt-free, i.e. a self-coercing adversary can still prove what his input was. The well known reason (see [4, 5]) is that the ability of a party to lie about its input relies on its ability to produce fake random coin flips that were used for the (probabilistic) encryption of the input. Thus a party who commits itself to the used randomness by choosing it as the output of a hash function lost its ability to lie and can thereby prove what its input was.

Nevertheless, incoercible protocols can be useful for the design of receipt-free protocols. We use the commital deniable proofs to ensure public verifiability in our protocol. The protocol is still receipt-free for the donors.

Benaloh and Tuinstra [4] initiated the study of receipt-freeness and incoercibility for secret ballot election schemes (see also [18, 16, 17, 15]). To achieve incoercibility these protocols typically have a “voting booth” (or make other physical assumptions) that guarantee that the voter is isolated from the coercer for one phase of the voting protocol. In principle every receipt-free or incoercible voting scheme can be mapped into a deniable payment scheme. Each potential contributor registers as many times as he likes, paying one unit of cash for each vote. The contributor later casts as many of these votes as he likes to contribute to a particular candidate. The voting authority moves cash to candidate accounts according to the outcome of the election. Our efficient deniable payment scheme is based on different principles than the previous ones for receipt-free and incoercible voting which use homomorphic encryption, Mixnets, or blind signatures.

Canetti and Ostrovsky [10] consider multiparty computation where all parties may diverge from the protocol as long as they can do so undetectably. They distinguish the cases of globally-honest-looking and locally-honest-looking misbehavior, i.e., whether any party’s deviation is undetectable by all parties or by any one party. The problems they face are similar to the problems of defending against a self-coercing adversary.

3 Requirements and Assumptions

Before describing our solution, we discuss the requirements and assumptions, and point to some of the potential problems one runs into when designing a deniable payment mechanism. Let’s first restate the main requirement for our protocol:

Req. 1: Receipt-Freeness for the Donor: A donor should not be able to prove to the candidate that he made a donation.

If we want to avoid the cumbersome necessity of a physical donation booth, in which the donor drops his dollars into a donation box we have to allow for mechanisms for remote donations. This creates potential problems that have been recognized in remote voting systems: a coercer may look over the voter's shoulder while he casts his ballot. This applies even more so to a donation system: Candidate and donor can always get together while the donor writes the check or initiates a payment with electronic cash. The effect of this behavior in the donation setting is potentially very effective as a few donors may already account for a large total sum of donations (in contrast to remote voting in a large scale election where it is much less feasible "to watch over the shoulder" of a sufficient number of voters to influence the outcome). This includes attacks where a donor tries to prove a donation was made via covert timing channels in the financial system. For example a donor could announce to the candidate that a specific amount is about to be contributed.

A way to defeat this kind of attacks is to give a donor the ability to cancel his donation. To enable the donor to send cancellation messages for his donation to the blind trust we make the physical assumption of the availability of an:

Ass. 1.: Untappable Channel: The donor has the ability to send one (unnoticed) message to the trust via an untappable channel in some time window (e.g., two weeks).

This assumption seems to be well implementable in real life systems. Note that the trust could set up various channels to receive cancellation messages, e.g., via (anonymized) email, the phone system, postal mail ... It seems realistic to assume that not all of these channels are under the control of parties colluding with a candidate.

Thus the first main ingredient to achieve receipt-freeness for the donor is to allow for (possible) overpayment via cancellation.

To ensure the overall correctness of the deniable donation process we require

Req. 2: Verifiability: It can be publicly verified that the trust paid out no less money to the candidate than if it would have followed the protocol honestly.

Note that accountability is an important feature to ensure the overall trustworthiness of a deniable payment mechanism. As individual payers must not have individualized receipts and as, e.g., in our campaign finance application large sums of money may be involved, the system would else become a lucrative target for insider attacks that may be hard to detect.

In our protocol each cancellation message consists of a secret that is sent by the donor to the trust. For public verification the trust will construct a proof of how many secrets it has learned in the cancellation phase. (The proof does not reveal which secrets the trust has learned). Once this proof has been publicly verified there is no longer a need that it can be derived from the trusts records which the actual secrets were that the trust has learnt. This should allow to secure the trust against later coercion attempts (e.g., by a curious and powerful politician *after* he has been elected). A simple, but reasonable solution to achieve

this is that the trust “forgets” which secrets it has learnt during the cancellation phase, i.e. it erases all its corresponding records.

Note that however reliable erasure of records in the presence of multiple operators seems to be hard to ensure. (To realize the untappable channel in our system records will e.g. be received by parties operating phones, postal mail and email). The alternative solution we suggest is based on deniable proofs of knowledge. It is no longer vulnerable to the revelation of accidentally or intentionally kept copies of the revealed secrets. Accidental or coerced revelation of a record after the deniability phase of the protocol does no longer prove anything.

To ensure this deniability property of the protocol it is sufficient that the trust makes sure that random bits used for the construction of the proof were in fact randomly chosen (and not e.g. as images of a hash function).

Req. 3: Incoercibility for the Trust Without Erasure: There is a deniability phase in the protocol such that after its completion the honest trust cannot be forced to prove to anybody who the actual individual donors were or what the individually donated amounts were, even if the trust performs no erasures.

We make the following assumption about the trust:

Ass. 2: No Pre-Coercion of Trust: The trust is not coerced (or corrupted) up to the deniability phase.

This assumption implies in particular that the trust itself does not collude with the candidate for whom it is collecting the donations.

The following requirement is motivated by the fact that, e.g., foreign nationals are not allowed to make donations to U.S. campaigns. Furthermore it should be prevented that money from criminal organizations is funneled to a candidate, or that the deniable payment mechanism is abused for money laundering activities.

Req. 4: Legitimacy of Funds: The candidate should only receive donations from “legitimate sources”.

What exactly “legitimate sources” are is beyond the (technical) scope of this paper and may depend on the particular election situation. We assume that the legitimacy of the origin of (non-anonymized) funds can be determined by traditional means.

4 Committal Deniable Proofs

In this section we introduce the notion of committal deniable proofs. We show that every predicate has a committal deniable proof, using techniques of Cramer et al. [13, 12]. That is, we show that the protocol from [12] has our new property of committal deniability. The protocol requires constant rounds and message complexity proportional to the size of the formula for the predicate. This protocol is a useful building block for our deniable payment scheme.

The general intuition behind our notion is that there is a predicate and a set of unconditionally hiding commitments. A party should be able to prove knowledge of certain decommitments that satisfy the predicate, without revealing which commitments he is using. Later, given witnesses to *any* decommitments that satisfy the predicate, he can later claim that these were the ones used to produce the proof. More formally, there is a faking algorithm that takes as input the new (claimed) decommitments, the old (actually used) decommitments, the transcript of the proof, and the coin flips of the prover during the proof. The output of the faking algorithm is a new set of coin flips that is consistent with the old transcript together with the new decommitments.

Let z_1, \dots, z_n be boolean variables. A “boolean circuit” is a directed acyclic graph where every node has in-degree 0 or 2, and one node has out-degree 0. A node with in-degree 0 is called an input node, and is labeled with some z_i or \bar{z}_i (possibly repeated). A node with in-degree 2 is called a gate, and is labeled with OR or AND. The node with out-degree 0 is called the output node. Let E, I, G denote the edges, input nodes, and gates of a circuit. A “boolean formula” is a boolean circuit where no node has out-degree greater than 1. Let p be a sufficiently large prime. Let g and h have large prime order q in Z_p^* , where the discrete log of h to the base g is unknown to all parties. Let $y_{i,j} = g^{b_{i,j}} h^{r_{i,j}} \bmod p$, where $b_{i,j} \in \{0, 1\}$ and $r_{i,j} \in_R Z_q$ (unconditionally hiding boolean commitments). We say that $b_{i,j}, r_{i,j}$ is a “decommitment” of $y_{i,j}$.

In a commital deniable proof of knowledge for a language L both prover and verifier are given a formula ϕ over n boolean variables and commitments $\{y_{i,j} : 1 \leq i \leq k, 1 \leq j \leq n\}$. The prover knows decommitments $\{(b_{i^*,j}, r_{i^*,j}) : 1 \leq j \leq n\}$ for some i^* such that $\phi(b_{i^*,1}, \dots, b_{i^*,n}) = 1$.

Definition 1. *A proof system is called commital deniable if the following conditions hold:*

Completeness *When executed with an honest prover P , an honest verifier V always accepts at the end of the protocol.*

Soundness *There is a knowledge extractor such that if V accepts the proof then the knowledge extractor can find in polynomial time w.v.h.p. decommitments of $y_{i,1}, \dots, y_{i,n}$ for some i that satisfy ϕ .*

Commital Deniability *There is a faking algorithm F that takes as an input the real decommitments $b_{i^*,1}, r_{i^*,1}, \dots, b_{i^*,n}, r_{i^*,n}$ and the new decommitments $b_{i',1}, r_{i',1}, \dots, b_{i',n}, r_{i',n}$, where $i^* \neq i'$ or $i^* = i'$ are both possible, and where $\phi(b_{i^*,1}, \dots, b_{i^*,n}) = \phi(b_{i',1}, \dots, b_{i',n}) = 1$. The faking algorithm is also given the transcript T of the proof protocol and the internal coin flips of the prover. The output of F is a new sequence of internal coin flips that make the (real) transcript consistent with the new decommitments.*

Theorem 1. *Every formula has a commital deniable proof of knowledge.*

Prover and Verifier both know a boolean formula ϕ , cryptographic parameters p, q, g, h , and boolean commitments $\{y_{i,j} : 1 \leq i \leq k, 1 \leq j \leq n\}$ for some $k \geq 2$.

The Prover secretly knows $\{b_{i^*,j}, r_{i^*,j} : 1 \leq j \leq n\}$ for some i^* . The Prover wants to demonstrate to the Verifier that $\phi(\{b_{i^*,j} : 1 \leq j \leq n\}) = 1$, without revealing any useful information about i^* or the satisfying assignment.

1. Prover \rightarrow Verifier: $\{u_{i,v} : 1 \leq i \leq k, v \in I\}$.
2. Verifier \rightarrow Prover: $c \in_R Z_q$.
3. Prover \rightarrow Verifier: $\{c_i : 1 \leq i \leq k\}, \{c_{i,e} : 1 \leq i \leq k, e \in E\}, \{\alpha_{i,v} : 1 \leq i \leq k, v \in I\}$.
4. Verifier accepts if and only if the following:
 - (a) $c = \sum_{i=1}^k c_i \bmod q$.
 - (b) $c_{i,e_1} + c_{i,e_2} = c_{i,e_3} \bmod q$ for every internal OR gate with incoming edges e_1, e_2 and outgoing edge e_3 , for every $1 \leq i \leq k$.
 - (c) $c_{i,e_1} = c_{i,e_2} = c_{i,e_3}$ for every internal AND gate with incoming edges e_1, e_2 and outgoing edge e_3 , for every $1 \leq i \leq k$.
 - (d) If the output node is an OR gate with incoming edges e_1, e_2 , then $c_{i,e_1} + c_{i,e_2} = c_i \bmod q$, for every $1 \leq i \leq k$.
 - (e) If the output node is an AND gate with incoming edges e_1, e_2 then $c_{i,e_1} = c_{i,e_2} = c_i$, for every $1 \leq i \leq k$.
 - (f) For every input node $v \in I$ with label z_i and outgoing edge e , $u_{i,v}^{c_{i,e}} y_{i,j} = gh^{\alpha_{i,v}} \bmod p$, for every $1 \leq i \leq k$.
 - (g) For every input node $v \in I$ with label \bar{z}_i and outgoing edge e , $u_{i,v}^{c_{i,e}} y_{i,j} = h^{\alpha_{i,v}} \bmod p$, for every $1 \leq i \leq k$.

Claim 1: An honest prover can execute this protocol so that an honest verifier always accepts.

Claim 2: This protocol is a witness indistinguishable proof of knowledge of a ϕ -satisfying decommital of $\{y_{ij} : 1 \leq j \leq n\}$ for some i .

Claim 3: This protocol is commital-deniable.

Claim 4: The message complexity of the protocol is $O(\#Ik)$. (The last message from Prover to Verifier appears to have size $O(\#Ek)$, but it was written this way for simplicity. In fact, all of the $c_{i,e}$ can be derived from $\{c_{i,e} : 1 \leq i \leq k, e \in E_I\}$ where E_I are the out-edges of input nodes.)

Claim 1-4 yield Theorem 2.

Proof of Claim 1:

1. Prover prepares the first message to Verifier as follows:
 - (a) Choose $c_i \in_R Z_q$ for all $i \neq i^*$.
 - (b) Choose $c_{i,e} \in Z_q$ for all $i \neq i^*$ and for all $e \in E$, subject to constraints 4b-e, but otherwise drawn from the uniform distribution.
 - (c) If input node v has outgoing edge e and label z_j :
 - i. Choose $\alpha_{i,v} \in_R Z_q$ for all $i \neq i^*$.
 - ii. Compute $u_{i,v} = (gh^{\alpha_{i,v}}/y_{i,j})^{-c_{i,e}} \bmod p$ for all $i \neq i^*$.
 - iii. If $b_{i^*,j} = 0$, then choose $c_{i^*,e}, \alpha_{i^*,v} \in_R Z_q$ and compute $u_{i^*,v} = (gh^{\alpha_{i^*,v}}/y_{i^*,j})^{-c_{i^*,e}} \bmod p$.

- iv. If $b_{i^*,j} = 1$, then choose $s_{i^*,v} \in_R Z_q$ and compute $u_{i^*,v} = h^{s_{i^*,v}}$ mod p .
- (d) If input node v has outgoing edge e and label \bar{z}_j :
 - i. Choose $\alpha_{i,v} \in_R Z_q$ for all $i \neq i^*$.
 - ii. Compute $u_{i,v} = (h^{\alpha_{i,v}}/y_{i,j})^{-c_{i,e}}$ mod p for all $i \neq i^*$.
 - iii. If $b_{i^*,j} = 1$, then choose $c_{i^*,e}, \alpha_{i^*,v} \in_R Z_q$ and compute $u_{i^*,v} = (h^{\alpha_{i^*,v}}/y_{i^*,j})^{-c_{i^*,e}}$ mod p .
 - iv. If $b_{i^*,j} = 0$, then choose $s_{i^*,v} \in_R Z_q$ and compute $u_{i^*,v} = h^{s_{i^*,v}}$ mod p .
- 2. Prover receives challenge c from Verifier.
- 3. Prover prepares his response to Verifier as follows:
 - (a) Compute $c_{i^*} = c - \sum_{i \neq i^*} c_i \text{ mod } q$.
 - (b) Choose $c_{i^*,e} \in Z_q$ for all $e \in E$ for which this is still unassigned, subject to constraints 4b-e, and otherwise drawn from the uniform distribution. Note that this must be possible given that $\{b_{i^*,j} : 1 \leq j \leq n\}$ is a satisfying assignment for ϕ .
 - (c) If input node v has outgoing edge e and label z_j , and $b_{i^*,j} = 1$, then compute $\alpha_{i^*,v} = r_{i^*,j} + s_{i^*,v} c_{i^*,e} \text{ mod } q$.
 - (d) If input node v has outgoing edge e and label \bar{z}_j , and $b_{i^*,j} = 0$, then compute $\alpha_{i^*,v} = r_{i^*,j} + s_{i^*,v} c_{i^*,e} \text{ mod } q$.

Proof of Claim 2: The proof follows from Theorem 8 of Cramer et al. [13]. Here the underlying secret sharing scheme is the dual of the Benaloh-Leichter scheme [3].

Proof of Claim 3: Given knowledge of $\{b_{i',j}, r_{i',j} : 1 \leq j \leq n\}$ for some i' , Prover can compute $s_{i',v}$ for every satisfied input node v with outgoing edge e : $s_{i',v} = c_{i',e}(r_{i',j} - \alpha_{i',v})$. (Here “satisfied” means that v is labeled with z_j and $b_{i',j} = 1$, or labeled with \bar{z}_j and $b_{i',j} = 0$.) This equation allows also to compute the needed $\alpha_{i^*,v}$ for previously satisfied nodes v . Given these discrete logs, the Prover can fake the internal transcript for its computation as in the proof of Claim 1.

Corollary 1: Claims 1-4 remain true when some of the boolean variables have a fixed assignment that is known to both the Prover and Verifier. Prover and Verifier simply replace ϕ with a smaller formula that “hardwires” the assignment $\{b_j : j \in F\}$, and then proceed with the earlier protocol.

Corollary 2: The protocol can be modified for the case where ϕ is a boolean circuit but not a formula. This can be viewed as applying a standard transformation to the circuit to convert it to an equivalent formula, and then executing the original protocol on the resulting formula.

Corollary 3: When $\phi = (z_1 \vee \bar{z}_1)$, then the protocol is a commital deniable proof of knowledge of a decommital for one of k committed bits. This can easily be modified into a commital deniable proof of knowledge for $\ell \leq k$ committed bits, by modifying Verifier’s test (4a): All of the (i, c_i) should lie on a degree

$k - \ell$ polynomial that passes through $(0, c)$. It is this version of the protocol that we will need in the next section for our deniable payment scheme.

Corollary 4: Consider the formula $\Phi(x_{1,1}, \dots, x_{k,n}) = \phi(x_{1,1}, \dots, x_{1,n}) \vee \dots \vee \phi(x_{k,1}, \dots, x_{k,n})$. Then our protocol can be viewed as a committal deniable proof of knowledge of certain decommitals of inputs to Φ that guarantee that it is satisfied. Our protocol can be modified to allow a prover to demonstrate this for any sufficient subset of decommitals of any formula (instead of just the k partition subsets that we need for our applications).

5 Our Deniable Payment Protocol

In this section we will describe a protocol that allows a party to receive deniable payments. The protocol is practical. As in [1] we call this receiving party a Blind Trust and describe a five phase protocol how such a trust can be used to collect deniable donations for one candidate.

5.1 The protocol

After an initial system setup phase, our protocol has five consecutive phases: (1) Pre-donation, (2) Cancellation, (3) Verification, (4) Deniability, and (5) Reimbursement.

System Setup: A trusted organization chooses a field F_p such that G_q is a cyclic subgroup of large prime order q of the multiplicative group of F_p and that $DLOG$ is hard in G_q . Furthermore generators g, h of G_q are chosen, s.t. $\log_g h$ is unknown. p, q, g, h are made public.

1. **Pre-Donation:** Every party $D_i, 1 \leq i \leq l$ who would like to make a (deniable) donation of d_i dollars to the candidate selects d_i elements $r_{i,j} \in_R Z_q$ and d_i elements $b_{i,j} \in_R Z_2$ and computes $y_{i,j} := g^{b_{i,j}} h^{r_{i,j}}$. He transfers via a non-anonymous, payment mechanism that has receipts (e.g., checks) the amount of d_i dollars to the trust. He additionally sends the list of elements $y_{i,1} \dots, y_{i,d_i}$ to the trust. The trust verifies the legitimacy of the origin of the received funds by traditional means and enters the fact that party D_i made a pre-donation of amount d_i and the elements $y_{i,1}, \dots, y_{i,d_i}$ into a public database. D_i checks that his pre-donated amount d_i was correctly entered into the database. (If this is not the case he can complain to a third party using the receipt.) After the pre-donation phase is closed no further pre-donations are accepted.
2. **Cancellation:** A donor who wishes to “cancel” an amount of $c_i \leq d_i$ dollars of his pre-donation sends a message to the trust that contains the quadruples $(i, j, b_{i,j}, r_{i,j}), 1 \leq j \leq c_i$. The trust stores secretly all the quadruples it receives during this phase.
3. **Verification:** Assume the trust received k quadruples of discrete logs of elements in the database D during the cancellation phase. In the verification

phase the trust proves with a commital deniable proof of knowledge that it knows decommitals for k of the $d := d_1 + \dots + d_l$ elements in the public database. The trust makes a payment of $d - k$ dollars to the candidate it collects the donations for. This uses the commital deniable proof of knowledge from the previous section, as modified in Corollary 3.

4. **Deniability:** All donors are required to reveal their secret values $b_{i,j}, r_{i,j}$ to the public.
5. **Reimbursement:** In the reimbursement phase each donor who made a cancellation can contact the trust and arrange for reimbursement, e.g., with electronic cash. Here it is important that a user undeniably identifies himself to the trust (e.g., in a personal contact and with a picture I.D.) to avoid impersonation attacks of a blackmailing candidate who tries to check if actually a donation was made by trying to get reimbursed.

Theorem 2. *Under the assumption of an untappable channel and that the trust is not pre-coerced, the protocol is receipt-free for the donor. It is incoercible for the trust without erasure after completion of the deniability phase. Every system participant can verify that the candidate did not receive less money than what a trust following the protocol honestly would have paid him. Furthermore the money the candidate obtains comes from legitimate sources.*

Before giving the proof sketch we make several remarks.

1. As the protocol is receipt-free for the donor, it defends against a blackmailing candidate as well as against an influence buying donor.
2. The receipt-freeness for the donor relies on the fact that he knows the representation of the elements he submitted during the pre-donation phase. Extra measures can be taken to assure this. E.g., if each donor holds a public/secret key pair (P_K, S_K) (of which one is sure the donors know the secret key), the protocol could require that the donor's additionally submit an encryption of the representations under their public key and a ZK proof that they encrypted the correct value. These data are additionally entered into the public database.
3. Using techniques from distributed cryptography [6, 14] the trust can be distributed over several agencies that cooperate in the execution of the protocol.
4. As the donors and the values $g^b h^r$ they submitted are publicly known they can be forced to reveal their secret values in the revelation phase by external means.
5. Our cryptography based solution improves on the earlier physical implementation of [1] which does not offer verifiability. There it was suggested to achieve some form of auditability by having the trust keep all the records that could later (e.g., 10 years after the election) be publicly audited. Besides the delay, record keeping has further disadvantages. It would make the agency vulnerable to coercion, e.g., by the candidate after he won the presidential elections. Furthermore sensitive information about which donor cancelled would be revealed in the auditing process.

6. Our protocol improves on the physical implementation of deniable donations, where a donor steps into a donation booth and drops dollar bills into a donation box, as this implementation does not offer verifiability.
7. To minimize the information that can be derived about individual donations from the publicly known values d_1, \dots, d_l, k , parties who have an interest in a well functioning deniable donation mechanism can deliberately pre donate and cancel. (Note that if, e.g., k were 0 it would be clear that each donor D_i made an actual donation d_i .)
8. The complexity for construction and verification of the of the committal deniable proof depends linearly on the number of witnesses. Although this still seems to be feasible it requires significant computing power at the side of the trust and the parties that verify the proof. We think it would be interesting to improve the efficiency of the proofs (e.g., with probabilistic techniques).
9. Our protocol as it stands does not ensure that donors who cancelled get reimbursed. The trust may refuse to pay them. We sketch a variant of the protocol that prevents this: in the pre-donation phase donor D_i sends a *pair* of commitments to values $(b_{i,j}, v_{i,j})$ for his j 'th donated dollar to the trust which enters these pairs into the public database. In the cancellation phase the donor cancels his j 'th dollar by sending as before $(b_{i,j}, r_{i,j})$ to the trust. Assuming the trust receives k decommitals it pays $d - k$ dollars to the candidate. The reimbursement phase follows in this variant directly the cancellation phase. When D_i obtains his j 'th dollar back from the trust he reveals $(v_{i,j}, s_{i,j})$ in return to the trust ($s_{i,j}$ is the randomness used when committing to $v_{i,j}$). In the verifiability phase the trust proves with a committal deniable proof that it knows k out of the d pairs that were entered into the public database. Then follows the deniability phase where all donors are required to reveal their secret values $b_{i,j}, r_{i,j}, v_{i,j}, s_{i,j}$.
10. Our protocol does not protect against third party attempts (e.g., by a competing candidate) to force a donor to cancel his donation.

Proof of Theorem 2 (Sketch)

Under the assumption that the donor has an untappable channel he could have sent an unnoticed cancellation notice to the trust. In particular he can not prove to the candidate that he did not send a cancellation message to the trust. A donor could be coerced to cancel a donation, but not to make a donation that cannot be canceled later. This gives receipt-freeness for the donor.

As the trust makes a sound proof of knowledge of k out of d representations during the verification phase public, and as the total amount of pre donated funds d is publicly known, the trust can not pay less money out to the candidate than d minus the number of distinct secrets it received during the cancellation phase which proves public verifiability.

As we assumed the trust is not pre-coerced. Thus in particular the “random” bits used to produce the proof of knowledge in the verification phase were in fact chosen honestly chosen at random. After the deniability phase is completed the trust can “open” his proof of knowledge as coming from any k elementary subset

of the secret values by the previously proved properties of a commital deniable proof. This shows that the trust is later unable to prove any more information about who the actual donors (resp. the donated amounts were) than what can already be derived from the publicly known values d_1, \dots, d_l and k . As this holds even in the presence of accidentally or intentionally kept cancellation messages our protocol does not require erasure of cancellation messages.

As the trust accepts only, non-anonymous pre-donations the legitimacy of these funds and consequently also of the funds that get paid out to the candidate can be determined. This concludes the proof.

6 Deniability of Chaumian E-Cash

6.1 An incoercible payment system

In this section we study the deniability properties of anonymous electronic cash. We show how the basic anonymous ecash system can be made incoercible. Deniability can be seen as a much stronger privacy enhancing property than pure anonymity as it additionally preserves the privacy of payments under external coercion.

We briefly review the protocol for Chaumian ecash [11]. The bank has generated an RSA modulus N and a public/secret key pair (e, d) . An electronic coin consists of a pair $(x, h(x)^d)$, where h is a fixed hash function. During withdrawal the user A obtains a blind signature on $h(x)$ from the bank: A picks a random serial number x and computes $h(x)$. A picks a random “blinding factor” r and computes $m = r^e h(x)$. A sends m to the bank. The bank computes the RSA signature m^d on m and sends m^d back to A. A computes $r^{-1} m^d$ and has obtained an RSA signature on $h(x)$. During payment the user sends the coin $(x, h(x)^d)$ to the merchant who passes the coin on to the bank. The bank verifies the validity of the signature and that the coin has not been spent before. If both conditions are met the bank credits the merchant’s account with the corresponding value and enters the serial number x into its database of spent coins.

A user could be coerced by the bank, or by the government to reveal how he spent the coin obtained during a particular withdrawal session. Thus in order to make this protocol incoercible a user has to be able to “open” the message m in a way that leads to a different coin than the one he actually withdrew. As the system is unforgeable, lying can not result in the presentation of a coin that has not in fact been obtained from the bank before. However we observe that a user can open the message $m = r^e h(x)$ he sent during withdrawal to come from any other coin $(y, h(y)^d)$ he is aware of, as the following simple algorithm shows:

FAKING-ALGORITHM:

Input: $r, h(x)^d, h(y)^d$

Output: an element s , s.t. $s^e h(y) = r^e h(x)$

Algorithm:

1. Compute: $s := \frac{r h(x)^d}{h(y)^d}$.

2. Output s .

Thus the only modification needed to make the Chaumian ecash system incoercible is to require that the bank makes a list of all spent coins $(x, h(x)^d)$ public. Under coercion a user could then choose any coin of this list to open his withdrawal transcript.¹

This incoercibility protects the privacy of payments even in the presence of a later coercion attempt. Another interpretation of this observation is that the classical version of Chaumian ecash does not allow users to prove or disprove how they spent their coins under external coercion, say if they are under investigation by the police.

6.2 Self-Coercion

The protocol is quite ineffective against a self-coercing user. A self-coercing user can deviate from the protocol by choosing his blinding factor r not as a random value, but as the image of a cryptographically strong hash function H at a randomly chosen t . To prove now to somebody else that in fact a payment with a coin with serial number x was initiated by him, he presents his withdrawal record, the coin and the preimage t of the blinding factor r . He can no longer make his withdrawal record look like that of another coin, as this would require him to find a preimage of $rh(x)^d/h(y)^d$ under H . (Note that even the knowledge of the secret key d of the bank does not seem to help to create withdrawal records that can be opened as two different coins as the serial numbers are also images of hash functions.)

7 Conclusions

In conclusion, it is possible to use cryptographic methods to implement the radical campaign proposal of Ayres and Bulow. The building blocks of committal deniable proofs and deniable payment schemes are interesting in their own right, and may well find other applications. It would be interesting to find efficient deniable versions for other cryptographic applications, e.g., for anonymous remailing where the traffic from the client to the mix is observed by a potential future coercer.

References

1. I. Ayres and J. Bulow, "The donation booth: Mandating donor anonymity to disrupt the market for political influence", *Stanford Law Review* 50(3), 1998.

¹ We remark that extra measures can and should be taken to make this protocol more fault tolerant. One obvious attack could be that a bank could enter fake records into the list of spent coins and thereby catch a user who lies that he made that payment. To make lying of users more convincing it could be additionally required that the banks publishes in the database the recipient of the coin.

2. D. Beaver, "Plug and play encryption", *Crypto '97*, pp. 75–89.
3. J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions", *Crypto '88*, pp. 27–35.
4. J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections", *ACM Symposium on the Theory of Computing*, 1994, pp. 544–552.
5. J. Benaloh and D. Tuinstra, "Incoercible communication", *Clarkson University Department of Mathematics and Computer Science Technical Report number TR-MCS-94-1*, Feb. 94.
6. J. Benaloh and M. Yung, "Distributing the power of a government to enhance the privacy of voters", *ACM Symposium on Principles of Distributed Computing*, 1985.
7. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable encryption", *Crypto '97*, pp. 90–104.
8. R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure computation", *ACM Symposium on the Theory of Computing*, 1996.
9. R. Canetti and R. Gennaro, "Incoercible multiparty computation", *IEEE Foundations of Computer Science*, 1996.
10. R. Canetti and R. Ostrovsky, "Secure computation with honest-looking parties: What if nobody is truly honest?", *ACM Symposium on the Theory of Computing*, 1999.
11. D. Chaum, "Blind signatures for untraceable payments", *Crypto'82*, pp. 199–203.
12. R. Cramer and I. Damgard, "Linear zero-knowledge – A note on efficient zero-knowledge proofs and arguments", *ACM Symposium on the Theory of Computing*, 1997, 436–445.
13. R. Cramer, I. Damgard, B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols", *Crypto '94*, pp. 174–187.
14. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game, or: A completeness theorem for protocols with honest majority", *ACM Symposium on Theory of Computing*, 1987, 218–229.
15. M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption", *Eurocrypt 2000*, pp.539–556.
16. V. Niemi and A. Renvall, "How to prevent buying of votes in computer elections", *Asiacrypt'94*, pp. 141–148.
17. T. Okamoto, "Receipt-free electronic voting schemes for large scale elections", *Security Protocol Workshop '97*, LNCS, pp. 25 –35.
18. K. Sako and J. Kilian, "Receipt-free mix-type voting schemes", *Eurocrypt '95*, pp. 393–403.