End-to-end verifiability in the standard model

Thomas Zacharias & Bingsheng Zhang

joint work with **Aggelos Kiayias**

National & Kapodistrian University of Athens
Cryptography Security Lab – http://crypto.di.uoa.gr

www.demos-voting.org

May 28th, 2015

EUROCRYPT

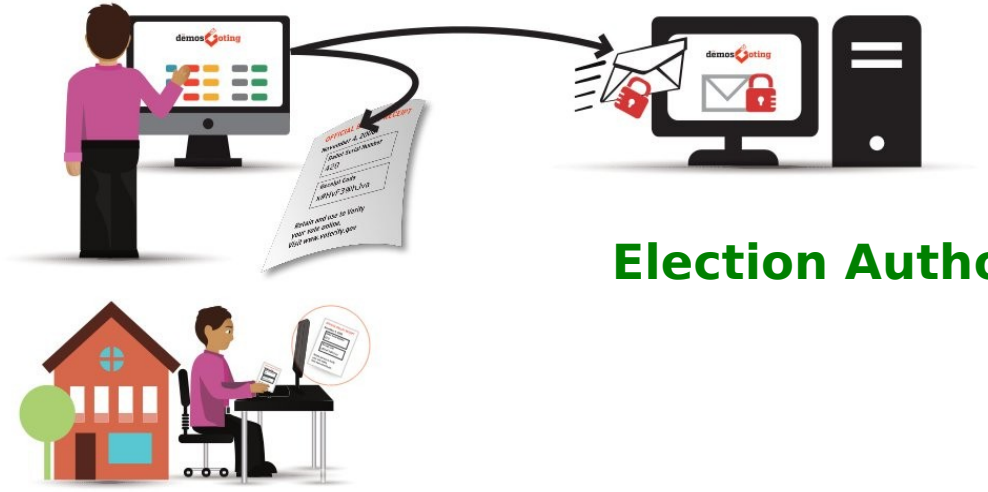
E-Voting systems in general

- Increase the participation of social groups that face considerable physical barriers.
- Reduce the financial cost of the elections.
- Increase the efficiency of the preparation of the election and the calculation of the final results.
- Preserve the fundamental requirements of a voting system (eligibility, integrity, fairness, secrecy etc.).
- They are divided into two main categories: (a) on-site e-voting systems and (b) remote e-voting systems.

Parties involved in an e-voting system

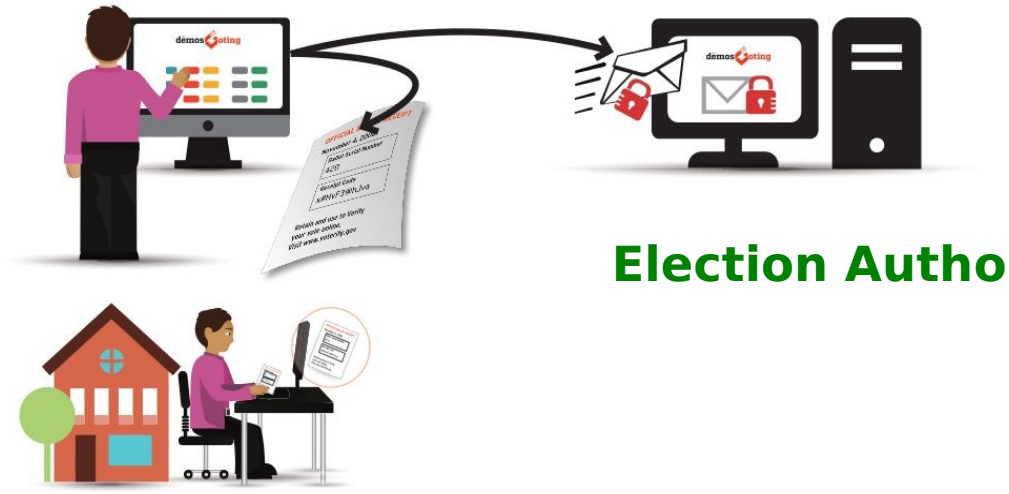
- **Voters**.
- An (a set of) **Election Authority**(ies) responsible for election preparation and tally announcement.
- A publicly accessible **Bulletin Board** (BB) where the tally is announced and voters can verify the election procedure.
- **Voter clients** used for vote submission.

End-to-end verifiability in e-voting systems



Election Authority (EA)

End-to-end verifiability in e-voting systems



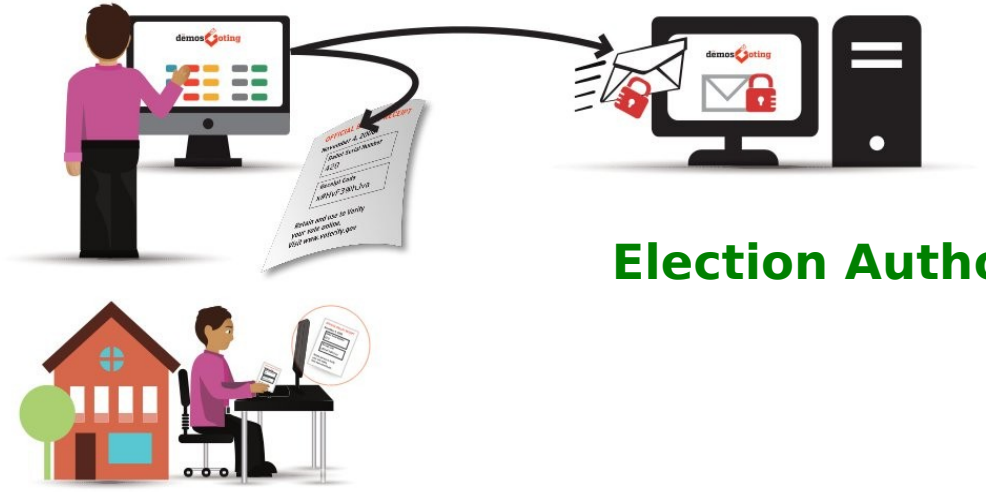
The voter obtains a receipt in order to verify that her vote was:

Cast-as-intended

Recorded-as-cast

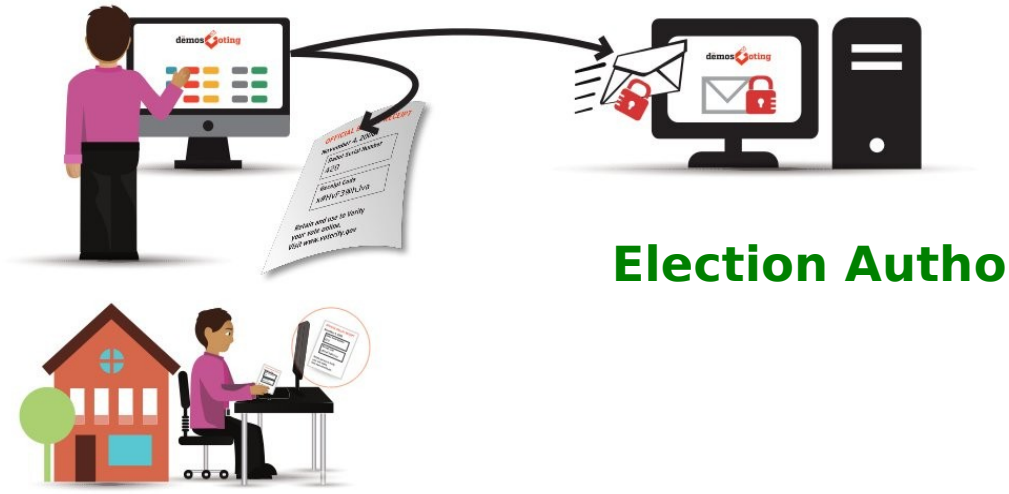
Tallied-as-recorded

Ideal standard for end-to-end verifiable e-voting



Election Authority (EA)

Ideal standard for end-to-end verifiable e-voting



1. Voters submit their votes to **EA** via an **authenticated channel**.

2. The **EA** publishes them all in the form (Name, Address, Vote) at the end of the election (together with the result).

End-to-end verifiability in e-voting systems

How close we can go to the ideal standard?

... while preserving secrecy /
universal suffrage / coercion resistance ...

Well known e-voting systems

On-site e-voting systems:

- SureVote [2001]
- Prêt à Voter [2005]
- Scantegrity [2008]
- STAR–Vote [2013]

Remote e-voting systems:

- Helios [2008]
- Scytl/Norwegian [2008]
- Civitas [2008]
- Remotegrity [2013]

Well known voter-side encryption e-voting systems

On-site e-voting systems:

- SureVote [2001]
- Prêt à Voter [2005]
- Scantegrity [2008]
- STAR–Vote [2013]

Remote e-voting systems:

- Helios [2008]
- Scytl/Norwegian [2008]
- Civitas [2008]
- Remotegrity [2013]

Well known vote-code e-voting systems

On-site e-voting systems:

- SureVote [2001]
- Prêt à Voter [2005]
- Scantegrity [2008]
- STAR–Vote [2013]

Remote e-voting systems:

- Helios [2008]
- Scytl/Norwegian [2008]
- Civitas [2008]
- Remotegrity [2013]

Till now, end-to-end verifiability could be achieved only if:

1) the voters are supposed to trust the client environment (e.g. **Scytl/Norwegian**,**Civitas**)

Till now, end-to-end verifiability could be achieved only if:

1) the voters are supposed to trust the client environment (e.g. **Scytl/Norwegian**,**Civitas**)

Till now, end-to-end verifiability could be achieved only if:

or **2)** the voters are supposed to trust an **unfalsifiable** assumption (called the **random oracle**)

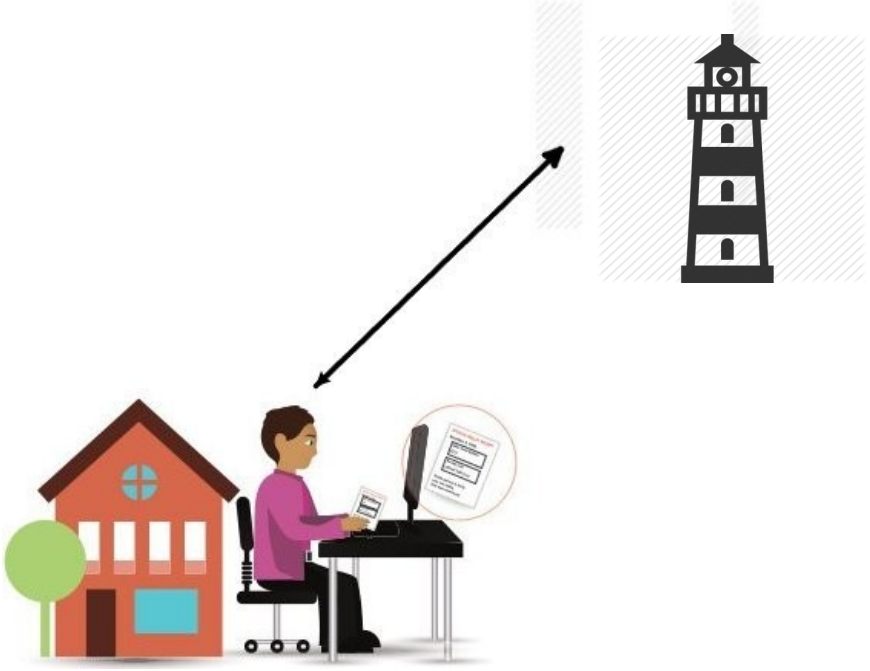
(e.g. **Helios, STAR-Vote**)

Till now, end-to-end verifiability could be achieved only if:

or **3)** the voters are supposed to trust a

randomness beacon

(**Code-voting systems Prêt à Voter/Scantegrity**)



Till now, end-to-end verifiability could be achieved only if:

or **3)** the voters are supposed to trust a

randomness beacon

(**Code-voting systems Prêt à Voter/Scantegrity**)

The fundamental question

The fundamental question

Can you **prove** that the election result is correct without requiring voters to believe in trusted hardware, random oracles, randomness beacons or even computational assumptions ?

The fundamental question

Can you **prove** that the election result is correct without requiring voters to believe in trusted hardware, random oracles, randomness beacons or even computational assumptions ?

We answer this question affirmatively!

Contributions of this work

1) We introduce a security framework where:

   (a) **End-to-end verifiability** is defined for adversaries that control the **entire election procedure** and a number of voters.

   (b) **Voter privacy & receipt-freeness** is defined for adversaries that **observe the network and obtain the honest voters' receipts**.

Contributions of this work

1) We introduce a security framework where:

   (a) **End-to-end verifiability** is defined for adversaries that control the **entire election procedure** and a number of voters.

   (b) **Voter privacy & receipt-freeness** is defined for adversaries that **observe the network and obtain the honest voters' receipts**.

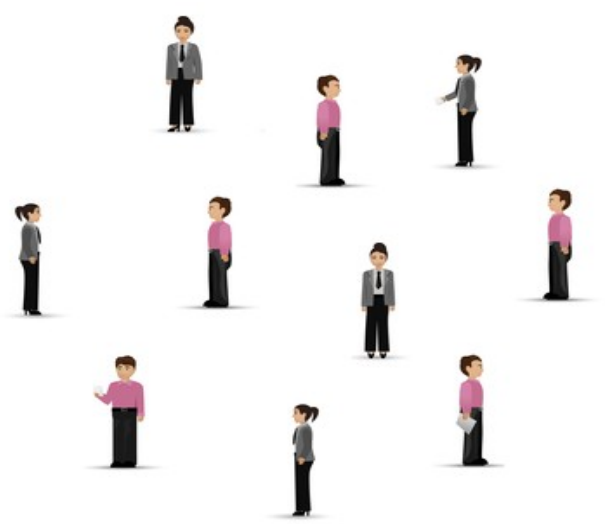
2) We construct an ***end-to-end verifiable remote code-voting*** system which achieves:

   (a) **End-to-end verifiability**, assuming only a **consistent bulletin board (BB)**.

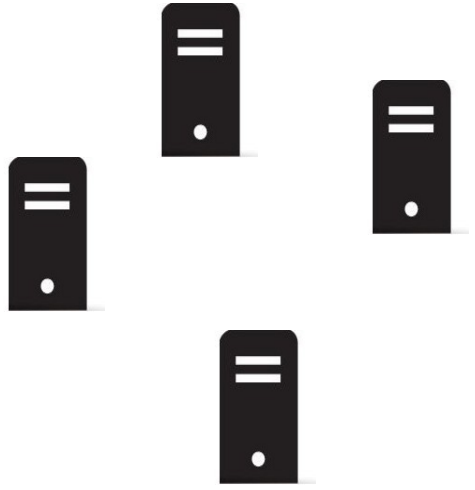
   (b) **Voter privacy & receipt freeness**, assuming the **subexponential hardness of the Decisional Diffie-Hellman problem**.

Security framework: End-to-end verifiability

The adversarial power

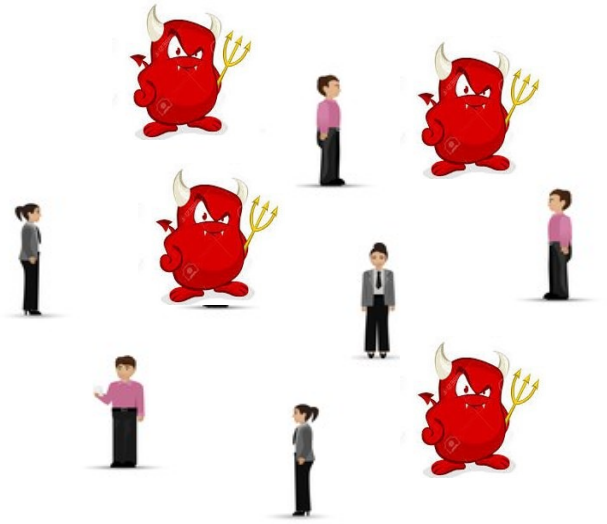


Bulletin Board



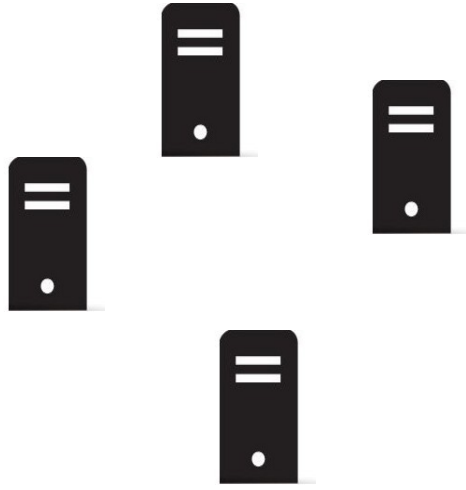
Security framework: End-to-end verifiability

The adversarial power



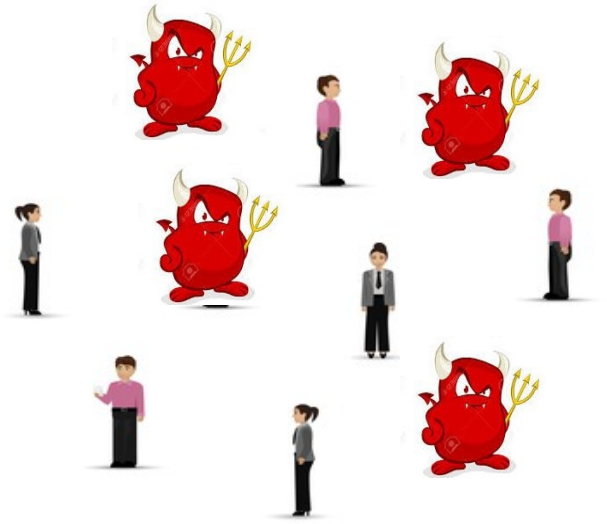
The adversary can corrupt a number of voters

Bulletin Board



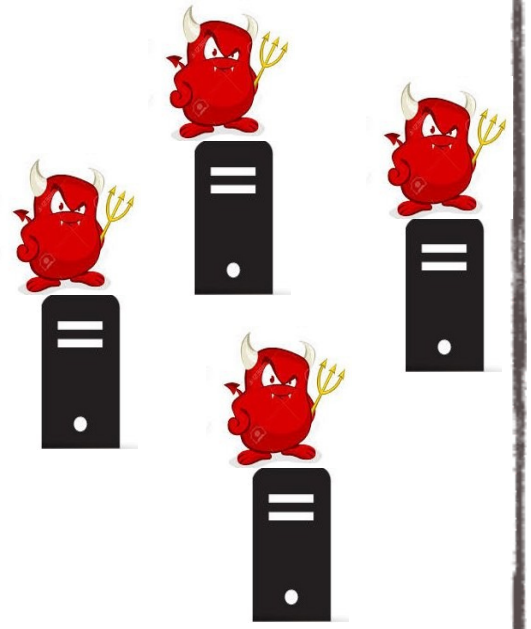
Security framework: End-to-end verifiability

The adversarial power



The adversary can corrupt a number of voters

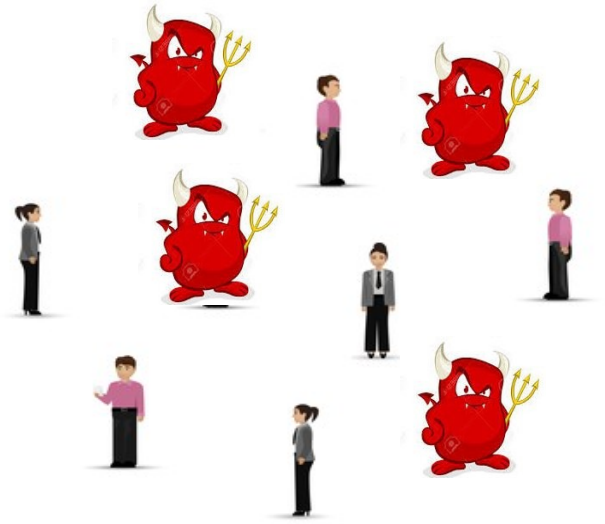
Bulletin Board



The adversary can corrupt all EA servers

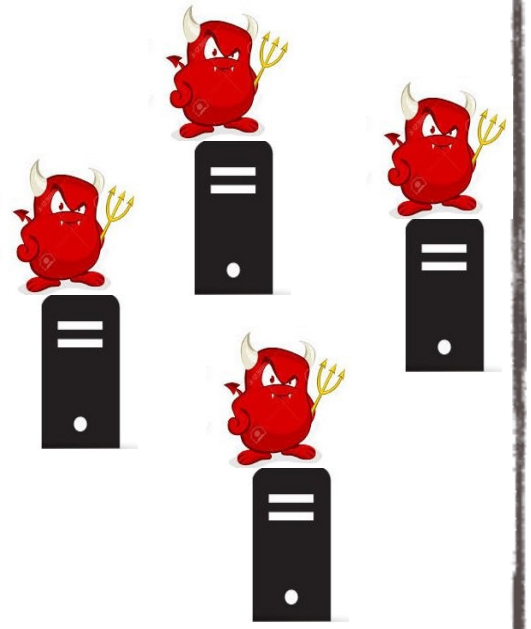
Security framework: End-to-end verifiability

The adversarial power

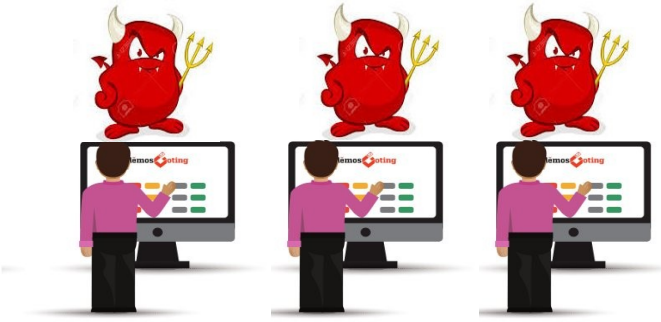


The adversary can corrupt a number of voters

Bulletin Board



The adversary can corrupt all EA servers



The adversary can corrupt all voters' clients

Security framework: End-to-end verifiability

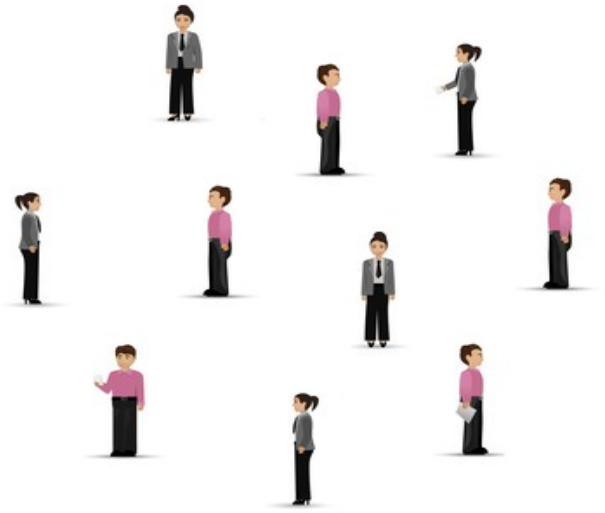
Definition

An e-voting system achieves **end-to-end verifiability with parameters (ε,θ,*d*)** *if every adversary that does not corrupt at least* ***θ*** *voters, cannot cause tally deviation more than* ***d*** *votes with more than* ***ε*** *probability*.

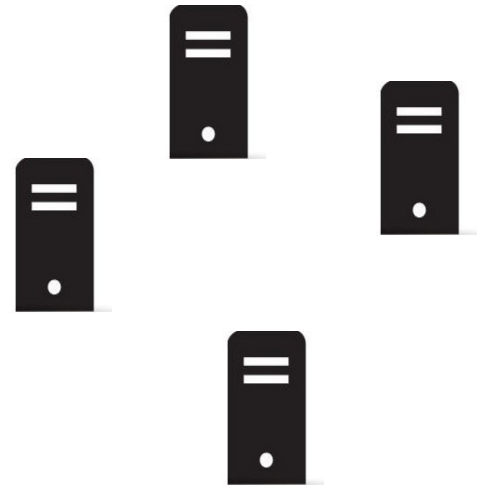
Key point: express ***ε*** as a function of ***d*** or ***θ***. It should be that ***ε*** decreases rapidly as ***d*** or ***θ*** become larger.

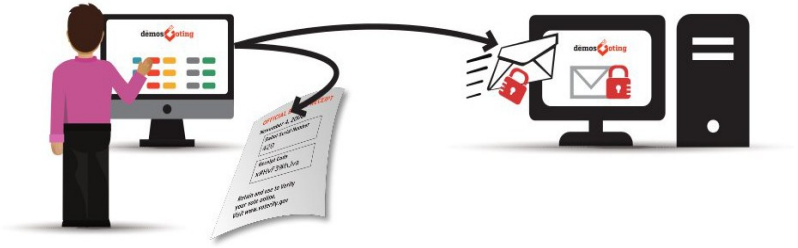
Security framework: Voter privacy & receipt-freeness

The adversarial power



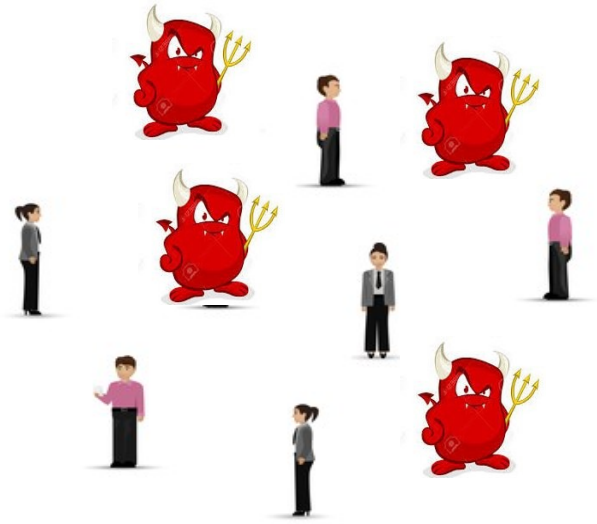
Bulletin Board





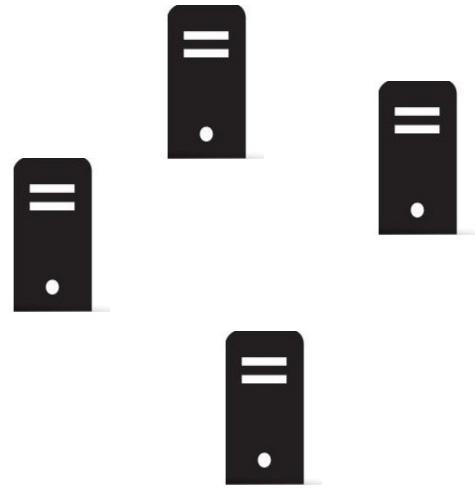
Security framework: Voter privacy & receipt-freeness

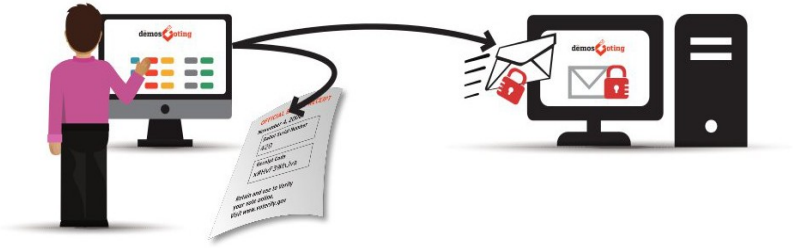
The adversarial power



The adversary can corrupt a number of voters

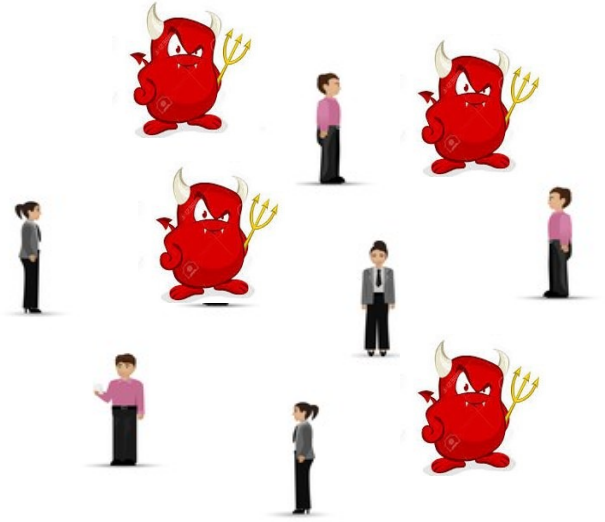
Bulletin Board





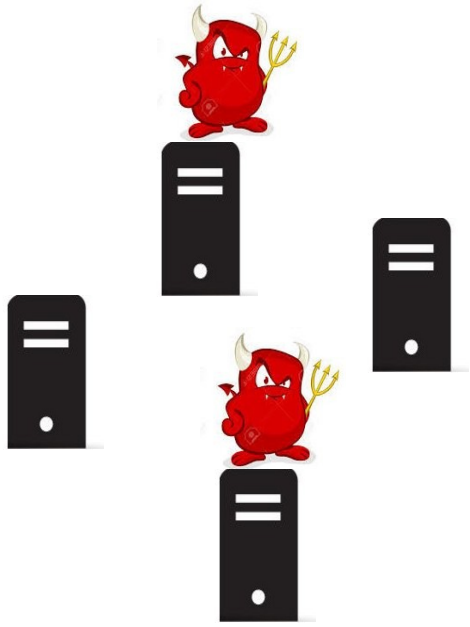
Security framework: Voter privacy & receipt-freeness

The adversarial power

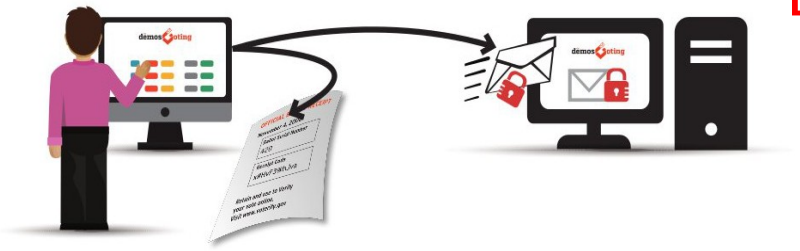


The adversary can corrupt a number of voters

Bulletin Board

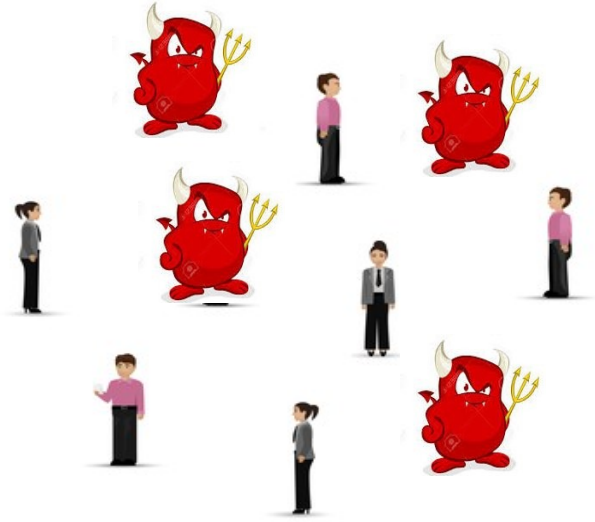


The adversary can corrupt a subset of EA servers



Security framework: Voter privacy & receipt-freeness

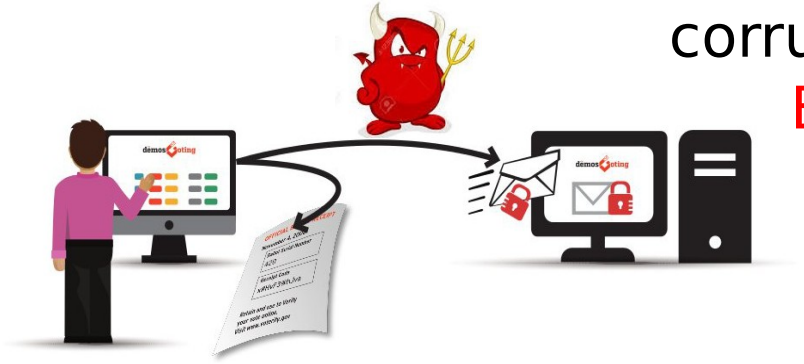
The adversarial power

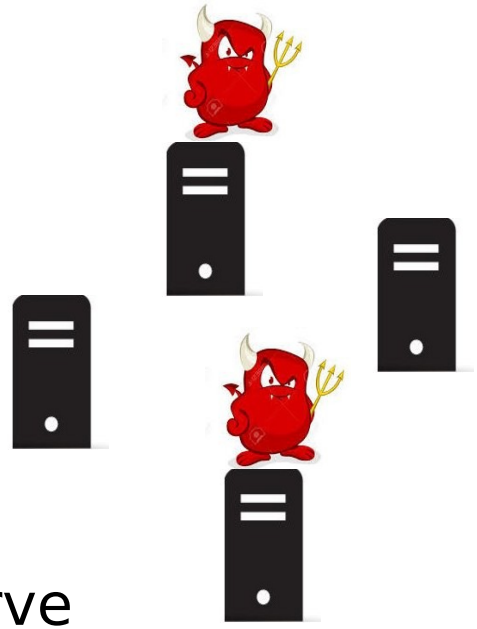


The adversary can corrupt a number of voters

Bulletin Board

The adversary can observe the network

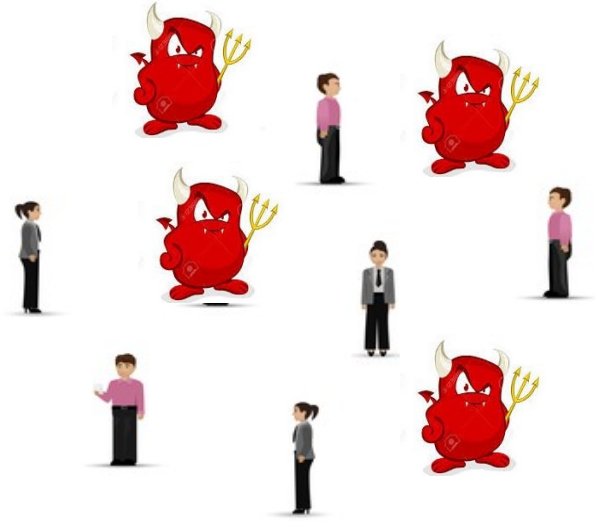




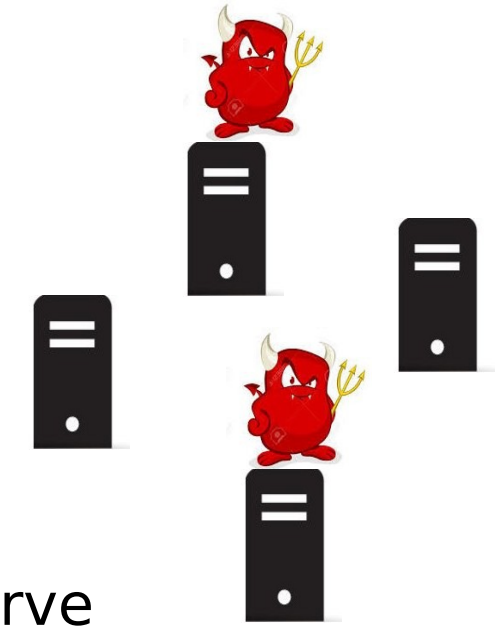
The adversary can corrupt a subset of EA servers

Security framework: Voter privacy & receipt-freeness

The adversarial power



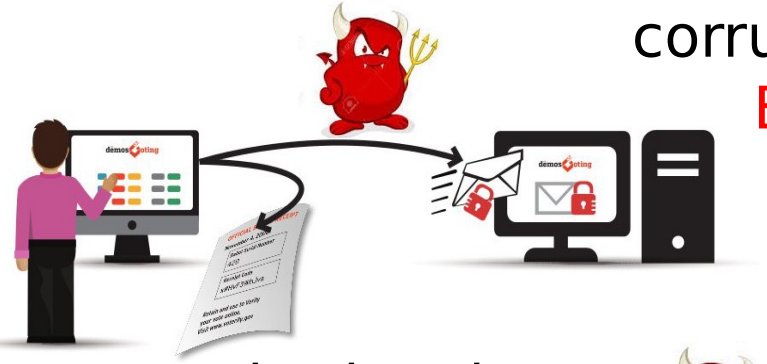
Bulletin Board



The adversary can observe the network

The adversary can corrupt a number of voters

The adversary can corrupt a subset of EA servers



The adversary obtains the honest voters' receipts

Security framework: Voter privacy & receipt-freeness

Definition

An e-voting system achieves **voter privacy & receipt-freeness with parameters (ε,t,k)**, *if every adversary that corrupts at most* ***t*** *voters and* ***k*** *servers cannot distinguish between any two plausible voter strategies with more than* ***ε*** *probability.*

Tools

Perfectly Binding Commitments

- Lifted ElGamal over elliptic curves
  - Additively homomorphic property.
  - Used to commit candidate encodings.

Perfectly Binding Commitments

- Lifted ElGamal over elliptic curves
  - Additively homomorphic property.
  - Used to commit candidate encodings.
  - The i-th candidate is encoded as N^{i-1}

N = #voters + 1

Commitment of the (i+1)-th candidate is $E = \left(g^r , g^{N^i} h^r \right)$

Min-entropy Schwartz-Zippel

Schwartz-Zippel lemma (Min-entropy variant):

Let f(x) be a non-zero univariate polynomial of degree d over Z_q .
Let D be a probability distribution on Z_q such that $H_\infty(D) \geq k$.
The probability of f(x) = 0 is at most $d/2^k$, where x is drawn randomly according to D.

A Sigma Protocol for Commitment Correctness

- Statement: $E = \left(g^r , g^{N^i} h^r \right)$
- Witness: (i,r) w.l.o.g. $0 \leq i < 2^k , i = \sum_{j=0}^{k-1} 2^j b_j$

A Sigma Protocol for Commitment Correctness

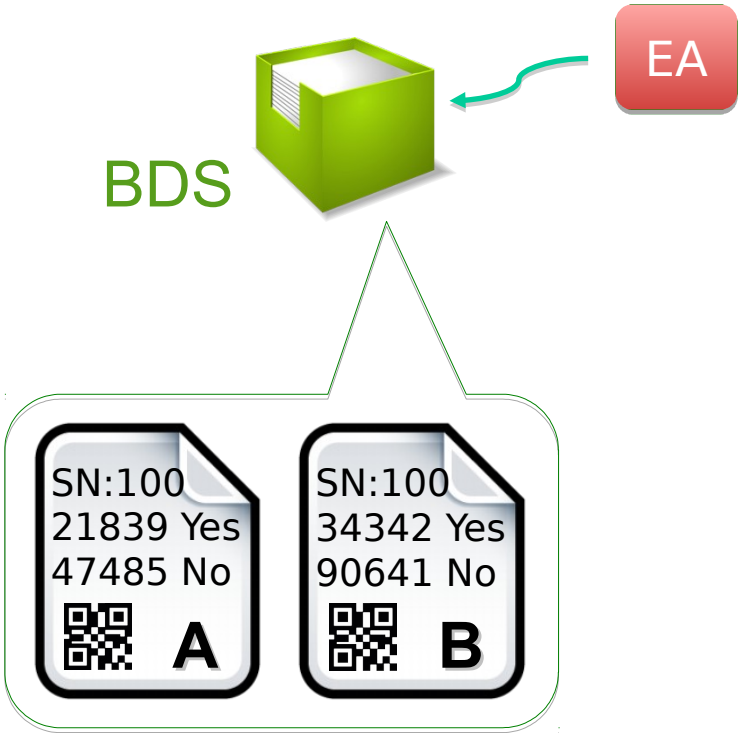
- Statement: $E=\left(g^r, g^{N^i} h^r\right)$
- Witness: (i,r) w.l.o.g. $0 \le i < 2^k, i=\sum_{j=0}^{k-1} 2^j b_j$
- Intuition:
  - Commit i bit-wisely: $B_j = Com\left(b_j\right)$
  - Show that $b_j\left(1-b_j\right)=0$
  - Set $A_j = Com\left(a_j\right) = B_j^{N^{2^j}-1} Com\left(1\right)$ so $a_j = b_j N^{2^j} - b_j + 1$
  - Show that $\prod_{j=0}^{k-1} a_j$ equals to the content of the the commitment
    - Construct and test polynomials

$$f(X)=\prod_{j=0}^{k-1}(a_j X + s_j)=\prod_{j=0}^{k} e_j X^j$$

System Description

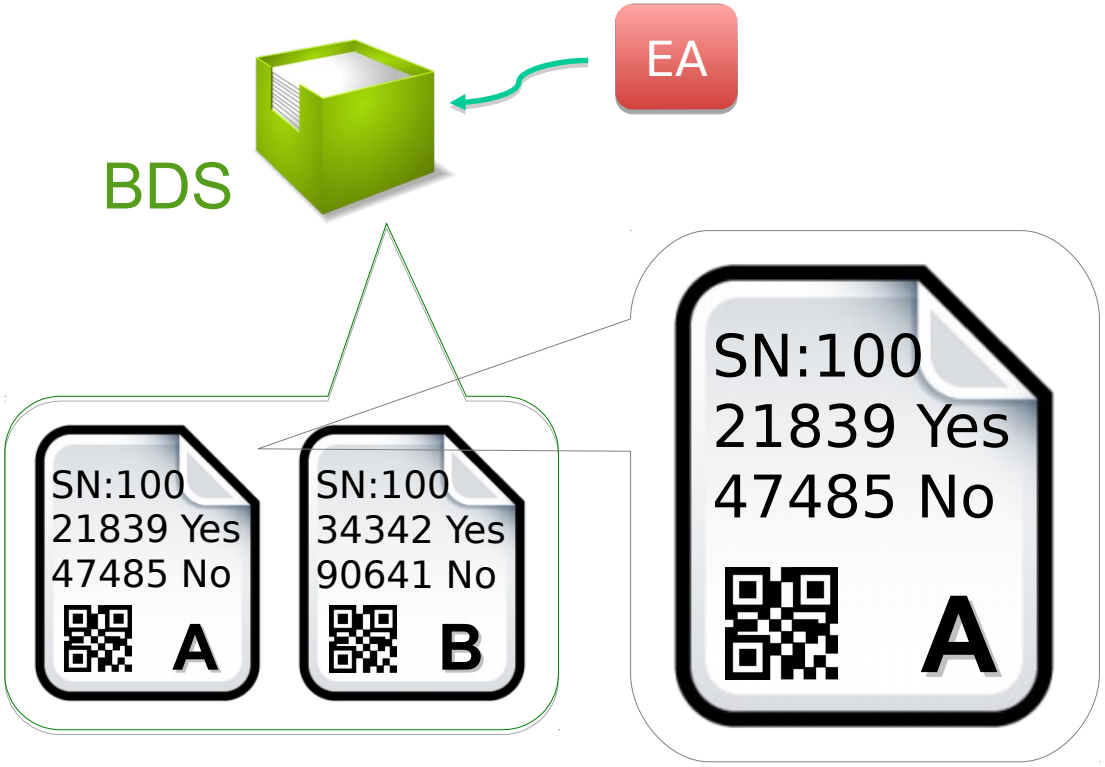
dēmos Voting

Step 1: Setup



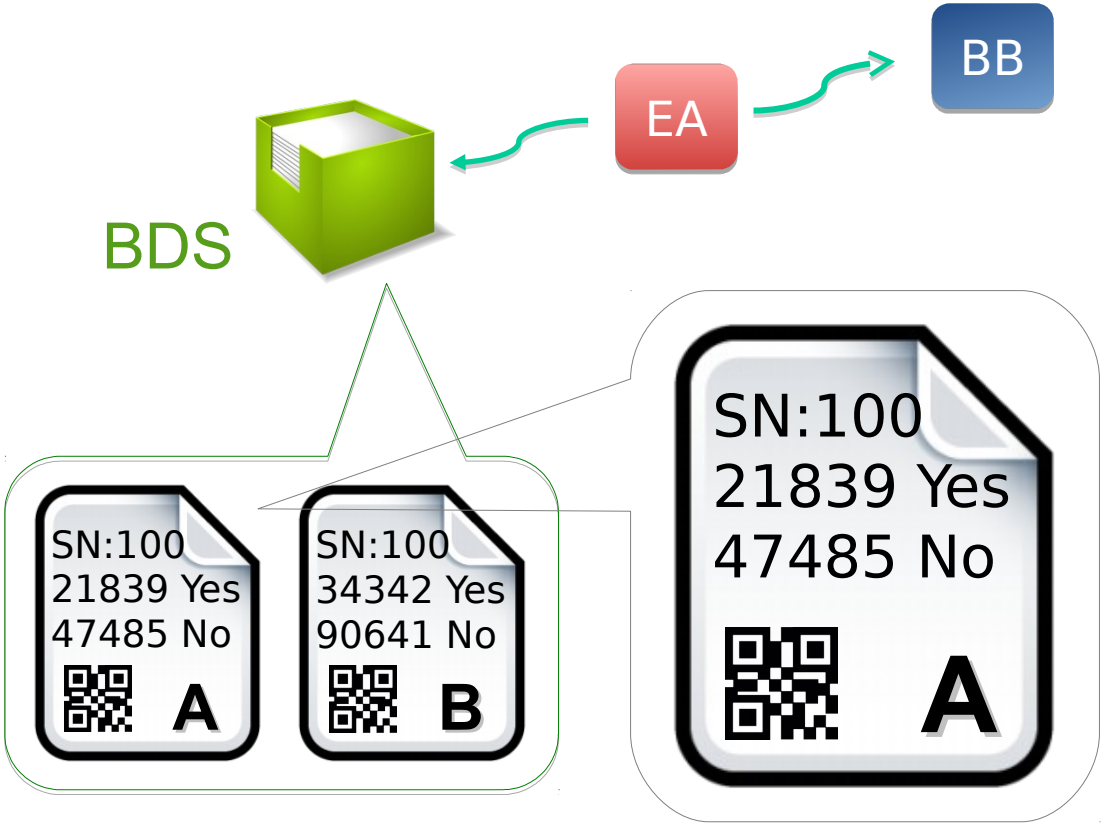
dēmos Voting

Step 1: Setup



dēmos Voting

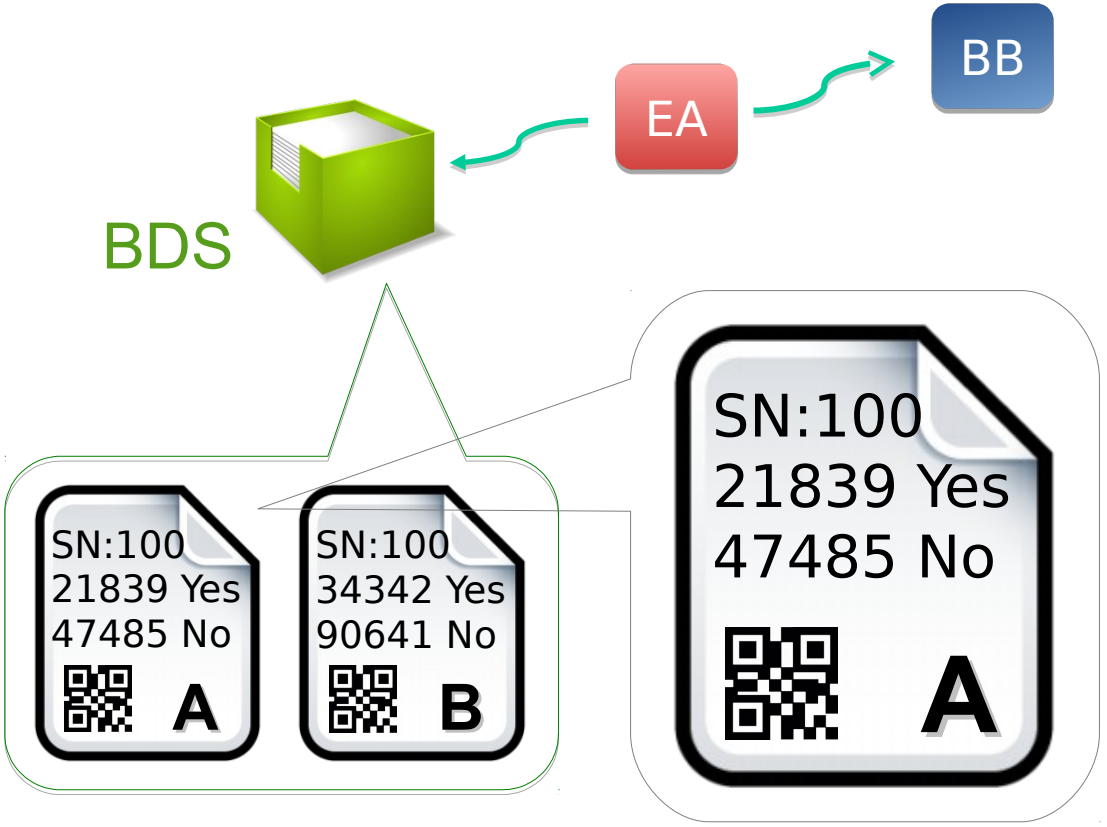
Step 1: Setup



SN: 100		
Side A	21839	Initial Data
	47485	Initial Data
Side B	90641	Initial Data
	34342	Initial Data
SN: 101		
Side A	50349	Initial Data
	22092	Initial Data
Side B	43547	Initial Data
	97651	Initial Data

dēmos Voting

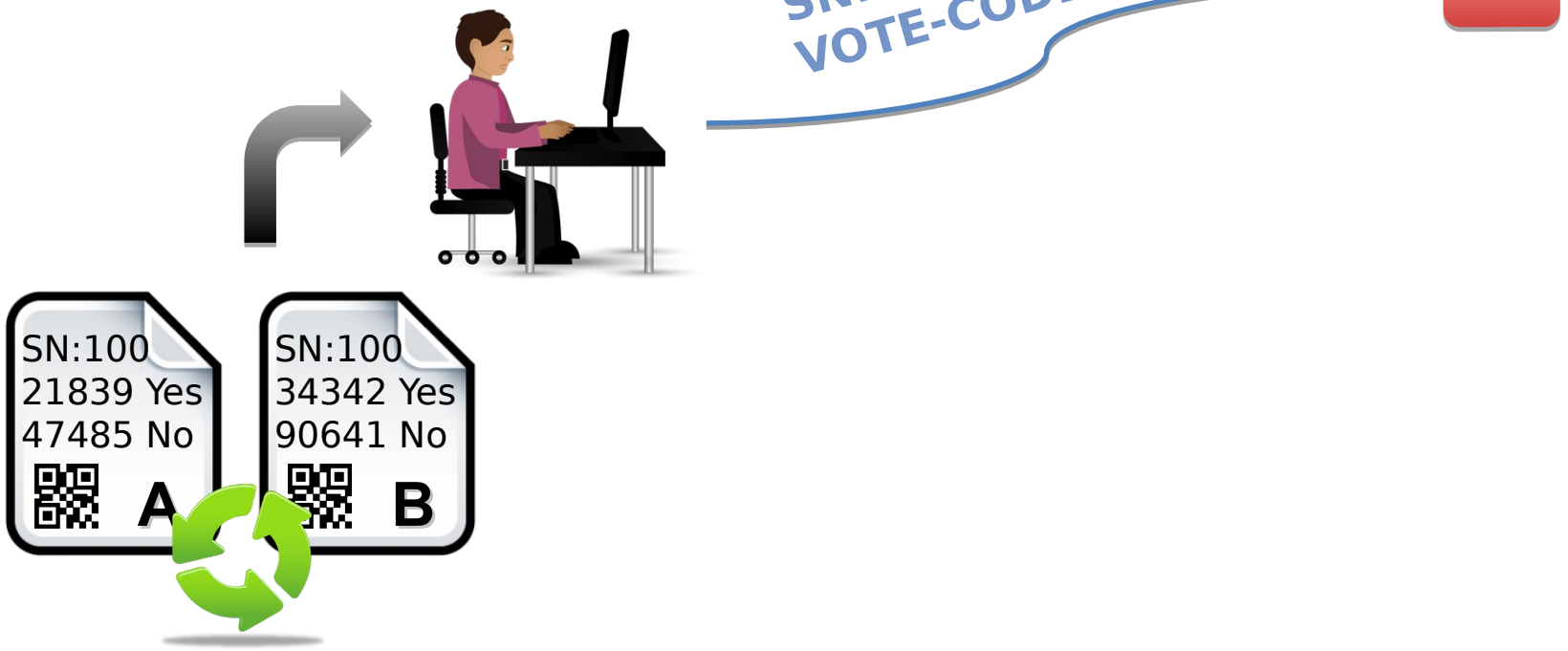
Step 1: Setup



SN: 100		
Side A	21839	Initial Data
	47485	Initial Data
Side B	90641	Initial Data
	34342	Initial Data
SN: 101		
Side A	50349	Initial Data
	22092	Initial Data
Side B	43547	Initial Data
	97651	Initial Data

dēmosVoting

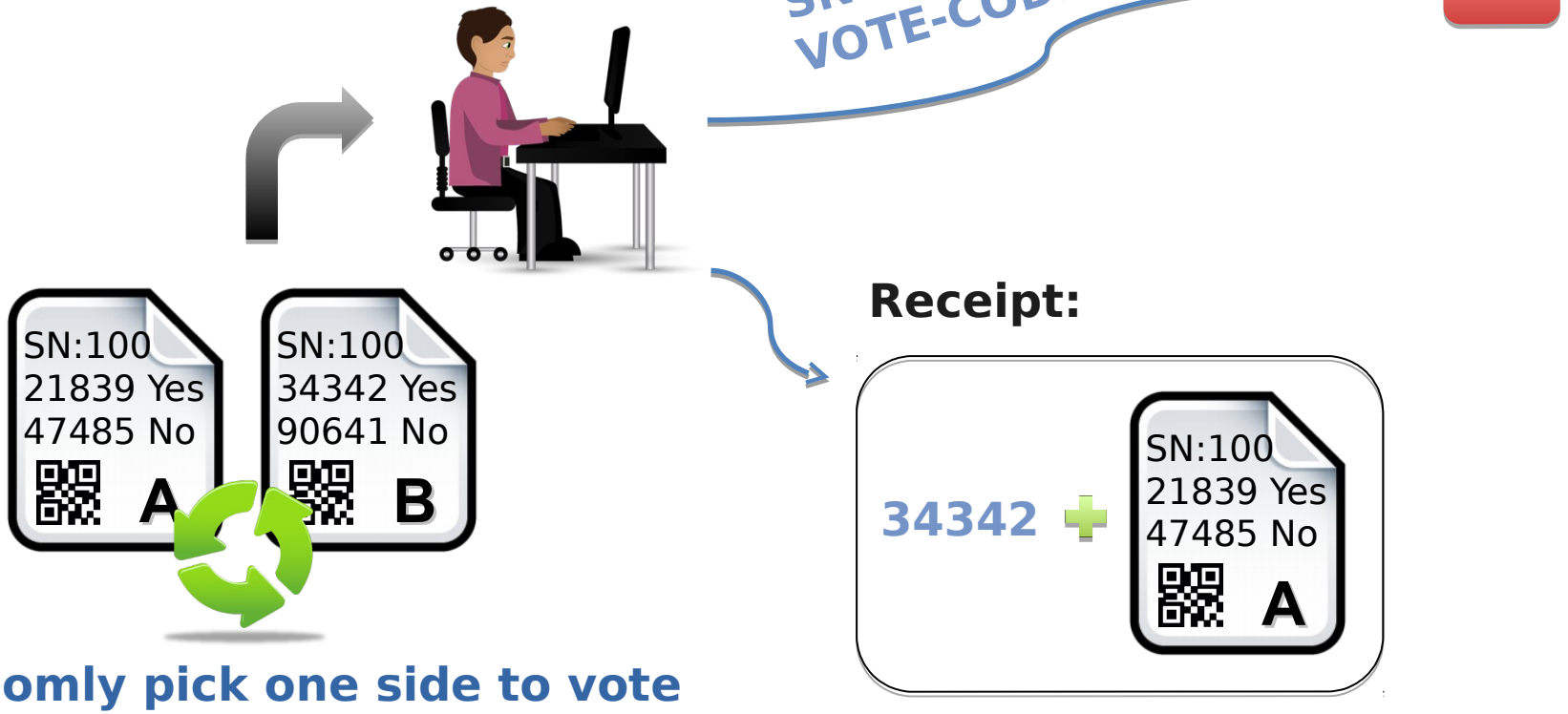
Step 2: Voting



Randomly pick one side to vote
Use the other side to audit

dēmosVoting

Step 2: Voting

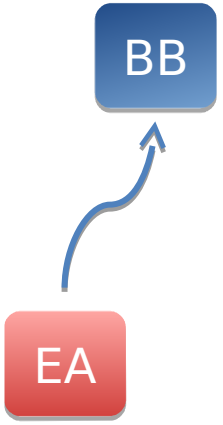


Randomly pick one side to vote
Use the other side to audit

dēmos Voting

Step 3: Finalizing

- Open vote-codes
- Mark “Voted”
- Open  of the unused side.
- Complete the Sigma protocols for “Voted”  .

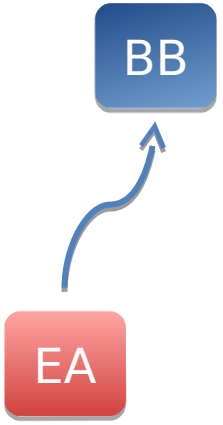


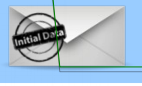
SN: 100			
Side A	21839		Yes
	47485		No
Side B	90641		
	34342		**Voted**
SN: 101			
Side A	50349		
	22092		**Voted**
Side B	43547		No
	97651		Yes

dēmos Voting

Step 3: Finalizing

- Open vote-codes
- Mark “Voted”
- Open  of the unused side.
- Complete the Sigma protocols for “Voted”  .

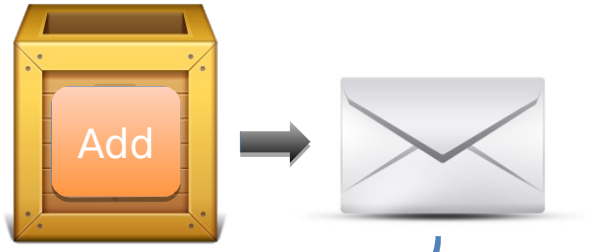


SN: 100			
Side A	21839		Yes
	47485		No
Side B	90641		
	34342		**Voted**
SN: 101			
Side A	50349		
	22092		**Voted**
Side B	43547		No
	97651		Yes

dēmos Voting

Step 4: Tally

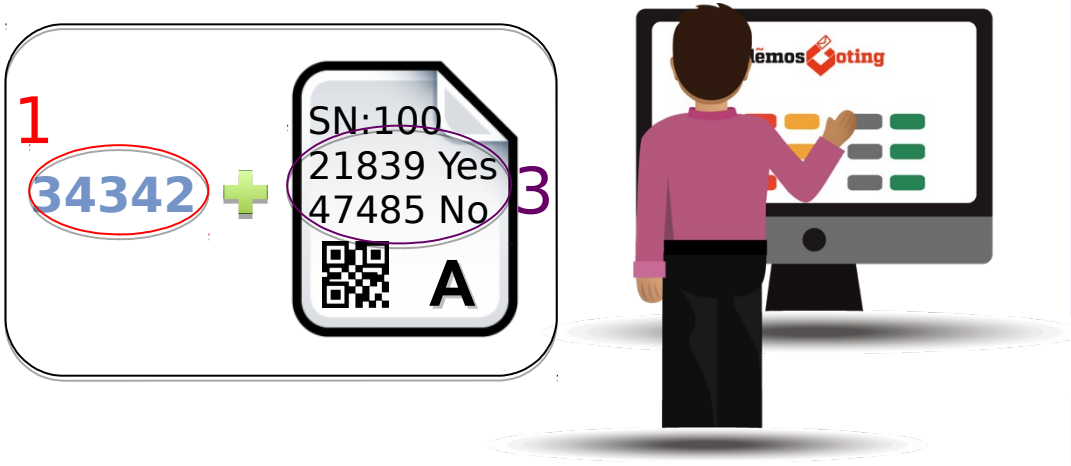
SN: 100			
Side A	21839		Yes
	47485		No
Side B	90641		
	34342		Voted
SN: 101			
Side A	50349		
	22092		Voted
Side B	43547		No
	97651		Yes



Election Result	
Yes	2 votes
No	0 votes

dēmos Voting

Step 5: Audit (optional)

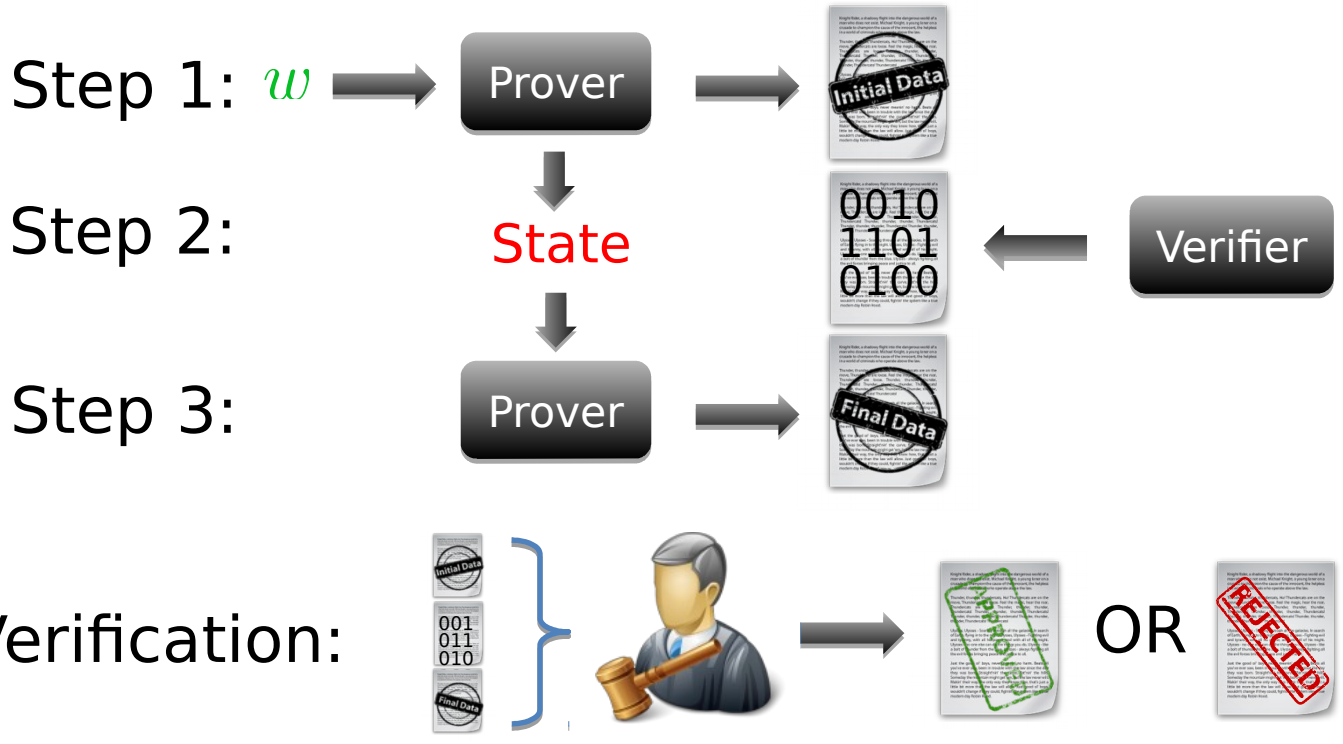


SN: 100			
Side A (4)	21839	Initial Data	Yes
	47485	Initial Data	No
Side B (2)	90641	Initial Data	
	34342	Initial Data, Final Data	**Voted**
SN: 101			
Side A	50349	Initial Data	
	22092	Initial Data, Final Data	**Voted**
Side B	43547	Initial Data	No
	97651	Initial Data	Yes

- Check 1 and 2 are consistent
- Check 3 and 4 are consistent
- Verify  (c.f. later)

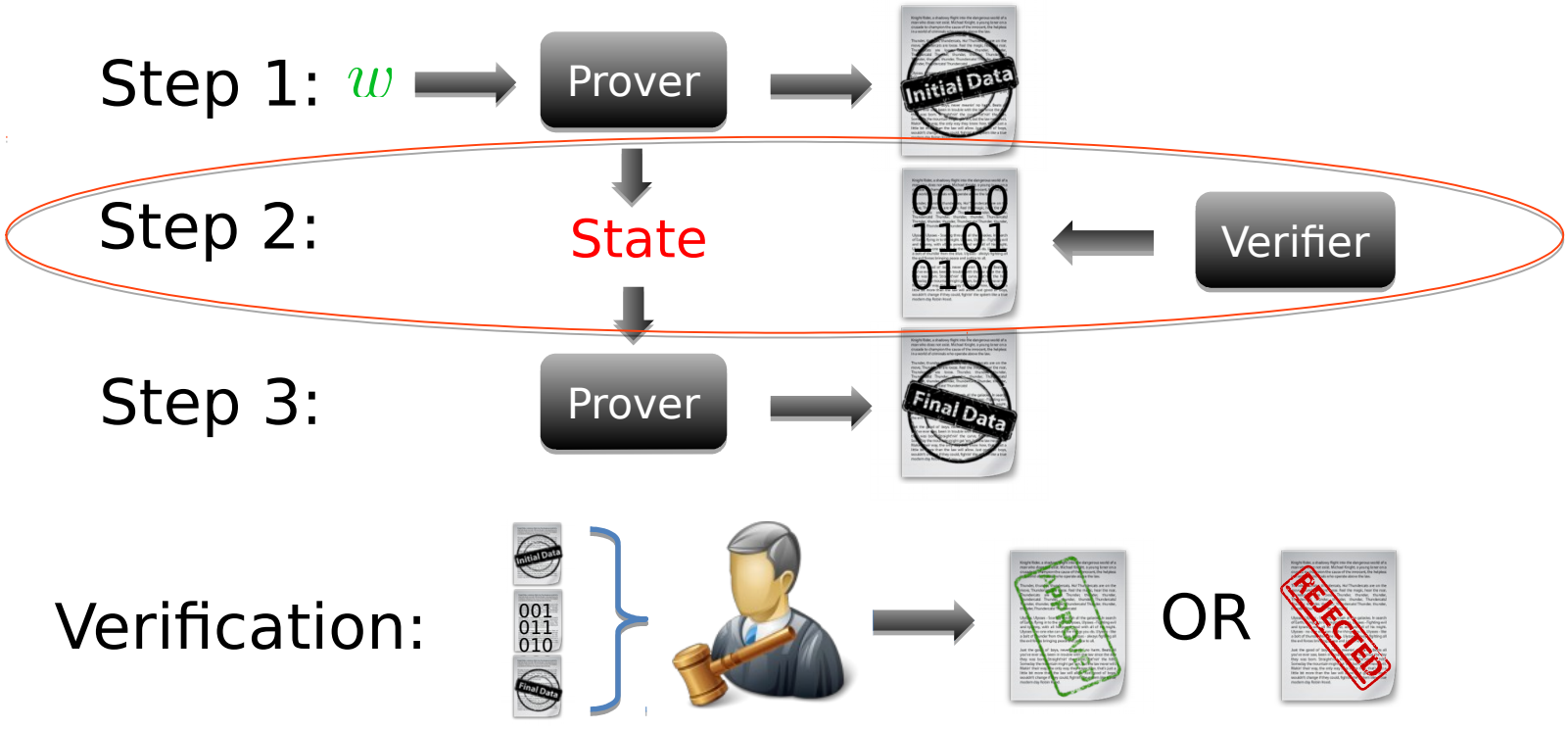
Sigma Protocols

$$x \in \mathcal{L} \leftrightarrow \exists w : (x, w) \in \mathcal{R}_{\mathcal{L}}$$



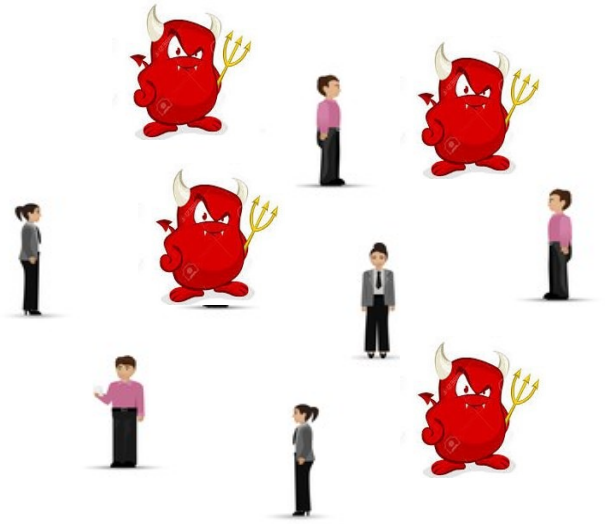
Sigma Protocols

$$x \in \mathcal{L} \leftrightarrow \exists w : (x, w) \in \mathcal{R}_{\mathcal{L}}$$



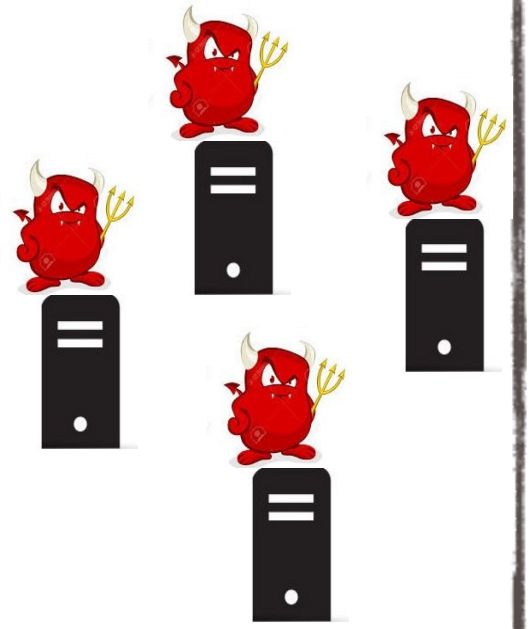
Security framework: End-to-end verifiability

The adversarial power

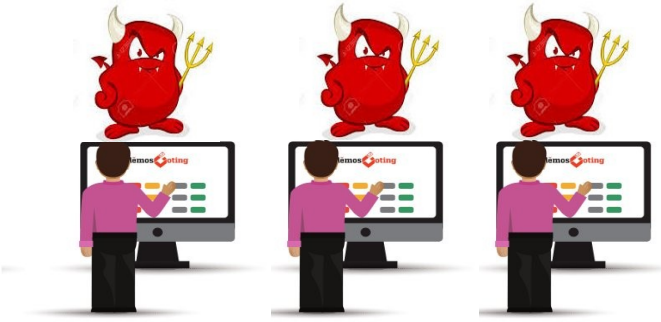


The adversary can corrupt a number of voters

Bulletin Board



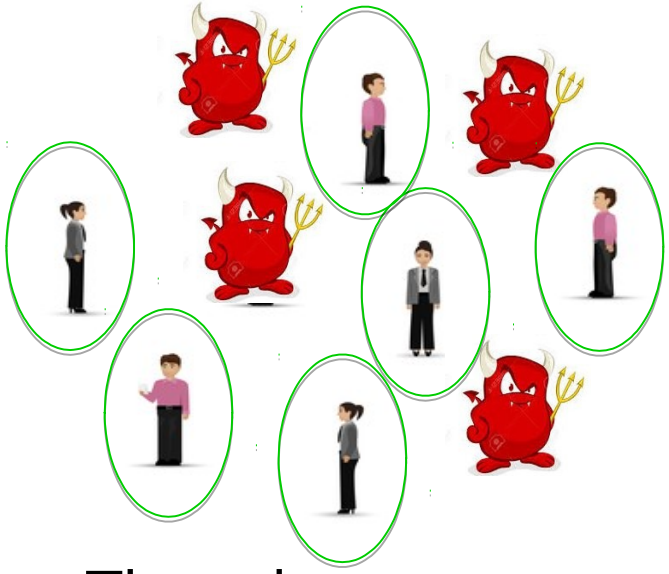
The adversary can corrupt all EA servers



The adversary can corrupt all voters' clients

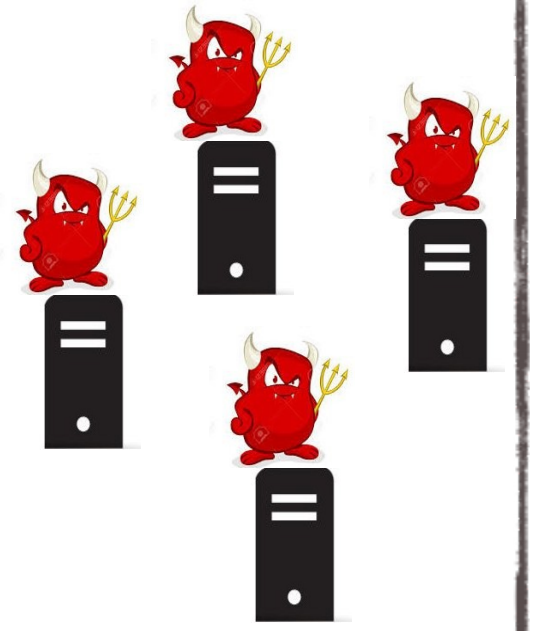
Security framework: End-to-end verifiability

The adversarial power

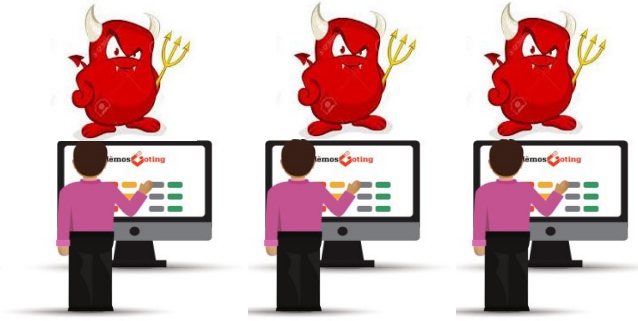


The adversary can corrupt a number of voters

Bulletin Board



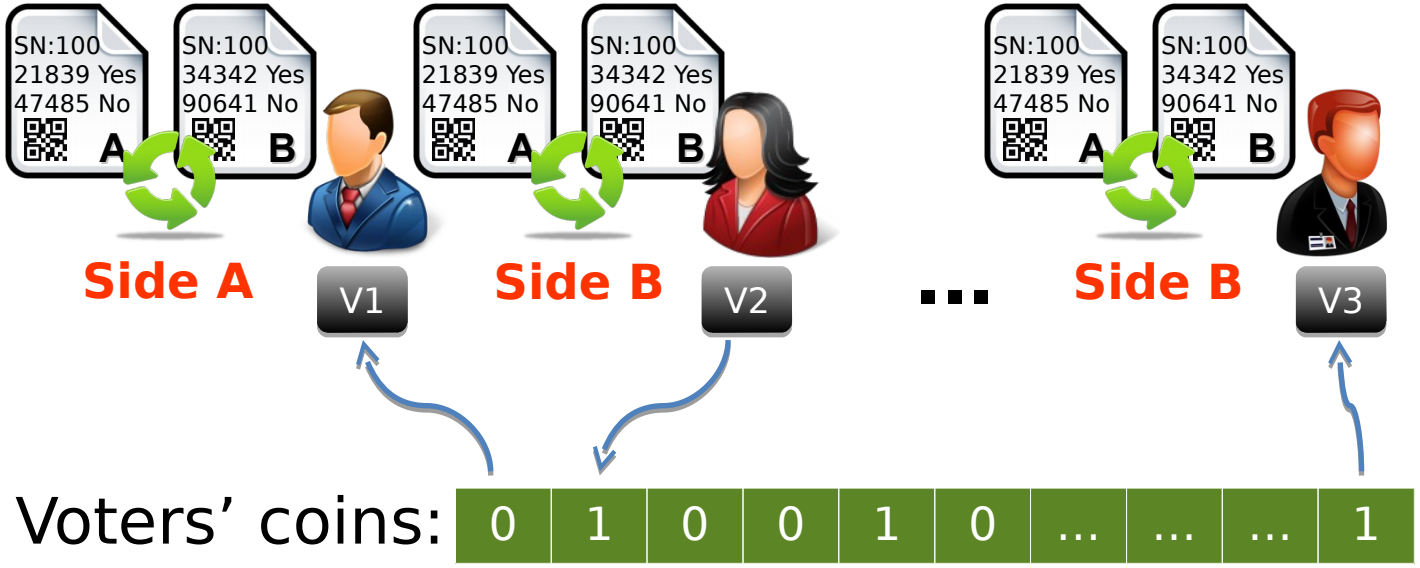
The adversary can corrupt all EA servers



The adversary can corrupt all voters' clients

Sigma Challenge Generation

- Recall that each voter should select side A or B at random in voting phase.



NB: Side A = 0 Side B = 1

Quality of the Voters' coins

- Most of them can be produced by the adversaries.
- The honest voters produce independent low entropy coins.
- The adversarial coins may depend on the honest voters' coins.
- For large elections, the length of the coins is too long to be presented as a unique group element.

Adaptive non-oblivious bit fixing source

How can we use this source?

- LHL extractors?
  - Who will produce the seed?

How can we use this source?

- LHL extractors?
  - Who will produce the seed?
- Deterministic extractors?
  - Kamp and Zuckerman showed that at most n/k bits can be extracted when k out of n bits of the source are fixed!

How can we use this source?

- LHL extractors?
  - Who will produce the seed?
- Deterministic extractors?
  - Kamp and Zuckerman showed that at most n/k bits can be extracted when k out of n bits of the source are fixed!
- Condensers?
  - Entropy loss

How can we use this source?

- LHL extractors?
  - Who will produce the seed?
- Deterministic extractors?
  - Kamp and Zuckerman showed that at most n/k bits can be extracted when k out of n bits of the source are fixed!
- Condensers?
  - Entropy loss

Can we do Better?

ZK Soundness Amplification

- The voters' coins are divided into k challenges.

- The prover will prove the statement according to all k challenges.
- The verifier will accept the proof is all of them are valid.

ZK Soundness Amplification

- The voters' coins are divided into k challenges.

- The prover will prove the statement according to all k challenges.
- The verifier will accept the proof is all of them are valid.

By min-entropy Schwartz-Zippel lemma, the soundness error of our Sigma protocol drops exponentially w.r.t. the min-entropy of the challenges.

Verifiability In The Standard Model

In Step 1, the EA posts the initial data (k copies of the sigma protocols).

In Step 2, voters' coins are divided into k challenges.

In Step 3, the EA posts the final data (k copies of the sigma protocols).

End-to-end verifiability of Demos

End-to-end verifiability of Demos

Possible attacks:

1) The adversary commits to an invalid encoded value (e.g. 1000 votes for “Yes”) and posts it on the BB.

2) The adversary commits to a different vote-code and candidate correspondence than the one in the honest voter's ballot.

3) The adversary performs a clash attack by linking a set of honest voters to the same audit position on the BB.

End-to-end verifiability of Demos

1) Defense against invalid commitments:

By the soundness of the Sigma protocol for ballot correctness, the probability that such an attack is successful is no more than

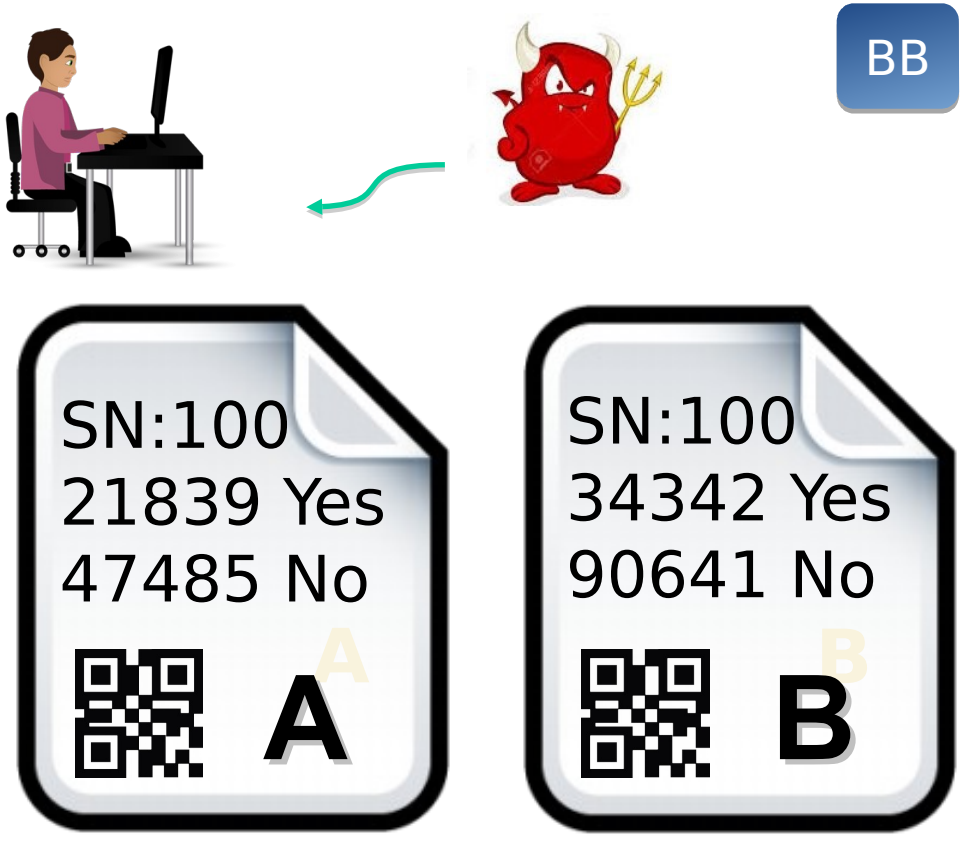
$$2^{-\left(\vartheta-\left(n/\log q+1\right)\log\log m\right)}$$

where q is the size of the modulo group, n is the number of voters, m is the number of candidates and θ is the number of honest voters.

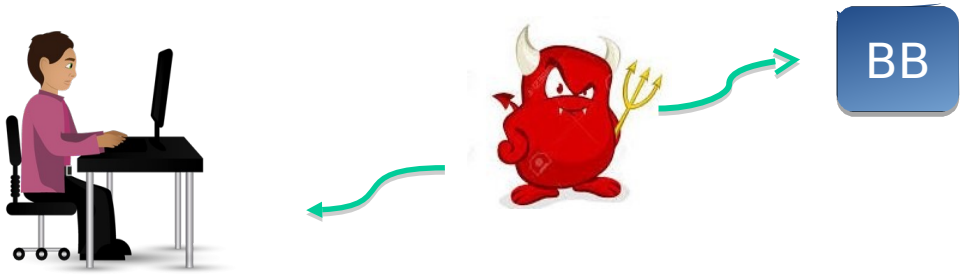
End-to-end verifiability of Demos

2)Defense against inconsistencies in vote-code and candidate correspondences:

End-to-end verifiability of Demos



End-to-end verifiability of Demos



SN: 100			
Side A	21839		No
	47485		Yes
Side B	90641		No
	34342		Yes

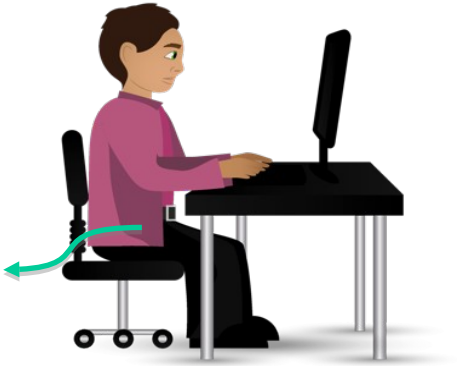
SN:100
21839 Yes
47485 No

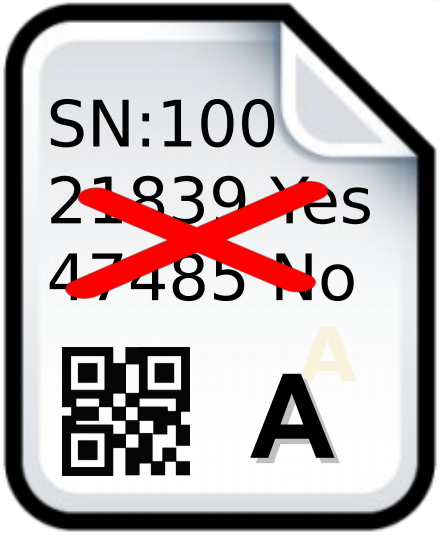
A

SN:100
34342 Yes
90641 No

B

End-to-end verifiability of Demos





34342

SN: 100			
Side A	21839		“No”
	47485		“Yes”
Side B	90641		
	34342		Voted

End-to-end verifiability of Demos

2)Defense against inconsistencies in vote-code and candidate correspondences:

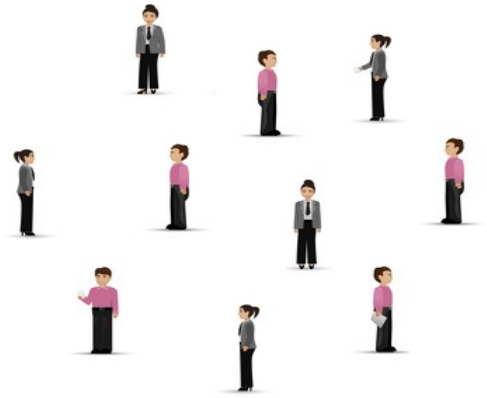
- The probability that the voter/auditor will detect the attack is 1/2.
- The probability that the adversary causes tally deviation ***x*** by launching these attacks is

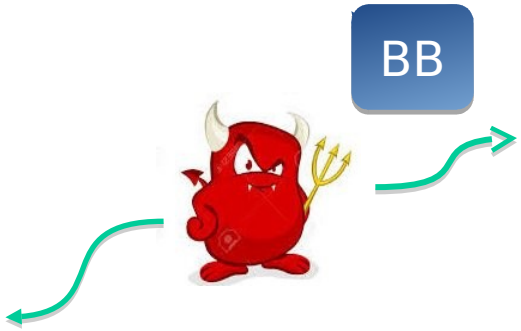
$$2^{-x}$$

End-to-end verifiability of Demos

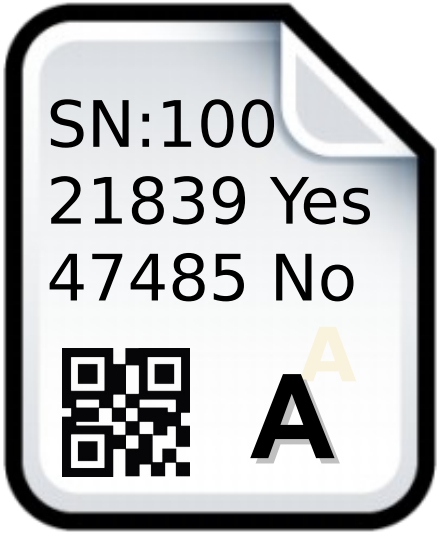
3)Defense against clash attacks:

End-to-end verifiability of Demos

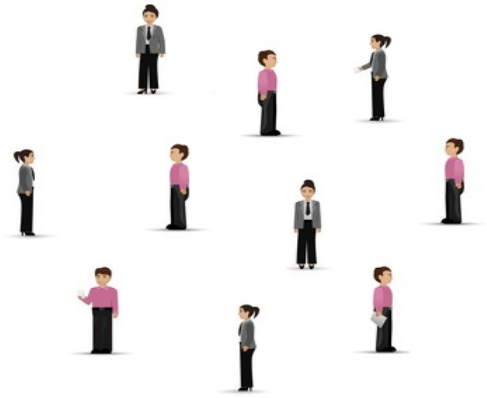


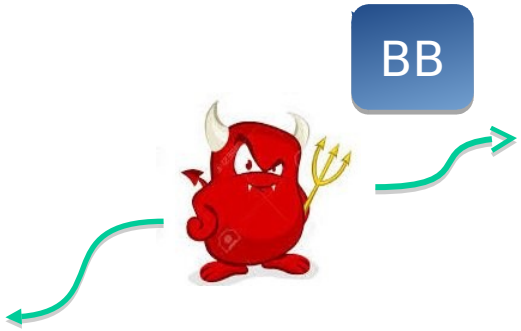


SN: 100		
Side A	21839	Initial Data
	47485	Initial Data
Side B	90641	Initial Data
	34342	Initial Data

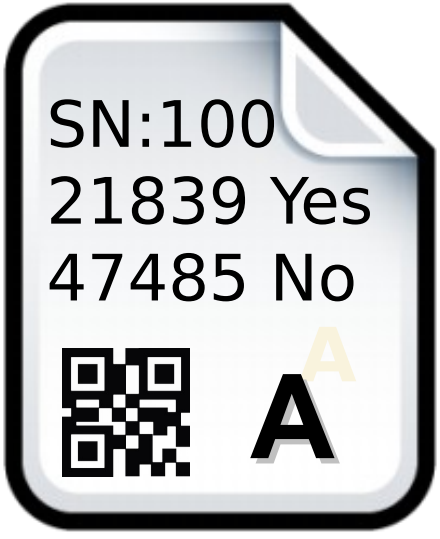


End-to-end verifiability of Demos



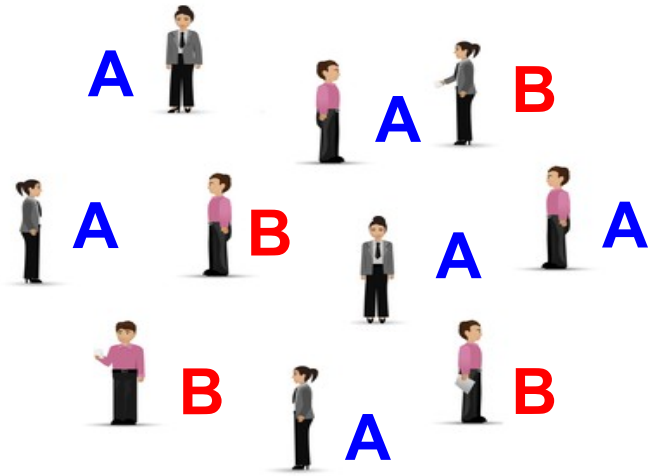


SN: 100		
Side A	21839	Initial Data
	47485	Initial Data
Side B	90641	Initial Data
	34342	Initial Data

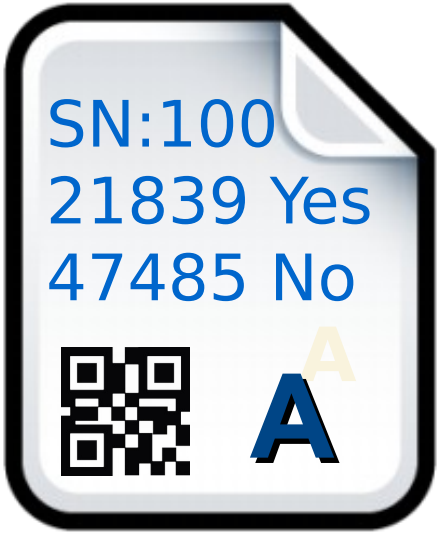


The adversary attacks by creating **unauditable positions** where it can place votes of its choice

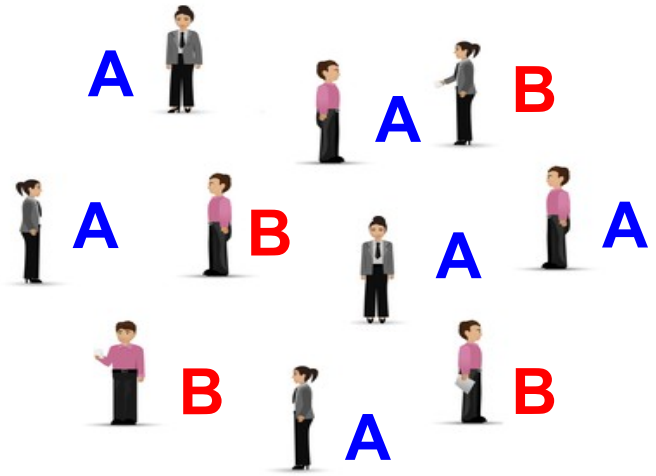
End-to-end verifiability of Demos



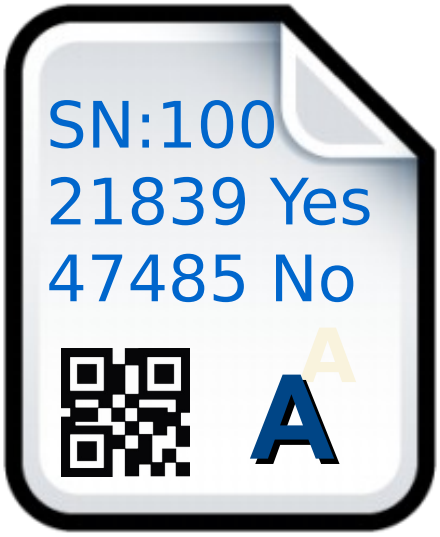
SN: 100			
Side A	21839		No
	47485		Yes
Side B	90641		No
	34342		Yes



End-to-end verifiability of Demos



SN: 100			
Side A	21839		No
	47485		Yes
Side B	90641		No
	34342		Yes



End-to-end verifiability of Demos

3)Defense against clash attacks:

- The probability that ***y*** clashed voters choose the same side to vote is

$$2^{-(y-1)}$$

- If this happens, the maximum tally deviation the adversary can achieve is ***y-1*** by exploiting **all unauditable positions**.

End-to-end verifiability of Demos

Theorem:

*Let **q** be the size of the modulo group, **n** be the number of voters and **m** be the number of candidates. Then, any adversary that does not corrupt at least **θ** voters cannot achieve tally deviation **d** with probability more than*

$$2^{-(\vartheta-(n/\log q+1)\log\log m)}+2^{-d}$$

Information theoretically!

Voter Privacy/Receipt Freeness

- Complexity Leveraging:
  - If the commitment scheme is hiding against 2^{λ^c} running time adversaries, then our e-voting system is voter private/receipt free for for at most $\lambda^{c'}$ corrupted voters, where $c' < c$ are constants.

  A Concrete example:

  #voters: n = 100 #candidates: m = 2

NIST Curve	Security	Max. Corrupted Voters
p192	96	82
p224	112	98

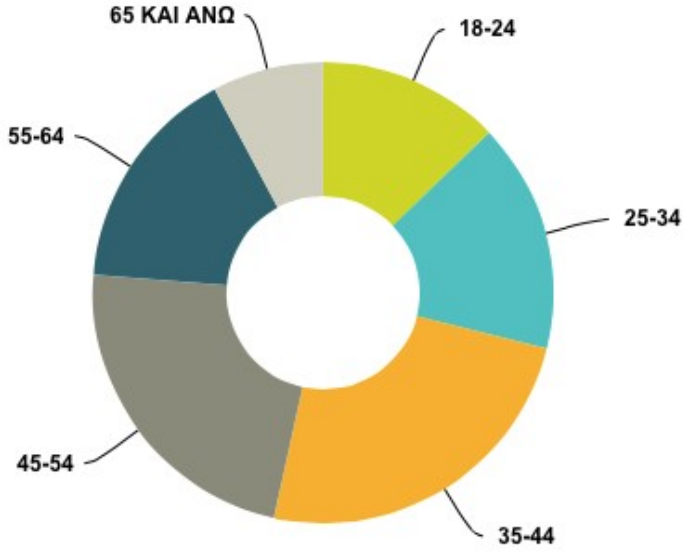
Implementation

- Web interface:
  - Django 1.6+
  - CSS: Bootstrap 3.3.0+
- Cryptography:
  - NIST curves
  - MIRACL (Multi-precision Integer and Rational Arithmetic)
  - Javascript: SJCL (Stanford Javascript Crypto Library)

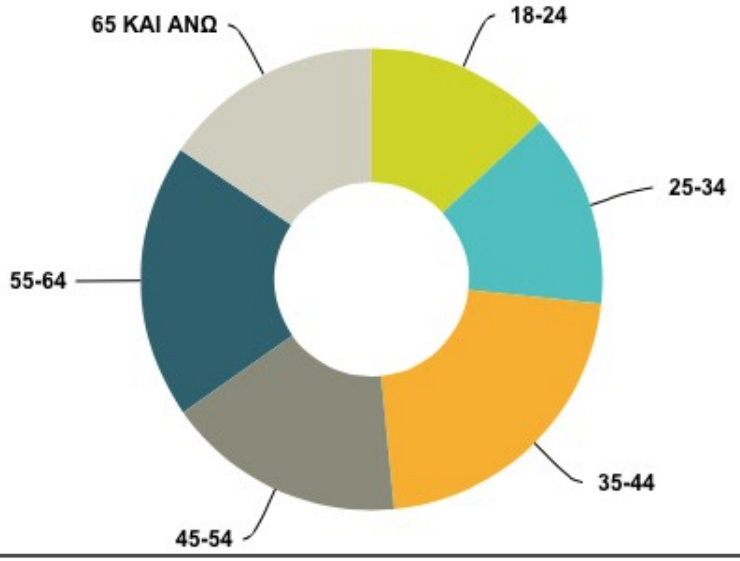
Experiments

- Our system is tested in the exit polls of

2014 Greek European parliament election (747 participants)



2015 Greek National election (400 participants)



Software Release

- Open source
- The Beta system is launched at University of Athens.
  - Available to all the student/faculty electronic elections.

Software Release

- Open source
- The Beta system is launched at University of Athens.
  - Available to all the student/faculty electronic elections.

Scheduled to release this summer

Coming Soon: a web-voting system for public use!

More information: www.demos-voting.org

End-to-end verifiability in the standard model

Thomas Zacharias & Bingsheng Zhang

joint work with **Aggelos Kiayias**

National & Kapodistrian University of Athens
Cryptography Security Lab – http://crypto.di.uoa.gr

www.demos-voting.org

May 28th, 2015

EUROCRYPT