CRYPTOGRAPHIC PROTOCOLS FOR TRANSPARENCY AND AUDITABILITY IN REMOTE ELECTRONIC VOTING SCHEMES

Spain Cryptography Days (SCD 2011)

Department of Mathematics Seminar

Sandra Guasch
Researcher
Sandra.Guasch@scytl.com

scytl
Secure Electronic Voting

Index

Background

- **Worldwide leader** in the research, development and implementation of **highly secure** solutions for **electoral modernization**.
- Founded as a spin-off from a leading University research group, author of the **first two European PhD theses** on security applied to electronic voting.
- **Patented** core technology based on groundbreaking cryptographic protocols developed over **16 years of research**, ensuring a transparent electoral process.
- Clear reference in the e-electoral market, having **advised international institutions** and **governmental agencies** and participated in breakthrough projects and studies.
- Scytl has provided its secure electronic voting technology to **14 out of 16 countries** that carry out binding elections using remote electronic voting.
- Efficient and reliable processes **certified ISO 9001-2000**

COUNCIL CONSEIL OF EUROPE DE L'EUROPE

NIST National Institute of Standards and Technology

OSCE

EUROPEAN COMMISSION DIRECTORATE GENERAL Joint Research Centre

REGISTERED COMPANY ISO 9001:2000

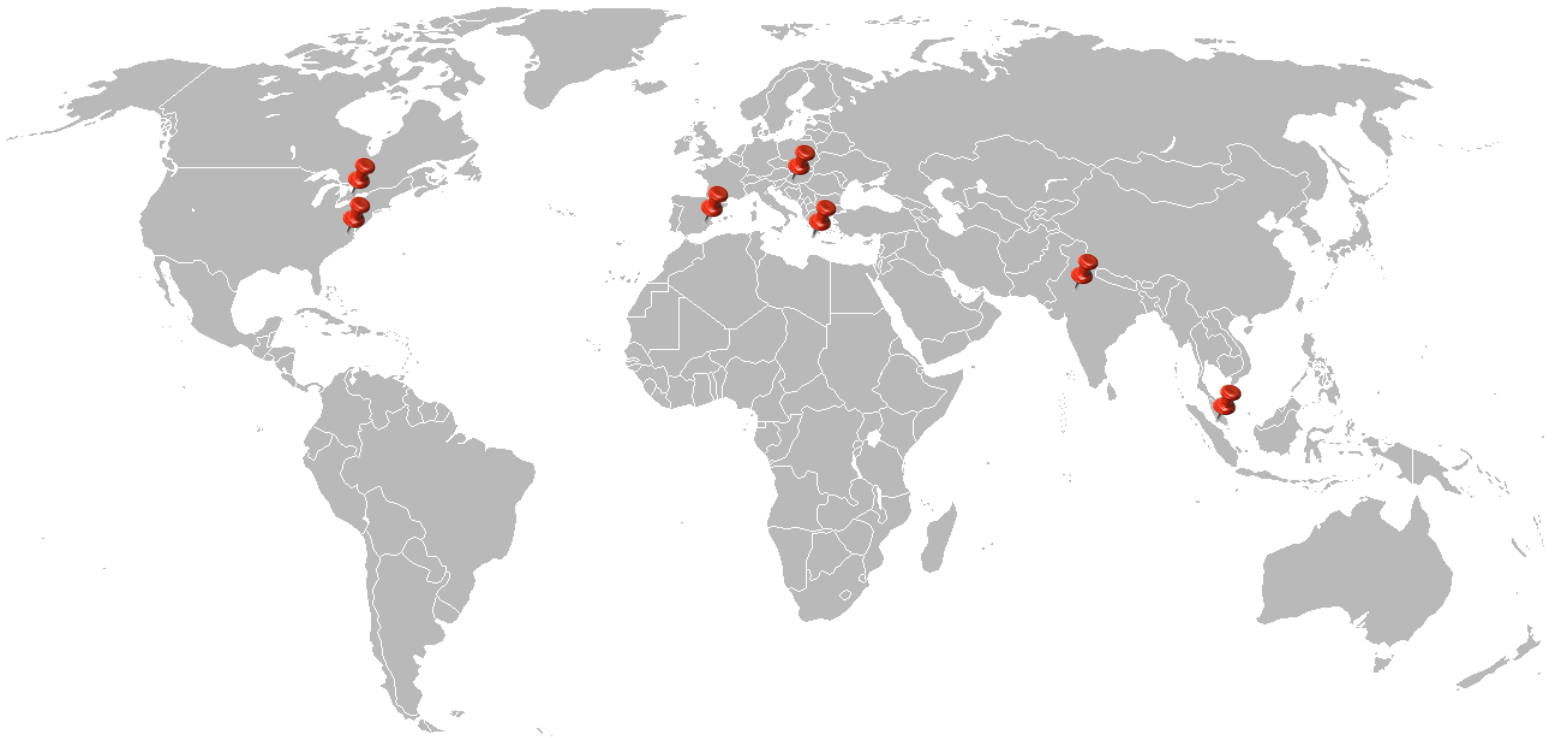
scytl Secure Electronic Voting

International Awards

Scytl has received multiple **international awards**, including:

- ***ICT Prize***, granted by the European Commission.
- ***European Venture Contest Award***, granted by the European Association of Venture Capital.
- ***Best Case Label***, granted by the European Commission.
- ***Leader de l'ITech-Economie***, granted by the French Chambers of Commerce.
- ***Global Innovator Award***, granted by The Guidewire Group.
- ***Red Herring 100***, granted *Red Herring Magazine.*
- ***Premi Ciutat de Barcelona***, granted by the City of Barcelona.
- ***ebiz egovernment award*****,** granted by the Austrian chancellery.

Worldwide Offices

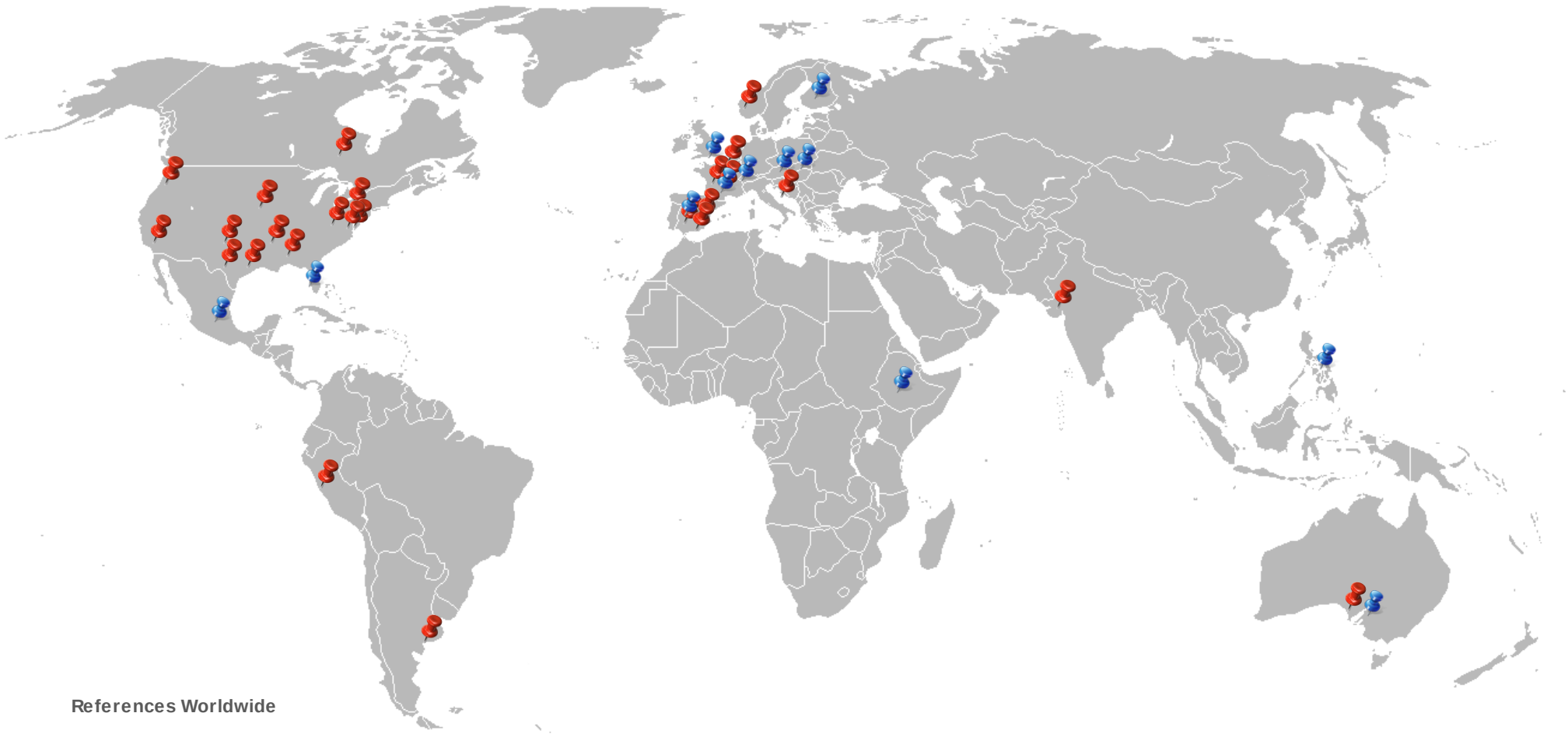


Headquarters
Barcelona (Spain)

Regional Offices
Toronto (Canada)
Washington, D.C. (USA)
Bratislava (Slovakia)
Athens (Greece)
New Delhi (India)
Singapore (Singapore)

About Scytl

Worldwide References



References Worldwide

- Province of Ontario (Canada)
- State of New York (USA)
- State of Alabama (USA)
- US Dep. of Defense (USA)
- DC Board of Elections (USA)
- State of Florida (USA)
- Parliament of Nuevo Leon (Mexico)
- Peruvian Office of Elections (Peru)
- Government of Mendoza (Argentina)
- Ministry of Justice (UK)
- City of Madrid (Spain)
- Spanish National Police (Spain)
- Catalan Universities (Catalonia)
- Catalan Government (Catalonia)
- Parliament of Catalonia (Catalonia)
- Catalan Force Police (Catalonia)
- Ministry of Higher Education (France)
- Ministry of Foreign Affairs (France)
- Ministry of Science and Research (Austria)
- Canton of Neuchatel (Switzerland)
- Ministry of Justice (Finland)
- Tradenomiliitto (Finland)
- Ministry of Local Government (Norway)
- BiH Central Election Commission (Bosnia and Hercegovina)
- European Union
- African Union Commission (Ethiopia)
- State of Gujarat (India)
- Commission on Elections (Philippines)
- Victorian Electoral Commission (Asutralia)

scytl
Secure Electronic Voting

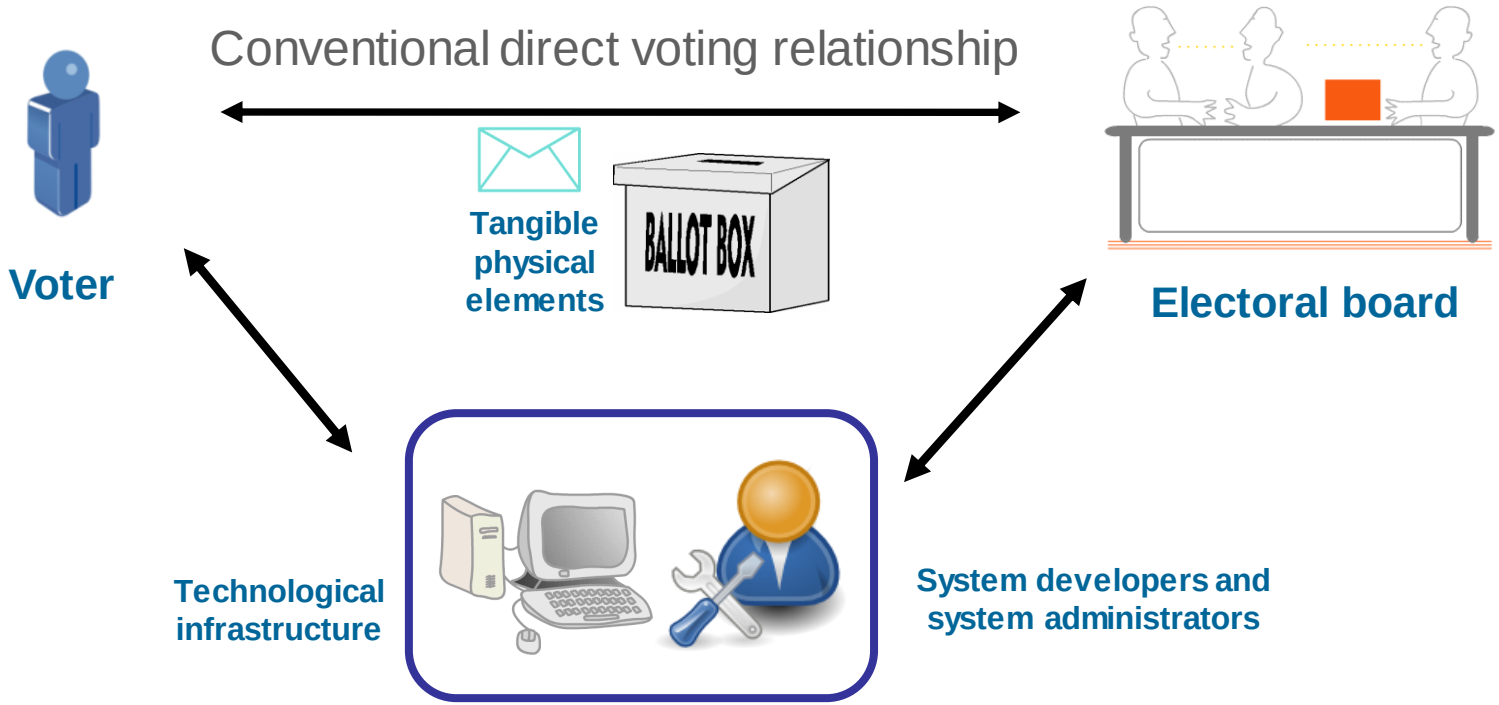
Index

Types

- Implements the traditional voting process by electronic means: the voter intent is captured and stored electronically

- Types of e-voting:
  - Poll-site:
    - Votes are cast only from polling stations
    - Voter identification follows current traditional methods
    - The proper use of computing technology could enable voters to use the polling station of their choice, in real time
  - Remote:
    - Remote electronic voting is a particular case of electronic voting in which digital votes are sent through a communication network from the voter's location to a remote digital urn
    - Analogous with postal voting
    - Two types of remote electronic voting:
      - Kiosk-based: voting from supervised locations
      - "Pyjama voting": voting anywhere, even from home

Change of voting paradigm



Electronic voting creates a new indirect voting relationship that brings **new security risks** that reduce the trustworthiness of the electoral process.

Security Risks

- Four main sources of security risks emerge due to the technical infrastructure interposed between the voter and the electoral board:
  - The digital (virtual) nature of the ballots
    - Ballots may be added, deleted or otherwise manipulated
    - Voters' privacy may be compromised on a large scale
  - The complexity of the systems used
    - Electronic equipment may malfunction
    - Software may contain programming errors
  - The lack of transparency of the systems used
    - The technical infrastructure is not easily audited
  - The introduction of people with privileges on the systems used
    - New players enter the scene
- These risks can be mitigated using adequate cryptographic voting protocols

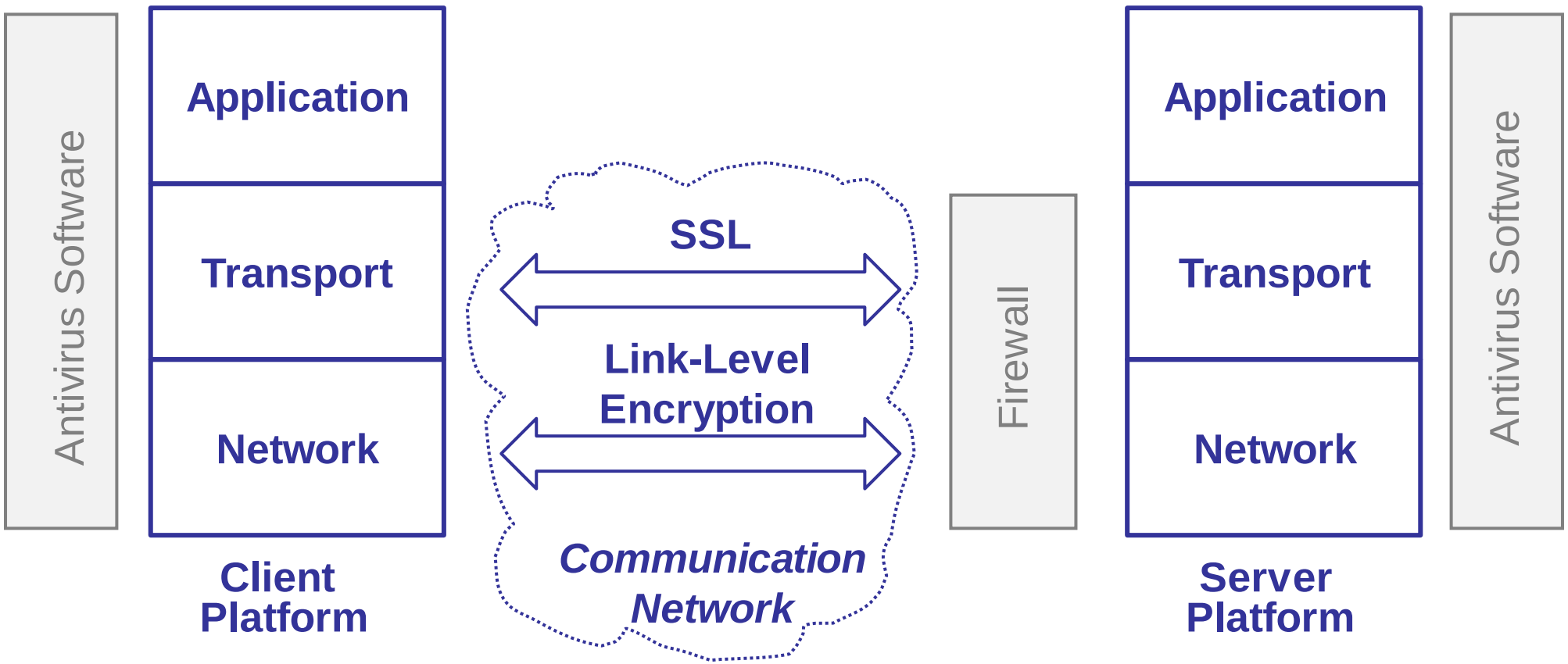
Security Requirements

- Any e-voting application must fulfill the following security requirements:
  - Guarantee voters' privacy while ensuring their proper identification
    - Voters should be strongly authenticated
    - Only eligible voters are able to vote
    - Privacy of voters must be preserved
  - Protect the digital ballot box to ensure
    - The secrecy of intermediate results
    - The integrity of the ballots cast
    - The impossibility of adding bogus ballots
  - Enable verifiability, while preventing coercion and vote-buying

> "one concern with Internet voting is that the voter has absolutely no control over the vote cast once it leaves his own computer system, and he cannot check whether it has been subverted on the way to the count"
>
> *ODPM meeting with Dr. Rebecca Mercuri*
> *17th October 2002*

Index

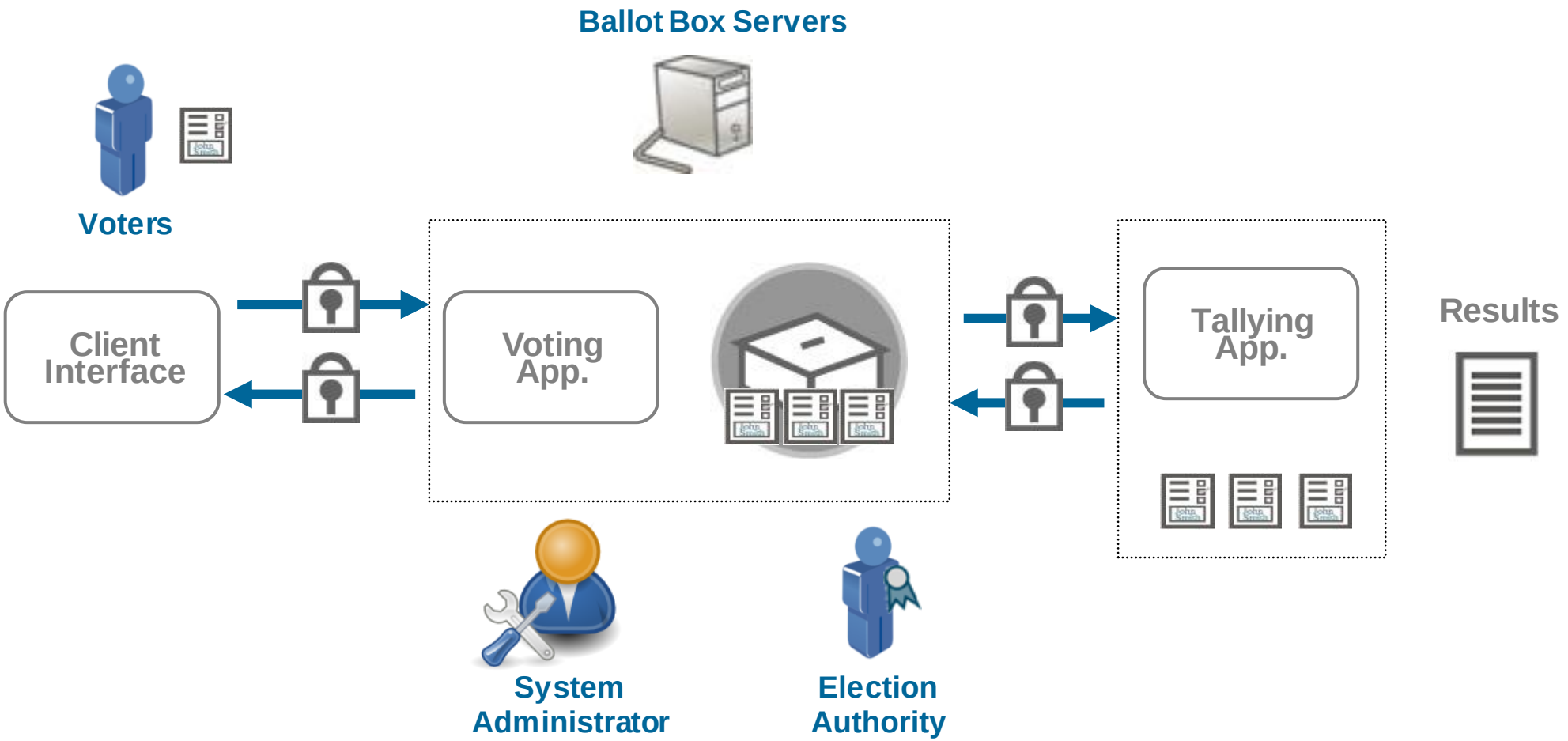
Electronic voting with standard security



Conventional digital security measures are completely generic, not addressing an application's specific needs.

scytl Secure Electronic Voting

Electronic voting with standard security



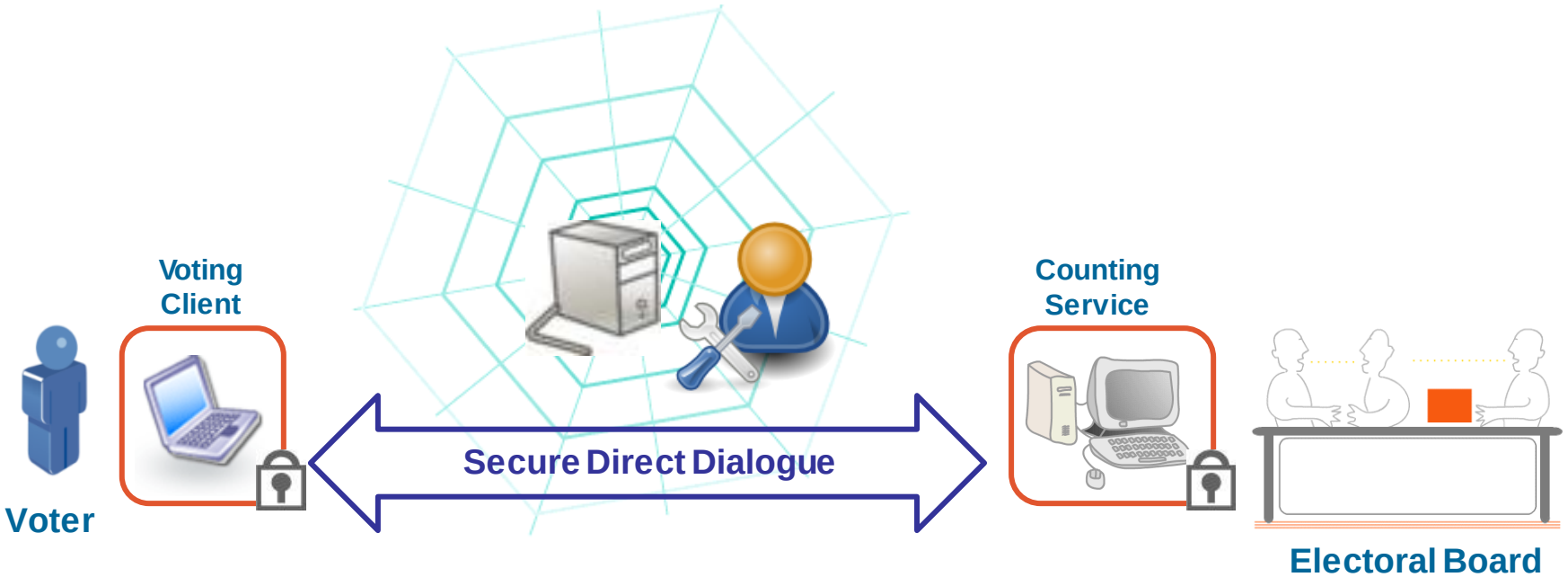
Standard security measures fail to cover the specific security needs of an electronic voting platform.

scytl Secure Electronic Voting

Fulfillment of Security Requirements

- **Voter privacy compromise**
  - Privileged actors can see the votes in the servers
- **Inaccurate auditability**
  - Logs and information can be easy to manipulate
- **Vote tampering**
  - Votes can be manipulated (no digital signature)
- **Vote deletion**
  - Votes can be easily eliminated (no verifiability)
- **Voter coercion and vote buying**
  - Privileged actors can check the vote contents in the ballot box
- **Unauthorized voters casting votes**
  - There is no strong authentication (digital certificates)
- **Voter impersonation / Ballot stuffing**
  - Authenticity /integrity of votes is not protected
- **Intermediate results**
  - Votes can be counted by privileged actors in the servers (no encryption)

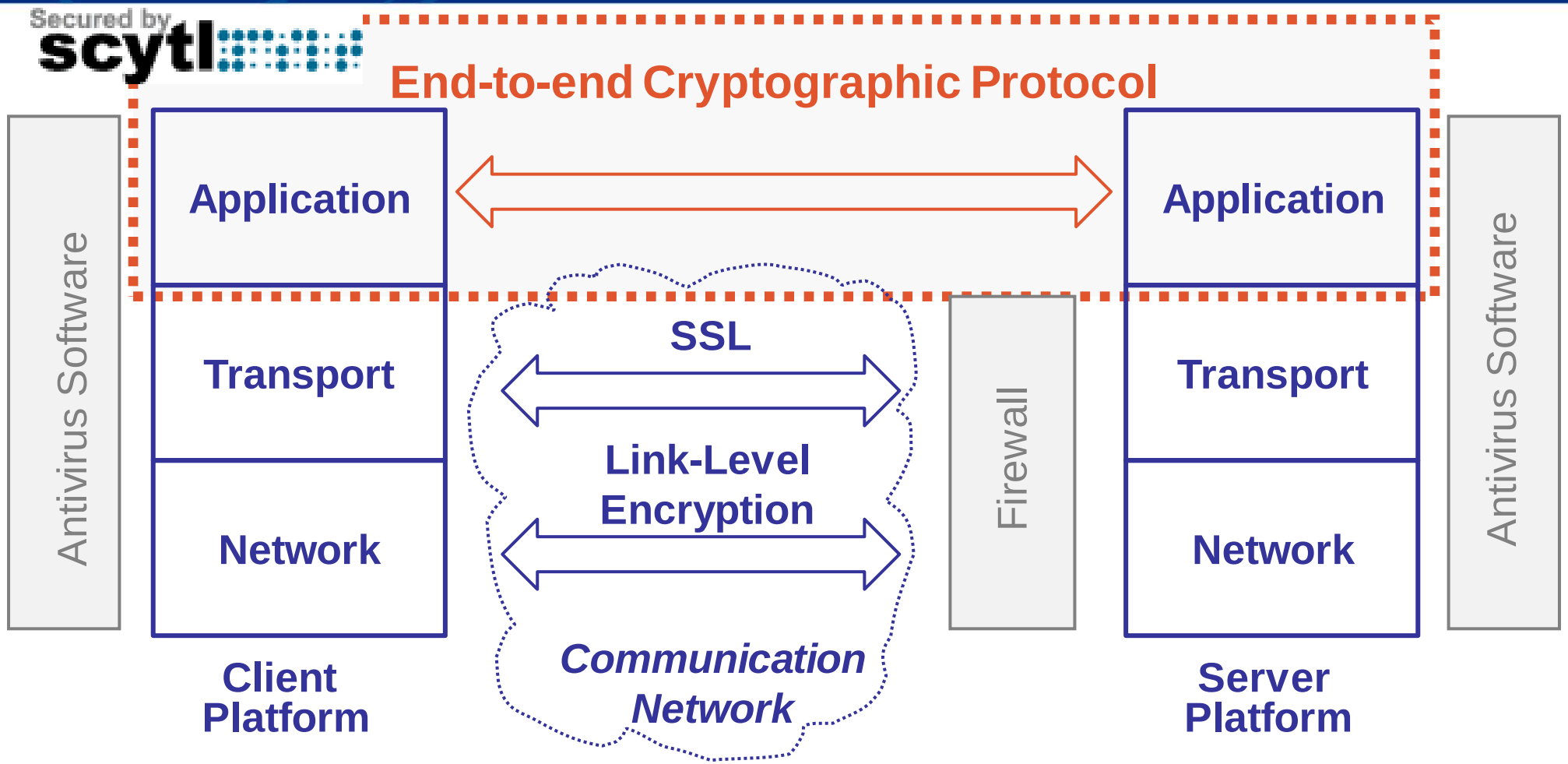
Electronic voting with end-to-end security



The main objective of a secure architecture is to allow a **secure direct dialog** between voter and Electoral Board, protecting them from attacks comming from the IT infrastructe between them.

scytl Secure Electronic Voting

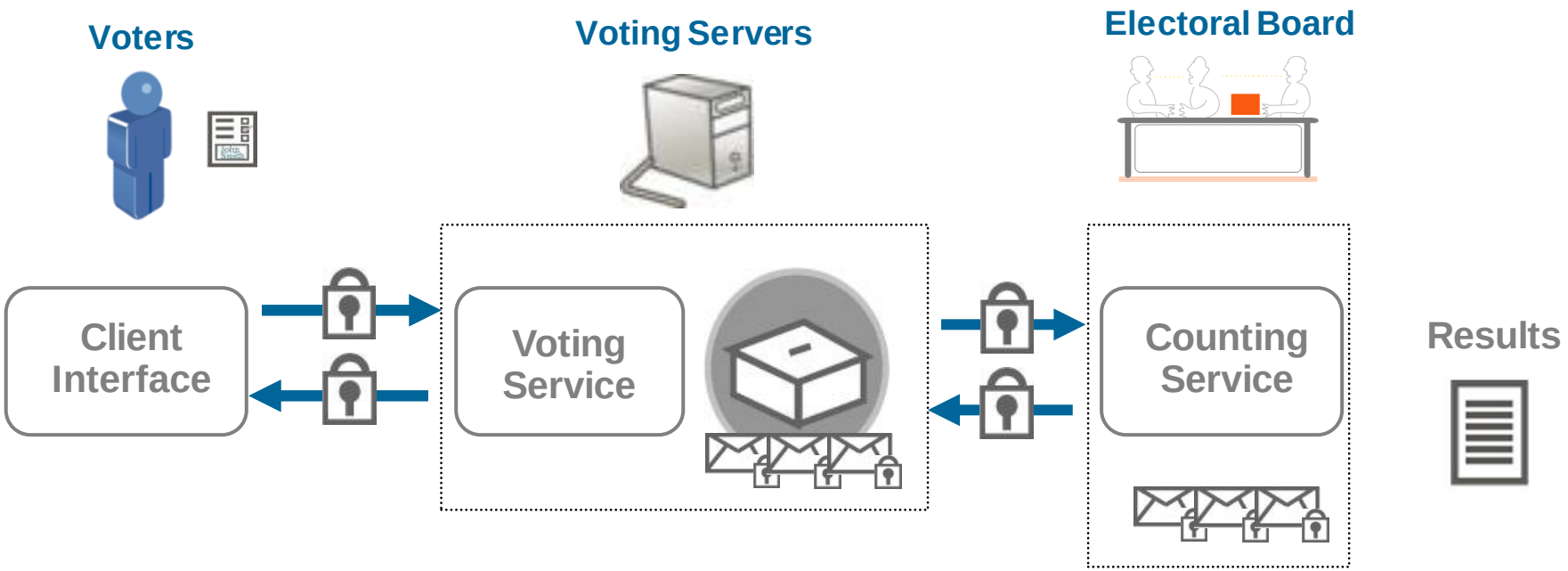
Electronic voting with end-to-end security



Scytl's application-level cryptography addresses the specific security requirements of e-voting.

scytl Secure Electronic Voting

Electronic voting with end-to-end security



- **Votes are encrypted** using the Electoral Board private key.
  - Only Electoral Board can decrypt the votes.
- **Encrypted votes are digitally signed** using voters' private key.
  - Votes cannot be manipulated or re-used after being cast.

scytl Secure Electronic Voting

Fulfillment of Security Requirements

Electronic voting with standard security

- Vote authenticity ✗
- Strong authentication of voters ✗
- Voters privacy ✗
- Accuracy of election results ✗
- Secrecy of intermediate results ✗
- Verifiability ✗
- Prevention of coercion and vote-selling ✗

Electronic voting with end-to-end security

- Vote authenticity ✓
- Strong authentication of voters ✓
- Voters privacy !
- Accuracy of election results !
- Secrecy of intermediate results ✓
- Verifiability ✗
- Prevention of coercion and vote-selling ✓

Some advanced cryptographic protocols are used to protect voter privacy, such as:

- Pollsterless
- Two-agencies model
- Mixnets
- Homomorphic tally

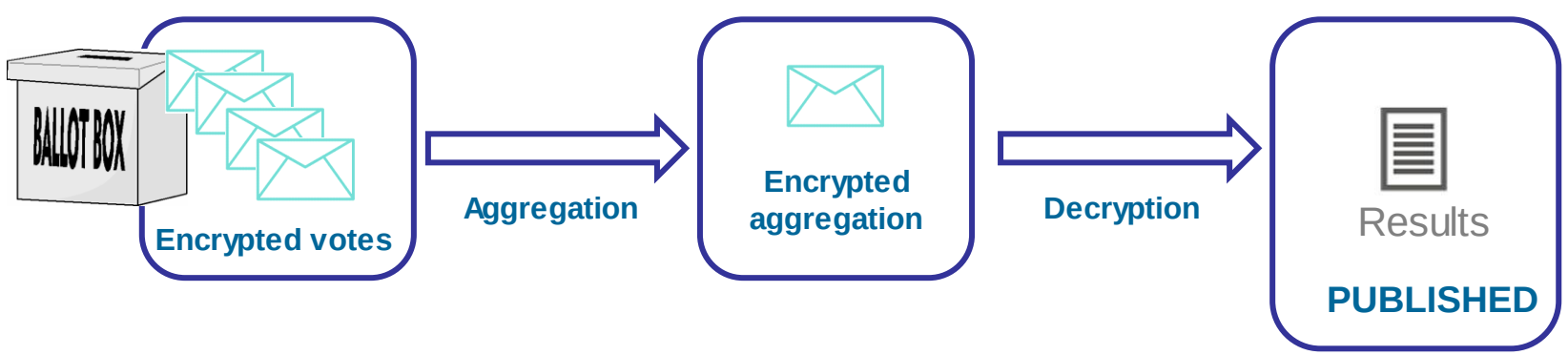
Accuracy has to do with verifiability.

Homomorphic tally (1/5)

- Votes are encrypted by voters using a cryptographic algorithm with homomorphic properties (e.g., ElGamal).

- Votes are digitally signed by voters before being cast.

- Encrypted votes are operated. The result of this operation is then decrypted, instead of the individual votes .
  - The decryption result is the operation (homomorphic properties) of the plaintext votes.
    - For instance, the number of the times each voting option has been selected.



Homomorphic tally (2/5)

- In homomorphic encryption algorithms the result of operating two encrypted messages is the encryption of the result of operating these messages:
  - $P(m_1) \varnothing P(m_2) = P(m_1 o m_2)$
- In the case of ElGamal the addition of two encrypted votes yields an encryption of the sum of the votes:
  - $E(v_1) \cdot E(v_2) = E(v_1 + v_2)$
- In homomorphic tally, the addition of the encrypted votes returns the encryption of the sum of the votes of each candidate (i.e., the encryption of the result).

Homomorphic tally (3/5)

- Using **ElGamal** as the encryption algorithm, we have the following components:

g generator of Z_p^*	p large prime p=2q+1
private key: x	x random number in Z_p
public key: (h, g, p)	$h = g^x \bmod p$
message: m	

- Encryption: $c = (a, b) = (m \cdot h^w, g^w)$, where w is a random number in Z_p
- Decryption: $m = a \cdot b^{-x} = m \cdot h^w / (g^w)^x = m \cdot h^w / h^w$

Homomorphic tally (4/5)

- Each voting option has a binary value *v* equal to 1 if the option has been selected or 0 if it hasn't.

Encrypted vote: $c = (\lambda^{v} \cdot h^{w}, g^{w})$ $\quad v=\{1,0\}$

- If two votes c' and c'' encrypted with the same public key are multiplied:

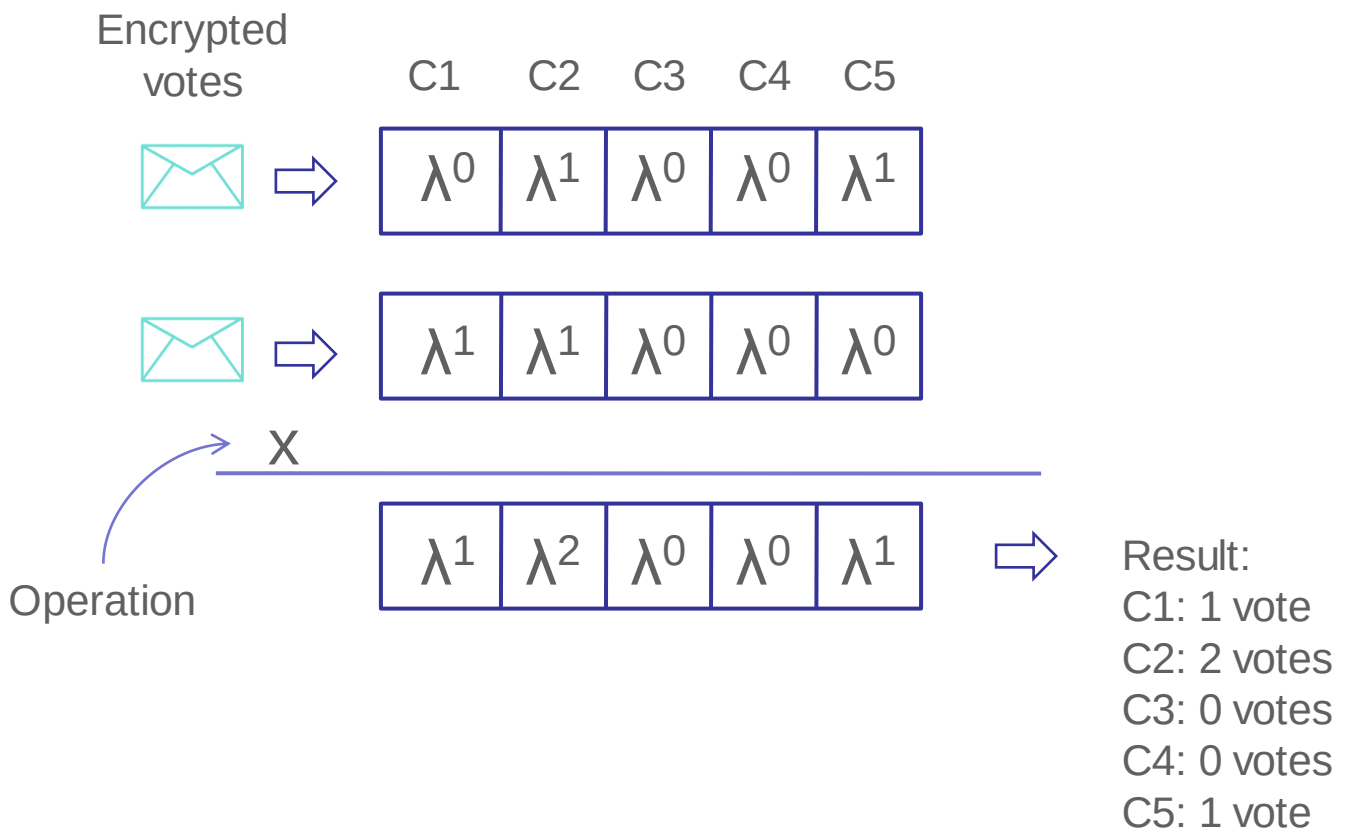
$$c' = (a', b') = (\lambda^{v'} \cdot h^{w'}, g^{w'})$$
$$c'' = (a'', b'') = (\lambda^{v''} \cdot h^{w''}, g^{w''})$$

$$c' \cdot c'' = (a', b') \cdot (a'', b'') = (\lambda^{v'} \cdot h^{w'}, g^{w'}) \cdot (\lambda^{v''} \cdot h^{w''}, g^{w''}) = (\lambda^{v'+v''} \cdot h^{w'+w''}, g^{w'+w''})$$

- Decryption $(c' \cdot c'') = \lambda^{v'+v''} \rightarrow \log_{\lambda}( ) \rightarrow v'+v''$

Number of times a voting option has been selected.

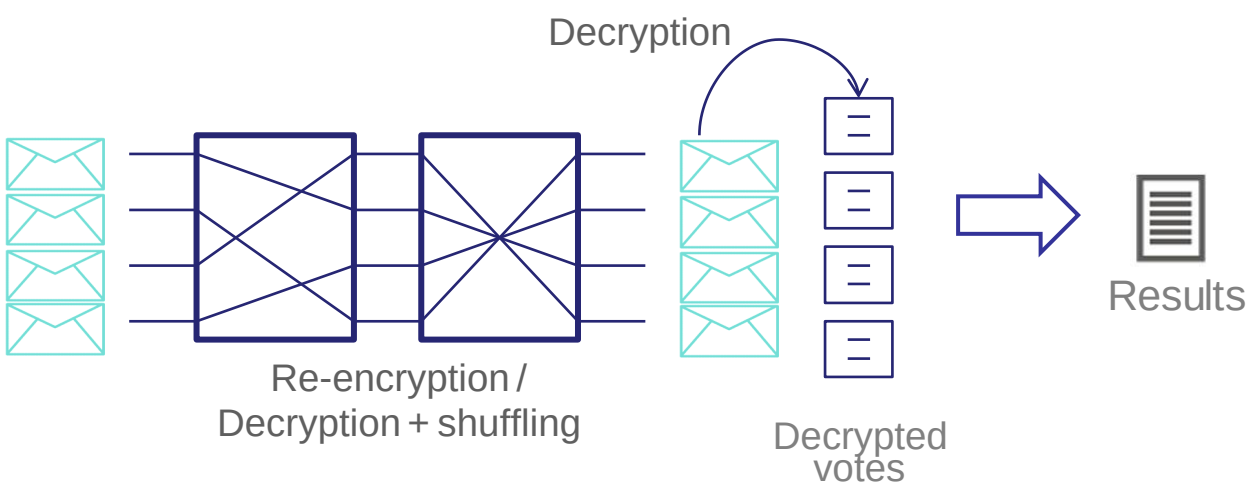
Homomorphic tally (5/5)



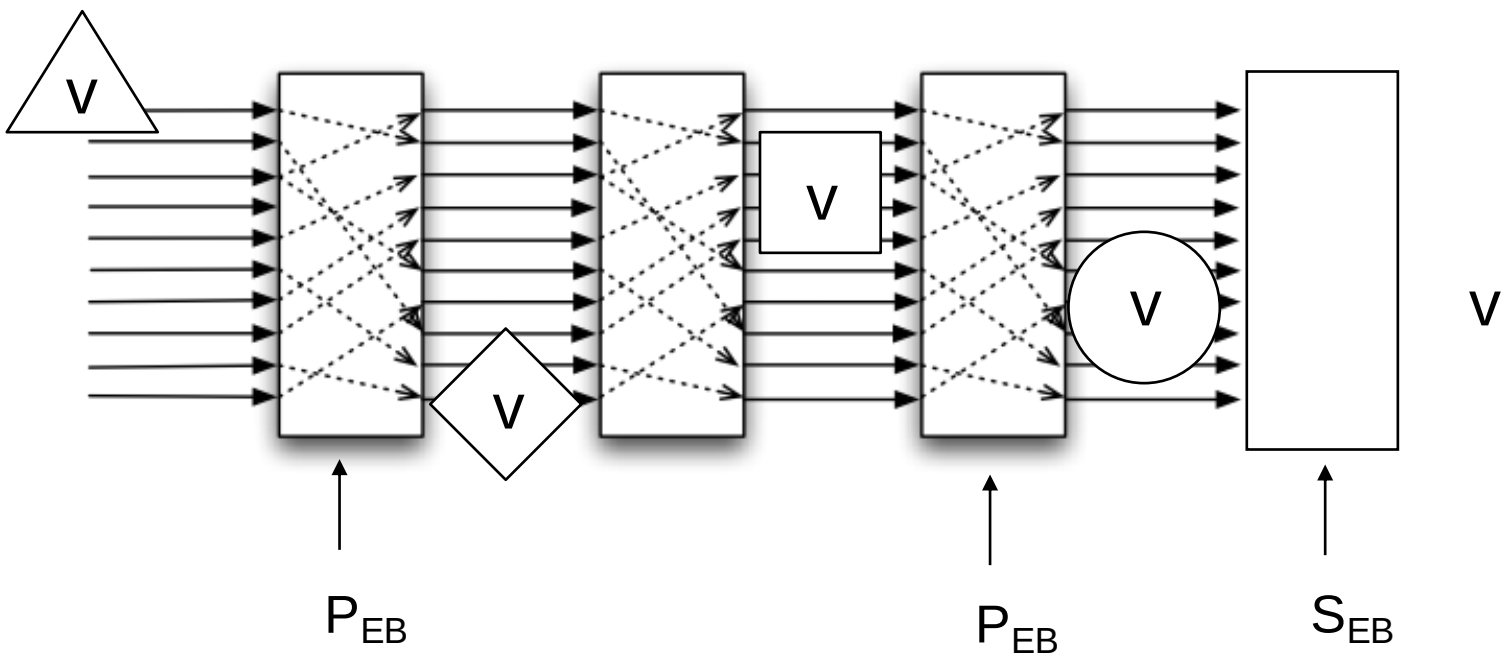
Mix-nets (1/2)

- Several nodes shuffle and re-encrypt/decrypt the votes for breaking the correlation between the original input order and the output one.
  - The shuffled and re-encrypted/decrypted vote output from one node is used as the input of another one.
  - The vote contents are obtained (decrypted) at the last node.

- Re-encryption:
  - $c = (m \cdot h^w, g^w)$
  - $c' = c \cdot (1 \cdot h^{w'}, g^{w'}) = (m \cdot h^w, g^w) \cdot (1 \cdot h^{w'}, g^{w'}) = (m \cdot h^{w+w'}, g^{w+w'})$

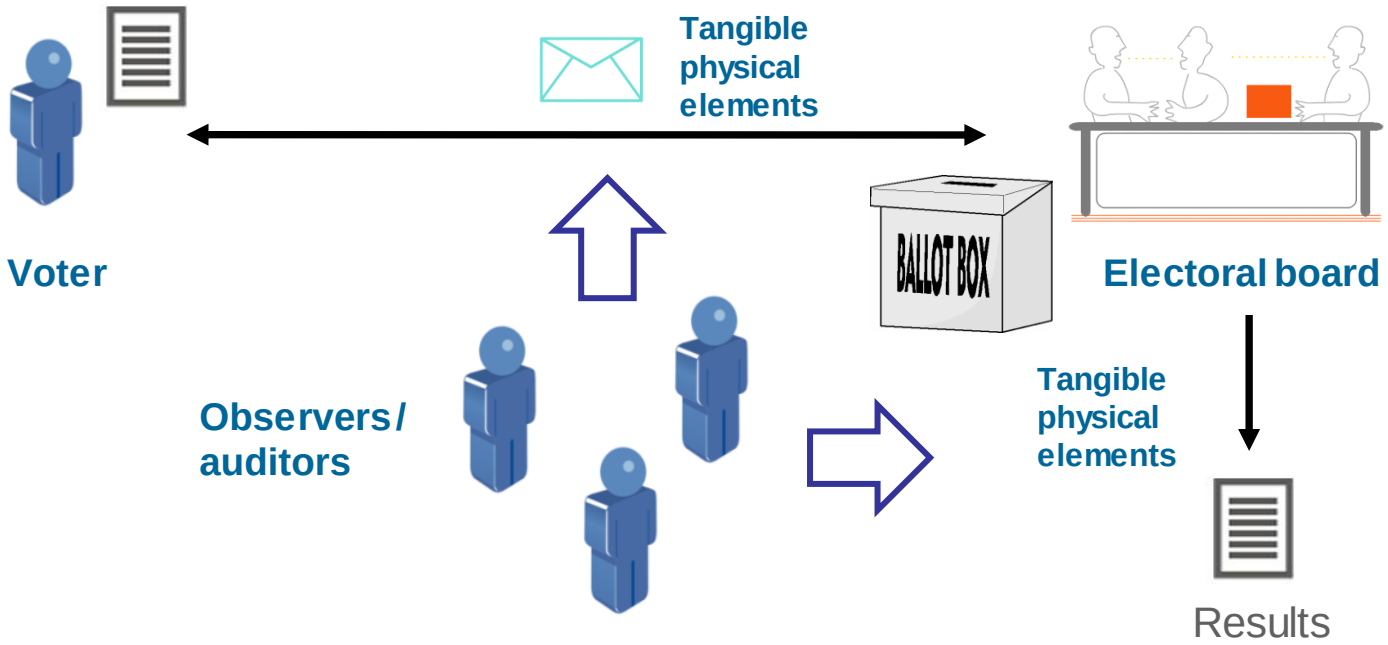


Mix-nets (2/2)



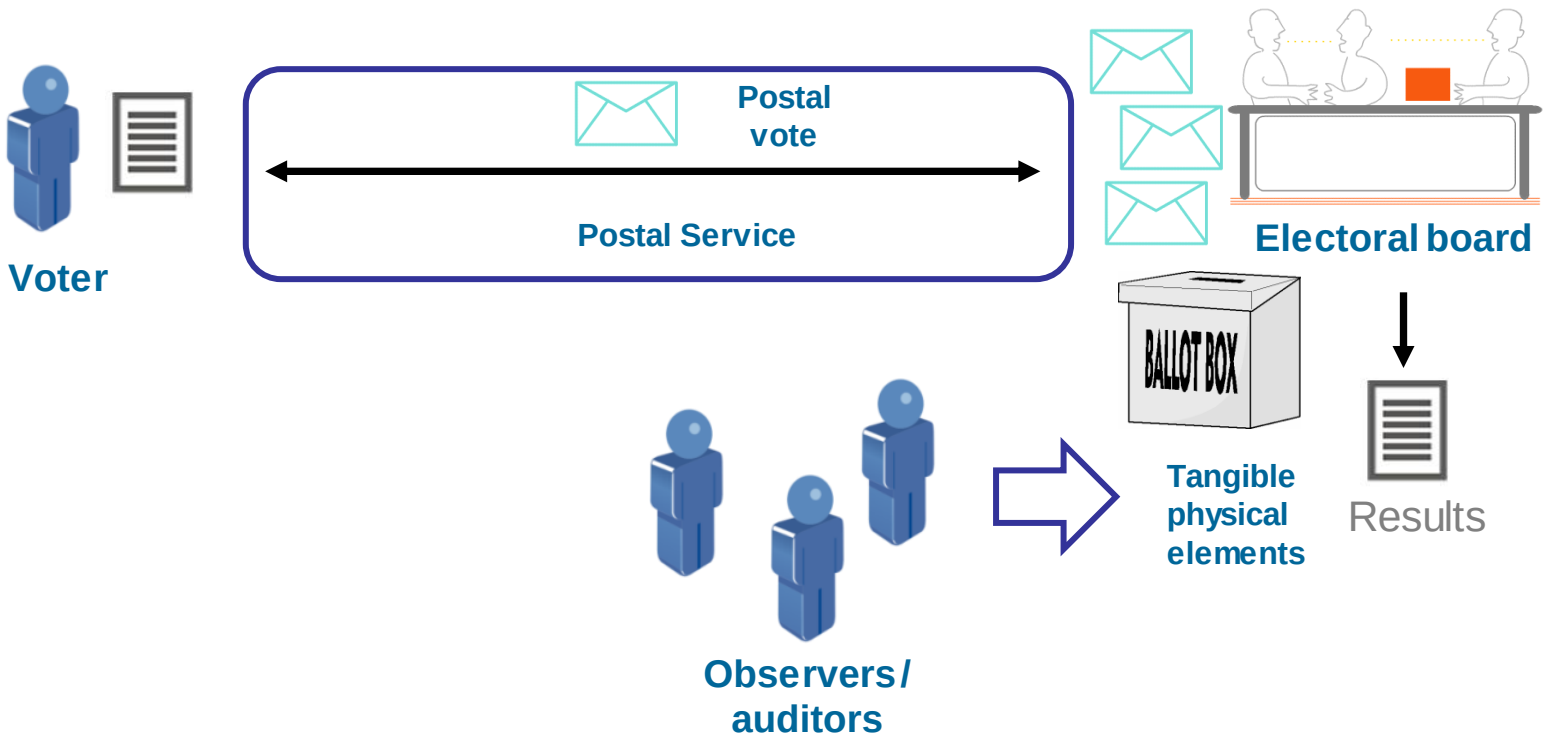
Index

Auditability in traditional voting



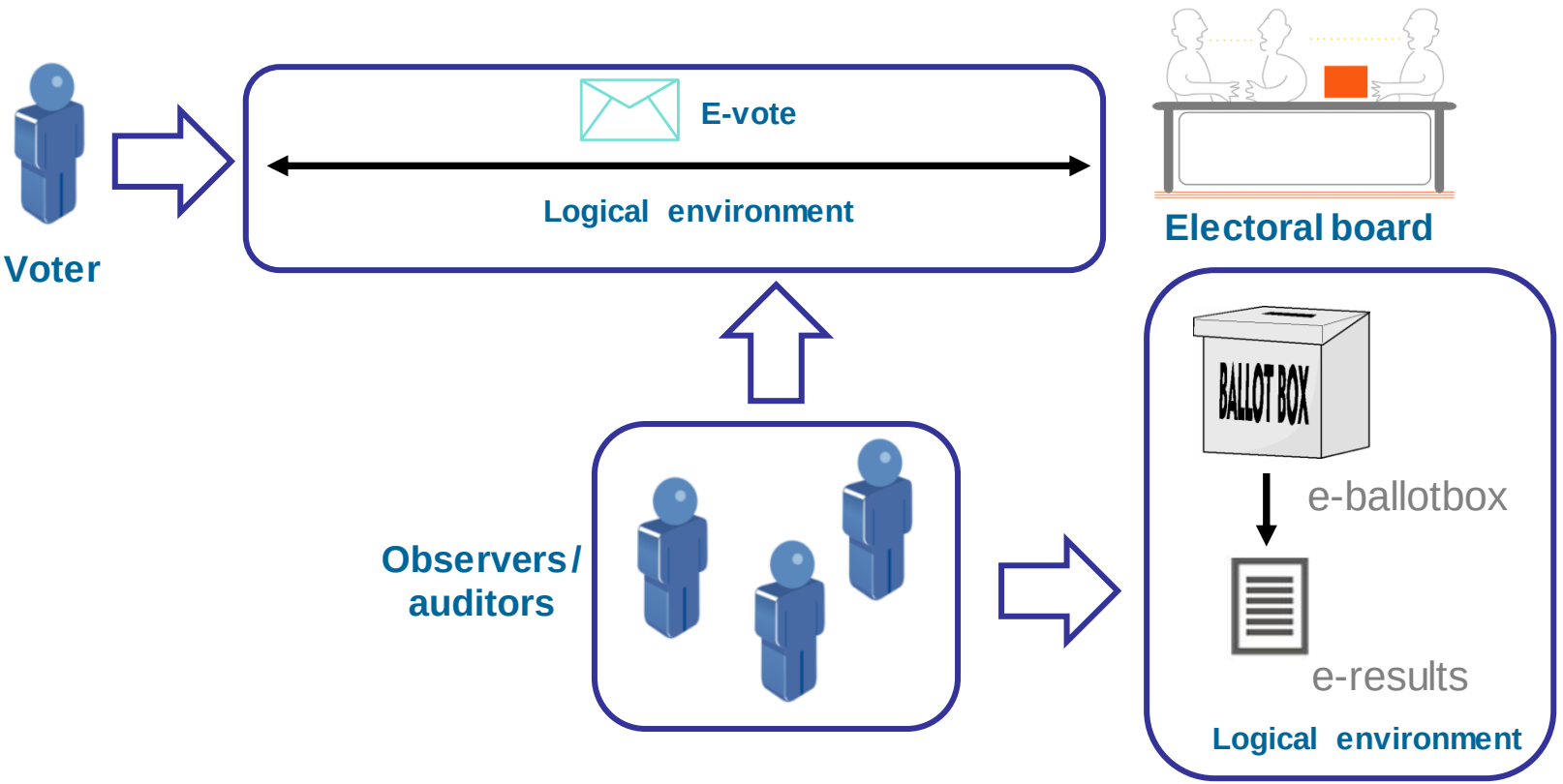
- Votes and processes (e.g., counting) are based on tangible elements:
  - Audit can be done by voters, observers and independent auditors by human means when the processes are carried out.
  - Observers can monitor the behavior of other observers to detect any fraud practices.

Auditability in postal voting



- The audit of the vote delivery process and storage in the ballot box is difficult if not impossible:
  - Voters only can verify the selection they made but cannot verify if the same vote is received by the Electoral Board.
  - Observers can audit the opening of the votes stored in the Ballot Box, but they have no access to the vote delivery process and have limited access to the process of storing the postal votes in the Ballot Box.

Auditability in e-voting



- Votes and processes are happening in a logical dimension:
  - Audit cannot be done by human means.
  - Difficult to monitor the behavior of other observers.

Index

Types of verifiability

- Individual verifiability
  - Focused on the voter: only the voter that casts the vote is able to implement the verification process.
  - Audit of the correct encoding of the voting options, correct vote reception, and presence of the vote on the final count.
  - Security concerns: preservation of voter privacy and prevention of vote selling/coercion practices.

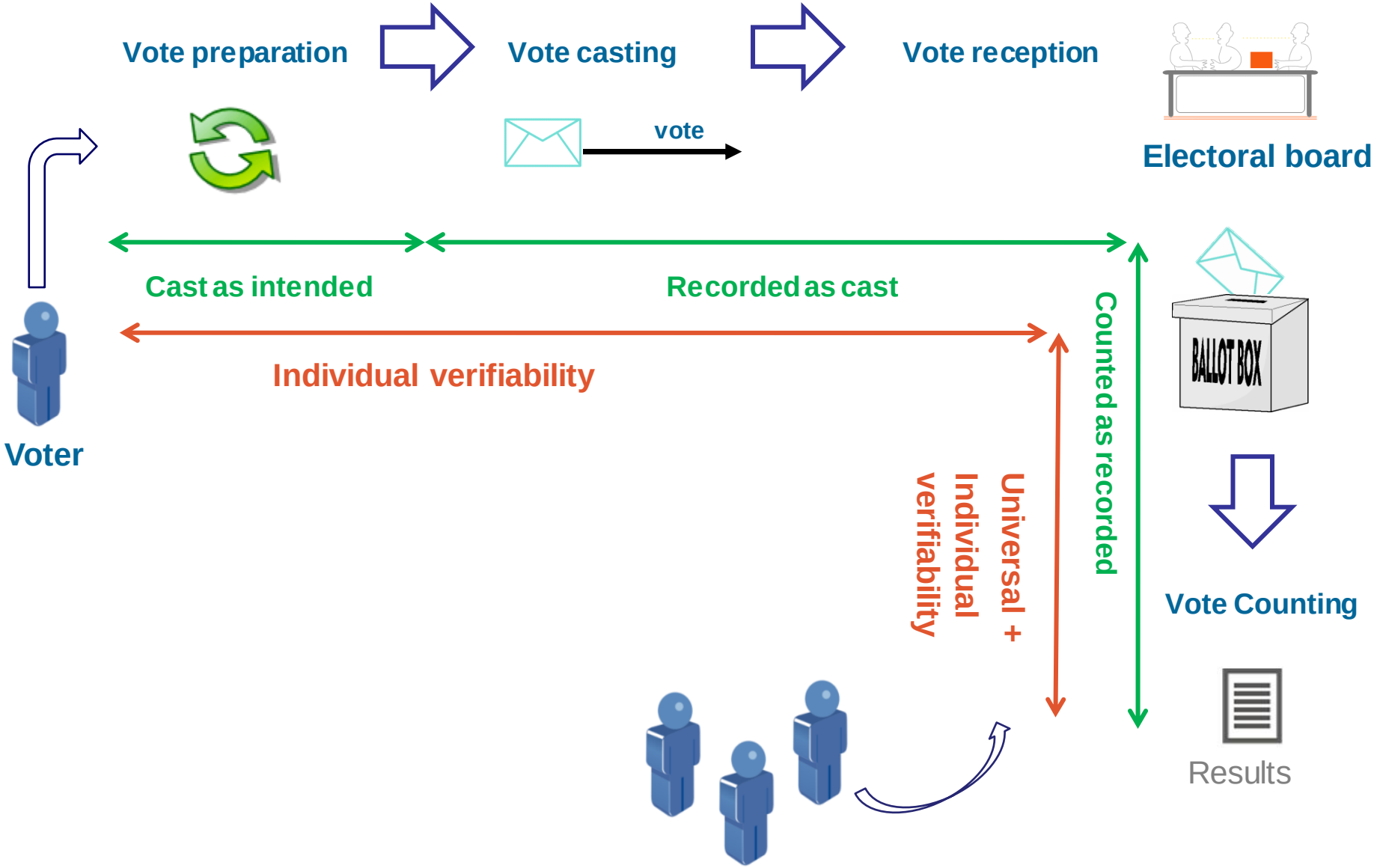
  - Cast as intended (Karlof et al.)
    - Voters can verify that their cast votes really represent their voter intent.

- Universal verifiability
  - Focused on the public, not restricted to voters.
  - Audit of the correct vote counting.
  - Security concerns: preservation of voter privacy.

  - Counted as cast (Karlof et al.)
    - Any observer can verify that the final tally is an accurate count of the ballots cast.

End-to End verifiability (Benaloh'06)= cast as intended + counted as cast

Verifiability and election processes



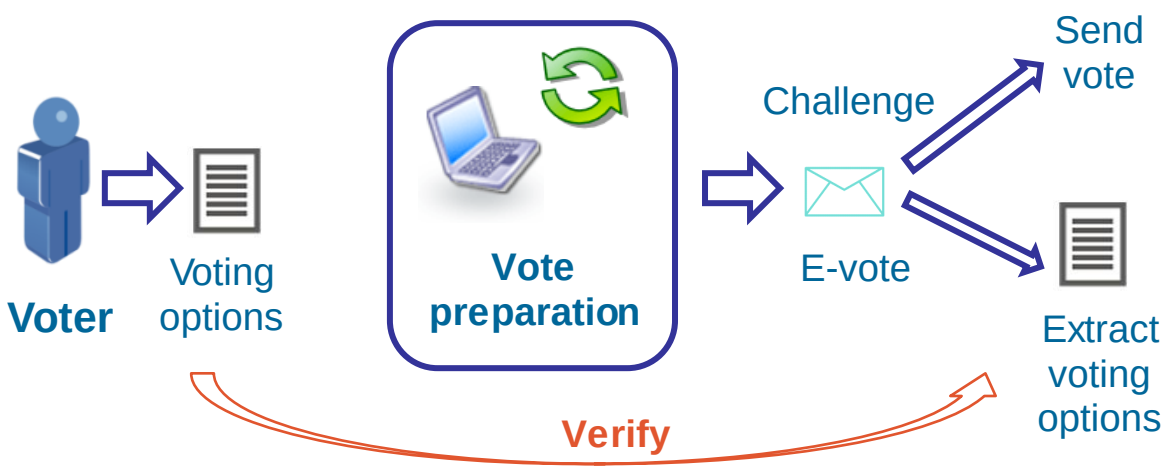
scytl Secure Electronic Voting

Index

- About Scytl
- Introduction to Electronic Voting
- Internet voting cryptography
- Auditability in e-voting
- Types of verifiability
- Verifiability methods for e-voting

Vote encryption challenge (1/2)

- The vote is encrypted and the application commits to the encryption (e.g., showing a hash of the encrypted vote).
- The voter can challenge the application to verify the proper encryption of the vote before casting it:
  - Challenge: voter asks the application for showing the secret random parameters used to encrypt the vote.
  - Verification: voter uses the random parameters and the encryption proof to verify if the encrypted vote contains her voter intent.
  - New encryption: the vote is encrypted again with new random parameters, and a new encryption proof is generated.
- Probabilistic verification.



Vote encryption challenge (2/2)

- Remember the **ElGamal** encryption algorithm:

g generator of Z_p^*	p large prime p=2q+1
private key: x	x random number in Z_p
public key: (h, g, p)	$h = g^x \bmod p$
message: m	

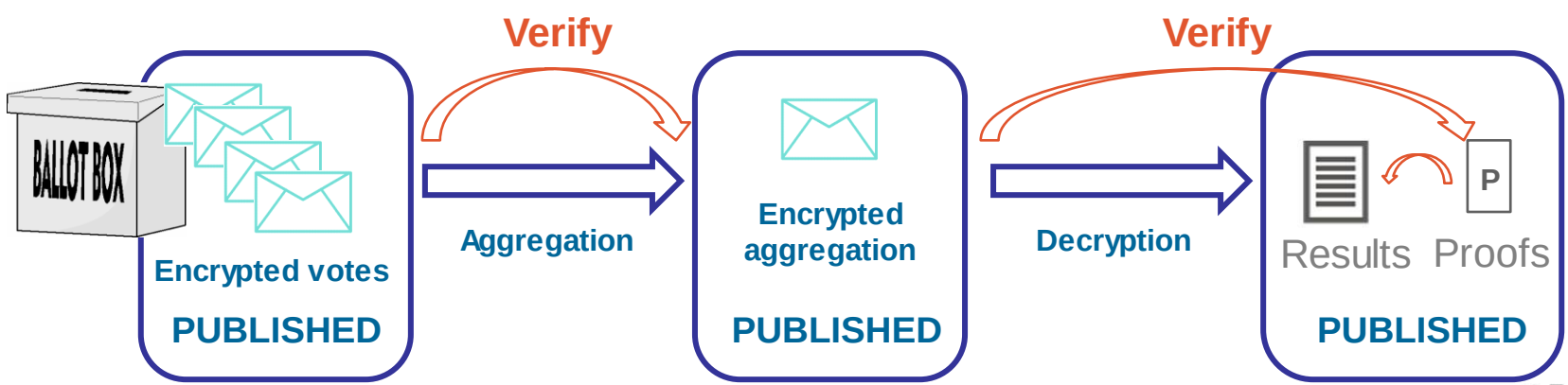
- Encryption: $c = (a, b) = (m \cdot h^w, g^w)$, where w is a random number in Z_p
- Decryption: $m = a \cdot b^{-x} = m \cdot h^w / (g^w)^x = m \cdot h^w / h^w$

- Verification in vote encryption challenge:
  - Software commits to the encryption: H(c)
  - Secret randomness is shown to the voter: w.
  - Voter can generate the encryption again and check the commitment:
    - $c' = (m \cdot h^w, g^w) \rightarrow H(c') == H(c)?$

Homomorphic tally

- **Zero Knowledge Proof of correct decryption**, based on the equality of discrete logarithms:
  - Remember $c = (a, b) = (m \cdot h^w, g^w)$. Decryption recovers m using the private key x.
  - Given a tuple (g, b, h, v), where v: encryption factor $h^w = a / m$.
  - The prover can prove that he knows the secret value x satisfying $x = \log_g h = \log_b v$, without giving this value x.

- **Verification:**
  - Anyone can calculate the result of the operation using the encrypted votes.
  - The process generates proofs of correct decryption of the result that can be verified by anyone.



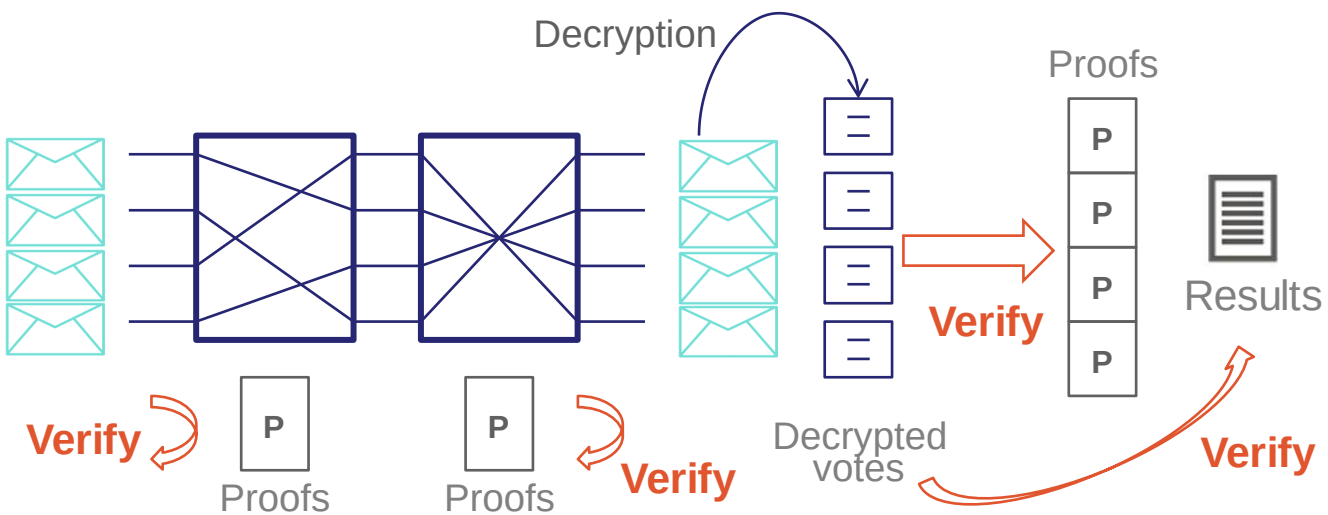
scytl Secure Electronic Voting

Universal verifiable Mix-nets (1/3)

- **Zero Knowledge Proof of plaintext equivalence** to demonstrate the correct re-encryption, based on the equality of discrete logarithms:
  - At one node, input is $c = (a, b) = (m \cdot h^w, g^w)$. Output is $c' = (a', b') = (m \cdot h^{w+w'}, g^{w+w'})$.
  - Given a tuple (g, u, h, v), where $u = b'/b = g^{w'}$, and $v = a'/a = h^{w'}$.
  - The prover can prove that he knows the secret value w' satisfying $w' = \log_g u = \log_h v$, without giving this value w'.
- **Zero Knowledge Proof of correct decryption**, based on the equality of discrete logarithms:
  - Remember $c = (a, b) = (m \cdot h^w, g^w)$. Decryption recovers m using the private key x.
  - Given a tuple (g, b, h, v), where v: encryption factor $h^w = a / m$.
  - The prover can prove that he knows the secret value x satisfying $x = \log_g h = \log_b v$, without giving this value x.

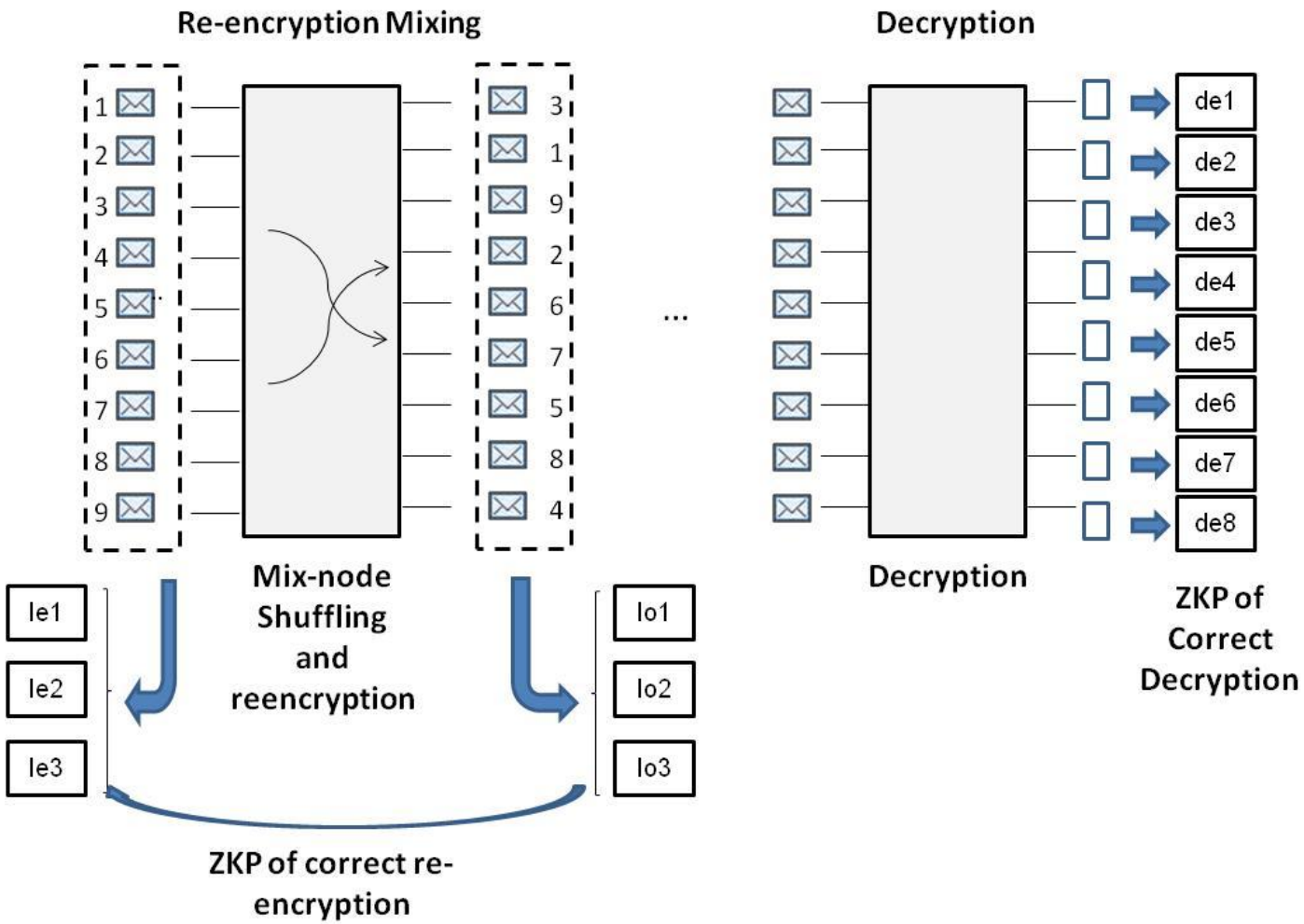
Universal verifiable Mix-nets (2/3)

- **Verification:**
  - Each mix-node calculates proofs of correct shuffling and correct re-encryption / decryption.
  - All the proofs are verifiable by anyone to detect that the input and output votes are based on the same original plaintexts (i.e., have not been changed).



Universal verifiable Mix-nets (3/3)

Mixing detail



scytl Secure Electronic Voting

www.scytl.com