

Receipt-Free Voting with Randomizers

Martin Hirt

ETH Zurich

1. Voting Introduction

- What is **voting**?
- Known **approaches**
- Motivation of **receipt-freeness**

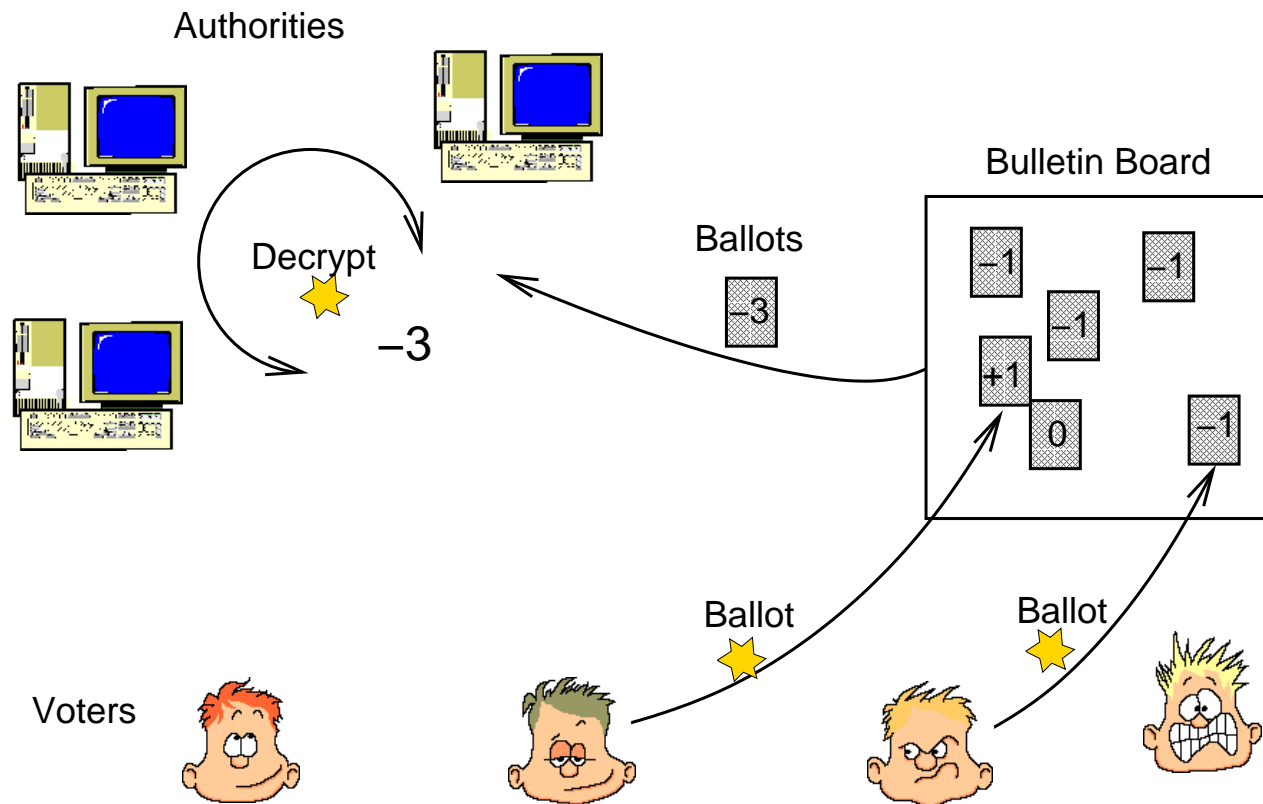
2. Protocol of Cramer/Gennaro/Schoenmakers [CGS97]

- Based on **homomorphic encryption**
- **Most efficient** known voting protocol

3. Receipt-Free Vote Generation

- Based on **randomizers** [LK00]
- **More efficient** than any previous receipt-free scheme

Part II: Voting Protocol of [CGS97]



- single SK/PK, **SK is shared** among authorities
- ballot = encrypted vote + validity proof
- public tallying with using homomorphism
- authorities **jointly decrypt** and prove tally (e.g. MPC)

Yes/No

- “yes” \rightarrow 1, “no” \rightarrow 0.
- tally \rightarrow number of “yes”, number of “no”.

Yes/No/Empty

- “yes” \rightarrow 1, “no” \rightarrow -1, “empty” \rightarrow 0.
- tally \rightarrow number of “yes” minus number of “no”.

L Candidates

- M : Upper bound on number of voters.
- candidate 1 \rightarrow 1, candidate 2 $\rightarrow M$, ..., candidate $L \rightarrow M^{L-1}$.
- tally \rightarrow number of votes per candidate.

Encryption function: $(v, \alpha) \mapsto E(v, \alpha)$

Requirements

- **semantically secure** (w.r.t. v)
- **homomorphic**: $E(v_1, \alpha_1) \otimes E(v_2, \alpha_2) = E(v_1 + v_2, \alpha_1 + \alpha_2)$
- **distributed set-up** (threshold security)
- **verifiable decryption** (threshold security)

Instances

- **[CGS97]**: variant of [ElGamal84], with [Pedersen91] setup
- **[DJ00], [FPS00]**: threshold setup for [Paillier99]

Setup [Ped91, CGS97]

- Cyclic group $G = \langle g \rangle = \langle \gamma \rangle$
- Shared SK z , PK $Z = g^z$

Encryption

- $E(v, \alpha) = (g^\alpha, \gamma^v Z^\alpha)$, for $\alpha \in_R \{0, \dots, |G| - 1\}$

Homomorphism

- $(x_1, y_1) \otimes (x_2, y_2) \stackrel{\text{def}}{=} (x_1 x_2, y_1 y_2)$
 $\Rightarrow E(v_1, \alpha_1) \otimes E(v_2, \alpha_2) = E(v_1 + v_2, \alpha_1 + \alpha_2)$

Decryption [ElGamal84, CGS97]

- $E(T) = (x, y) \rightarrow \frac{y}{x^z} = \frac{\gamma^T Z^\alpha}{(g^\alpha)^z} = \frac{\gamma^T (g^z)^\alpha}{(g^\alpha)^z} = \gamma^T$
- $\gamma^T \rightarrow T$, with costs $O(T)$

Part III: Receipt-Free Vote Generation

Entities

- N authorities, at least t of them remain honest.
- M voters.

Communication

- Bulletin board (public channels).
- Untappable channels from authorities to voters.

PKI

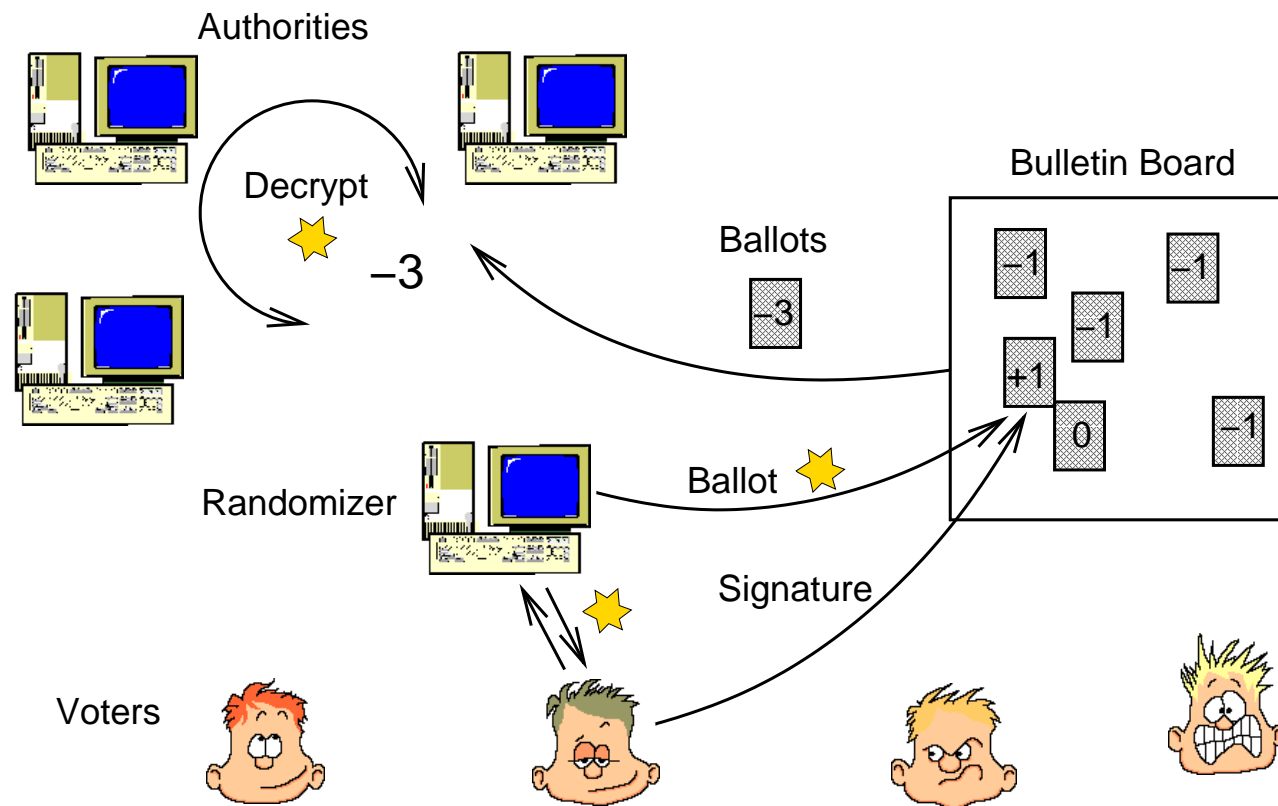
- Each voter has a SK z_v and a PK $Z_v = g^{z_v}$.

Generality

- 1-out-of- L voting scheme, set of valid votes \mathcal{V} .

Security

- Correctness \Leftarrow at least t honest authorities tally.
- Privacy \Leftarrow less than t authorities are curious.
- Receipt-free \Leftarrow no collaboration with coercer (can be weakened).



- randomizer re-encrypts and proves ballot: **randomization proof**
- voter and randomizer **jointly** generate **validity proof**
- **several randomizers** help avoiding denial-of-service attacks

Given: $f : (G, \oplus) \rightarrow (H, \otimes)$ with $f(x \oplus x') = f(x) \otimes f(x')$.

Prover

Verifier

knows x with $f(x) = y$

knows y

$r \in_R G, t = f(r)$



$c \in_R \mathbb{Z}_q$

$s = r \oplus cx$



$f(s) \stackrel{?}{=} t \otimes y^c$

- honest-verifier zero-knowledge
- addition requirement (“ q -invertibility”)

Prover

knows i, x with $f(x) = y_i$

$$r_i \in_R G, t_i = f(r_i)$$

For $j = 1, \dots, L, j \neq i$:

$$c_j, s_j \in_R \mathbb{Z}_q$$

$$t_j = f(s_j) \otimes y^{c_j}$$

$$c_i = c - \sum_{j=1, j \neq i}^L c_j$$

$$s_i = r_i \oplus c_i x$$

Verifier

knows y_1, \dots, y_L

$$\xrightarrow{t_1, \dots, t_L}$$

$$\xleftarrow{c} \quad c \in_R \mathbb{Z}_q$$

$$\xrightarrow{s_1, \dots, s_L, c_1, \dots, c_L} \quad c \stackrel{?}{=} \sum_{j=1}^L c_j$$

For $j = 1, \dots, L$:

$$f(s_j) \stackrel{?}{=} t_j \otimes y^{c_j}$$

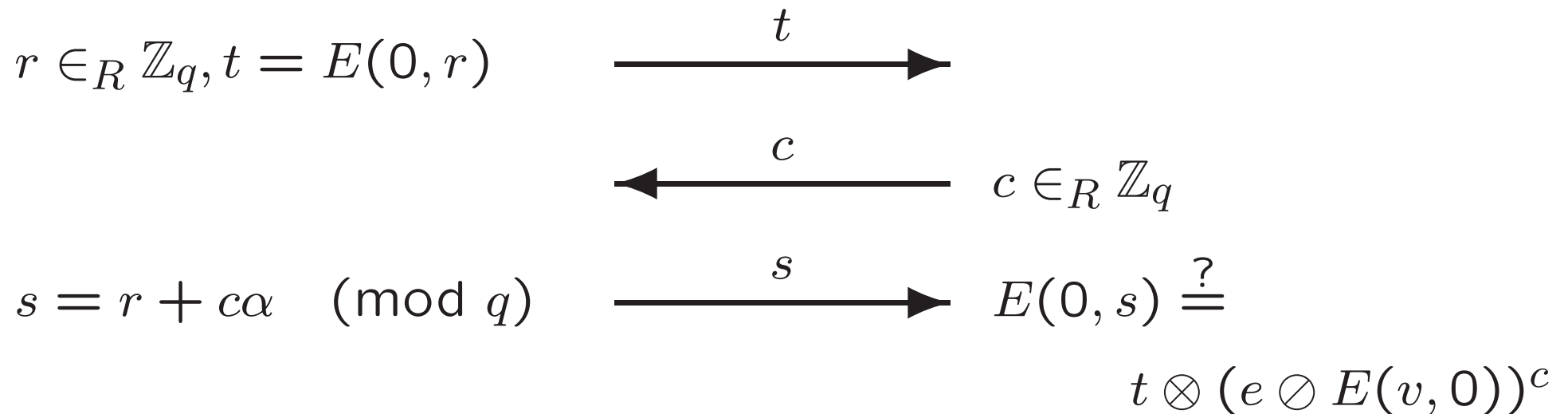
Given: Encryption e , vote v , voter knows α with $e = E(v, \alpha)$.

Idea: $f : r \mapsto E(0, r)$

prove knowledge of pre-image β with $f(\beta) = e \otimes E(v, 0)$.

Randomizer

Voter



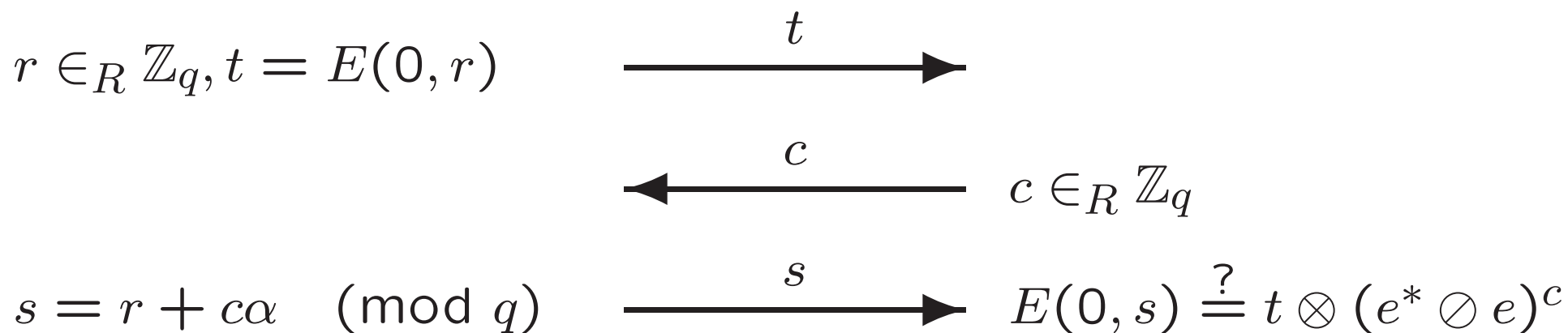
Given: Randomizer knows α with $e^* = e \otimes E(0, \alpha)$.

Idea: $f : r \mapsto E(0, r)$,

prove knowledge of pre-image β with $f(\beta) = e^* \oslash e$.

Randomizer

Voter



non-interactive: pair (t, s) , accepting for $c = H(t)$.

Problem: Voter can give proof to vote-buyer.

Prover knows:

- secret value x , satisfies predicate $P(x)$.

Verifier knows:

- secret key z , satisfies predicate $Q_Z(z)$.

Goal:

- prover proves knowledge of χ with $P(\chi)$.
- verifier is convinced.
- verifier cannot convince anyone that $\exists \chi : P(\chi)$.

Idea:

- prove knowledge of (χ, ζ) with $P(\chi) \vee Q_Z(\zeta)$.

Given: SK z_v of voter, PK $Z_v = g^{z_v}$

Idea: $f : r \mapsto g^r$,

prove knowledge of pre-image ζ with $f(\zeta) = Z_v$.

Voter

Verifier

$$r \in_R \mathbb{Z}_q, t = g^r$$



$$c \in_R \mathbb{Z}_q$$

$$s = r + cz_v \pmod{q}$$



$$g^s \stackrel{?}{=} t \cdot Z_v^c$$

Randomizer

Voter

knows α with

$$E(0, \alpha) = e^* \otimes e$$

knows e, e^*

$$r_1 \in_R \mathbb{Z}_q, t_1 = E(0, r_1)$$

$$c_2, s_2 \in_R \mathbb{Z}_q, t_2 = g^{s_2} / Z_v^{c_2}$$

 t_1, t_2
 c

$$c \in_R \mathbb{Z}_q$$

$$c_1 = c - c_2 \pmod{q}$$

$$s_1 = r_1 + c_1 \alpha \pmod{q}$$

 s_1, s_2, c_1, c_2

$$c_1 + c_2 \stackrel{?}{=} c \pmod{q}$$

$$E(0, s_1) \stackrel{?}{=} t_1 \otimes (e^* \otimes e)^{c_1}$$

$$g^{s_2} \stackrel{?}{=} t_2 \cdot Z_v^{c_2}$$

Given: Encryption e , Valid votes $\mathcal{V} = \{v_1, \dots, v_L\}$

Homomorphism: $f : r \mapsto E(0, r)$

Prove:

Know α_1 with $f(\alpha_1) = e \oslash E(v_1, 0)$

OR

know α_2 with $f(\alpha_2) = e \oslash E(v_2, 0)$

OR

...

OR

know α_L with $f(\alpha_L) = e \oslash E(v_L, 0)$

Voter

$$r_i \in_R G, t_i = E(0, r_i)$$

For $j = 1, \dots, L, j \neq i$:

$$c_j, s_j \in_R \mathbb{Z}_q$$

$$t_j = E(0, s_j) \otimes$$

$$(e \otimes E(v_j, 0))^{c_j}$$

$$\xrightarrow{t_1, \dots, t_L}$$

$$\xleftarrow{c}$$

$$c \in_R \mathbb{Z}_q$$

$$c_i = c - \sum_{j=1, j \neq i}^L c_j \pmod{q}$$

$$s_i = r_i + c_i \alpha \pmod{q}$$

$$\xrightarrow{s_1, \dots, s_L, c_1, \dots, c_L}$$

$$c \stackrel{?}{=} \sum_{j=1}^L c_j \pmod{q}$$

For $j = 1, \dots, L$:

$$E(0, s_j) \stackrel{?}{=} t_j \otimes$$

$$(e \otimes E(v_j, 0))^{c_j}$$

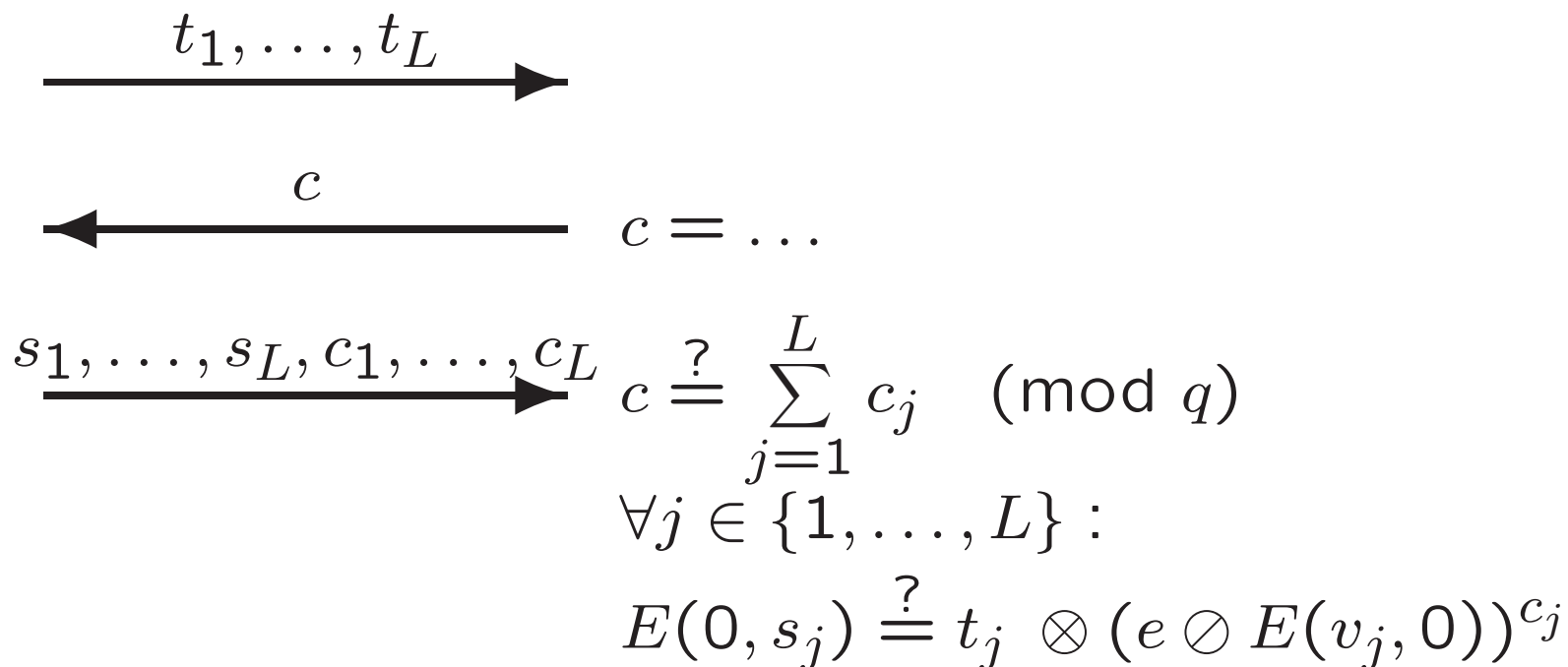
Use Fiat-Shamir heuristics

Tuple $(t_1, \dots, t_L, s_1, \dots, s_L, c_1, \dots, c_L)$ such that

- $c_1 + \dots + c_L = \mathcal{H}(t_1, \dots, t_L)$, and
- s_1, \dots, s_L is accepting.

Voter

Randomizer



Randomize:

$$\left. \begin{array}{l} \bullet s_j \rightarrow s_j + s'_j \\ \bullet c_j \rightarrow c_j + c'_j \end{array} \right\} \Rightarrow t_j \rightarrow t_j \otimes E(c'_j v_j, s'_j) \otimes e^{c'_j}$$

Adjust:

$$\bullet e \rightarrow e + E(0, \alpha) \Rightarrow s_j \rightarrow s_j + \alpha c_j$$

- **Very efficient**
almost as [CGS97] (up to constant factor)
- **Receipt-free**
as long as randomizer does not cooperate with vote-buyer
- **Private** and **correct**
as long as honest quorum
- Conceptually very **simple**
slight adaption of [CGS97]