Cryptographic Protocols for Electronic Voting

David Wagner
UC Berkeley

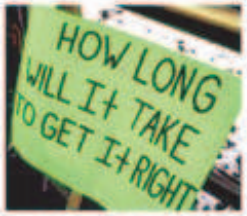
The Problem with Paperless Voting

- Unverified software must be presumed malicious
- How do you know whether your vote will be counted correctly, when voting machine software can record one thing and tell you another?

$\Rightarrow$ No rational basis for trust in election results

http://denvervoice.org

LIVE FREE
ELECTRONIC VOTING IS UNRELIABLE
PAPER BALLOTS
THE COMPUTER ATE MY VOTE!


HOW LONG WILL IT TAKE TO GET IT RIGHT

ELECTION FRAUD
MEDIA BLACKOUT

VOTE

VOTING WITHOUT AUDITING ARE WE INSANE?
SECRET SOFTWARE SECRET GOVERNMENT

LIVE FREE OR DIE BOLD
VISUALIZE TRUE DEMOCRACY

C Berkeley

Problem Statement

- The problem: With today's paperless voting machines, the integrity of the election relies completely on software.

- Goal: The integrity of the election should not be dependent upon the correctness of software.

Security Goals for an Election

- **Integrity**: No election fraud
- **Transparency**: Everyone must be able to verify that the election was conducted properly
- **Privacy**: No one learns how the voter has voted
- **Secret ballot**: Voter cannot prove how she voted

In This Talk…

- "The early years"
  - How to prove ballots were counted correctly (using crypto)
  - But: fails to address ballot preparation

- Modern cryptographic voting systems
  - End-to-end integrity: proving that ballots were cast and counted as the voter intended (using crypto)

Featuring Work By...

Andy Neff
David Chaum

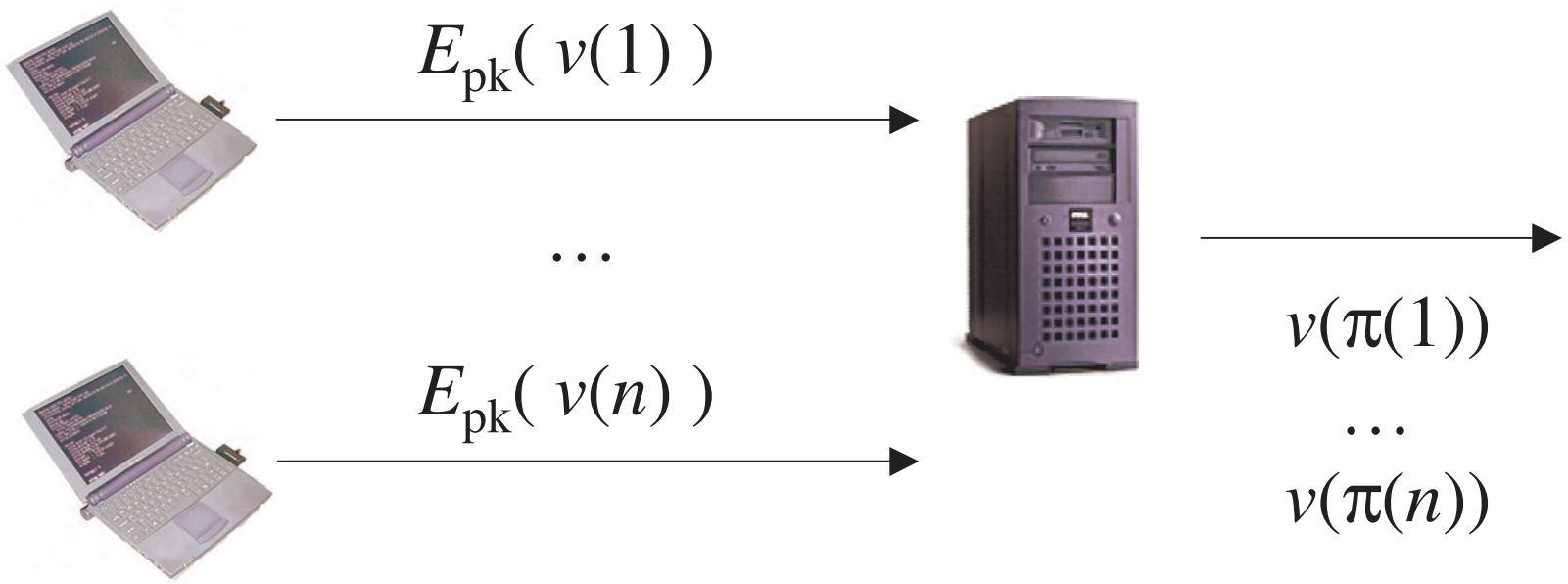
and

Josh Benaloh
Steve Schneider

Peter Ryan
and many others

All ideas in this talk were discovered by others.
Any errors are my fault.

Cryptographic Voting with Trusted Server



El Gamal Encryption

- Encrypt votes using El Gamal:

  $E(v) = (g^r, h^r\, v)$ $\qquad\qquad$ $r \leftarrow \mathbf{Z}/q\mathbf{Z}$

- Ciphertexts can be blinded (re-randomized):

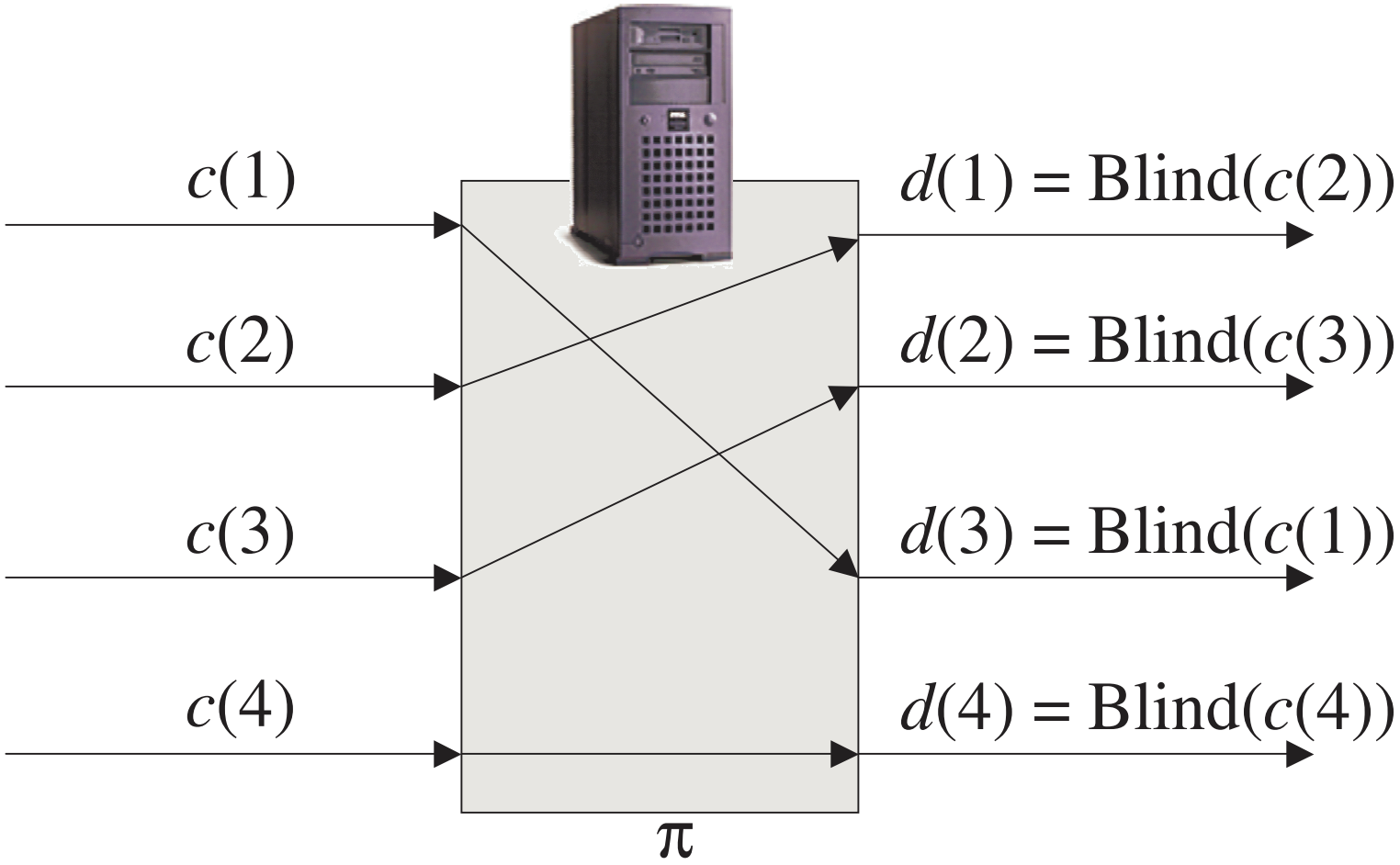
  $\text{Blind}(x, y) = (g^s\, x, h^s\, y)$ $\quad s \leftarrow \mathbf{Z}/q\mathbf{Z}$

- Blinding forms a group:

  $\text{Blind}_s(\text{Blind}_{s'}(c)) = \text{Blind}_{s+s'}(c)$

- Supports threshold decryption

Re-encryption Mixnet

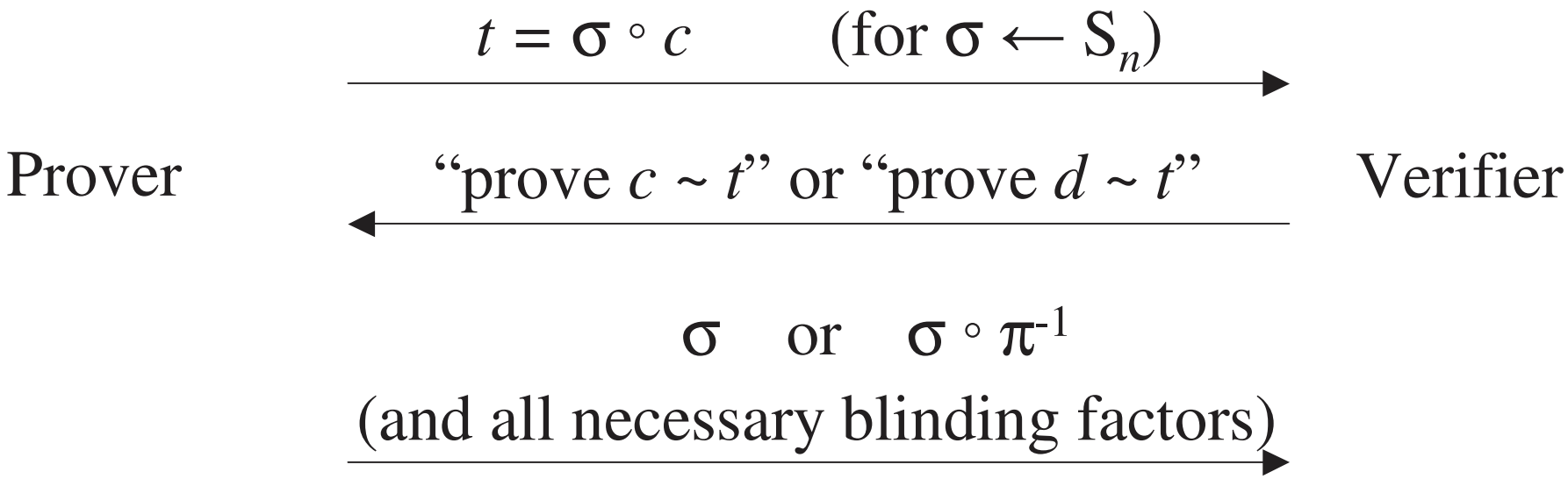


$c(i) = E(v(i))$

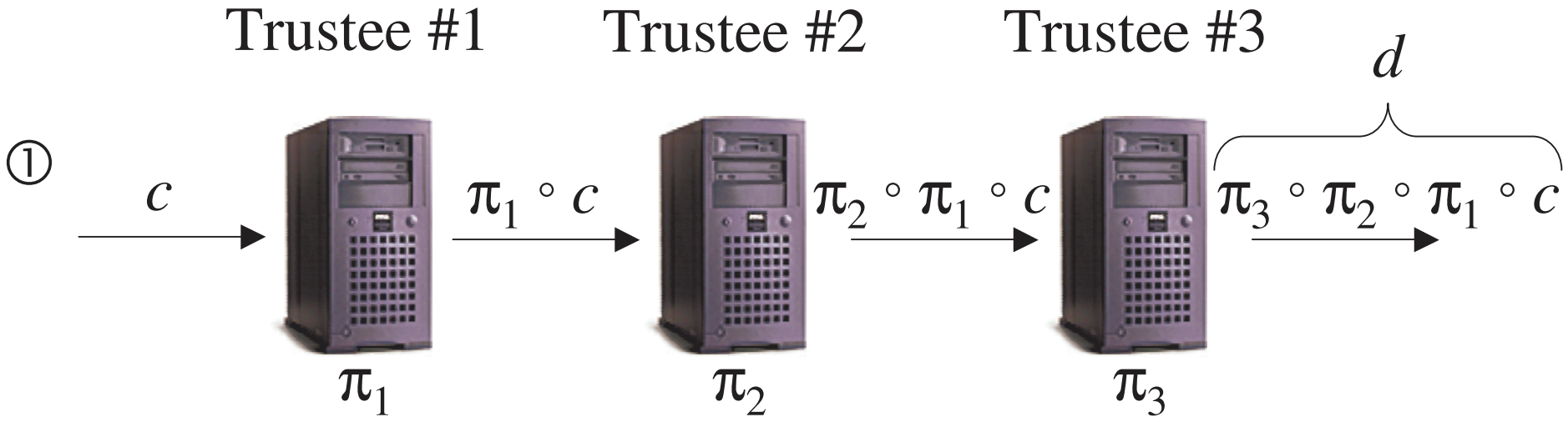
$d(i) = \text{Blind}(c(\pi(i)))$

ZK Proof of Correct Shuffling [Benaloh]

- Given: $c(1..n)$, $d(1..n)$
- To prove: $c \sim d$ (i.e., $d = \pi \circ c$)



Distributing Trust During Vote-Counting



② Trustees perform threshold decryption of d, and provide ZK proof of correct mixing and correct decryption.

Unconditional integrity (even if all trustees collude).
Computational privacy, assuming one honest trustee.

Criticisms of Early Voting Protocols

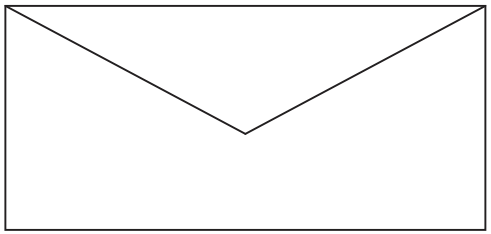
- Early protocols got the threat model wrong.
  - In reality, trust in voter's computer is unwarranted.
- Early protocols ignored ballot preparation—which turns out to be the hard problem.

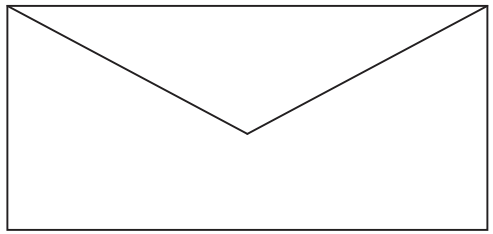
A Better Voting Machine [Neff]

Voting machine with untrusted software

Receipt
(enables voter to check that their vote was counted as intended)

Proof of Equality



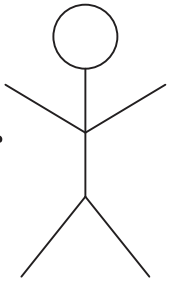


"Both envelopes contain the same number"

Prover

"Oh yeah? Prove it!"

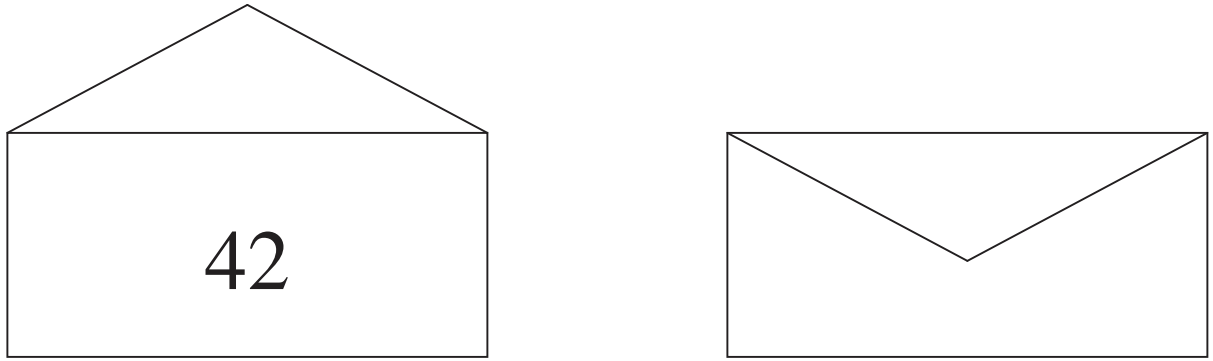
Verifier



"They both contain 42"

"Show me what's in the left one"

Proof of Equality

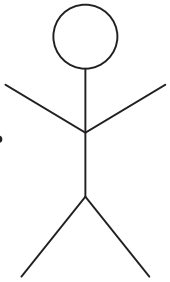


"Both envelopes contain the same number"

Prover

"Oh yeah? Prove it!"

Verifier



"They both contain 42"

"Show me what's in the left one"

Notation

$\boxed{b}$ = encryption of b
= commitment to b

(e.g., $\boxed{b} = (g^r, h^r\, g^b)$)

$\textcircled{b}$ = randomness used in $\boxed{b}$
= opened commitment to b

(e.g., $\textcircled{b} = (r, b)$)

A Special Ballot Encoding

Unencrypted ballot:

GIULIANI	0	1	1	0	1	0	0	1
CLINTON	1	1	0	0	0	0	0	0

This is a vote for Clinton

Encrypting The Ballot

Encrypted ballot:

GIULIANI	0	1	1	0	1	0	0	1
CLINTON	1	1	0	0	0	0	0	0

An encrypted vote for Clinton

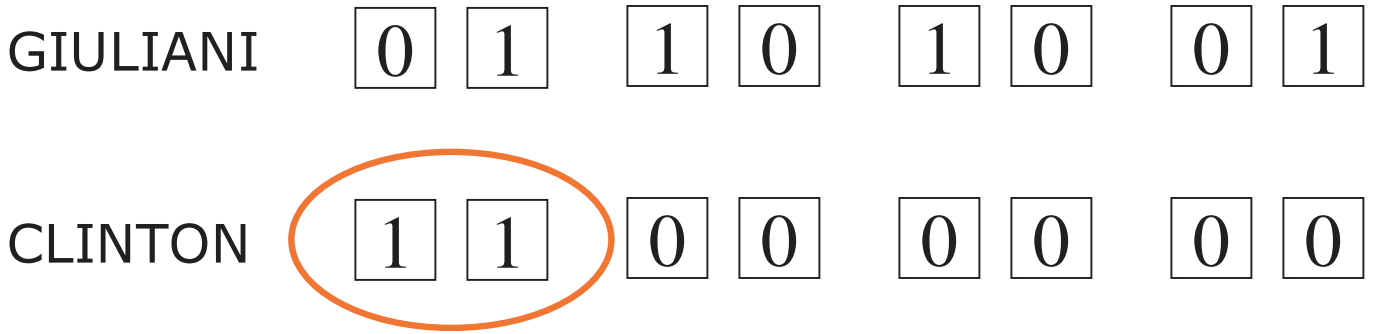
Encrypting The Ballot

Encrypted ballot:

GIULIANI	0	1	1	0	1	0	0	1
CLINTON	1	1	0	0	0	0	0	0

Proving the Ballot Was Encrypted Correctly

Encrypted ballot:



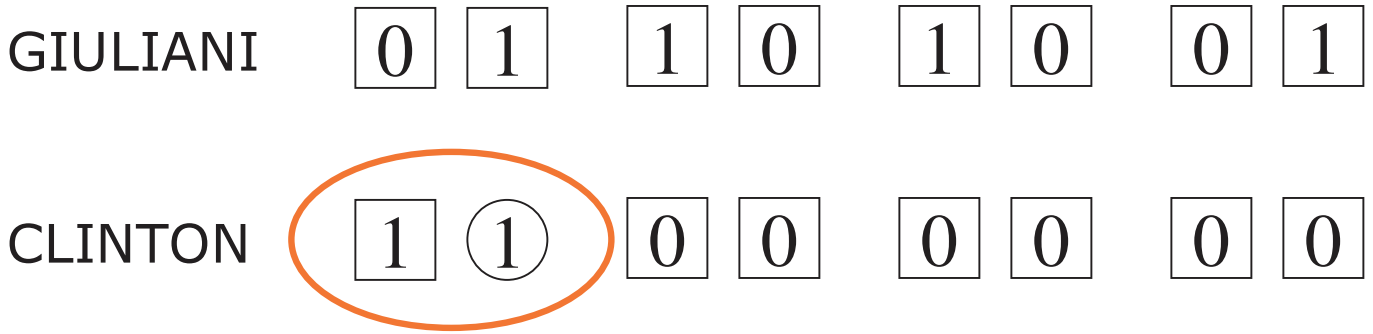
Machine proves both bits are the same.

"Both bits are 1" →

← "Open up the right commitment"

Proving the Ballot Was Encrypted Correctly

Encrypted ballot:



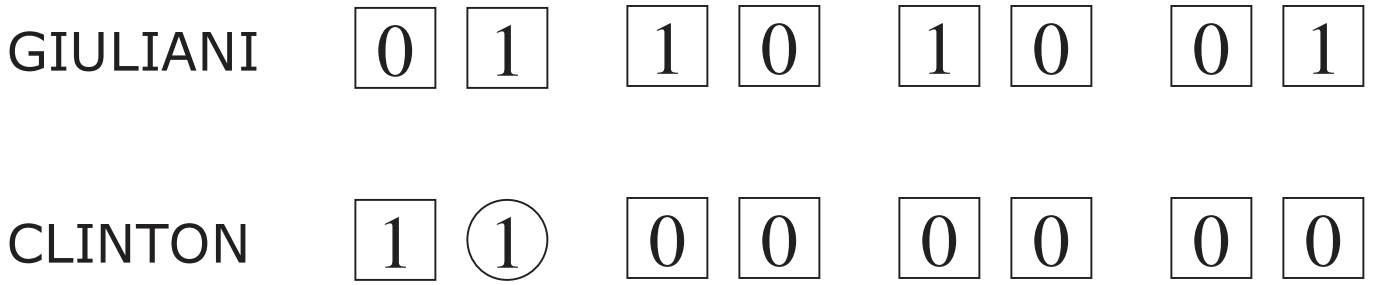
Machine proves both bits are the same.

"Both bits are 1" →

← "Open up the right commitment"

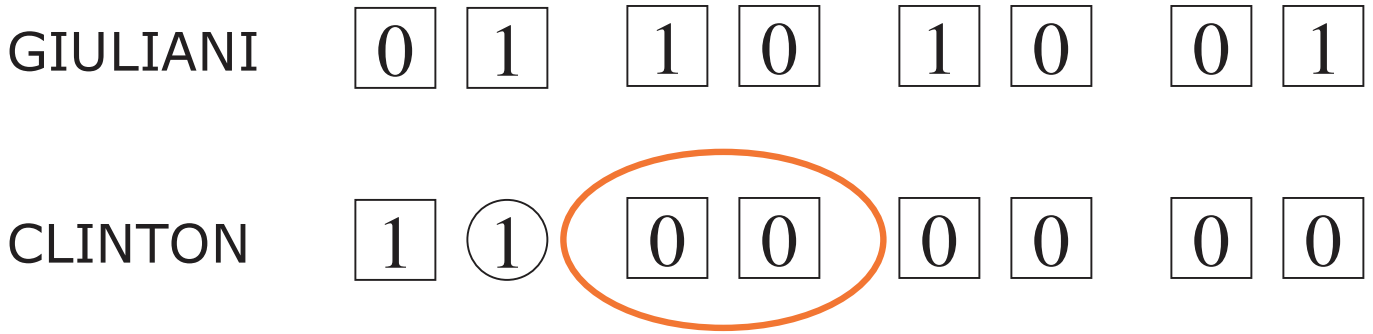
Proving the Ballot Was Encrypted Correctly

Encrypted ballot:



Proving the Ballot Was Encrypted Correctly

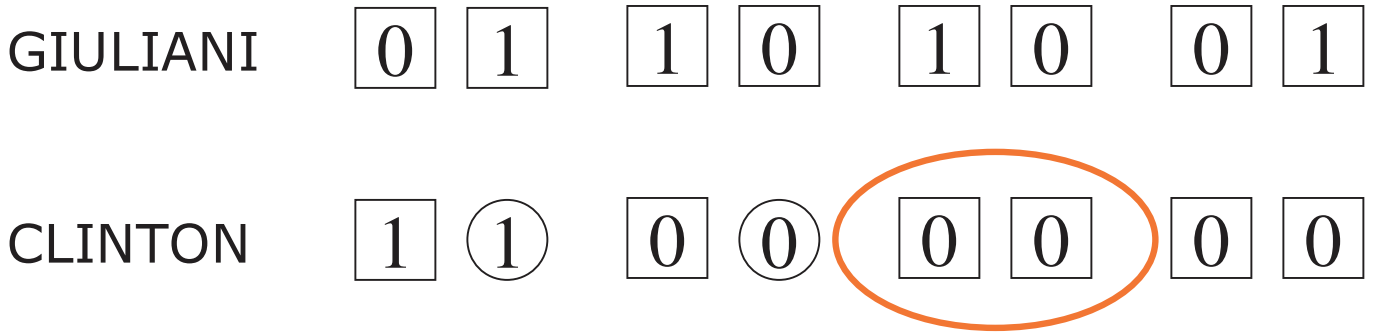
Encrypted ballot:



Machine proves both bits are the same.

Proving the Ballot Was Encrypted Correctly

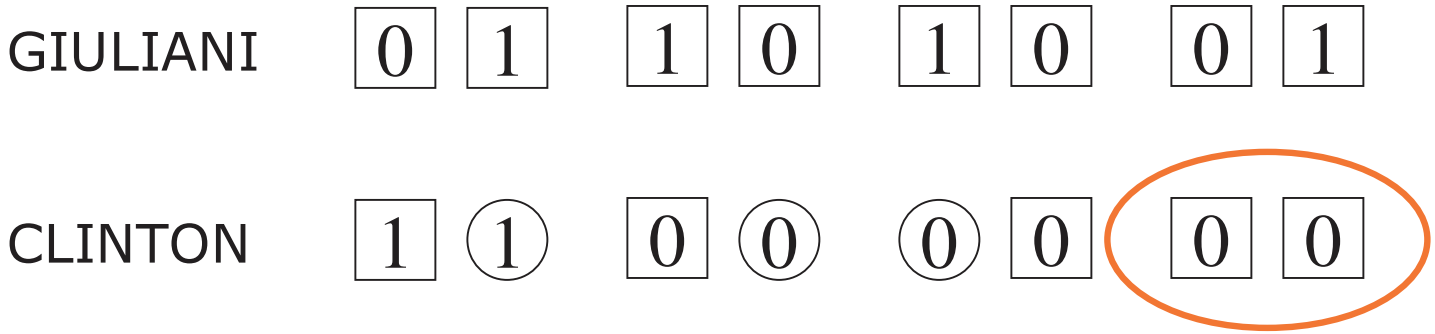
Encrypted ballot:



Machine proves both bits are the same.

Proving the Ballot Was Encrypted Correctly

Encrypted ballot:



Machine proves both bits are the same.

Proving the Ballot Was Encrypted Correctly

Partially encrypted ballot:

GIULIANI	0	1	1	0	1	0	0	1
CLINTON	1	1	0	0	0	0	0	0

(A transcript of an interactive proof that this contains a valid vote for Clinton)

Receipts That Reveal Nothing

Printed on the receipt:

(A fake transcript of an interactive proof that this contains a valid vote for Giuliani)

GIULIANI	[0]	(1)	[1]	(0)	[1]	(0)	(0)	[1]
CLINTON	[1]	(1)	[0]	(0)	(0)	[0]	[0]	(0)

Putting it Together: Neff's Scheme

① Machine interactively proves that the encrypted ballot accurately captures the voter's intent

② Machine prints (real and fake) proof-transcripts onto a paper receipt retained by the voter

③ Machine publicly posts image of receipt

④ Voter checks that her receipt was publicly posted

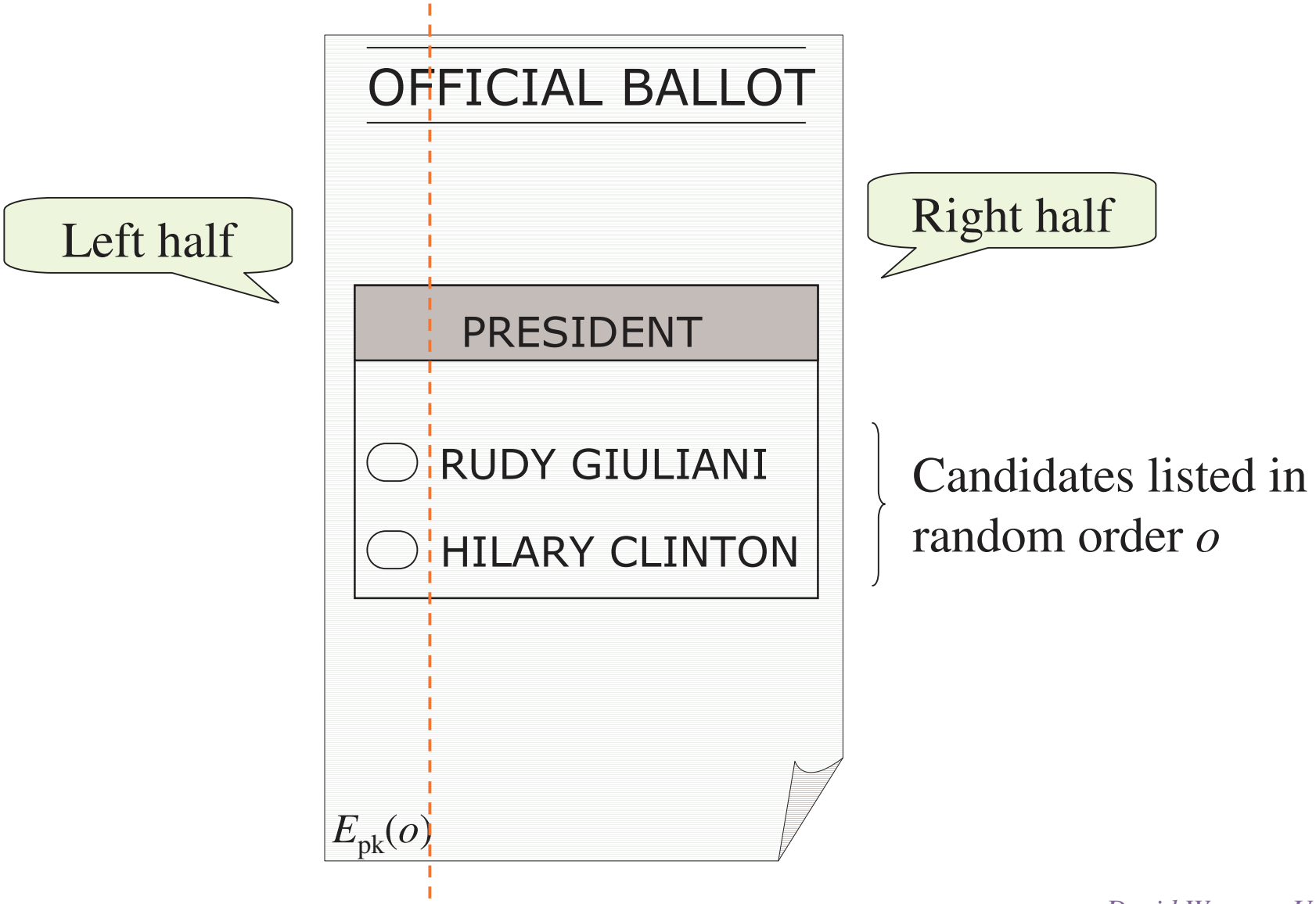
⑤ Trustees decrypt and tally all posted receipts using re-encryption mixes and threshold decryption

Security Properties of [Neff]

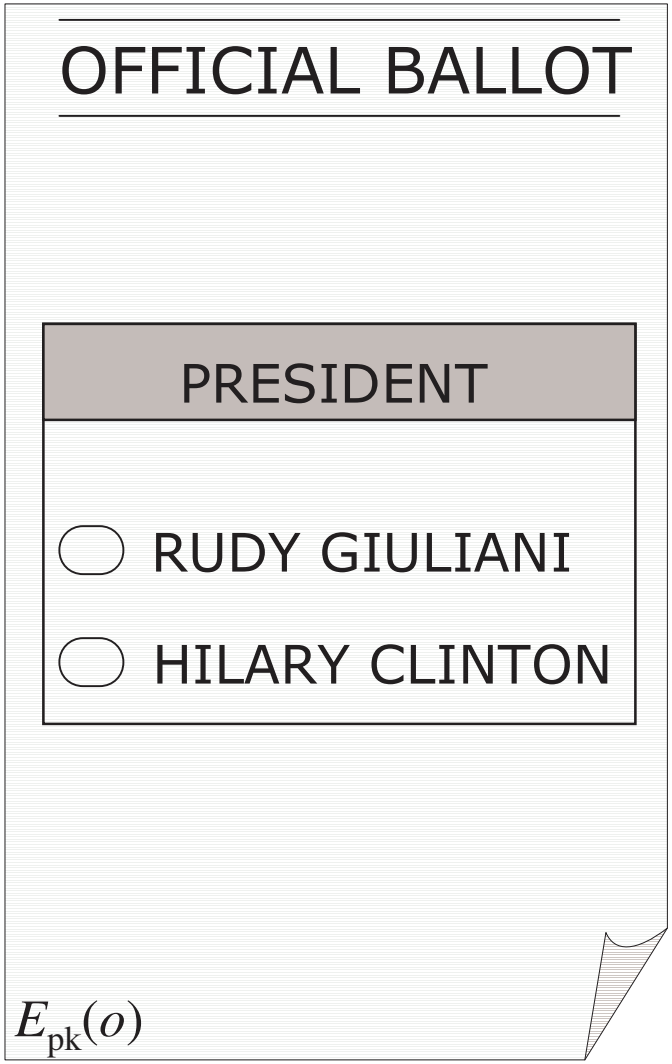
- Integrity: Voters can use their receipt to confirm that their votes were recorded *and counted* as intended
- Privacy: Voters cannot sell their vote or be coerced (the receipt provides no information about their vote, since all transcripts on receipt can be simulated)

- No reliance on software!

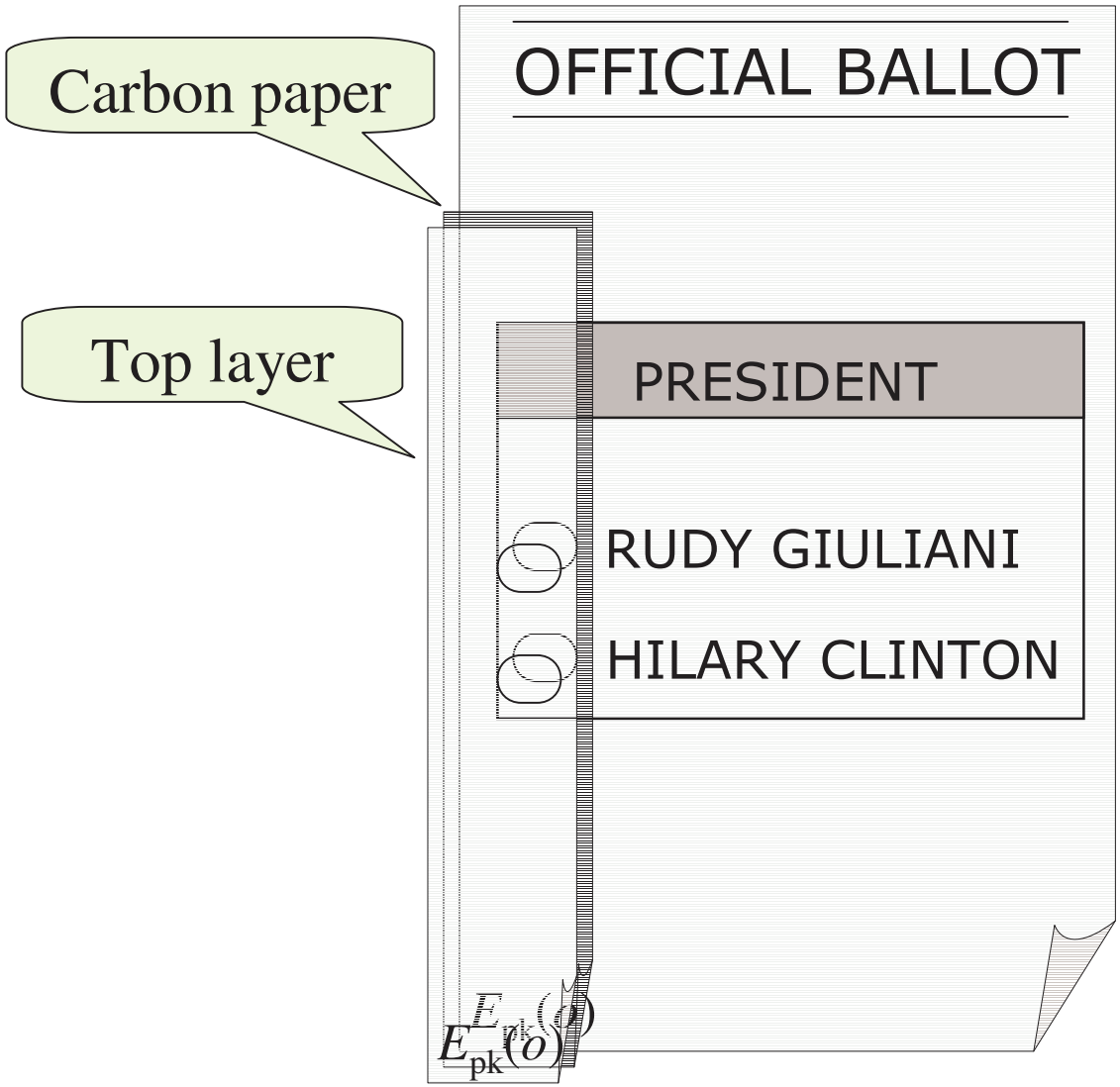
A Better Paper Ballot [CRS]



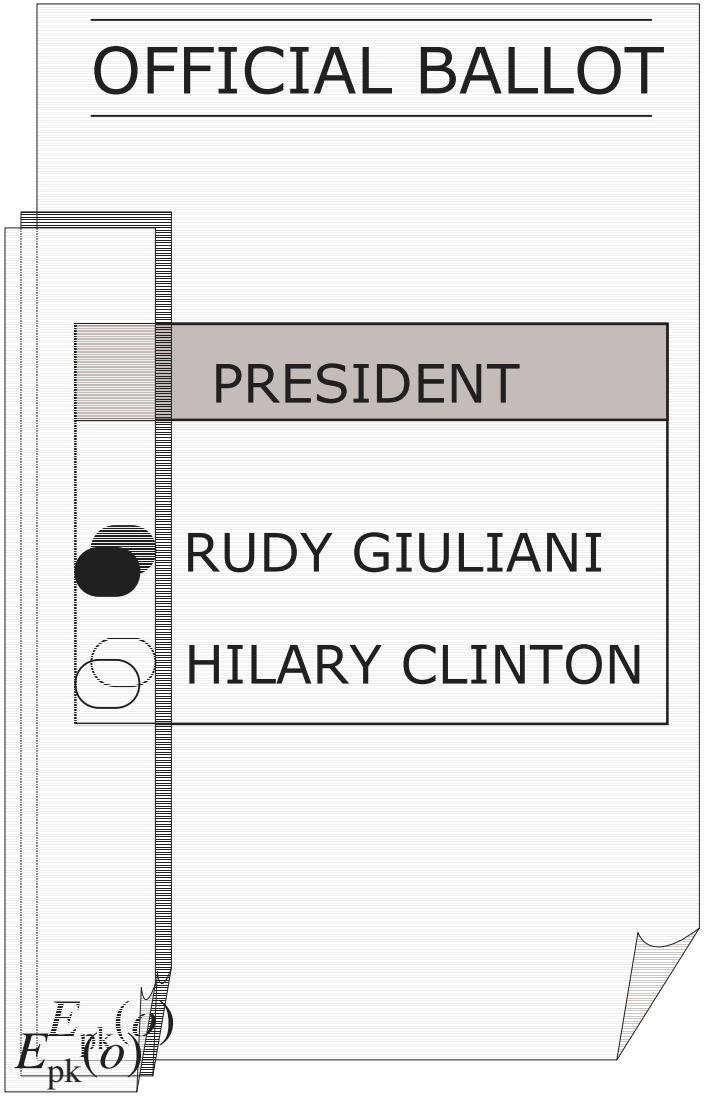
A Better Paper Ballot [CRS]



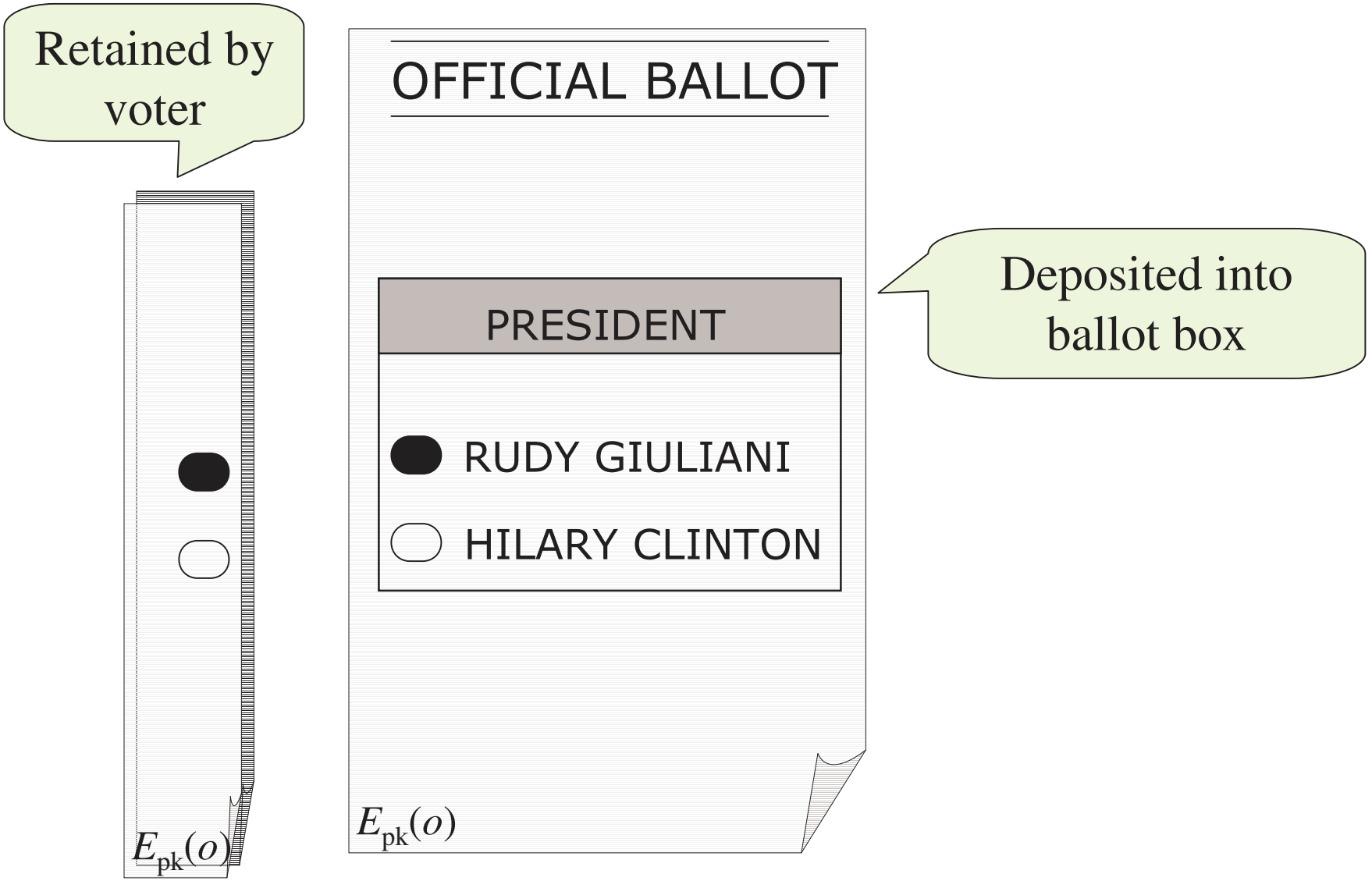
A Better Paper Ballot, With Receipt



A Marked Ballot

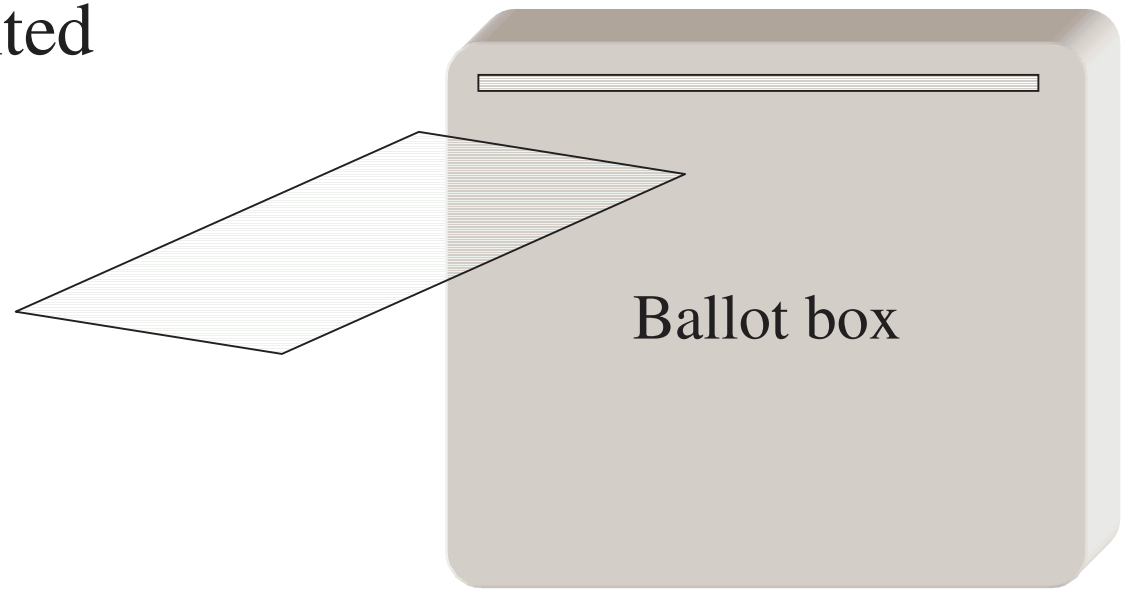


The Receipt Is Torn Off



Casting the Ballot

- The ballot is deposited into the ballot box
- The left side of the ballot is digitally scanned and this image is posted publicly
- Ballots can be hand-counted or electronically counted



Verfiably Correct Tallying

- Voters check that a picture of their receipt appears on the public bulletin board
- Trustees shuffle and decrypt receipts using re-encryption mixes and threshold decryption
- Everyone verifies that trustees performed tallying correctly by checking ZK proofs

Security Properties of [CRS]

- Integrity: Voters can use their receipt to confirm that their votes were recorded *and counted* as intended
- Privacy: Voters cannot sell their vote or be coerced (the receipt provides no information about their vote)

- No reliance on software!

Potential Challenges in the Real World

- Human factors and voter training
  (voters will have to learn how to use new ballots;
  will voters make more mistakes?)

- Accessibility
  (lacks verifiability for visually impaired voters)

- Public confidence in hairy math
  (most voters and officials won't understand the crypto)

In Summary

- Can build voting machines whose correctness is—at least in principle—not dependent on software.
- Practical feasibility still uncertain, but worth a shot. An exciting field with many beautiful ideas.
- Humans can verify that complex cryptographic computations were performed correctly. Wow!