Private Elections in the Smartphone Era_

Practical CodeVoting Revisited


Helvetia 100
Stanserhorn-Bahn
Peter Wähler
Haupstrasse 100
3000 Bern

VOTING CARD #1823

Do you accept the tax law?

Yes

Verification Code:
CB14

CH23-03

Scan QR code on back and check verification code

VOTING CARD #1823

Do you accept the tax law?

No

Verification Code:
F142

CH23-03

Scan QR code on back and check verification code

VOTING CARD #1823

Do you accept the tax law?

Abstain

Verification Code:
0B1D

CH23-03

Scan QR code on back and check verification code

CONFIRMATION CARD #1823

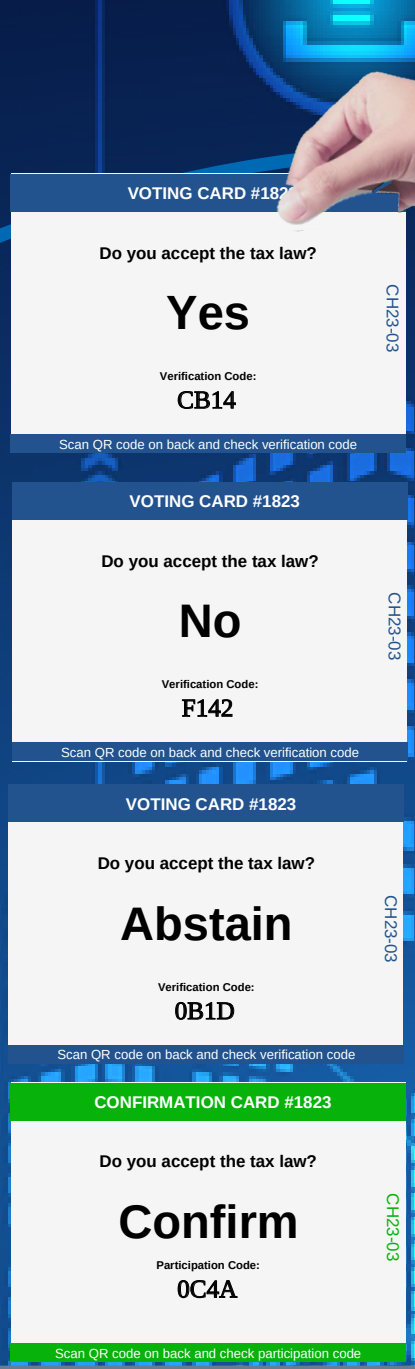
Do you accept the tax law?

Confirm

Participation Code:
0C4A

CH23-03

Scan QR code on back and check participation code


VOTING CARD #182
Do you accept the tax law?
Yes
Verification Code:
CB14
CH23-03
Scan QR code on back and check verification code
VOTING CARD #1823
Do you accept the tax law?
No
Verification Code:
F142
CH23-03
Scan QR code on back and check verification code
VOTING CARD #1823
Do you accept the tax law?
Abstain
Verification Code:
0B1D
CH23-03
Scan QR code on back and check verification code
CONFIRMATION CARD #1823
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code

VOTING CARD #1823

Do you accept the tax law?

CH23-03

Scan QR code and check verification code on front

VOTING CARD #1823

Do you accept the tax law?

No

Verification Code:

F142

CH23-03

Scan QR code on back and check verification code

VOTING CARD #1823

Do you accept the tax law?

Abstain

Verification Code:

0B1D

CH23-03

Scan QR code on back and check verification code

CONFIRMATION CARD #1823

Do you accept the tax law?

Confirm

Participation Code:

0C4A

CH23-03

Scan QR code on back and check participation code

VOTING CARD #1823

Do you accept the tax law?

CH23-03

Scan QR code and check verification code on front

VOTING CARD #1823

Do you accept the tax law?

CH23-03

Scan QR code and check verification code on front

VOTING CARD #1823

Do you accept the tax law?

CH23-03

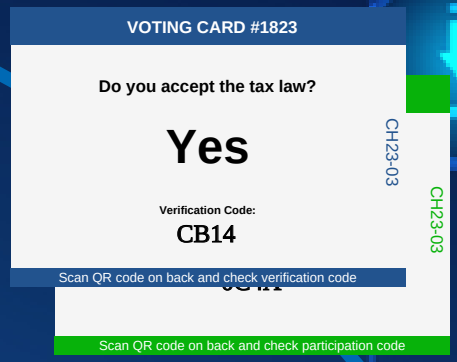
Scan QR code and check verification code on front

CONFIRMATION CARD #1823

Do you accept the tax law?

CH23-03

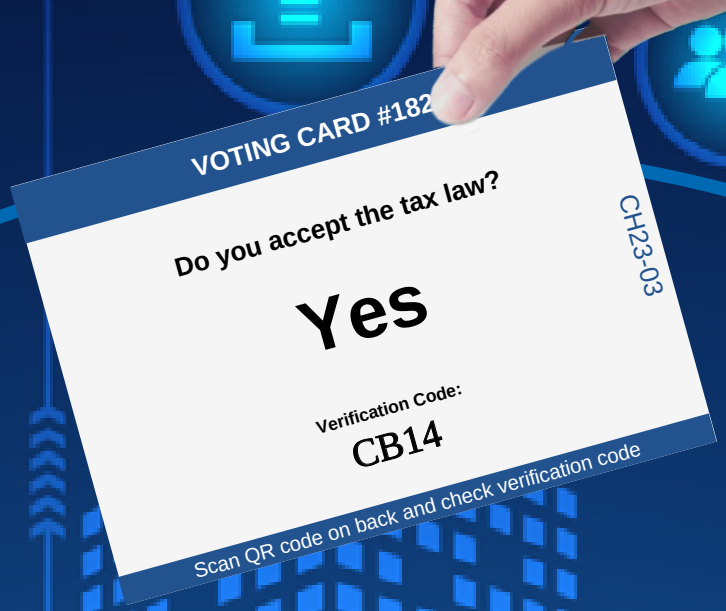
Scan QR code and check participation code on front


VOTING CARD #1823
Do you accept the tax law?
Yes
Verification Code:
CB14
Scan QR code on back and check verification code
CH23-03
CH23-03
Scan QR code on back and check participation code


CONFIRMATION CARD #1823
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code
VOTING CARD #182
Do you accept the tax law?
Yes
Verification Code:
CB14
CH23-03
Scan QR code on back and check verification code
CH23-03
Scan QR code and check verification code on front


CONFIRMATION CARD #1823
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code
VOTING CARD #182
Do you accept the tax law?
Yes
Verification Code:
CB14
CH23-03
Scan QR code on back and check verification code
i=1823&j=1&s=ug76JGjnkd16778
CH23-03
Scan QR code and check verification code on front


CONFIRMATION CARD #1823
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code
VOTING CARD #182
Do you accept the tax law?
Yes
Verification Code:
CB14
CH23-03
Scan QR code on back and check verification code
i=1823&j=1&s=ug76JGjnkd16778
CB14


VOTING CARD #182
Do you accept the tax law?
Yes
Verification Code:
CB14
CH23-03
Scan QR code on back and check verification code


CONFIRMATION CARD #1823
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code


qrvote.ch?U=CH23-03&i=1823&j=1&s=ug76JGjnkd16778
CB14


CONFIRMATION
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code


CONFIRMATION
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code
qrvote.ch?U=CH23-0
Do you accep
CH23-03


CONFIRMATION
Do you accept the tax law?
Confirm
Participation Code:
0C4A
CH23-03
Scan QR code on back and check participation code
qrvote.ch?U=CH23-0
Do you accep
0C4A
CH23-03
Scan QR code and che

Usable Verifiable Secrecy-Preserving E-Voting

Oksana Kulyk[1], Jonas Ludwig[2], Melanie Volkamer[2], Reto E. Koenig[3], and Philipp Locher[3]

[1] IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S
[2] Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen
[3] Bern University of Applied Sciences, Quellgasse 21, 2501 Biel

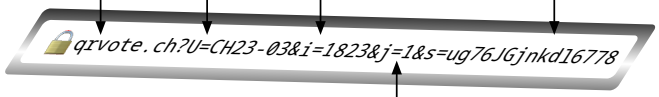
Abstract. In this paper we propose the usage of QR-Codes to enable usable verifiable e-voting schemes based on code voting. The idea – from a voter's perspective – is to combine code voting proposed by Chaum with the cast-as-intended verification mechanism used e.g. in Switzerland (using a personal initialization code, return codes per option, a confirmation code and a finalisation code); while all codes to be entered into the e-voting system by voters are available as QR-Code (i.e. one personalised QR voting code per voting option and one personal confirmation QR-Code). We conduct a user study to evaluate the usability and user experience of such an approach: both the code sheets and the election webpage are based on usability research in this area but adopted for our idea. As our proposal performs good wrt. usability, we discuss how such usable front-ends enable more secure e-voting systems in respect to end-to-end verifiability and vote secrecy.

Election Identifier *U*

BLS Signature *σ* of *(U,i,j)*

URL

Voting Card Index *i*



$$(U, i, j, \sigma)$$

Permuted voting-option index *j*

Step1

Initialize $B_{ik} \leftarrow \emptyset$, for $i \in [1, N]$
Initialize $C_k \leftarrow \emptyset$

qrDecode(QR_s)

b →

Step2c

$(\hat{U}, i, j, \sigma) \leftarrow b_s$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i, j)) = 0$, abort
If $b_s \in B_{ik}$, abort
If $|B_{ik}| = m$, abort
If $(i, \cdot) \in C_k$, abort

Voting Client	Election Authority $k \in [1, s]$

Step1

Initialize $B_{ik} \leftarrow \emptyset$, for $i \in [1, N]$
Initialize $C_k \leftarrow \emptyset$

Step2b

$QR_s \longrightarrow$ $b_s \leftarrow \mathsf{qrDecode}(QR_s)$ $\xrightarrow{\;b\;}$

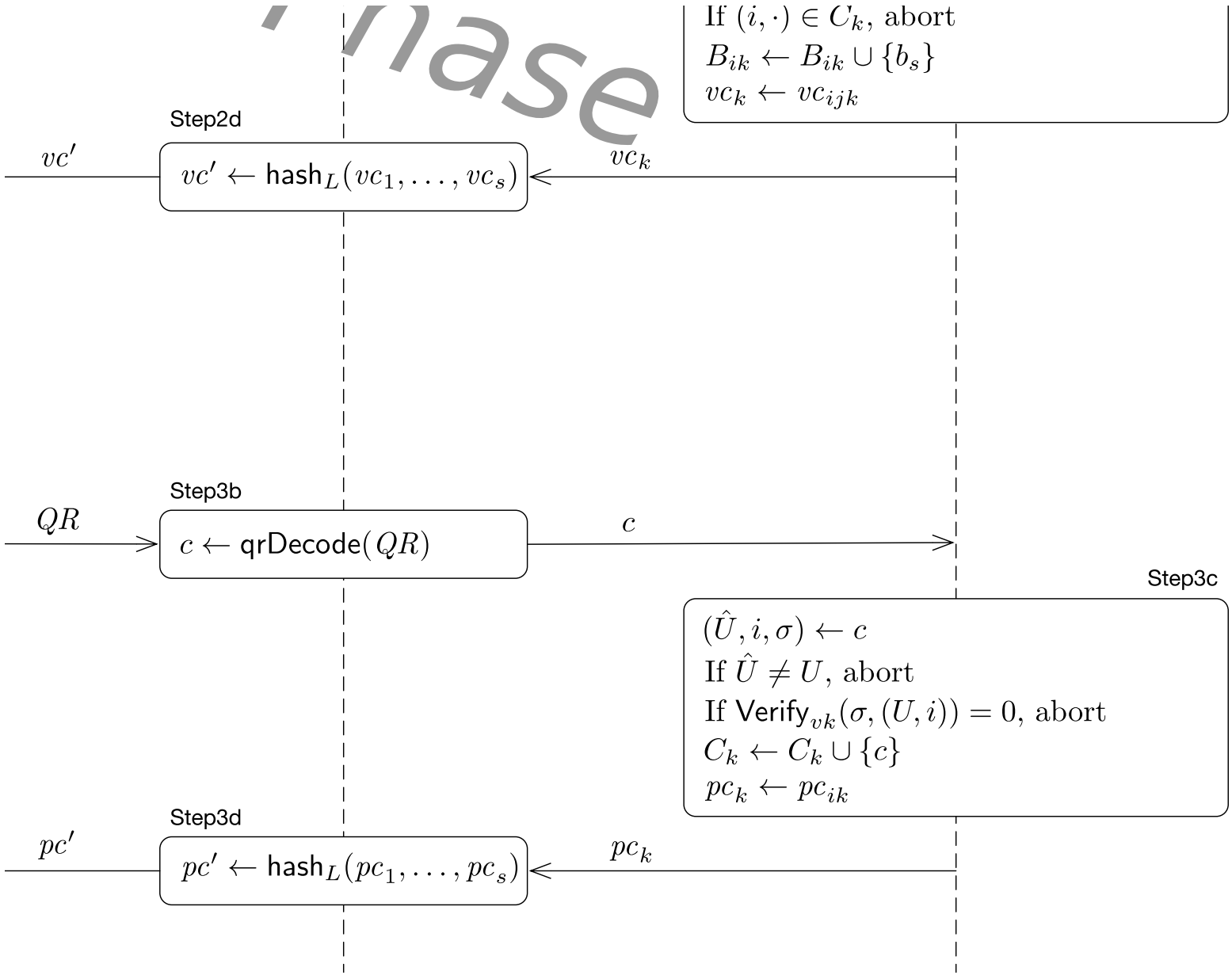
Step2c

$(\hat{U}, i, j, \sigma) \leftarrow b_s$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i, j)) = 0$, abort
If $b_s \in B_{ik}$, abort
If $|B_{ik}| = m$, abort
If $(i, \cdot) \in C_k$, abort
$B_{ik} \leftarrow B_{ik} \cup \{b_s\}$
$vc_k \leftarrow vc_{ijk}$

Step2d

$vc' \longleftarrow$ $vc' \leftarrow \mathsf{hash}_L(vc_1, \ldots, vc_s)$ $\xleftarrow{\;vc_k\;}$


Peter Wähler
Haupstrasse 100
3000 Bern


If $(i, \cdot) \in C_k$, abort
$B_{ik} \leftarrow B_{ik} \cup \{b_s\}$
$vc_k \leftarrow vc_{ijk}$
Step2d
vc'
$vc' \leftarrow \mathsf{hash}_L(vc_1, \ldots, vc_s)$
vc_k
Step3b
QR
$c \leftarrow \mathsf{qrDecode}(QR)$
c
Step3c
$(\hat{U}, i, \sigma) \leftarrow c$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i)) = 0$, abort
$C_k \leftarrow C_k \cup \{c\}$
$pc_k \leftarrow pc_{ik}$
Step3d
pc'
$pc' \leftarrow \mathsf{hash}_L(pc_1, \ldots, pc_s)$
pc_k

Phase

If $(i, \cdot) \in C_k$, abort
$B_{ik} \leftarrow B_{ik} \cup \{b_s\}$
$vc_k \leftarrow vc_{ijk}$

Step2d

vc' — $vc' \leftarrow \mathsf{hash}_L(vc_1, \ldots, vc_s)$ — vc_k

Step3b

QR — $c \leftarrow \mathsf{qrDecode}(QR)$ — c

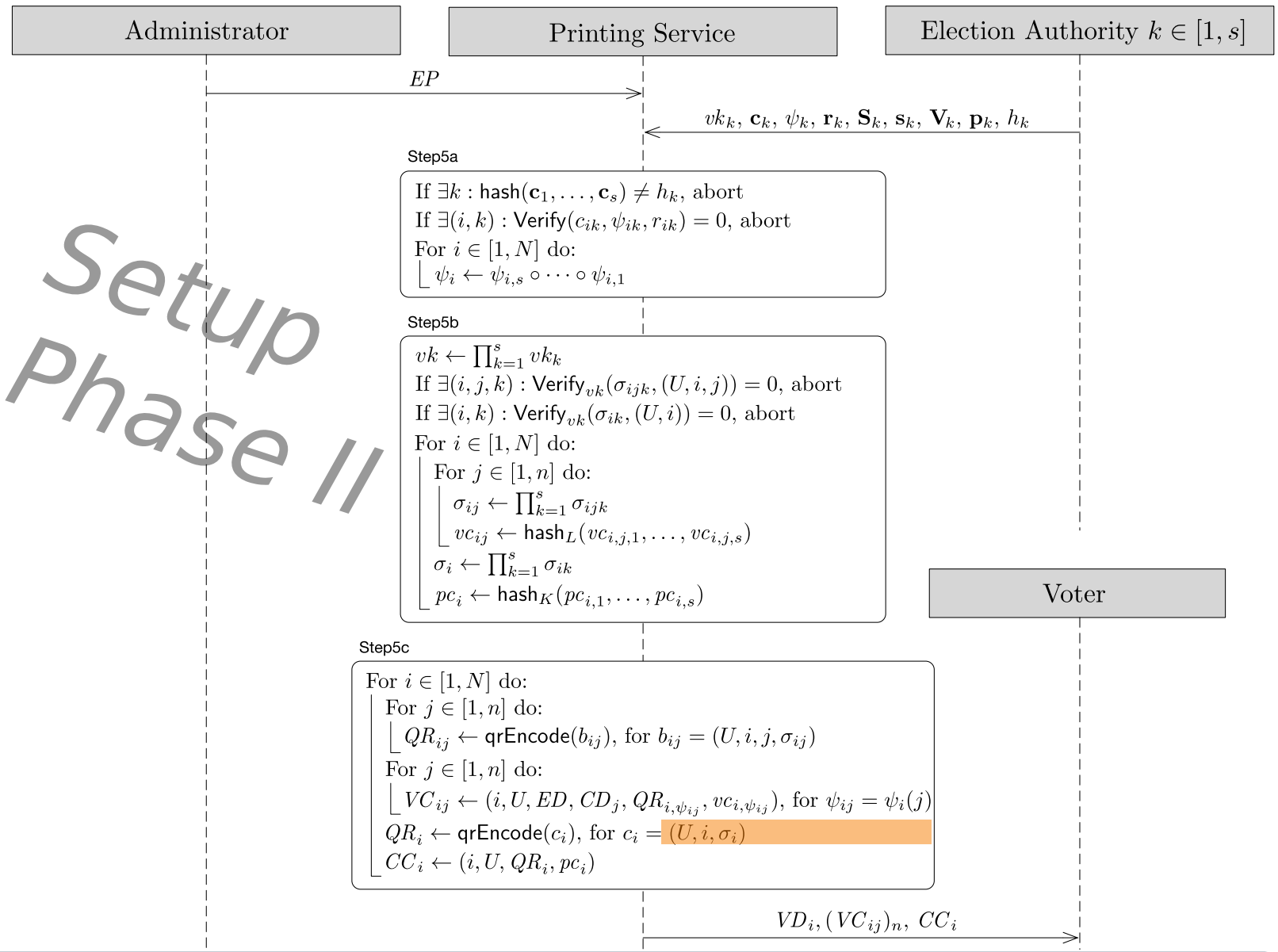
Step3c

$(\hat{U}, i, \sigma) \leftarrow c$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i)) = 0$, abort
$C_k \leftarrow C_k \cup \{c\}$
$pc_k \leftarrow pc_{ik}$

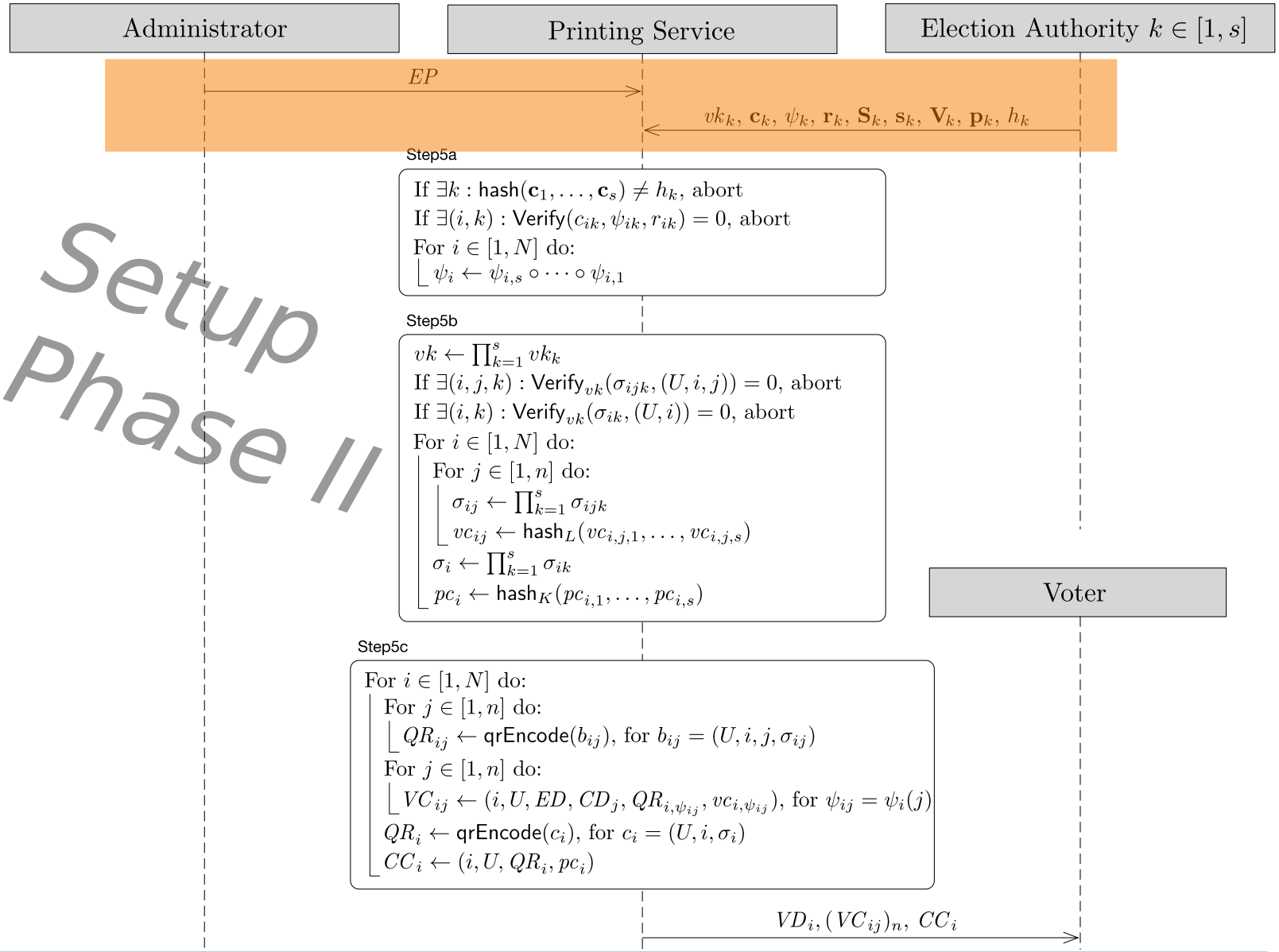
Step3d

pc' — $pc' \leftarrow \mathsf{hash}_L(pc_1, \ldots, pc_s)$ — pc_k

Setup Phase II



Setup Phase II



Setup Phase I

Administrator	Election Authority $k \in [1, s]$	Election Authority $\ell \neq k$

Step1

Select $EP = (U, ED, m, n, N, \mathbf{c}, \mathbf{v})$

$\xrightarrow{EP}$

Step2a

$(dk_k, ek_k) \xleftarrow{\$} \mathsf{KeyGenEnc}()$

$(sk_k, vk_k) \xleftarrow{\$} \mathsf{KeyGenSig}()$

$\pi^{\mathsf{key}} \xleftarrow{\$} \mathsf{NIZKP}[(dk, sk) : ek = g^{dk} \wedge vk = g_2^{sk}]$

$\xrightarrow{ek_k,\ vk_k,\ \pi_k^{\mathsf{key}}}$

$\xleftarrow{ek_\ell,\ vk_\ell,\ \pi_\ell^{\mathsf{key}}}$

Step2b

If $\exists \ell : \mathsf{Verify}(\pi_\ell^{\mathsf{key}}, ek_\ell, vk_\ell) = 0$, abort

$ek \leftarrow \prod_{k=1}^{s} ek_k, \quad vk \leftarrow \prod_{k=1}^{s} vk_k$

Step3a

For $j \in [1, n]$ do:

$\quad e_j \leftarrow \mathsf{Enc}_{ek}(\Gamma(j), 0)$

For $i \in [1, N]$ do:

$\quad \tilde{\mathbf{e}}_{i,0} \leftarrow (e_j)_n$

Step3b (upon receiving previous shuffle)

For $i \in [1, N]$ do:

$\quad \psi_{ik} \xleftarrow{\$} \Psi_n,\ \mathbf{r}_{ik} \xleftarrow{\$} \mathbb{Z}_q^n$

$\quad \tilde{\mathbf{e}}_{ik} \leftarrow \mathsf{Shuffle}_{ek}(\tilde{\mathbf{e}}_{i,k-1}, \psi_{ik}, \mathbf{r}_{ik})$

$\quad (\pi_{ik}^{\mathsf{shuffle}}, c_{ik}, r_{ik}) \xleftarrow{\$} \mathsf{NIZKP}[(\psi_{ik}, \mathbf{r}_{ik}) : \tilde{\mathbf{e}}_{ik} = \mathsf{Shuffle}_{ek}(\tilde{\mathbf{e}}_{i,k-1}, \psi_{ik}, \mathbf{r}_{ik})]$

$\mathbf{c}_k \leftarrow (c_{ik})_N,\ \psi_k \leftarrow (\psi_{ik})_N,\ \mathbf{r}_k \leftarrow (r_{ik})_N$

$\xrightarrow{(\tilde{\mathbf{e}}_{ik})_N,\ (\pi_{ik}^{\mathsf{shuffle}})_N,\ \mathbf{c}_k}$

Step2b

If $\exists \ell : \mathsf{Verify}(\pi_\ell^{\mathsf{key}}, ek_\ell, vk_\ell) = 0$, abort
$ek \leftarrow \prod_{k=1}^{s} ek_k, \;\; vk \leftarrow \prod_{k=1}^{s} vk_k$

Step3a

For $j \in [1, n]$ do:
  $e_j \leftarrow \mathsf{Enc}_{ek}(\Gamma(j), 0)$
For $i \in [1, N]$ do:
  $\tilde{\mathbf{e}}_{i,0} \leftarrow (e_j)_n$

Step3b (upon receiving previous shuffle)

For $i \in [1, N]$ do:
  $\psi_{ik} \xleftarrow{\$} \Psi_n, \; \mathbf{r}_{ik} \xleftarrow{\$} \mathbb{Z}_q^n$
  $\tilde{\mathbf{e}}_{ik} \leftarrow \mathsf{Shuffle}_{ek}(\tilde{\mathbf{e}}_{i,k-1}, \psi_{ik}, \mathbf{r}_{ik})$
  $(\pi_{ik}^{\mathsf{shuffle}}, c_{ik}, r_{ik}) \xleftarrow{\$} \mathsf{NIZKP}[(\psi_{ik}, \mathbf{r}_{ik}) : \tilde{\mathbf{e}}_{ik} = \mathsf{Shuffle}_{ek}(\tilde{\mathbf{e}}_{i,k-1}, \psi_{ik}, \mathbf{r}_{ik})]$
$\mathbf{c}_k \leftarrow (c_{ik})_N, \; \psi_k \leftarrow (\psi_{ik})_N, \; \mathbf{r}_k \leftarrow (r_{ik})_N$

$(\tilde{\mathbf{e}}_{ik})_N, \; (\pi_{ik}^{\mathsf{shuffle}})_N, \mathbf{c}_k \longrightarrow$

$\longleftarrow (\tilde{\mathbf{e}}_{i\ell})_N, \; (\pi_{i\ell}^{\mathsf{shuffle}})_N, \mathbf{c}_k$

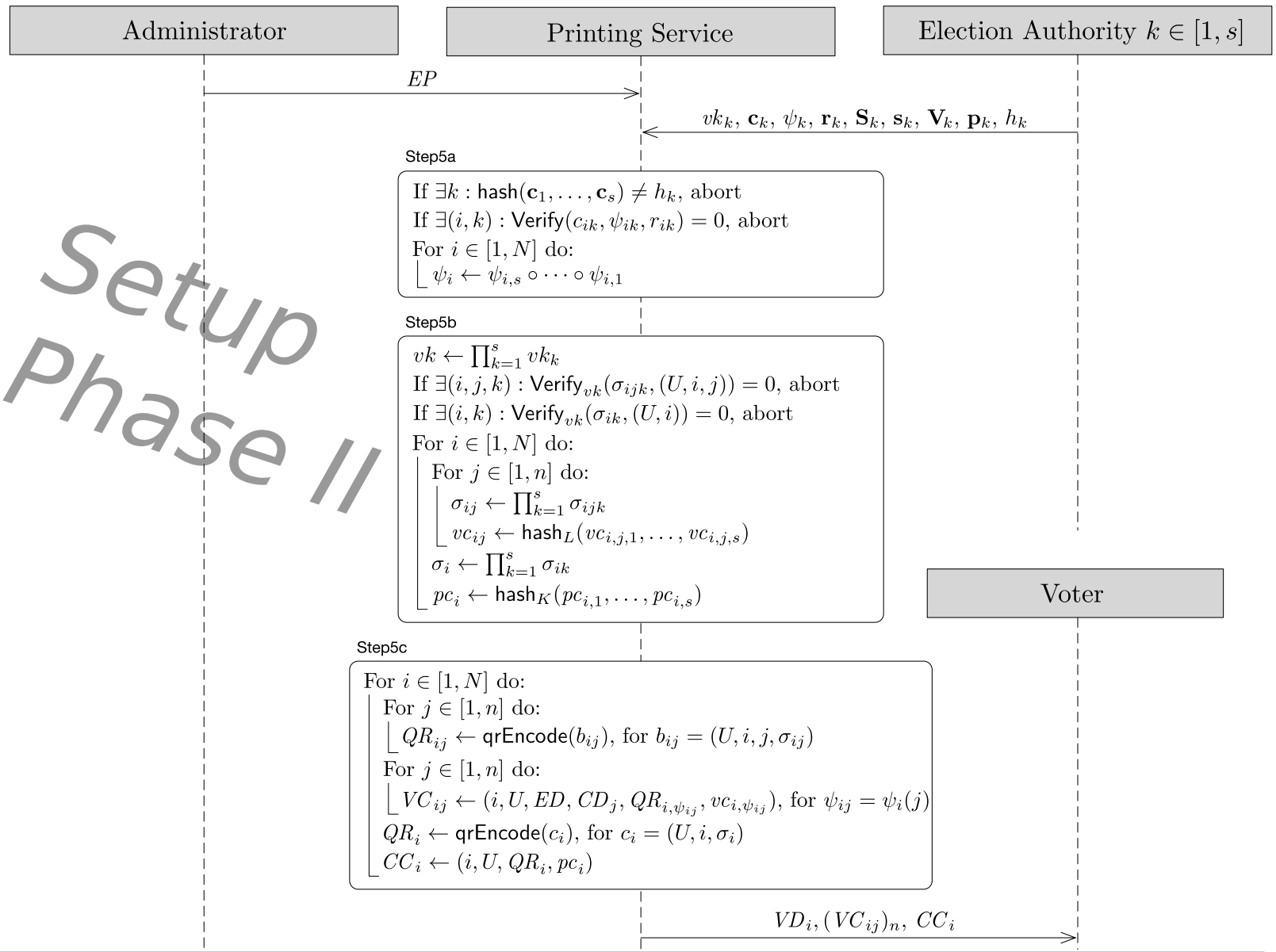
Step3c

If $\exists (i, \ell) : \mathsf{Verify}(\pi_{i\ell}^{\mathsf{shuffle}}, ek, \tilde{\mathbf{e}}_{i,\ell-1}, \tilde{\mathbf{e}}_{i,\ell}, c_{i\ell}) = 0$, abort
$\tilde{\mathbf{E}} \leftarrow (\tilde{\mathbf{e}}_{i,s})_N = (\tilde{e}_{ij})_{N \times n}$
$h_k \leftarrow \mathsf{hash}(\mathbf{c}_1, \ldots, \mathbf{c}_s)$

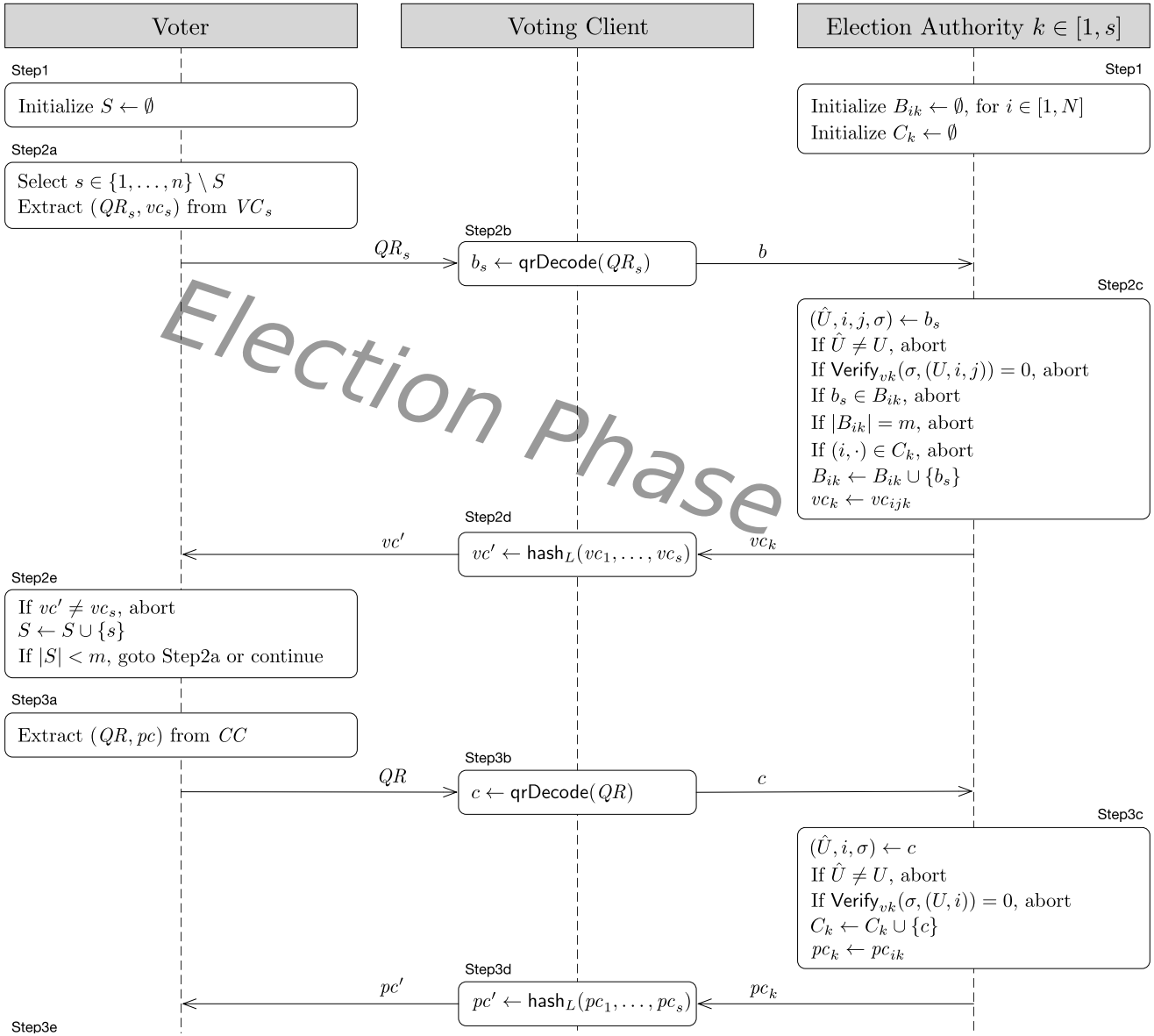
Step4

For $i \in [1, N]$ do:
  For $j \in [1, n]$ do:
    $\sigma_{ijk} \leftarrow \mathsf{Sign}_{sk_k}(U, i, j)$
    $vc_{ijk} \xleftarrow{\$} \{0,1\}^L$
  $\sigma_{ik} \leftarrow \mathsf{Sign}_{sk_k}(U, i)$
  $pc_{ik} \xleftarrow{\$} \{0,1\}^K$
$\mathbf{S}_k \leftarrow (\sigma_{ijk})_{N \times n}, \; \mathbf{s}_k = (\sigma_{ik})_N$
$\mathbf{V}_k \leftarrow (vc_{ijk})_{N \times n}, \; \mathbf{p}_k \leftarrow (pc_{ik})_N$

Setup Phase II




Peter Wähler
Haupstrasse 100
3000 Bern


Voter
Voting Client
Election Authority $k \in [1, s]$
Step1
Initialize $S \leftarrow \emptyset$
Step1
Initialize $B_{ik} \leftarrow \emptyset$, for $i \in [1, N]$
Initialize $C_k \leftarrow \emptyset$
Step2a
Select $s \in \{1, \ldots, n\} \setminus S$
Extract (QR_s, vc_s) from VC_s
QR_s
Step2b
$b_s \leftarrow \mathsf{qrDecode}(QR_s)$
b
Step2c
$(\hat{U}, i, j, \sigma) \leftarrow b_s$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i, j)) = 0$, abort
If $b_s \in B_{ik}$, abort
If $|B_{ik}| = m$, abort
If $(i, \cdot) \in C_k$, abort
$B_{ik} \leftarrow B_{ik} \cup \{b_s\}$
$vc_k \leftarrow vc_{ijk}$
Step2d
vc'
$vc' \leftarrow \mathsf{hash}_L(vc_1, \ldots, vc_s)$
vc_k
Step2e
If $vc' \neq vc_s$, abort
$S \leftarrow S \cup \{s\}$
If $|S| < m$, goto Step2a or continue
Step3a
Extract (QR, pc) from CC
QR
Step3b
$c \leftarrow \mathsf{qrDecode}(QR)$
c
Step3c
$(\hat{U}, i, \sigma) \leftarrow c$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i)) = 0$, abort
$C_k \leftarrow C_k \cup \{c\}$
$pc_k \leftarrow pc_{ik}$
Step3d
pc'
$pc' \leftarrow \mathsf{hash}_L(pc_1, \ldots, pc_s)$
pc_k
Step3e
Election Phase

Tally Phase

Administrator	Election Authority $k \in [1, s]$	Election Authority $\ell \neq k$

$(B_{ik})_N,\ C_k$ →

← $(B_{i\ell})_N,\ C_\ell$

Step1

For $i \in [1, N]$ do:
  $B_i \leftarrow \cup_{k=1}^{s} B_{ik}$
$C \leftarrow \cup_{k=1}^{s} C_k$

Step2

$E \leftarrow \emptyset$
For $c \in C$ do:
  $(\hat{U}, i, \sigma) \leftarrow c$
  If $(\hat{U}{=}U) \wedge \mathsf{Verify}_{vk}(\sigma, (U, i)) = 1$, do:
    $E_i \leftarrow \emptyset$
    For $b \in B_i$ do:
      $(\hat{U}, i, j, \sigma) \leftarrow b$
      If $(\hat{U}{=}U) \wedge \mathsf{Verify}_{vk}(\sigma, (U, i, j)) = 1$, do:
        Select $\tilde{e}_{ij}$ from $\tilde{\mathbf{E}}$
        $E_i \leftarrow E_i \cup \{\tilde{e}_{ij}\}$
    If $|E_i| \leq m$, do:
      $\tilde{e}_i \leftarrow \prod_{\tilde{e}_{ij} \in E_i} \tilde{e}_{ij}$
      $E \leftarrow E \cup \{\tilde{e}_i\}$
$\mathbf{e} \leftarrow \mathsf{sort}(E)$

Step3

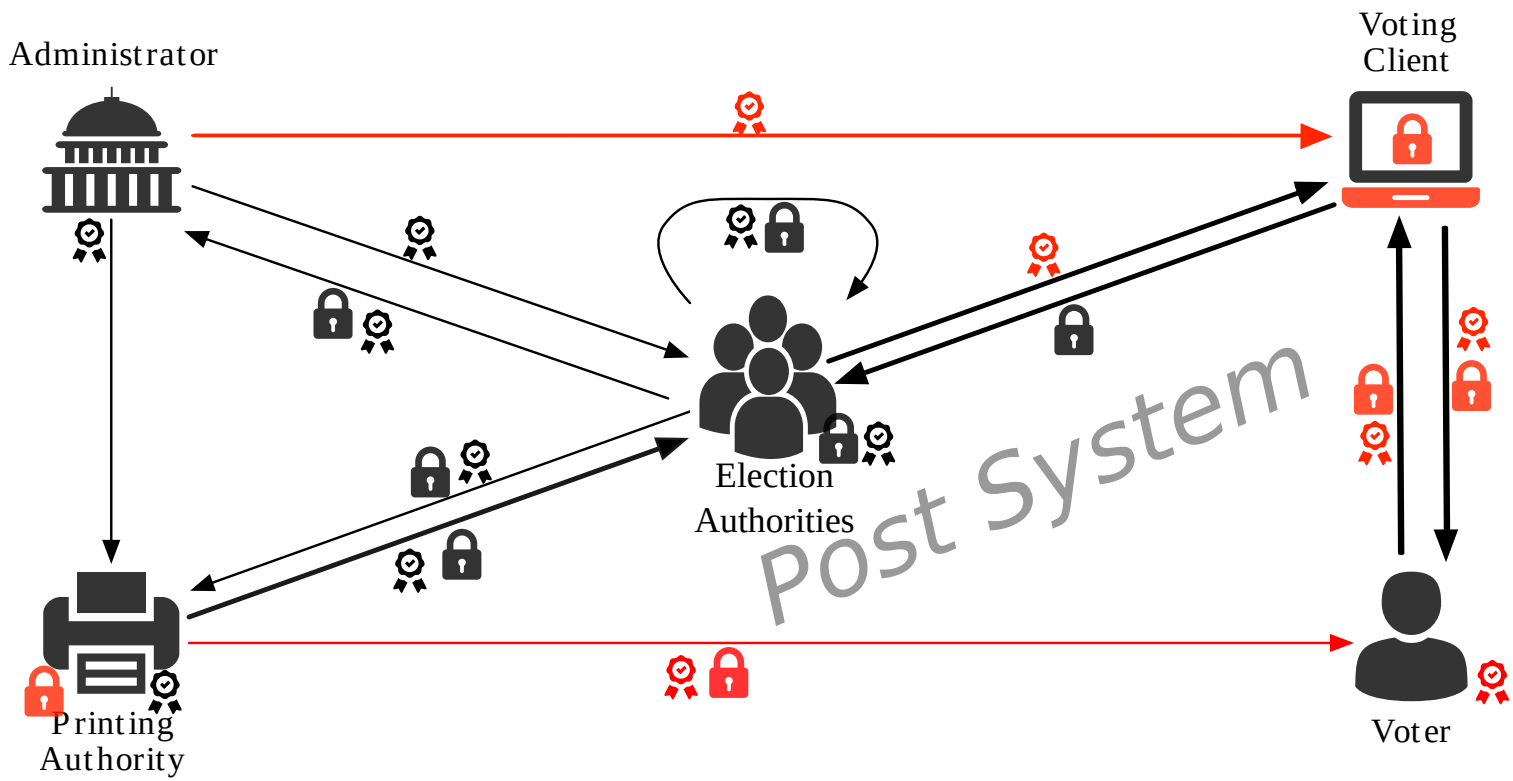
Mix and decrypt $\mathbf{e}$

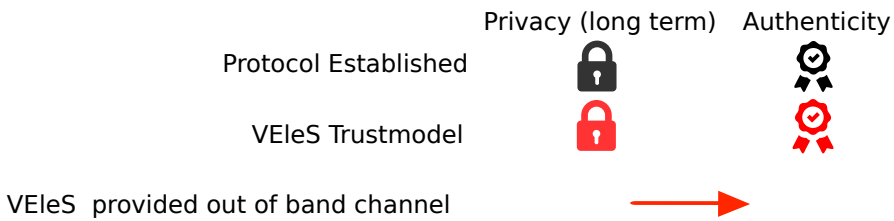
Partial decryptions and decryption proofs →

Step4

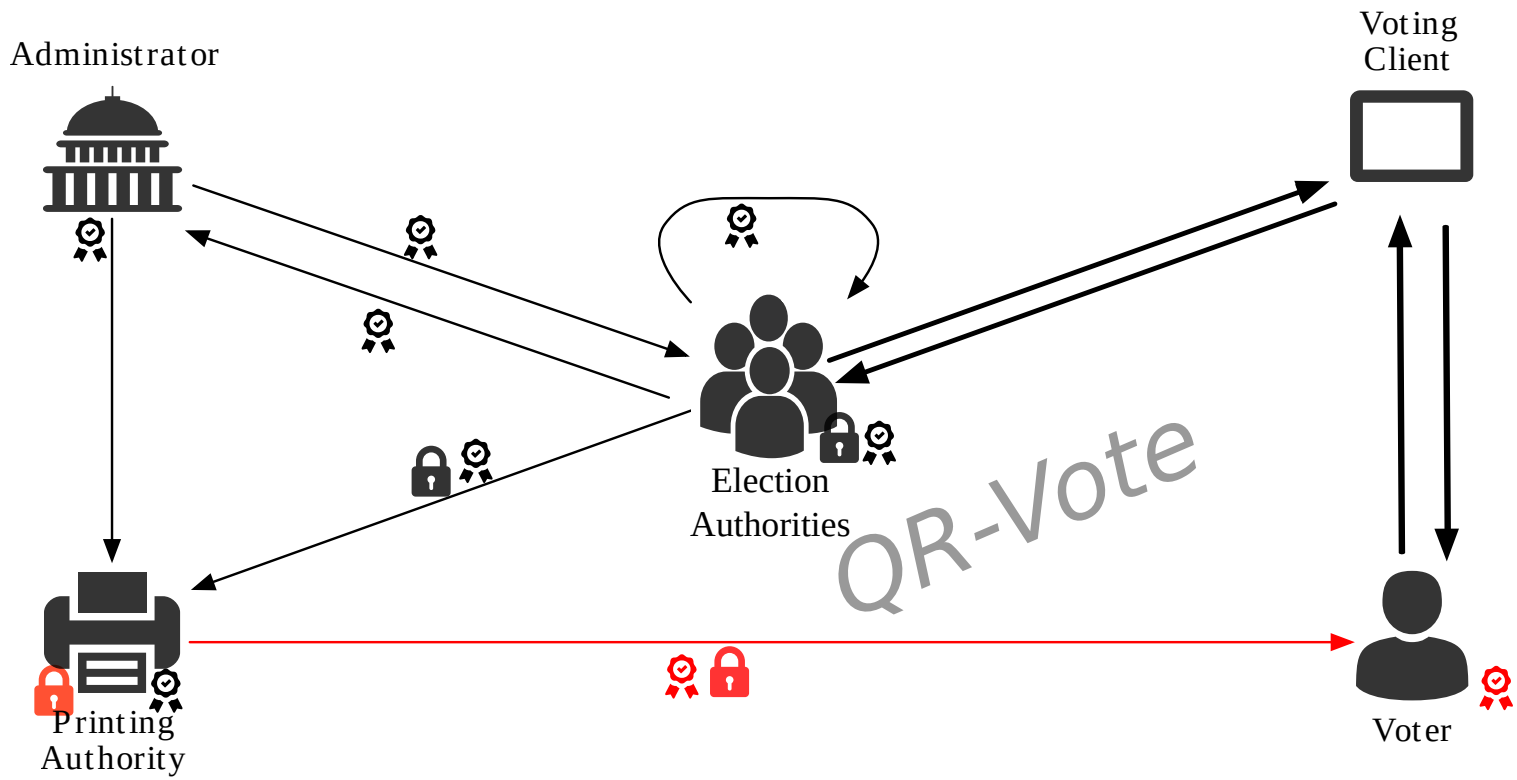
Verify proofs, decode and tally votes

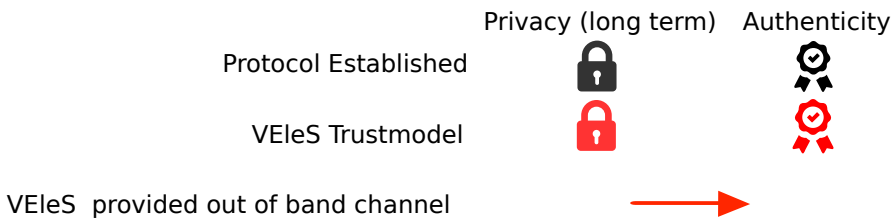

Peter Wähler
Haupstrasse 100
3000 Bern


Administrator
Voting Client
Election Authorities
Post System
Printing Authority
Voter


Privacy (long term)
Authenticity
Protocol Established
VEleS Trustmodel
VEleS provided out of band channel


Peter Wähler
Haupstrasse 100
3000 Bern


Administrator
Voting Client
Election Authorities
QR-Vote
Printing Authority
Voter


Privacy (long term)
Authenticity
Protocol Established
VEleS Trustmodel
VEleS provided out of band channel

Election Phase

Step1

Initialize $S \leftarrow \emptyset$

Step2a

Select $s \in \{1, \ldots, n\} \setminus S$
Extract (QR_s, vc_s) from VC_s

QR_s →

Step2b

$b_s \leftarrow \mathsf{qrDecode}(QR_s)$

→ b

Step1

Initialize $B_{ik} \leftarrow \emptyset$, for $i \in [1, N]$
Initialize $C_k \leftarrow \emptyset$

Step2c

$(\hat{U}, i, j, \sigma) \leftarrow b_s$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i, j)) = 0$, abort
If $b_s \in B_{ik}$, abort
If $|B_{ik}| = m$, abort
If $(i, \cdot) \in C_k$, abort
$B_{ik} \leftarrow B_{ik} \cup \{b_s\}$
$vc_k \leftarrow vc_{ijk}$

vc_k ←

Step2d

$vc' \leftarrow \mathsf{hash}_L(vc_1, \ldots, vc_s)$

← vc'

Step2e

If $vc' \neq vc_s$, abort
$S \leftarrow S \cup \{s\}$
If $|S| < m$, goto Step2a or continue

Step3a

Extract (QR, pc) from CC

QR →

Step3b

$c \leftarrow \mathsf{qrDecode}(QR)$

→ c

Step3c

$(\hat{U}, i, \sigma) \leftarrow c$
If $\hat{U} \neq U$, abort
If $\mathsf{Verify}_{vk}(\sigma, (U, i)) = 0$, abort
$C_k \leftarrow C_k \cup \{c\}$
$pc_k \leftarrow pc_{ik}$

pc_k ←

Step3d

$pc' \leftarrow \mathsf{hash}_L(pc_1, \ldots, pc_s)$

← pc'

Step3e

If $pc' \neq pc$, goto Step3a or complain

Step1

Initialize $S \leftarrow \emptyset$

Step2a

Select $s \in \{1, \ldots,$
Extract $(QR_s, vc_s$

Step2e

If $vc' \neq vc_s$, abor
$S \leftarrow S \cup \{s\}$
If $|S| < m$, goto S

Step3a

Extract (QR, pc)

Step3e

If $pc' \neq pc$, goto

Step1

Initialize $S \leftarrow \emptyset$

Step2a

Select $s \in \{1, \ldots,$
Extract $(QR_s, vc_$

Step2e

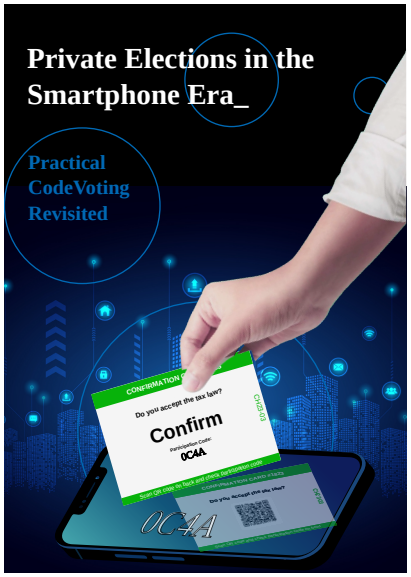
If $vc' \neq vc_s$, abor
$S \leftarrow S \cup \{s\}$
If $|S| < m$, goto S

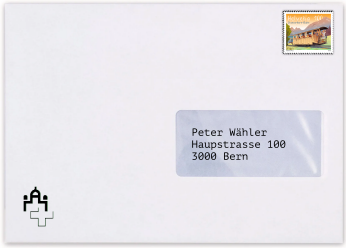
Step3a

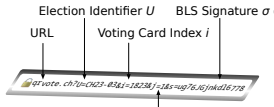
Extract (QR, pc)

Step3e

If $pc' \neq pc$, goto







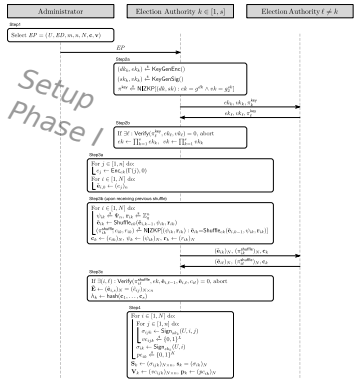


Fig. 2: First part of the pre-election phase: Election setup (Step1), key generation (Step2), ciphertext preparation (Step3), signature and code generation (Step4).

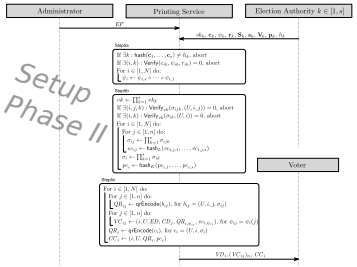


Fig. 3: Second part of the pre-election phase: Permutation composition (Step5a), signature and code aggregation (Step5b), voting card preparation (Step5c).

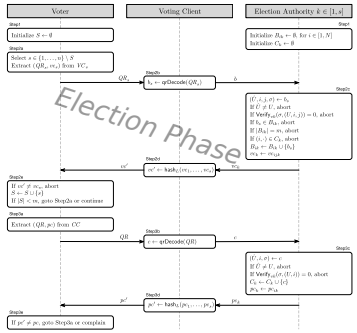


Fig. 4: Overview of the election phase: Initialization (Step1), vote casting (Step2), vote confirmation (Step3).

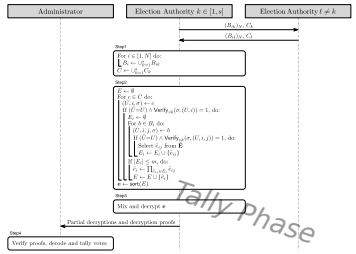


Fig. 5: Overview of the post-election phase: ballot and confirmation box synchronization (Step1), extraction of valid ciphertexts (Step2), mixing and decryption (Step3), decoding and tallying (Step4).