

Formalising Receipt-freeness

(joint work with Erik de Vink)

Hugo Jonker

hjonker@win.tue.nl

TU/e overview

Introduction
 overview
Real world voting
E-voting
Receipts
Formalisation

More concretely

Application

Final Thoughts

- How does voting work in the "real world"?
- Why vote digital?
- What is this "receipt-freeness" anyway?
- What is this "receipt-freeness" anyway in a more formal sense?
- Aha! But how would you use this?

TU/e typical elections

Introduction
Real world voting
● typical elections
preventing cheating
E-voting
Receipts
Formalisation
More concretely
Application

Final Thoughts

The type of elections we consider (1V1V):

- Various candidates
- Each voter may cast one vote
- All votes carry equal weight
- The result can be seen as the collection (multiset) of cast votes (ballots)

E.g. national elections in the Netherlands.

TU/e preventing cheating

Introduction

Real world voting

• typical elections

• preventing cheating

E-voting

Receipts

Formalisation

More concretely

Application

Final Thoughts

Cheating in elections is prevented by law, procedures and regulations, e.g.:

At all times during the elections, the chairman and two members of the voting bureau are present *Kieswet, Artikel J lid 12 sub 1*

This provides (some) protection against incorrect voting, multiple voting, incorrect counting, etc. etc.

Introduction

Real world voting

E-voting

●pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Advantages:

Disadvantage:

Introduction

Real world voting

E-voting

● pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Advantages:

• Greater convenience for voter ($\stackrel{?}{\Longrightarrow}$ greater voter turnout)

Less overhead to set up elections

Disadvantage:

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Advantages:

• Greater convenience for voter ($\stackrel{?}{\Longrightarrow}$ greater voter turnout)

Less overhead to set up elections

Disadvantage: Re-invent the wheel:

How to do elections in a digital environment?

What attacks are possible?

How to prevent those attacks?

Which means:

Introduction

Real world voting

E-voting

●pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Advantages:

• Greater convenience for voter ($\stackrel{?}{\Longrightarrow}$ greater voter turnout)

Less overhead to set up elections

Disadvantage: Re-invent the wheel:

How to do elections in a digital environment?

What attacks are possible?

How to prevent those attacks?

Which means:

Danger of introducing new flaws

Introduction

Real world voting

E-voting

•	pro's	&	con's
	nrone	ort	ioc

R	ece	eip	ts

Formalisation

More concretely

Application

Final Thoughts

Advantages:

• Greater convenience for voter ($\stackrel{?}{\Longrightarrow}$ greater voter turnout)

Less overhead to set up elections

Disadvantage: Re-invent the wheel:

How to do elections in a digital environment?

What attacks are possible?

How to prevent those attacks?

Which means:

- Danger of introducing new flaws
- Risc of forgetting about known flaws

Introduction

Real world voting
E-voting
• pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Introduction

Real world voting

E-voting

• pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Several properties have been established for e-voting protocols, such as:

Democracy

Introduction

Real world voting

E-voting

• pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Several properties have been established for e-voting protocols, such as:

Democracy

Eligibility

Introduction

Real world voting

E-voting

• pro's & con's

properties

Receipts

Formalisation

More concretely

Application

Final Thoughts

- Democracy
- Eligibility
- Accuracy

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

- Democracy
- Eligibility
- Accuracy
- Verifiability
 - Individual
 - Universal

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

- Democracy
- Eligibility
- Accuracy
- Verifiability
 - Individual
 - Universal
- Privacy

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

- Democracy
- Eligibility
- Accuracy
- Verifiability
 - Individual
 - Universal
- Privacy
- Fairness

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

Several properties have been established for e-voting protocols, such as:

- Democracy
- Eligibility
- Accuracy
- Verifiability
 - Individual
 - Universal
- Privacy
- Fairness

— ...

Introduction

Real world voting

E-voting

pro's & con'sproperties

Receipts

Formalisation

More concretely

Application

Final Thoughts

- Democracy
- Eligibility
- Accuracy
- Verifiability
 - Individual
 - Universal
- Privacy
- Fairness
- **•** ...
- Receipt-freeness(!)

TU/e intuition

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

A receipt is an object which enables a voter to prove how she voted.

TU/e intuition

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

A receipt is an object which enables a voter to prove how she voted.

Examples:

Everyone signs their vote.

TU/e intuition

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

example: FOO

Formalisation

More concretely

Application

Final Thoughts

A receipt is an object which enables a voter to prove how she voted.

Examples:

Everyone signs their vote.

In Italy, simultaneous elections were held for various posts, using one ballot. The order of posts listed is up to the voter, and is preserved. An attacker (El Mafiosi) can assign each voter a specific order of posts. *Benaloh & Tuinstra*

n	tr	0	Ы		0	ti	0	n
	u	υ	u	u	C	u	υ	

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

More precisely: a receipt r proves that a voter v cast a vote for candidate c.

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

More precisely: a receipt r proves that a voter v cast a vote for candidate c.

R1: r authenticates v

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

More precisely: a receipt r proves that a voter v cast a vote for candidate c.

- **R1:** r authenticates v
- **R2:** r proves that v chose candidate c

Intr	odu	ctio	n

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

More precisely: a receipt r proves that a voter v cast a vote for candidate c.

- **R1:** r authenticates v
- **R2:** r proves that v chose candidate c
- **R3:** r proves that v cast her vote

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

More precisely: a receipt r proves that a voter v cast a vote for candidate c.

- **R1:** r authenticates v
- **R2:** r proves that v chose candidate c
- **R3:** r proves that v cast her vote

Note:

- Specific for 1V1V elections
- Quite strict

Int	rod	uct	ion
		aor	

Real world voting

E-voting

Receipts

intuition

requirements

● example: FOO

Formalisation

More concretely

Application

Final Thoughts

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

Rough sketch of the FOO protocol for voter v, admin a and counter cnt:

1. v: create a blinded, encrypted vote

Introduction

Real world voting

E-voting

Receipts

intuition

requirements

• example: FOO

Formalisation

More concretely

Application

Final Thoughts

- 1. v: create a blinded, encrypted vote
- 2. $v \rightarrow a$: blinded, encrypted vote signed by v

Introduction Ref Real world voting CC E-voting 1. Receipts 2. • intuition 2. • example: FOO 3. Formalisation More concretely Application CC

Final Thoughts

- 1. v: create a blinded, encrypted vote
- 2. $v \rightarrow a$: blinded, encrypted vote signed by v
- 3. $a \rightarrow v$: blinded, encrypted vote signed by a

Introduction Real world voting E-voting Receipts intuition requirements example: FOO Formalisation More concretely

Application

Final Thoughts

- 1. v: create a blinded, encrypted vote
- 2. $v \rightarrow a$: blinded, encrypted vote signed by v
- 3. $a \rightarrow v$: blinded, encrypted vote signed by a
- 4. $v \rightarrow cnt$: encrypted vote signed by a

ntroduction	
Real world voting	
E-voting	
Receipts	
● intuition	
requirements	
example: FOO	
Formalisation	
More concretely	

Application

Final Thoughts

- 1. v: create a blinded, encrypted vote
- 2. $v \rightarrow a$: blinded, encrypted vote signed by v
- 3. $a \rightarrow v$: blinded, encrypted vote signed by a
- 4. $v \rightarrow cnt$: encrypted vote signed by a
- 5. cnt: collect all votes

TU example: FOO **'e**

Introduction	Rough sk
Real world voting	counter c
E-voting	
Receipts	1. v: crea
● intuition	2 $v \rightarrow a$
requirements	
● example: FOO	3. $a \rightarrow v$:
Formalisation	_
More concretely	4. $v \rightarrow cr$
Application	5. <i>cnt</i> : co
Final Thoughts	6. <i>cnt</i> : pi

ketch of the FOO protocol for voter v, admin a and ent:

- ate a blinded, encrypted vote
- blinded, encrypted vote signed by v
- blinded, encrypted vote signed by a
- nt: encrypted vote signed by a
- ollect all votes
- ublish list of received votes

example: FOO TU ρ

Introduction	Rough
Real world voting	counter
E-voting	
Receipts	1. <i>v</i> : cr
• intuition	$2 a \rightarrow$
requirements	
● example: FOO	3. $a \rightarrow$
Formalisation	
More concretely	4. $v \rightarrow$
Application	5. <i>cnt</i> :
Final Thoughts	6. <i>cnt</i> :
	7

- eate a blinded, encrypted vote
- a: blinded, encrypted vote signed by v
- v: blinded, encrypted vote signed by a
- cnt: encrypted vote signed by a
- collect all votes
- publish list of received votes
- 7. $v \rightarrow cnt$: decryption key, index of vote in list

example: FOO ΓU ρ

Introduction	Roughs
Real world voting	counter a
E-voting	
Receipts	1. v: cre
● intuition	$2 \rightarrow a$
requirements	\mathbf{Z} . $v \rightarrow u$
● example: FOO	3. $a \rightarrow v$
Formalisation	1
More concretely	$4. \ \ U \rightarrow C$
Application	5. <i>cnt</i> : c
Final Thoughts	6. <i>cnt</i> : p
	7 $v \rightarrow c$

- ate a blinded, encrypted vote
- : blinded, encrypted vote signed by v
- : blinded, encrypted vote signed by a
- *ent*: encrypted vote signed by *a*
- collect all votes
- ublish list of received votes
- *cnt*: decryption key, index of vote in list
- 8. *cnt*: publish list of received keys

example: FOO ρ

Introduction	Rough
Real world voting	counter
E-voting	
Receipts	1. <i>v</i> : cre
• intuition	$2 \rightarrow 2$
requirements	\mathbf{Z} . $U \rightarrow 0$
● example: FOO	3. $a \rightarrow a$
Formalisation	_
More concretely	4. $v \rightarrow c$
Application	5. <i>cnt</i> : 0
Final Thoughts	6. <i>cnt</i> : p
	7. $v \rightarrow c$

sketch of the FOO protocol for voter v, admin a and cnt:

- eate a blinded, encrypted vote
- a: blinded, encrypted vote signed by v
- v: blinded, encrypted vote signed by a
- *cnt*: encrypted vote signed by *a*
- collect all votes
- oublish list of received votes
- *cnt*: decryption key, index of vote in list
- 8. *cnt*: publish list of received keys

Obvious receipt... but it seems to lose its validity
TU/e example: FOO

ction	Rough s	
orld voting	counter	
<u>g</u>	4	
ts	1. <i>v</i> : cre	
ion	$2 \rightarrow c$	
irements	\mathbf{Z} . $U \rightarrow u$	
nple: FOO	3. $a \rightarrow i$	
isation	•••••	
oncretely	4. $v \rightarrow c$	
ation	5. <i>cnt</i> : c	
houghts	6. <i>cnt</i> : p	
	7. $v \rightarrow c$	

Introdu

Real w

E-votir

Receip

• requ

Forma

More c

Applica

Final T

Rough sketch of the FOO protocol for voter v, admin a and counter cnt:

- 1. v: create a blinded, encrypted vote
- 2. $v \rightarrow a$: blinded, encrypted vote signed by v
- **B.** $a \rightarrow v$: blinded, encrypted vote signed by a
- I. $v \rightarrow cnt$: encrypted vote signed by a
- 5. *cnt*: collect all votes
- 6. cnt: publish list of received votes
- 7. $v \rightarrow cnt$: decryption key, index of vote in list
- 8. *cnt*: publish list of received keys

Obvious receipt... but it seems to lose its validity Timestamping \implies no it doesn't!

TU/e ingredients

ntroduction
Real world voting
E-voting
Receipts
Formalisation

ingredients

decomposing receipts

More concretely

Application

Final Thoughts

- voters $v \in \mathcal{V}$, choices $c \in \mathcal{C}$
- ballots \mathcal{B} and results (multisets of choices) $\mathcal{M}(\mathcal{C})$
- a set of received ballots RB, from which the result will be computed
- a choice function $\Gamma: \mathcal{V} \to \mathcal{C}$, which specifies how the voters vote

TU/e ingredients

	voter
ntroduction	
Real world voting	■ ballo
E-voting	■ a set
Receipts	comp
Formalisation	∎ a cho
ingredients	
decomposing receipts	vote
More concretely	
Application	To den
Final Thoughts	■ the s
	■ Term
	auther
	$t\in \mathcal{A}$

- voters $v \in \mathcal{V}$, choices $c \in \mathcal{C}$
- ballots \mathcal{B} and results (multisets of choices) $\mathcal{M}(\mathcal{C})$
- a set of received ballots RB, from which the result will be computed
- a choice function $\Gamma: \mathcal{V} \to \mathcal{C}$, which specifies how the voters vote

To denote receipts, the following syntax is used: ■ the set of receipts *R*

- Terms(v), the set of all terms that a voter $v \in \mathcal{V}$ can generate
- authentication terms $\mathcal{AT}(v)$: $t \in \mathcal{AT}(v) \implies \forall w \neq v : t \notin Terms(w)$
- $auth: \mathcal{AT} \to \mathcal{V}$, the unique voter that created an AT

TU/e decomposing receipts

	The
Introduction	
Real world voting	
	$\square \alpha$
E-voting	
Receipts	■ β
Formalisation	$\square \gamma$
ingredients	/
decomposing receipts	
More concretely	For
Application	
Final Thoughts	

The following functions are used to decompose receipts:

• $\alpha : \mathcal{R} \to \mathcal{AT}$, extract authentication term from receipt • $\beta : \mathcal{R} \to \mathcal{RB}$, extract ballot from receipt • $\gamma : \mathcal{R} \to \mathcal{C}$, extract candidate from receipt

Formalisation of the requirements:

TU/e decomposing receipts

	The
ntroduction	
Real world voting	
-voting	$\blacksquare \alpha$
_ voung	
Receipts	$\square p$
Formalisation	$\square \gamma$
• ingredients	/
decomposing receipts	
More concretely	Forr
Application	
Final Thoughts	■ R

The following functions are used to decompose receipts:

 $\alpha \colon \mathcal{R} \to \mathcal{AT}$, extract authentication term from receipt $\beta \colon \mathcal{R} \to \mathcal{RB}$, extract ballot from receipt $\gamma \colon \mathcal{R} \to \mathcal{C}$, extract candidate from receipt

Formalisation of the requirements:

```
R1: \alpha(r) \in \mathcal{AT}(v)
```

TU/e decomposing receipts

	The
Introduction	
Real world voting	
E-voting	$\blacksquare \alpha$
Receipts	■ β
Formalisation	$\blacksquare \gamma$
ingredients	/
decomposing receipts	
More concretely	Forr
Application	
Final Thoughts	■ R
	- D
	■ R.

The following functions are used to decompose receipts:

• $\alpha : \mathcal{R} \to \mathcal{AT}$, extract authentication term from receipt • $\beta : \mathcal{R} \to \mathcal{RB}$, extract ballot from receipt • $\gamma : \mathcal{R} \to \mathcal{C}$, extract candidate from receipt

Formalisation of the requirements:

```
R1: \alpha(r) \in \mathcal{AT}(v)
R2: \gamma(r) = \Gamma(v)
```

decomposing receipts ρ

	The following function
Introduction	The following function
Real world voting	
E-voting	• $\alpha \colon \mathcal{R} \to \mathcal{AT}$, extra
Receipts	• $\beta \colon \mathcal{R} \to \mathcal{RB}$, extra
Formalisation	• $\gamma: \mathcal{R} \to \mathcal{C}$, extract
● ingredients	, , , , , , , , , , , , , , , , , , , ,
 decomposing receipts 	
More concretely	Formalisation of the
Application	
Final Thoughts	R1: $\alpha(r) \in \mathcal{AT}(v)$
	R2: $\gamma(r) = \Gamma(v)$
	R3: $\beta(r) \in \mathcal{RB}$

Þ

following functions are used to decompose receipts:

- : $\mathcal{R} \rightarrow \mathcal{AT}$, extract authentication term from receipt : $\mathcal{R} \to \mathcal{RB}$, extract ballot from receipt : $\mathcal{R} \rightarrow \mathcal{C}$, extract candidate from receipt
- malisation of the requirements:

decomposing receipts ρ

	Tł
Introduction	• •
Real world voting	
E-voting	
Receipts	
Formalisation	
● ingredients	
decomposing receipts	
More concretely	Fo
More concretely Application	Fo
More concretely Application Final Thoughts	Fα
More concretely Application Final Thoughts	Fo
More concretely Application Final Thoughts	F

Þ

he following functions are used to decompose receipts:

 $\alpha \colon \mathcal{R} \to \mathcal{AT}$, extract authentication term from receipt $\beta \colon \mathcal{R} \to \mathcal{RB}$, extract ballot from receipt $\gamma \colon \mathcal{R} \to \mathcal{C}$, extract candidate from receipt

ormalisation of the requirements:

```
R1: \alpha(r) \in \mathcal{AT}(v)
R2: \gamma(r) = \Gamma(v)
```

```
R3: \beta(r) \in \mathcal{RB}
```

So, for valid receipts: $auth(\alpha(r)) = v \implies \gamma(r) = \Gamma(v)$, which is satisfied by $\gamma = \Gamma \circ auth \circ \alpha$.

Introduction
Real world voting
E-voting
Receipts
Formalisation
More concretely
• receipts as terms
• receipts as terms II

Application

Final Thoughts

Intuitively, a receipt must be derivable from an actual execution of a voting protocol (i.e. receipts generated outside a protocol do not invalidate that protocol).

To facilitate detection of receipts, limit the notion of receipts to terms (i.e. $\mathcal{R} = \emptyset \lor \mathcal{R} \subseteq Terms$).

Now:

Model the protocol in ACP

Test suitability of communicated terms as receipts

Pronounce judgment

Introduction
Real world voting
E-voting
Receipts
Formalisation
More concretely
• receipts as terms
• receipts as terms II

Application

Final Thoughts

Intuitively, a receipt must be derivable from an actual execution of a voting protocol (i.e. receipts generated outside a protocol do not invalidate that protocol).

To facilitate detection of receipts, limit the notion of receipts to terms (i.e. $\mathcal{R} = \emptyset \lor \mathcal{R} \subseteq Terms$).

Now:

Model the protocol in ACP (+ tweaks)

- Test suitability of communicated terms as receipts
- Pronounce judgment

Inti	od.	I Cti	on
11111	out	JUU	

Real world voting

E-voting

Receipts

Formalisation

More concretely

receipts as terms

• receipts as terms II

Application

Final Thoughts

Write $t \in t'$ if t is a subterm of t'.

 α, β extract terms from terms, i.e. they deal with subterms.

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

receipts as terms

receipts as terms II

Application

Final Thoughts

Write $t \in t'$ if t is a subterm of t'.

 α,β extract terms from terms, i.e. they deal with subterms.

Lemma $\forall t \in \mathcal{R} \colon \alpha(t) \in t \land \beta(t) \in t$

Introduction
Real world voting
E-voting
Receipts

Formalisation

More concretely

receipts as terms
 receipts as terms II

Application

Final Thoughts

Write $t \in t'$ if t is a subterm of t'.

 α, β extract terms from terms, i.e. they deal with subterms. Lemma $\forall t \in \mathcal{R} \colon \alpha(t) \in t \land \beta(t) \in t$

(Note that, by definition: $t \in t' \land t \in \mathcal{AT}(v) \implies t' \in \mathcal{AT}(v)$. So receipts are themselves authentication terms)

Introduction
Real world voting
E-voting
Receipts
Formalisation

More concretely

receipts as terms
 receipts as terms II

Application

Final Thoughts

Write $t \in t'$ if t is a subterm of t'.

 α, β extract terms from terms, i.e. they deal with subterms. Lemma $\forall t \in \mathcal{R} \colon \alpha(t) \in t \land \beta(t) \in t$

(Note that, by definition: $t \in t' \land t \in \mathcal{AT}(v) \implies t' \in \mathcal{AT}(v)$. So receipts are themselves authentication terms)

Although this does not capture the entire notion of receipts, it turns out to be strong enough in the examined cases.

TU/e in advance

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

● in advance

• BT

BT: receipt-free

• RIES

receipts in RIES

Final Thoughts

Formalisation not yet complete (for terms)

■ Goal in this talk is a high-level analysis using the formalism

TU/e BT

Introduction	
Real world voting	
F-voting	
Receipts	
Formalisation	
More concretely	6
	r
Application	
● in advance	
●BT	
● BT: receipt-free	
• RIES	
• receipts in RIES	
Final Thoughts	

Original receipt-freeness paper by Benaloh & Tuinstra

- Attack found... but not on the main scheme
- Assumes untappable channels and a voting booth
- Uses randomised encryption and "ZKP"

Process for voting authority:

Process for a voter:

TU/e BT

Formalisation

More concretely

Application

in advanceBT

• BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

Original receipt-freeness paper by Benaloh & Tuinstra

Attack found... but not on the main scheme
Assumes untappable channels and a voting booth
Uses randomised encryption and "ZKP"

Process for voting authority:

$$A(v) = \sum_{x \in E(0), y \in E(1)} s_{a \to v}(\min(x, y), \max(x, y)) \cdot p_{a \to v}^*(x \in E(0) \land y \in E(1)) \cdot (r_{v \to a}(x) + r_{v \to a}(y))$$

Process for a voter:

TU/e BT

Introduction
Real world voting
E-voting
Receipts

Formalisation

More concretely

Application

in advanceBT

• BT: receipt-free

• RIES

receipts in RIES

Final Thoughts

Original receipt-freeness paper by Benaloh & Tuinstra

Attack found... but not on the main scheme
Assumes untappable channels and a voting booth
Uses randomised encryption and "ZKP"

Process for voting authority:

$$A(v) = \sum_{x \in E(0), y \in E(1)} s_{a \to v}(\min(x, y), \max(x, y)) \cdot p_{a \to v}^*(x \in E(0) \land y \in E(1)) \cdot (r_{v \to a}(x) + r_{v \to a}(y))$$

Process for a voter:

$$V = \sum_{x,y} r_{a \to v}(x,y) \cdot \sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)) \cdot (\Gamma(v) = i \to s_{v \to a}(x) + \Gamma(v) = 1 - i \to s_{v \to a}(y))$$

Introduction	
Real world voting	
Evoting	
E-voting	

Receipts

Formalisation

More concretely

Application

● in advance

●BT

● BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

Let's examine the voter process:

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \to v}(x,y)$

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

in advance

●BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

• in advance

• BT

● BT: receipt-free

● RIES

• receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \to v}(x,y) \cdot$ Not an authentication term

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

• in advance

• BT

● BT: receipt-free

● RIES

• receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \rightarrow v}(x,y) \cdot$ Not an authentication term

$$\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)).$$

BT: receipt-free TU/e

Real world voting

Introduction

E-voting

Receipts

Formalisation

More concretely

Application

• in advance

• BT

• BT: receipt-free

RIES

receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \to v}(x,y) \cdot$

Not an authentication term

 $\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)).$

No ballot as a subterm

Let's examine the voter process:

Real world voting

Introduction

E-voting

Receipts

Formalisation

More concretely

Application

in advance

• BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

 $V = \sum_{x,y} r_{a \rightarrow v}(x,y) \cdot$ Not an authentication term

$$\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)) \cdot$$

No ballot as a subterm

$$(\Gamma(v) = i \to s_{v \to a}(x) + \Gamma(v) = 1 - i \to s_{v \to a}(y))$$

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

In advance

• BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \rightarrow v}(x,y) \cdot$ Not an authentication term

$$\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)) \cdot$$

No ballot as a subterm

$$\begin{pmatrix} \Gamma(v) = i \to s_{v \to a}(x) + \Gamma(v) = 1 - i \to s_{v \to a}(y) \end{pmatrix}$$

Subterm of first term!

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

in advance

• BT

● BT: receipt-free

• RIES

receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \to v}(x,y) \cdot$ Not an authentication term

> $\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)) \cdot$ No ballot as a subterm

$$\begin{pmatrix} \Gamma(v) = i \to s_{v \to a}(x) + \Gamma(v) = 1 - i \to s_{v \to a}(y) \end{pmatrix}$$

Subterm of first term!

None of the terms from the voter can satisfy $\alpha(t) \in t \land \beta(t) \in t$

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

in advance

• BT

● BT: receipt-free

• RIES

receipts in RIES

Final Thoughts

Let's examine the voter process:

 $V = \sum_{x,y} r_{a \to v}(x,y) \cdot$ Not an authentication term

> $\sum_{i \in \{0,1\}} p_{a \to v}^* (x \in E(i) \land y \in E(1-i)) \cdot$ No ballot as a subterm

$$\begin{pmatrix} \Gamma(v) = i \to s_{v \to a}(x) + \Gamma(v) = 1 - i \to s_{v \to a}(y) \end{pmatrix}$$

Subterm of first term!

None of the terms from the voter can satisfy $\alpha(t) \in t \land \beta(t) \in t$ \implies BT is receipt-free!

Introduction	
Real world voting	
E-voting	
Receints	

Formalisation

More concretely

Application

in advance

• BT

• BT: receipt-free

●RIES

receipts in RIES

Final Thoughts

- Used in Dutch water management board elections
- Handled over 70,000 votes
- Uses a publicly-known hash-function and voter-specific keys
- Obvious receipt

How it works:

Introduction	
Real world voting	
E-voting	
Receipts	
Formalisation	
More concretely	
Application	
● in advance	

●BT

- BT: receipt-free
- ●RIES
- receipts in RIES

Final Thoughts

- Used in Dutch water management board elections
- Handled over 70,000 votes
- Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

1. $a \rightarrow v$: key(v)

Introduction
Real world voting
E-voting
Receipts
Formalisation
More concretely
Application
● in advance
●BT

BT: receipt-free

●RIES

• receipts in RIES

Final Thoughts

- Used in Dutch water management board elections
- Handled over 70,000 votes
- Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

- **1.** $a \rightarrow v$: key(v)
- 2. *a*: publish list of all possible encrypted votes, hashed:

$$\mathcal{L} = \bigcup_{v \in \mathcal{V}} \{ < h(\{c\}_{key(v)}), c > \mid c \in \mathcal{C} \}$$

ntroduction	
	_
Real world voting	
E-voting	
Receipts	
	_
Formalisation	
More concretely	H
	1 1
Application	1
in advance	
BT	2
BT: receipt-free	Ζ.
RIES	
Preceipts in RIES	

Final Thoughts

- Used in Dutch water management board elections
- Handled over 70,000 votes
- Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

- $I. a \to v: key(v)$
- 2. *a*: publish list of all possible encrypted votes, hashed:

$$\mathcal{L} = \bigcup_{v \in \mathcal{V}} \{ < h(\{c\}_{key(v)}), c > \mid c \in \mathcal{C} \}$$

3. $p_{v \to a}$: $\{\Gamma(v)\}_{key(v)}$

Itroduction	
eal world voting	
-voting	■ U:
eceipts	
ormalisation	= 0
lore concretely	How
pplication	1 <i>a</i>
in advance	1. 0
BT	つ
BT: receipt-free	Z. a
RIES	(
receipts in RIES	2
inal Thoughts	3. p
	-

- Used in Dutch water management board elections
- Handled over 70,000 votes
 - Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

- I. $a \rightarrow v$: key(v)
- 2. *a*: publish list of all possible encrypted votes, hashed:

$$\mathcal{L} = \bigcup_{v \in \mathcal{V}} \{ < h(\{c\}_{key(v)}), c > \mid c \in \mathcal{C} \}$$

3.
$$p_{v \to a}$$
: $\{\Gamma(v)\}_{key(v)}$

4. *a*: collect all votes

ntroduction	
Real world voting	■⊢
E-voting	∎ L
Receipts	
Formalisation	- (
More concretely	Hov
	1101
Application	1 (
in advance	1. (
BT	2
BT: receipt-free	Ζ. (
RIES	
ereceipts in RIES	7
Final Thoughts	3 . <i>j</i>
	Λ

- Used in Dutch water management board elections
- Handled over 70,000 votes
 - Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

- 1. $a \rightarrow v$: key(v)
- 2. *a*: publish list of all possible encrypted votes, hashed:

$$\mathcal{L} = \bigcup_{v \in \mathcal{V}} \{ < h(\{c\}_{key(v)}), c > \mid c \in \mathcal{C} \}$$

3. $p_{v \to a}$: $\{\Gamma(v)\}_{key(v)}$

- 4. *a*: collect all votes
- 5. *a*: publish outcome

Introduction Real world voting E-voting Receipts Formalisation More concretely Application in advance BT BT: receipt-free ● RIES receipts in RIES **Final Thoughts**

- Used in Dutch water management board elections
- Handled over 70,000 votes
- Uses a publicly-known hash-function and voter-specific keys
 Obvious receipt

How it works:

- **1.** $a \rightarrow v$: key(v)
- 2. *a*: publish list of all possible encrypted votes, hashed:

$$\mathcal{L} = \bigcup_{v \in \mathcal{V}} \{ < h(\{c\}_{key(v)}), c > \mid c \in \mathcal{C} \}$$

3. $p_{v \to a}$: $\{\Gamma(v)\}_{key(v)}$

- 4. a: collect all votes
- 5. *a*: publish outcome

Notice a receipt?

TU/e receipts in RIES

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

in advance

• BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

To prove that v cast a vote for candidate c, it suffices to show an k such that $< h(\{c\}_k), c > \in \mathcal{L}$.

This is precisely the voter's key!

TU/e receipts in RIES

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

In advance

• BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

To prove that v cast a vote for candidate c, it suffices to show an k such that $< h(\{c\}_k), c > \in \mathcal{L}$.

This is precisely the voter's key!

This means the following in the formalism:

 $\bullet \ \alpha(x) = x$

 $\blacksquare \ \beta(x) = x$
TU/e receipts in RIES

Introduction

Real world voting

E-voting

Receipts

Formalisation

More concretely

Application

in advance

• BT

BT: receipt-free

● RIES

receipts in RIES

Final Thoughts

To prove that v cast a vote for candidate c, it suffices to show an k such that $< h(\{c\}_k), c > \in \mathcal{L}$.

This is precisely the voter's key!

This means the following in the formalism:

 $\bullet \ \alpha(x) = x$

• $\beta(x) = x \dots$ for suitable \mathcal{RB}

TU/e Conclusions

Introduction
Real world voting
E-voting
Receipts
Formalisation
More concretely
Application

- We're doing nice work here!
- ... but we're not yet done
- BT, SK95, HS and ALBD analysis indicates receipt-freeness
- RIES and FOO analysis demonstrates receipts
- More information in paper (submitted)...
- ... or the tech report (to appear)

TU/e Conclusions

Introduction
Real world voting
E-voting
Receipts
Formalisation
More concretely

Application

Final Thoughts	
 Conclusions 	ſ

- We're doing nice work here!
- ... but we're not yet done
- BT, SK95, HS and ALBD analysis indicates receipt-freeness
- RIES and FOO analysis demonstrates receipts
- More information in paper (submitted)...
- ... or the tech report (to appear)

Questions?

TU/e Conclusions

Introduction
Real world voting
E-voting
Receipts
Formalisation

More concretely

Application

Final Thoughts	
 Conclusions 	

We're doing nice work here!

... but we're not yet done

- BT, SK95, HS and ALBD analysis indicates receipt-freeness
- RIES and FOO analysis demonstrates receipts
- More information in paper (submitted)...

... or the tech report (to appear)

Questions?

Take care of yourself... ... and each other!

Jerry Springer