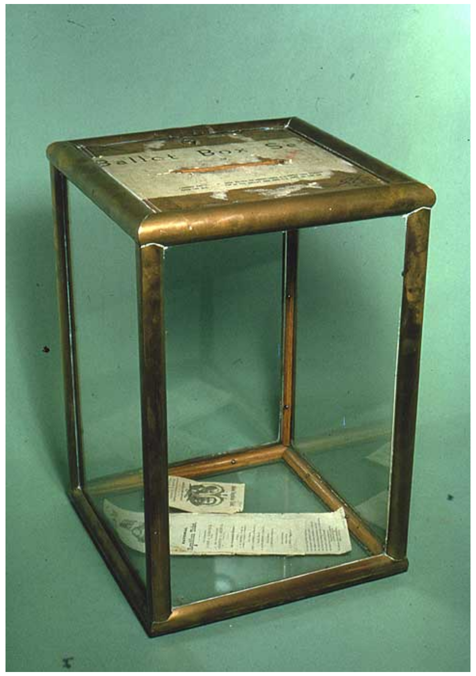
VERIFIABILITY
in electronic voting

Michael Clarkson
George Washington University

International Summer School on Secure Voting
July 16, 2012

Secret Ballot



HANDICAPPED


CLERK OF THE CIRCUIT COURT
Vote for One
Republican
Terry Bork
Democratic
Write in
BOARD OF EDUCATION
Vote for One
Write in
BOARD OF EDUCATION
Vote for One
Write in
SHERIFF
Vote for One
Republican

“Flawless”

Security FAIL

Analysis of an electronic voting system.

[Kohno, Stubblefield, Rubin, and Wallach 2004]

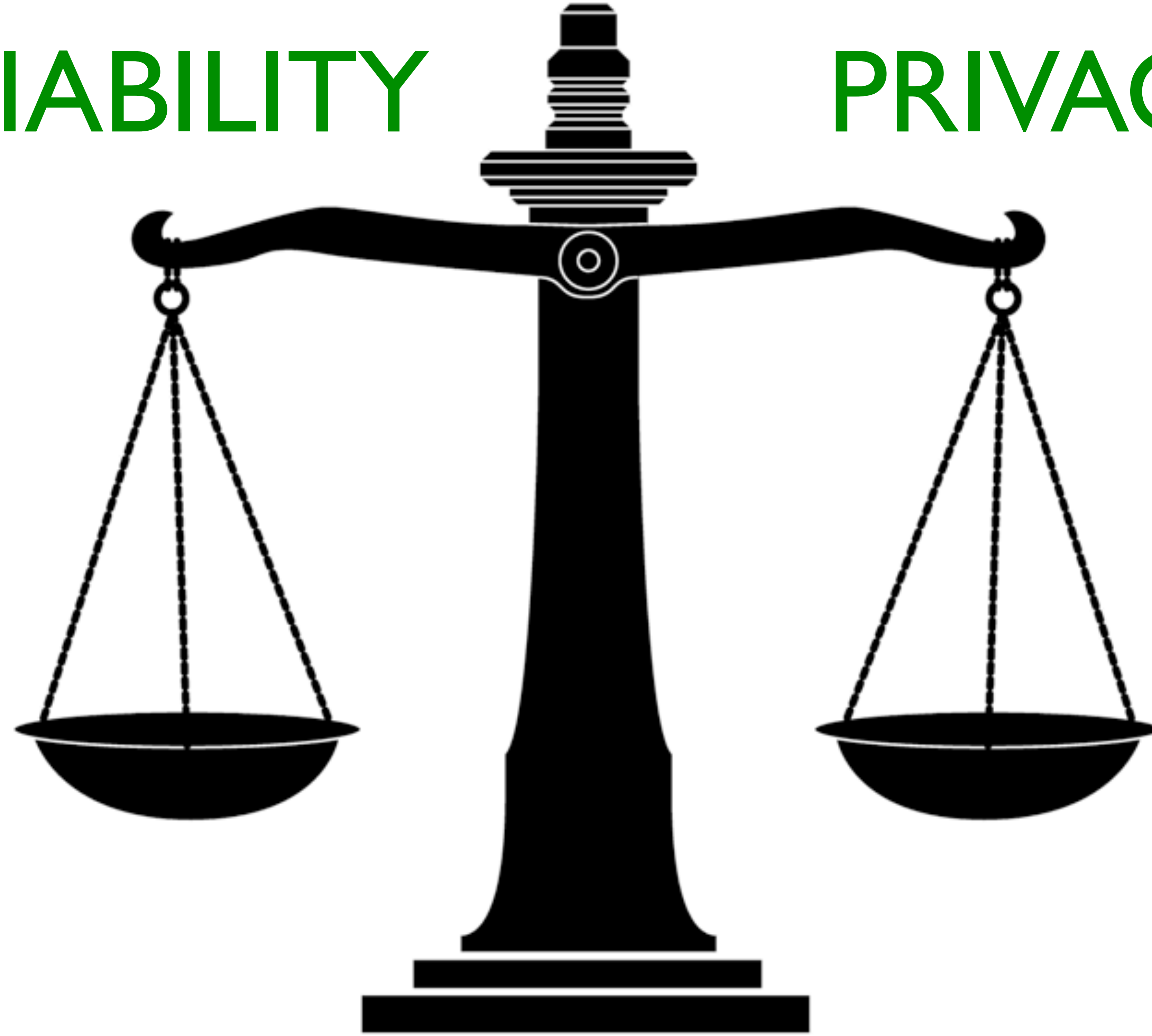
- DRE trusts smartcards
- Hardcoded keys and initialization vectors
- Weak message integrity
- Cryptographically insecure random number generator
- ...

California top-to-bottom reviews [Wagner et al. 2007]

- *"Virtually every important software security mechanism is vulnerable to circumvention."*
- *"An attacker could subvert a single polling place device...then reprogram every polling place device in the county."*
- *"We could not find a single instance of correctly used cryptography that successfully accomplished the security purposes for which it was apparently intended."*

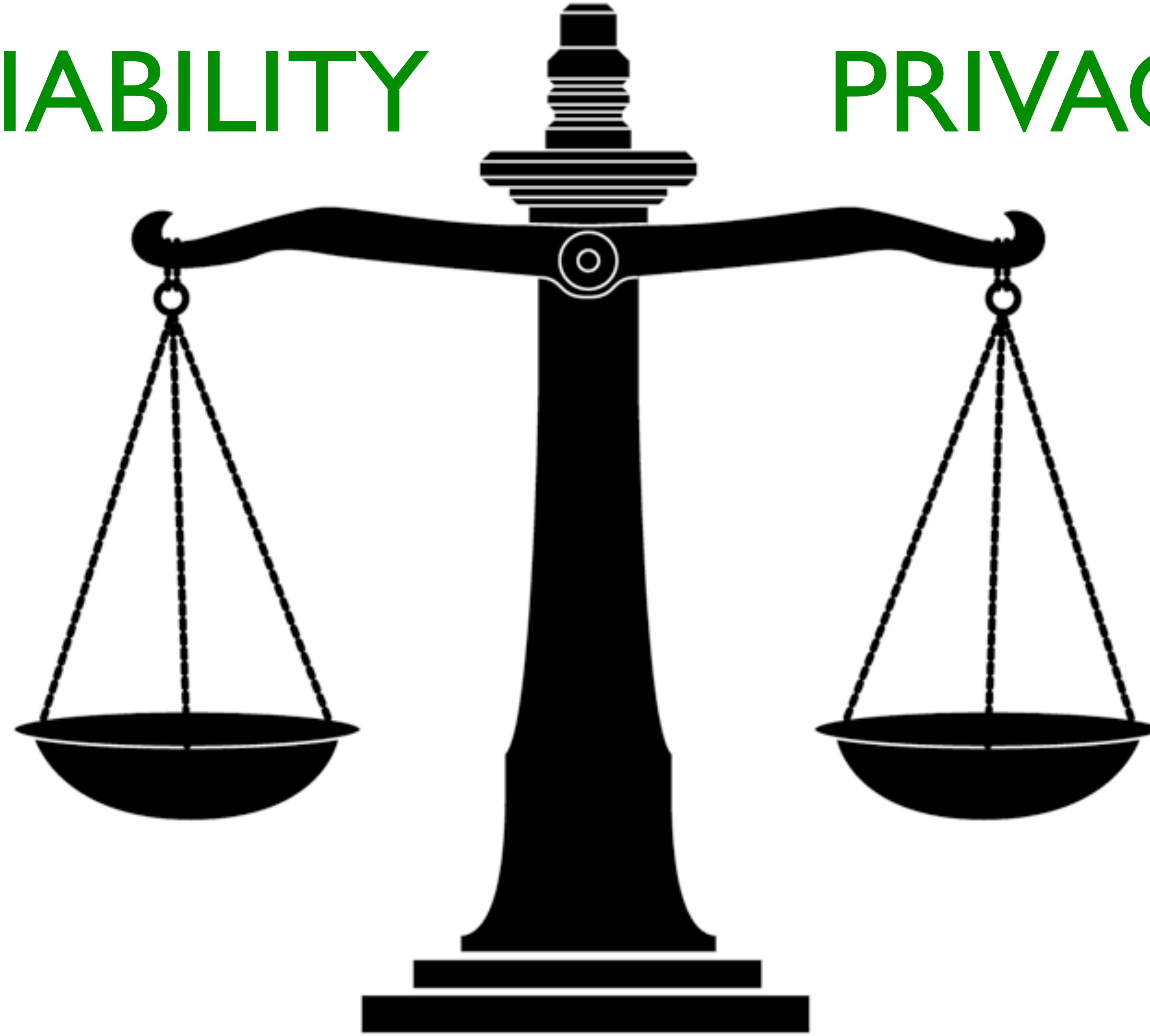
Why is this so hard?

VERIFIABILITY PRIVACY



What to verify?
What to keep private?

VERIFIABILITY PRIVACY



Why is this so hard?

Key differences:

Adversarial models

Fault detection and recovery

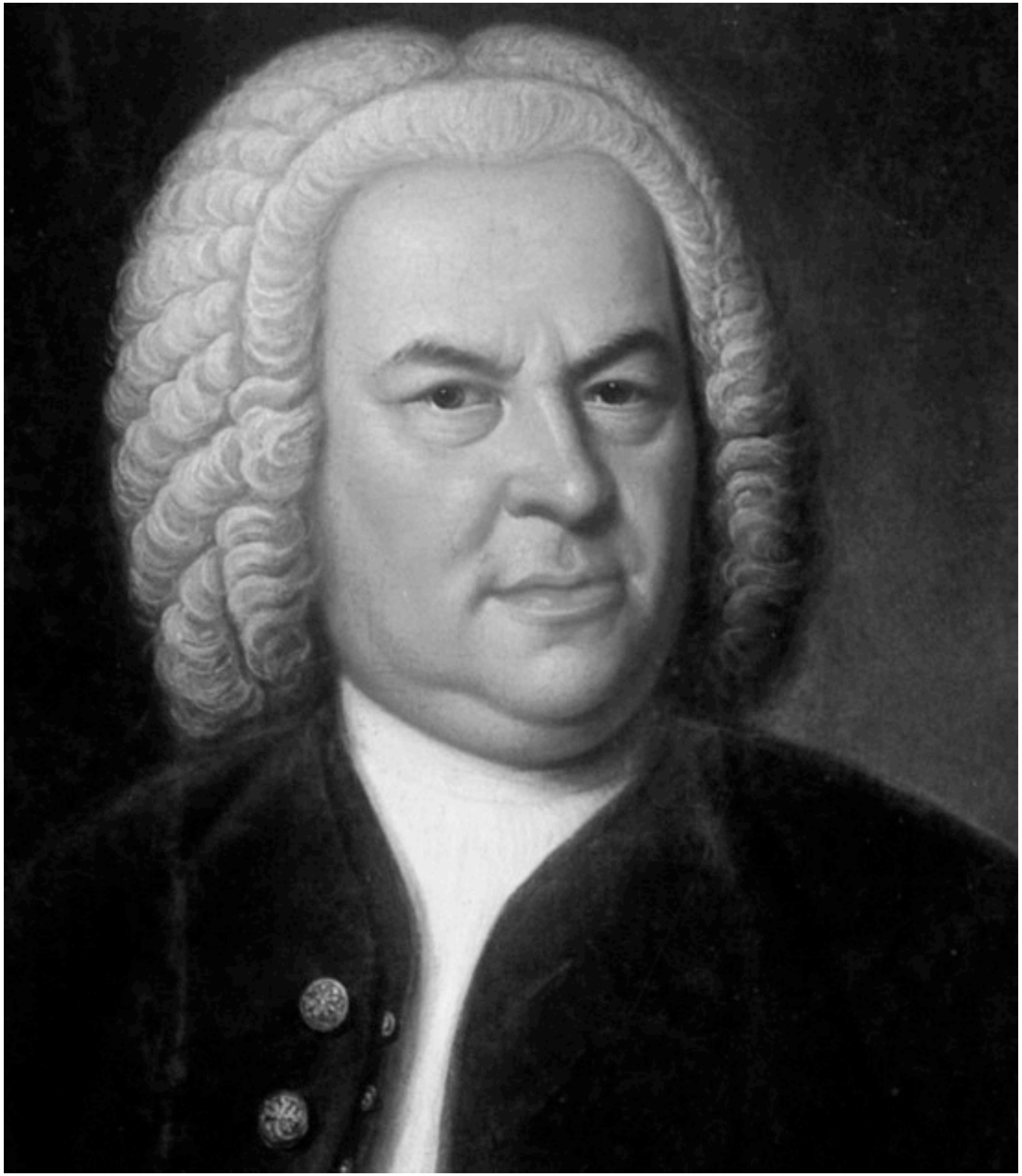
[Schneier 2001, Adida 2006]

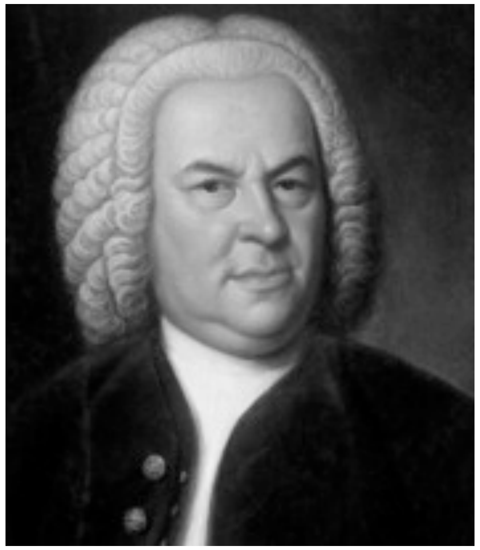
VERIFIABILITY
in electronic voting

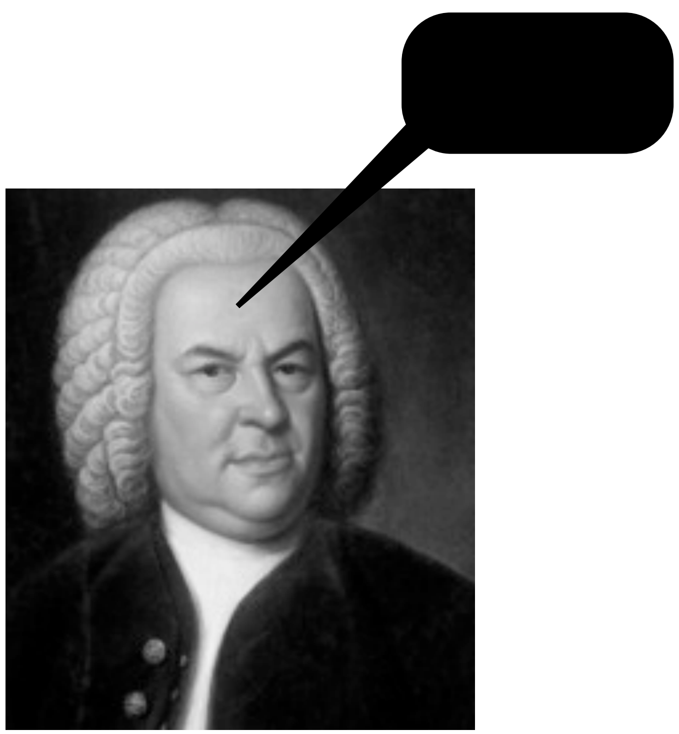
Michael Clarkson
George Washington University

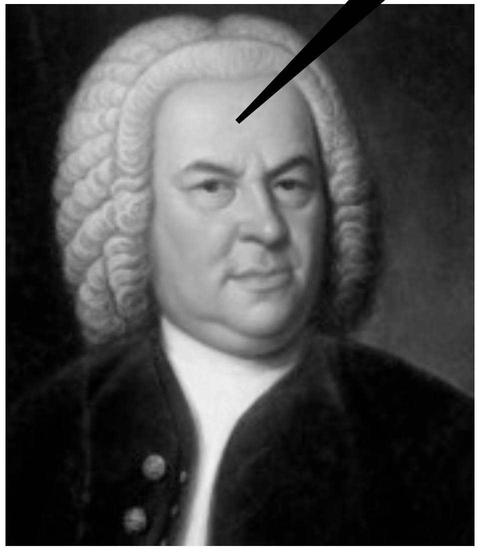
International Summer School on Secure Voting
July 16, 2012

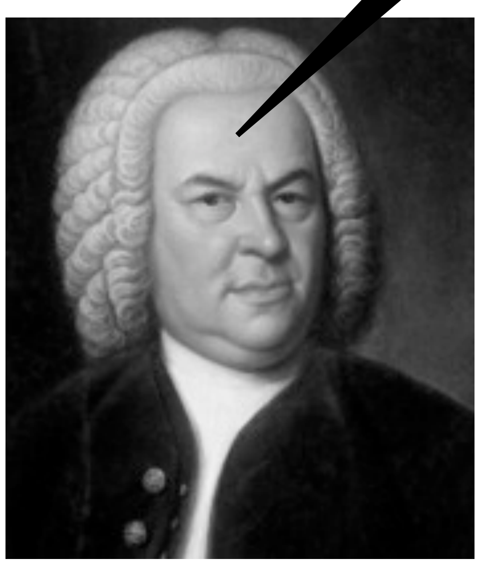
ACT I

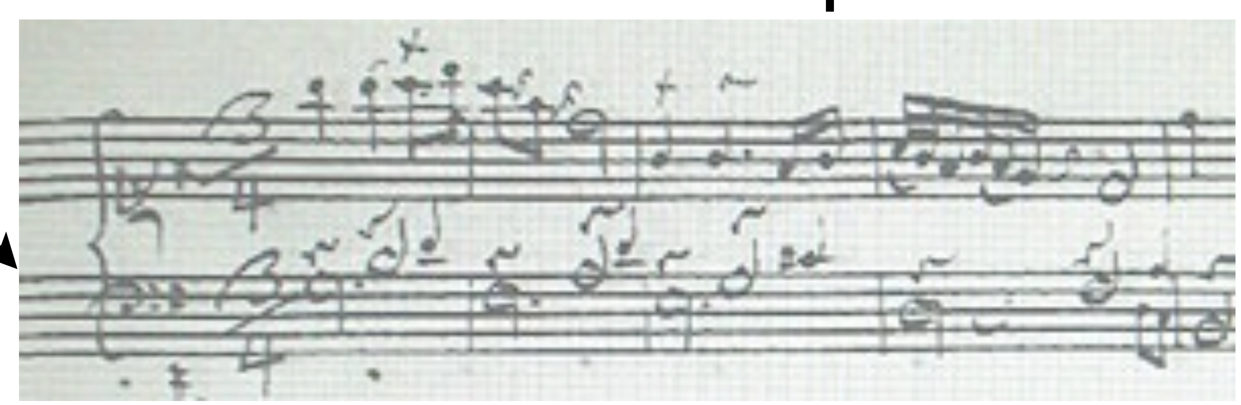






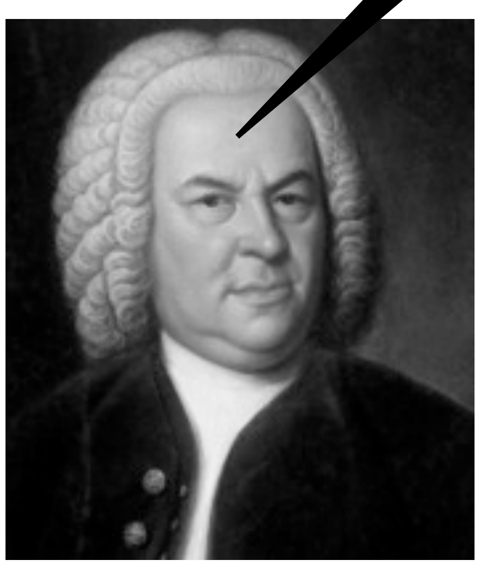


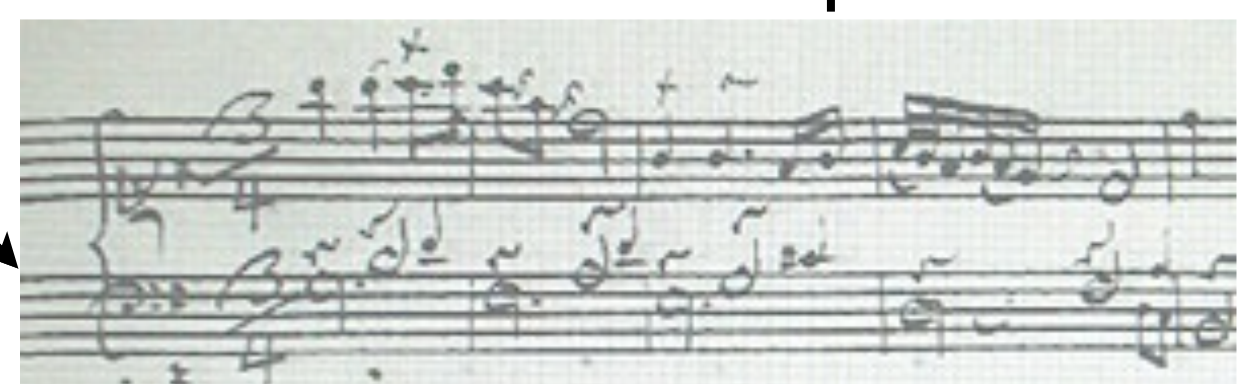




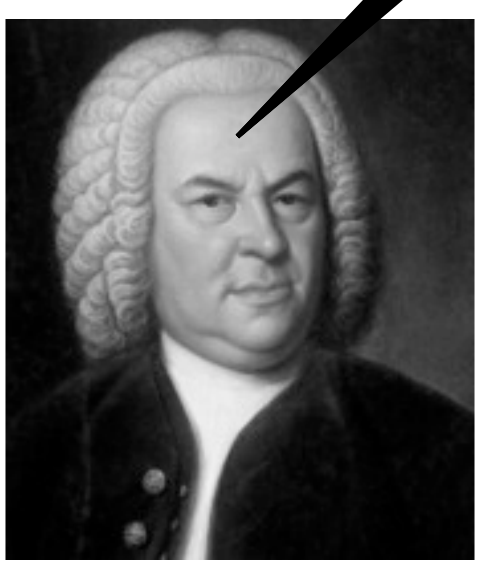

Goldberg Variations
(Air with 30 Variations)
BWV 988


Goldberg Variations
(Air with 30 Variations)
BWV 988



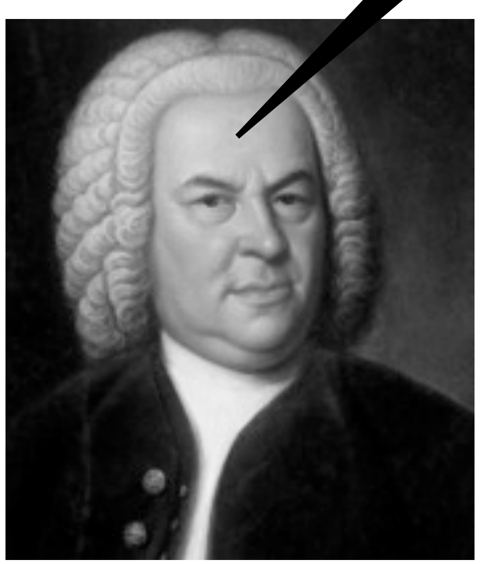



Goldberg Variations
(Air with 30 Variations)
BWV 988

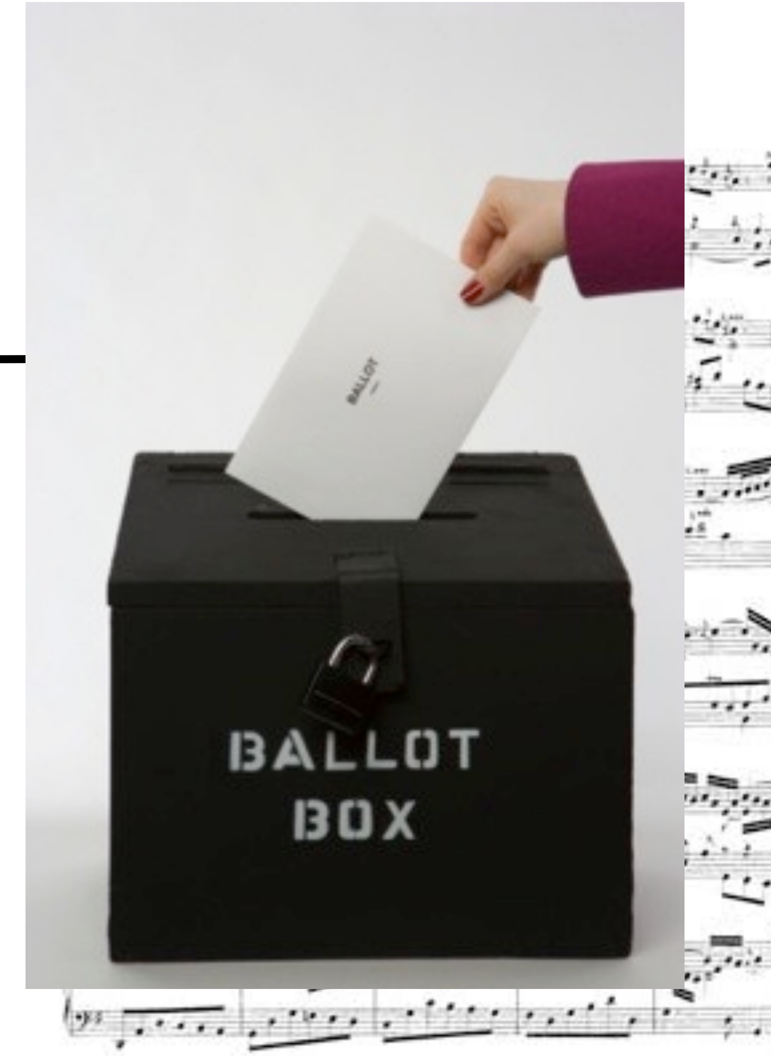


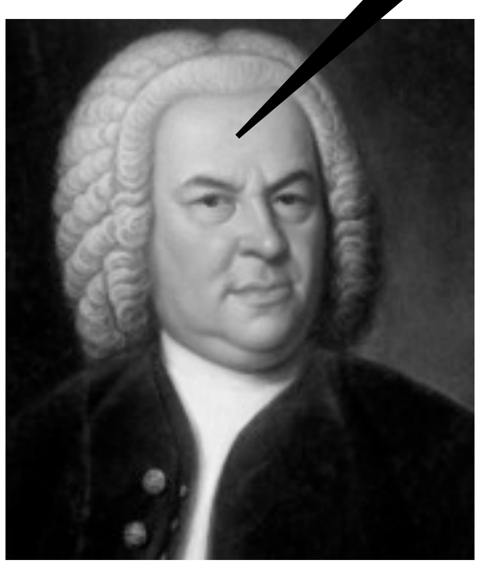

VOTE


BALLOT
BOX

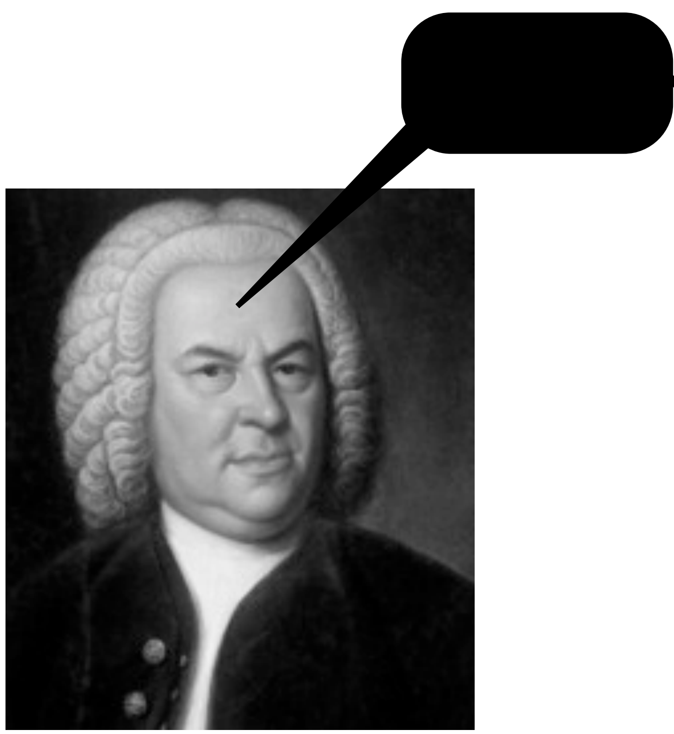


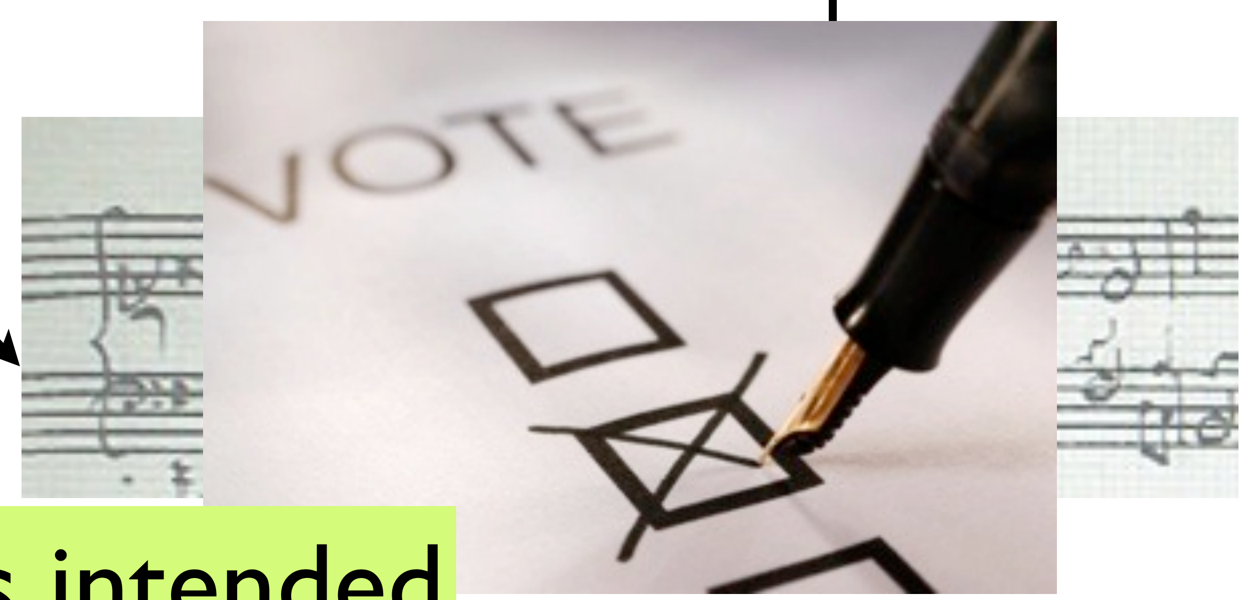

VOTE


BALLOT
BOX



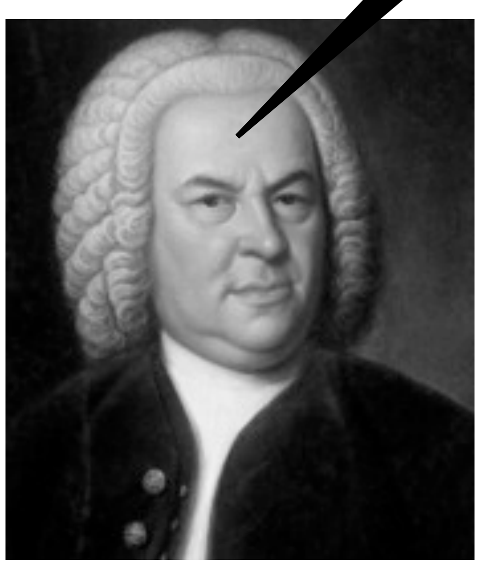

VOTE





Cast as intended

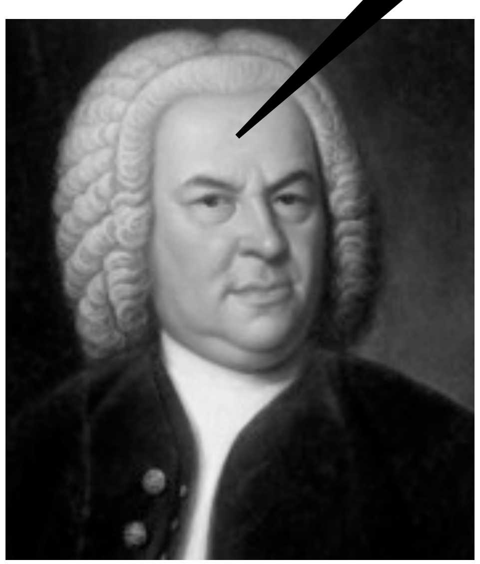
Recorded as cast



Cast as intended

Counted as recorded

Recorded as cast



Cast as intended

Verification Tasks

- Cast as intended
- Recorded as cast
- Counted as recorded

Formal Definitions of Counted-as-recorded VERIFIABILITY

Privacy?

Verifiability in Early Work

Definition: "Verifiability: Anyone can verify the correctness of the results."

Proof: "Verifiability holds assuming there is no collusion."

[XXXXXX 19XX]

Verifiability

[Juels, Catalano, Jakobsson 2005]

Election protocol is verifiable if adversary cannot concoct a BB that verifies with an incorrect tally, even if given access to all secret keys.

BB: bulletin board

Verifiability

[Juels, Catalano, Jakobsson 2005]

tabulate: $BB \times k \times \{VK\} \rightarrow tally \times zkpf$

Verifiability

[Juels, Catalano, Jakobsson 2005]

tabulate: BB × k × {VK} → tally × zkpf

verify: BB × K × {VK} × tally × zkpf → boolean

Verifiability

[Juels, Catalano, Jakobsson 2005]

tabulate: BB × k × {VK} → tally × zkpf

verify: BB × K × {VK} × tally × zkpf → boolean

fake-election: k × {Vk} → BB × tally × zkpf

Verifiability

[Juels, Catalano, Jakobsson 2005]

tabulate: $BB \times k \times \{VK\} \rightarrow tally \times zkpf$

verify: $BB \times K \times \{VK\} \times tally \times zkpf \rightarrow boolean$

fake-election: $k \times \{Vk\} \rightarrow BB \times tally \times zkpf$

(actually in computational model)

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})
and (tally, zkpf) = tabulate(BB, k, {VK}).

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})
and (tally, zkpf) = tabulate(BB, k, {VK}).

If verify(BB, K, {VK}, ftally, fzkpf),

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})
and (tally, zkpf) = tabulate(BB, k, {VK}).

If verify(BB, K, {VK}, ftally, fzkpf),
then ftally = tally.

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})
and (tally, zkpf) = tabulate(BB, k, {VK}).

If verify(BB, K, {VK}, ftally, fzkpf),
then ftally = tally. (prob. of inequality is neg.)

Verifiability

[Juels, Catalano, Jakobsson 2005]

Let (BB, ftally, fzkpf) = fake-election(k, {Vk})
and (tally, zkpf) = tabulate(BB, k, {VK}).

If verify(BB, K, {VK}, ftally, fzkpf),
then ftally = tally. (prob. of inequality is neg.)

...purely about "counted as recorded"

Verifiability

[Kremer, Ryan, Smyth 2010]

IV(vote, cred, ballot, privstate) : boolean

Verifiability

[Kremer, Ryan, Smyth 2010]

IV(vote, cred, ballot, privstate) : boolean

UV(votes, ballots, pfs) : boolean

Verifiability

[Kremer, Ryan, Smyth 2010]

IV(vote, cred, ballot, privstate) : boolean

UV(votes, ballots, pfs) : boolean

(actually in symbolic model)

Verifiability

[Kremer, Ryan, Smyth 2010]

1. If IV(vote1, cred, ballot, privstate1)
   and IV(vote2, cred, ballot, privstate2)
   then vote1=vote2
   and privstate1=privstate2

Verifiability

[Kremer, Ryan, Smyth 2010]

1. If IV(vote1, cred, ballot, privstate1)
   and IV(vote2, cred, ballot, privstate2)
   then vote1=vote2
   and privstate1=privstate2

...no ballot on BB can verify as more than one vote

Verifiability

[Kremer, Ryan, Smyth 2010]

2. If UV(votes, ballots, pfs)
   and UV(votes', ballots, pfs)
   then votes=votes'.

Verifiability

[Kremer, Ryan, Smyth 2010]

2. If UV(votes, ballots, pfs)
   and UV(votes’, ballots, pfs)
   then votes=votes’.

...ballots on BB can verify only as one set of votes

Verifiability

[Kremer, Ryan, Smyth 2010]

3. If for all i, IV(vote[i], cred[i], ballot[i], privstate[i]) and UV(votes, ballots, pfs) and ballots = [ballot[i] | i], then votes = [vote[i] | i].

Verifiability

[Kremer, Ryan, Smyth 2010]

3. If for all i, IV(vote[i], cred[i], ballot[i], privstate[i])
   and UV(votes, ballots, pfs)
   and ballots = [ballot[i] | i],
   then votes = [vote[i] | i].

...ballots on BB really do contain votes expected by voters

Verifiability

[Kremer, Ryan, Smyth 2010]

Verifiability

[Kremer, Ryan, Smyth 2010]

EV(creds, ballots, pfs) : boolean

Verifiability

[Kremer, Ryan, Smyth 2010]

EV(creds, ballots, pfs) : boolean

Three more conditions to formalize
that EV holds only if all votes are authorized

Accountability

[Küsters, Truderung, Vogt 2010]

Need to assign blame when protocol run fails to verify.

Accountability

[Küsters, Truderung, Vogt 2010]

- Fairness: Judge never blames protocol participants who run their honest program.
- Completeness: If misbehavior of participants causes protocol goal to fail, judge blames some subset of those participants.

Accountability

[Küsters, Truderung, Vogt 2010]

$$!G \Rightarrow v_1 \mid v_2 \mid \ldots \mid v_n$$

G is goal, a set of protocol traces
v is verdict, which assigns blame to subset

Accountability

[Küsters, Truderung, Vogt 2010]

verdict could be... $\mathrm{dis}(A) \mid \mathrm{dis}(V_1) \mid \mathrm{dis}(V_2)$

$\mathrm{dis}(A) \vee \mathrm{dis}(V_1) \vee \mathrm{dis}(V_2)$

$\mathrm{dis}(A) \mid \mathrm{dis}(V_1) \wedge \mathrm{dis}(V_2)$

Accountability

[Küsters, Truderung, Vogt 2010]

Generalizes a definition of verifiability

Accountability...

Verifiability

[Küsters, Truderung, Vogt 2010]

- Adequacy: If some subset of participants are honest in a run, judge accepts run.
- Soundness: If judge accepts a run, then run satisfies protocol goal.

Accountability...

Verifiability

[Küsters, Truderung, Vogt 2010]

$$h \Rightarrow G$$

h is honesty constraint

G is goal

Accountability...

Verifiability

[Küsters, Truderung, Vogt 2010]

honesty contraint could be...

$\mathrm{hon}(A) \vee \mathrm{hon}(V_1) \vee \mathrm{hon}(V_2)$

$\mathrm{hon}(A) \vee (\mathrm{hon}(V_1) \wedge \mathrm{hon}(V_2))$

honesty contraints are negations of (class of) verdicts, where $\mathrm{hon}(A) = !\mathrm{dis}(A)$

Verifiability vs. Accountability

[Küsters, Truderung, Vogt 2010]

If judge provides $!G \Rightarrow !h$ accountability,

then judge provides $h \Rightarrow G$ verifiability.

Verifiability vs. Accountability

[Küsters, Truderung, Vogt 2010]

If judge provides !G ⇒ !h accountability,

then judge provides h ⇒ G verifiability.

(converse holds with additional restrictions)

Verifiability vs. Accountability

[Küsters, Truderung, Vogt 2010]

If judge provides !G ⇒ !h accountability,

then judge provides h ⇒ G verifiability.

(converse holds with additional restrictions)

...accountability generalizes verifiability

Verifiability Verified

- Juels et al.: JCJ, Civitas
- Kremer et al.: FOO'92, Helios 2.0, Civitas
- Küsters et al.: Bingo, ThreeBallot, VAV, Wombat, Helios 2.0

VERIFIABILITY
in electronic voting

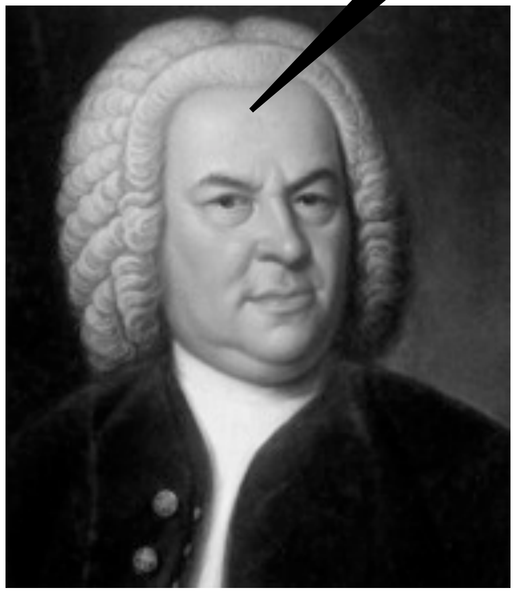
Michael Clarkson
George Washington University

International Summer School on Secure Voting
July 16, 2012

ACT II

Verification Tasks

- Cast as intended
- Recorded as cast
- Counted as recorded

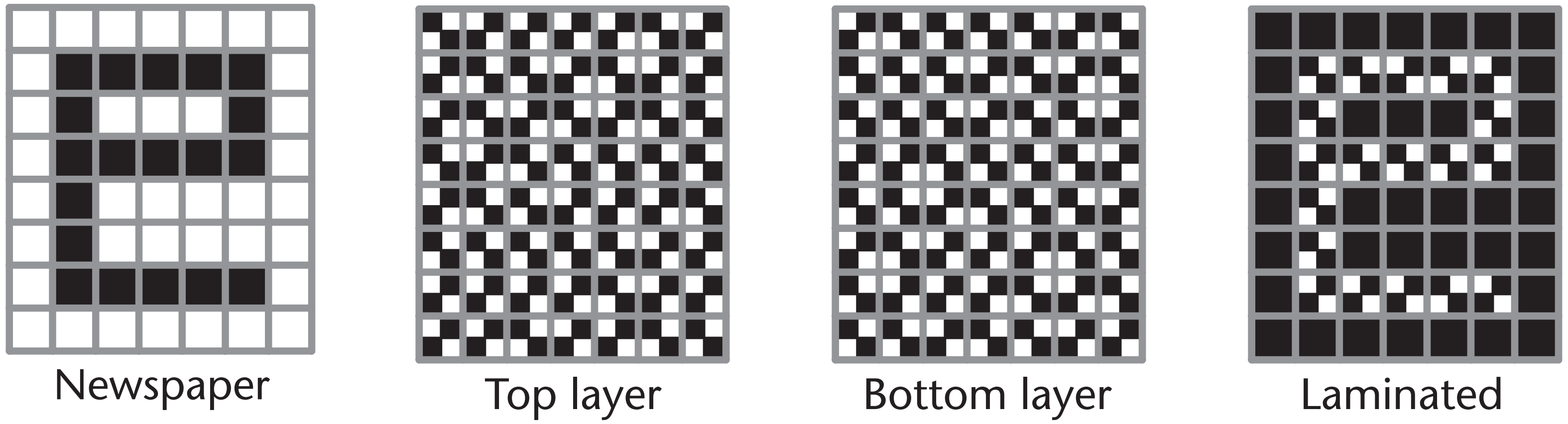


1. Bach → BB: enc(vote)

Recorded as Intended

- Two-part ballots [Chaum 2004]
- Cast NAND audit [Benaloh 2006]
- Proofs for people [Neff 2004]

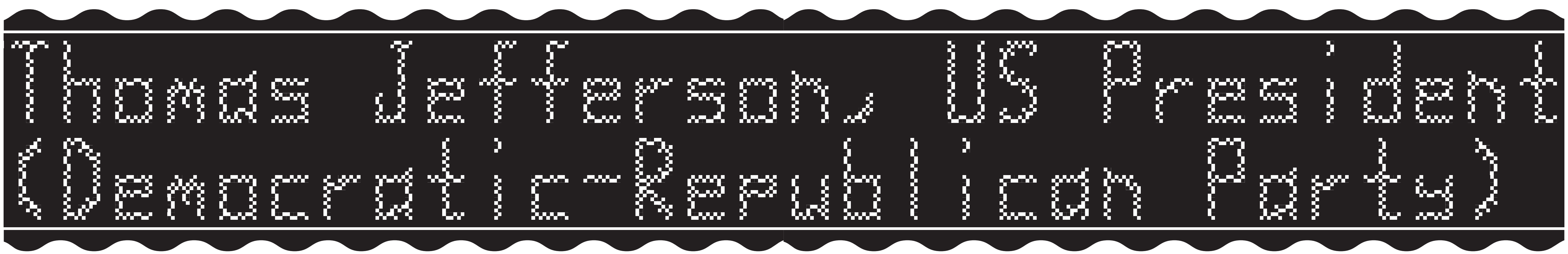
Two-part Ballots



Visual cryptography [Naor and Shamir 1994]

Two-part Ballots

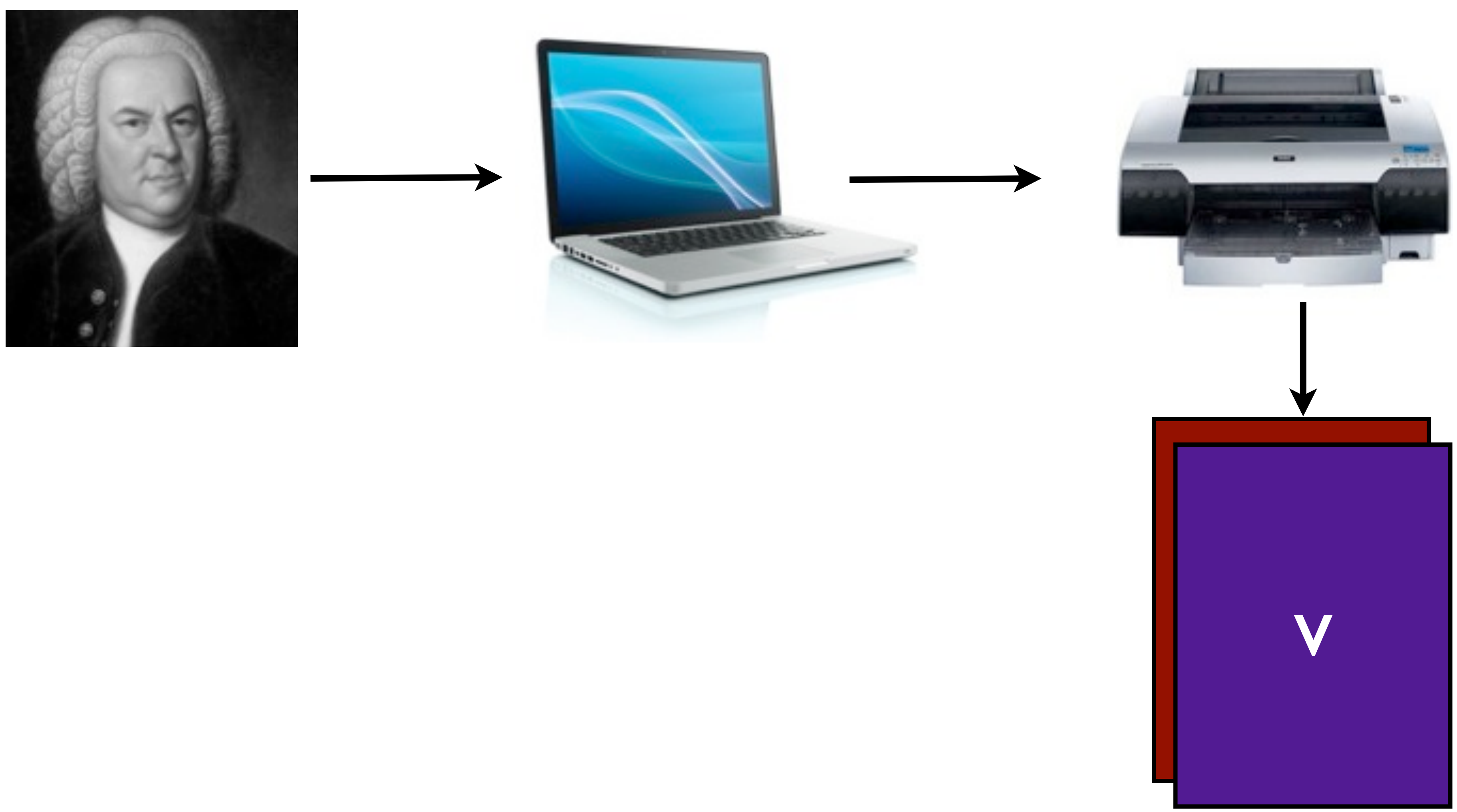
[Chaum 2004]



elaborated into non-visual form by Ryan (2004);
idea now a basis for Pret à Voter (Ryan et al.)
and Scantegrity II (Chaum et al.)

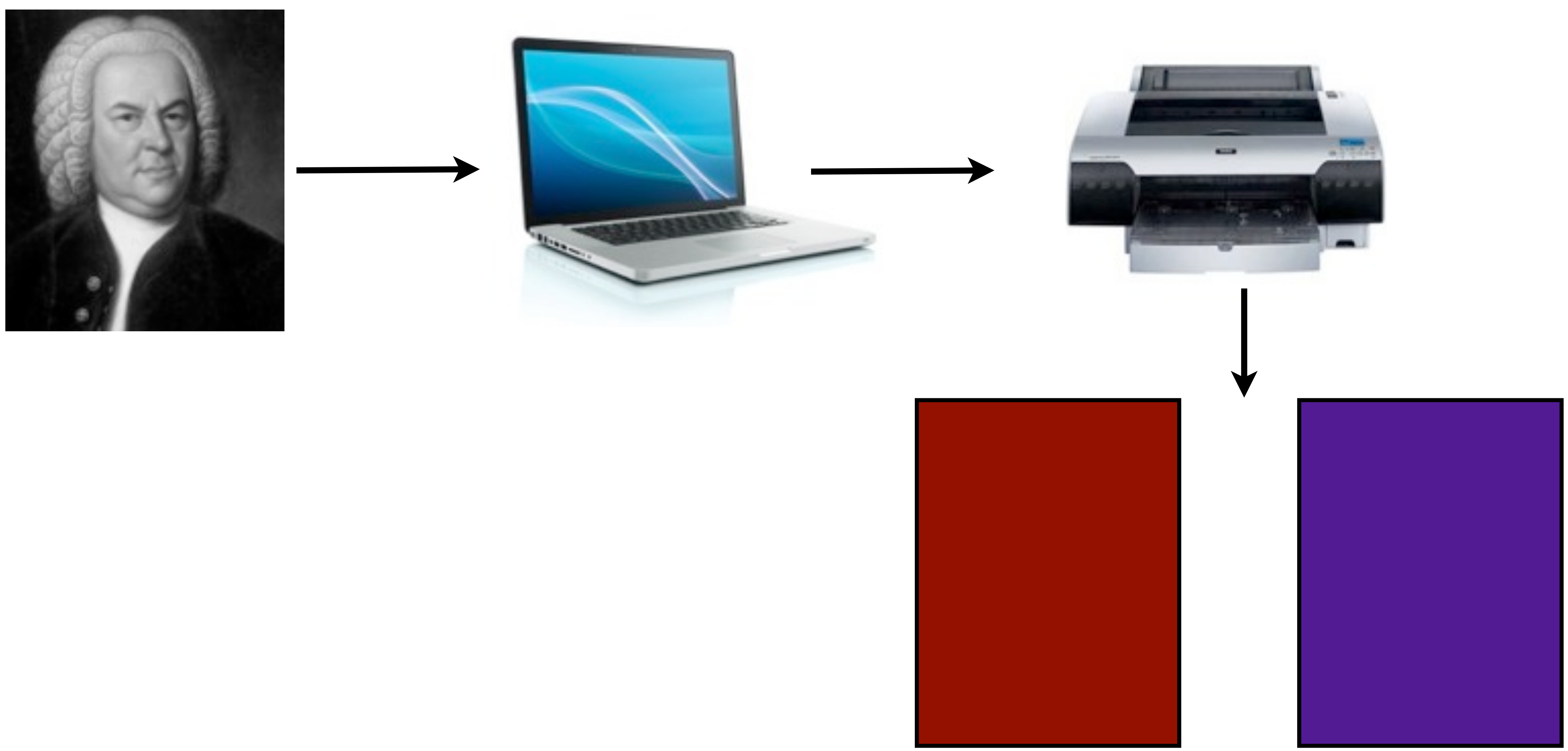
Two-part Ballots

[Chaum 2004]



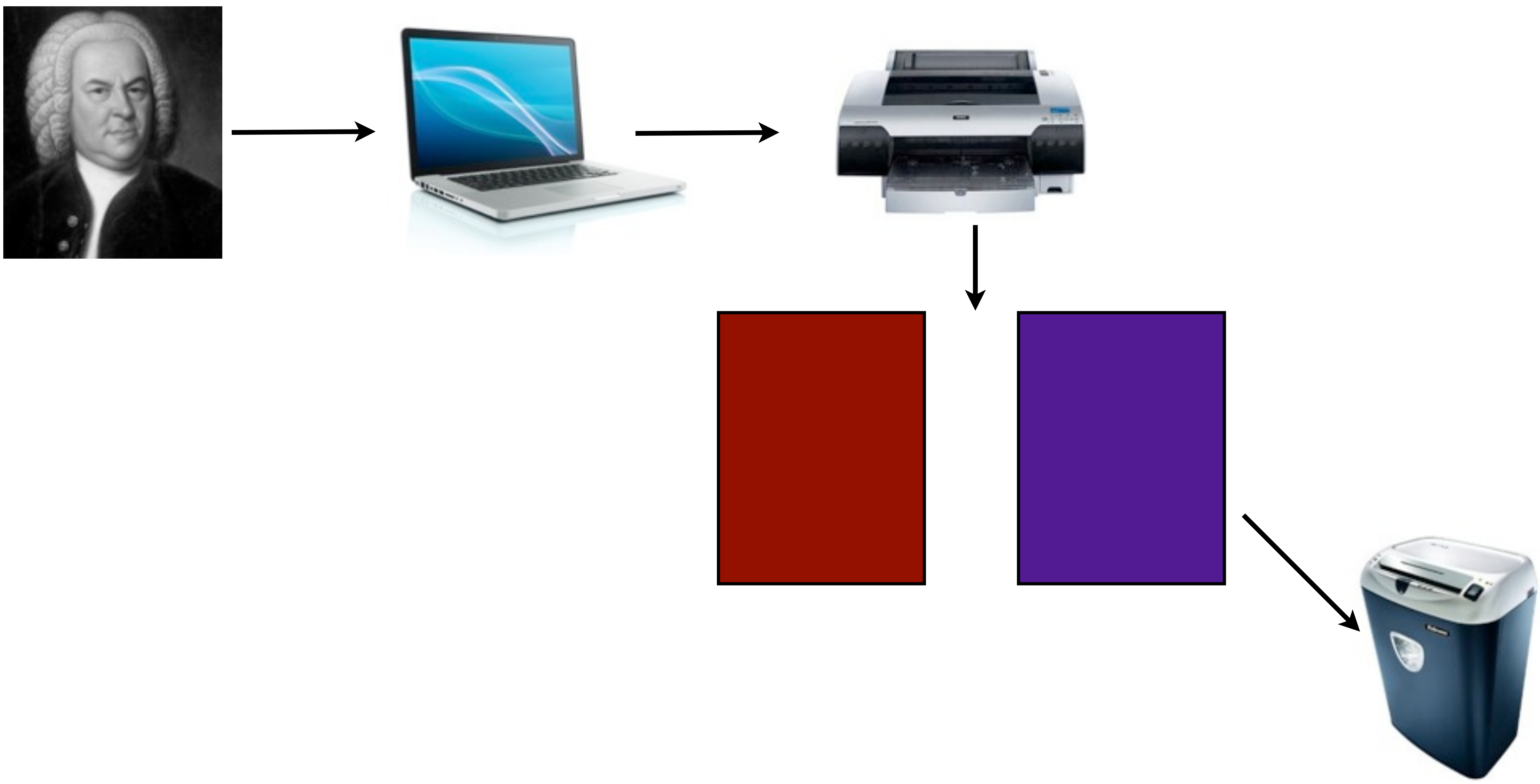
Two-part Ballots

[Chaum 2004]



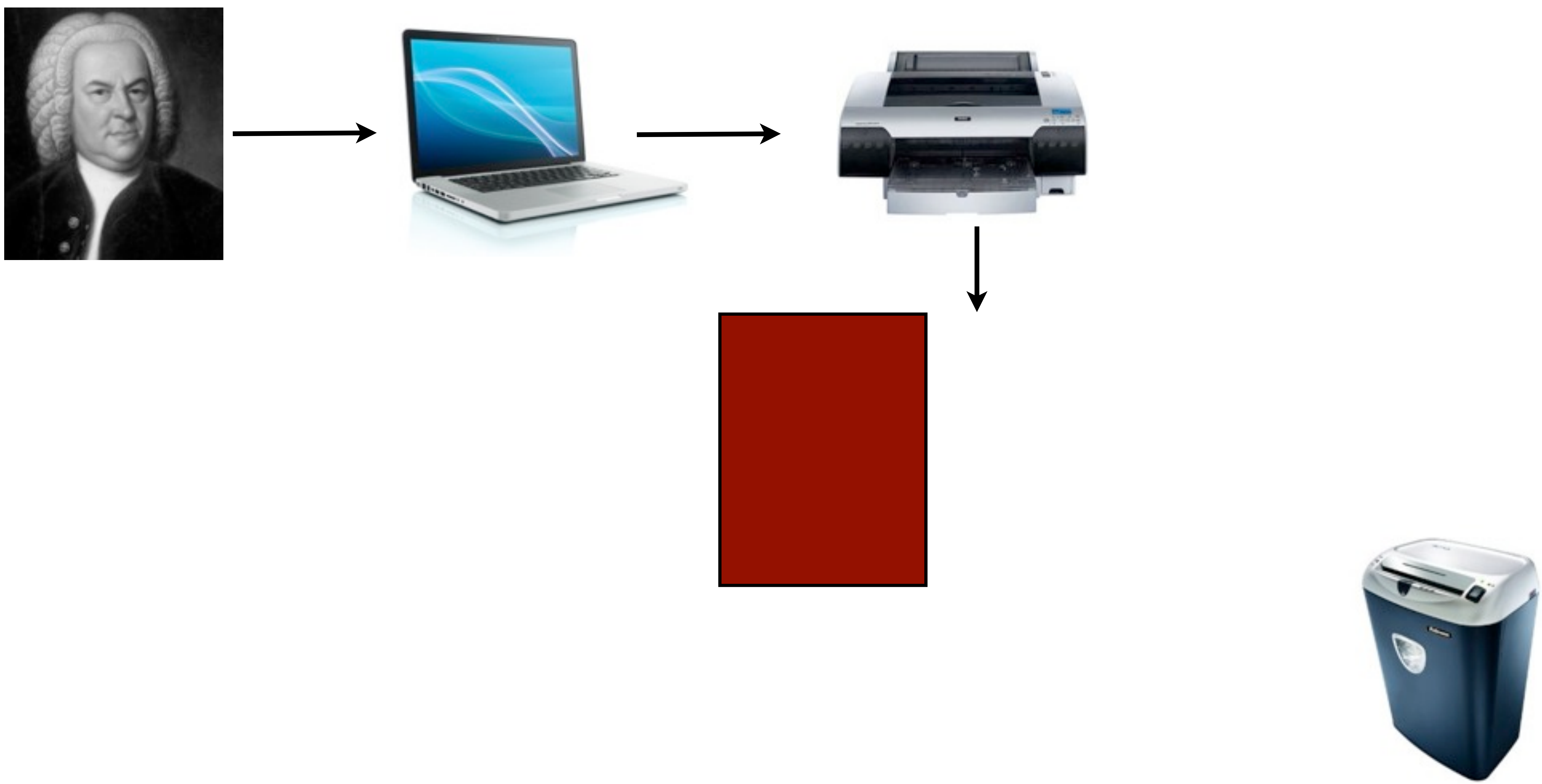
Two-part Ballots

[Chaum 2004]



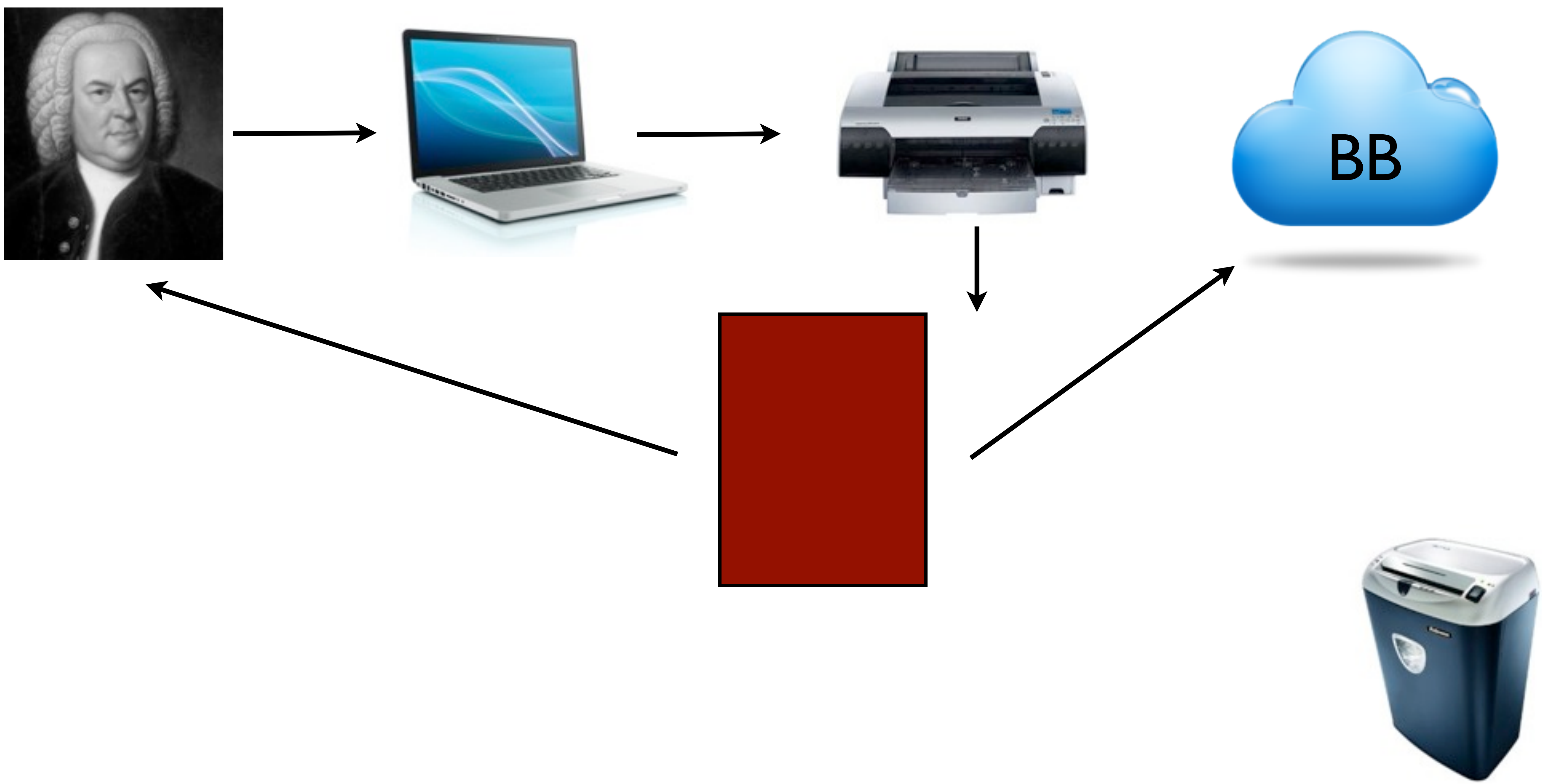
Two-part Ballots

[Chaum 2004]



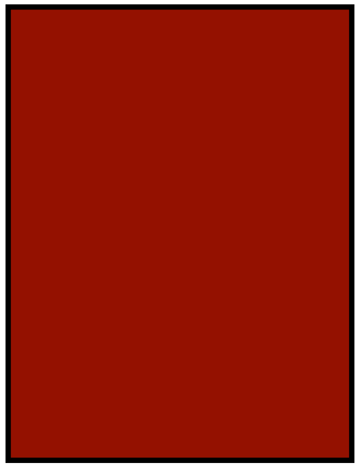
Two-part Ballots

[Chaum 2004]



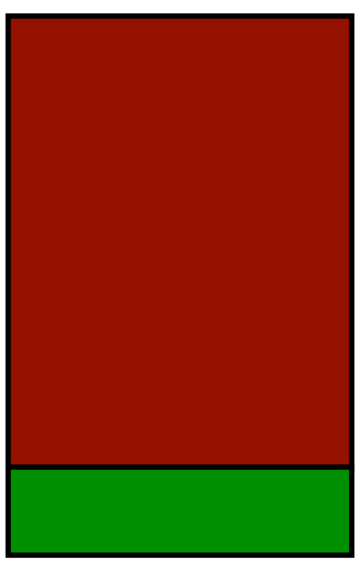
Two-part Ballots

[Chaum 2004]



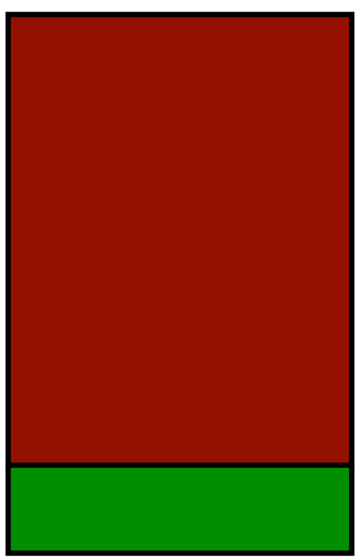
Two-part Ballots

[Chaum 2004]



Two-part Ballots

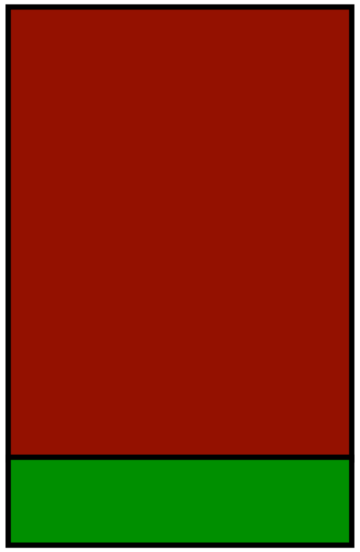
[Chaum 2004]



+

Two-part Ballots

[Chaum 2004]

 +  = V

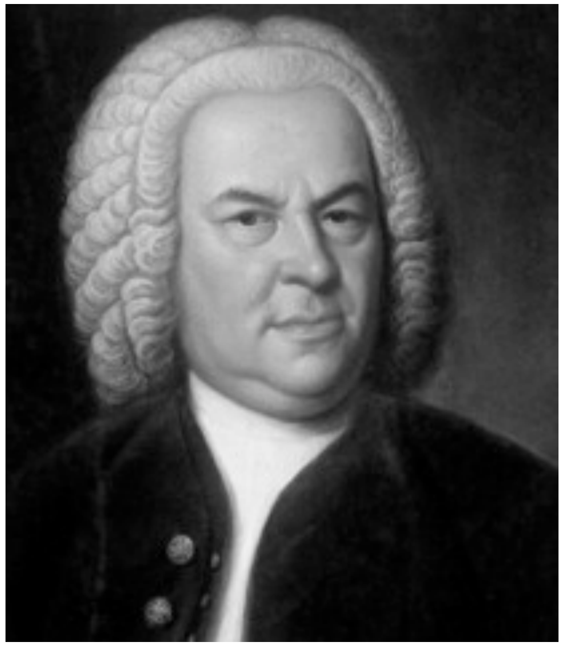
Cast NAND Audit

[Benaloh 2006]

Used in Helios 1.0, 2.0 [Adida]
and VoteBox [Sander, Derr, and Wallach 2008]

Cast NAND Audit

[Benaloh 2006]

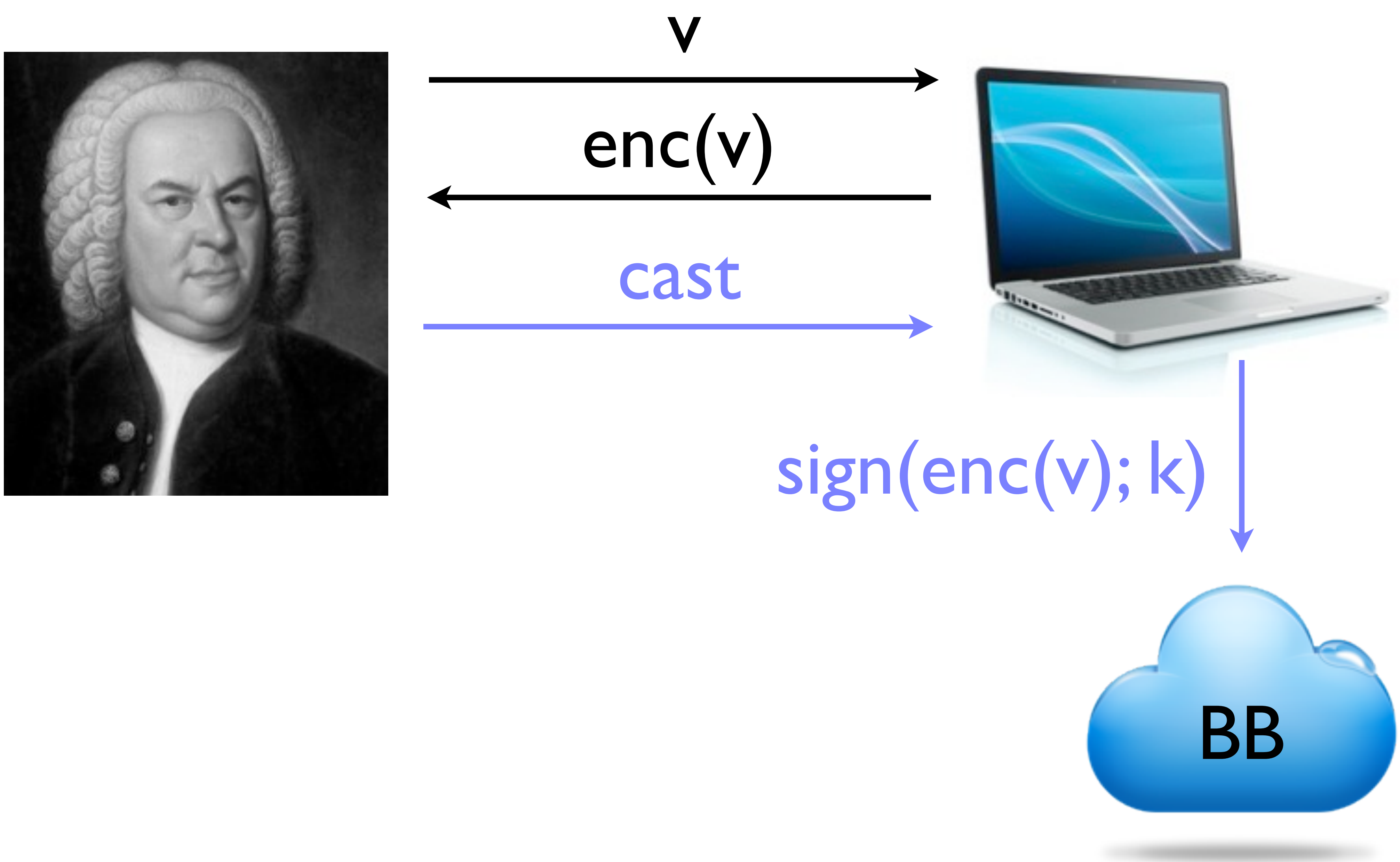


v →

enc(v) ←

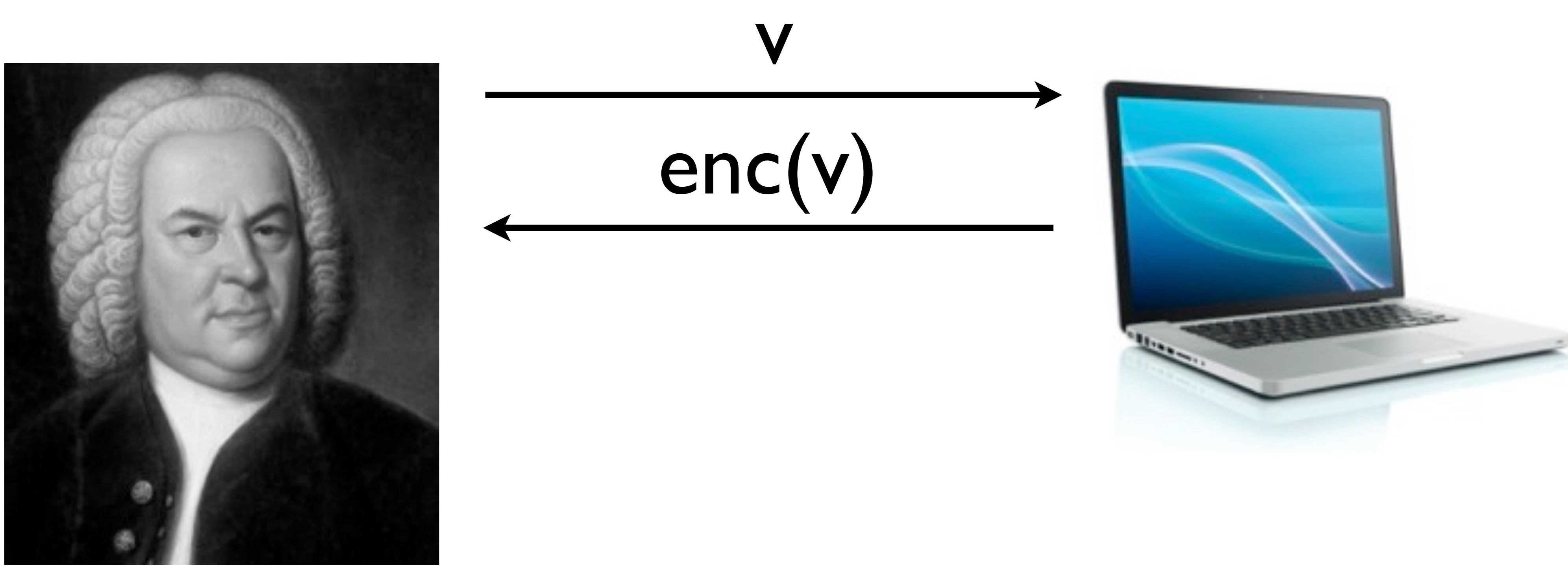
Cast NAND Audit

[Benaloh 2006]



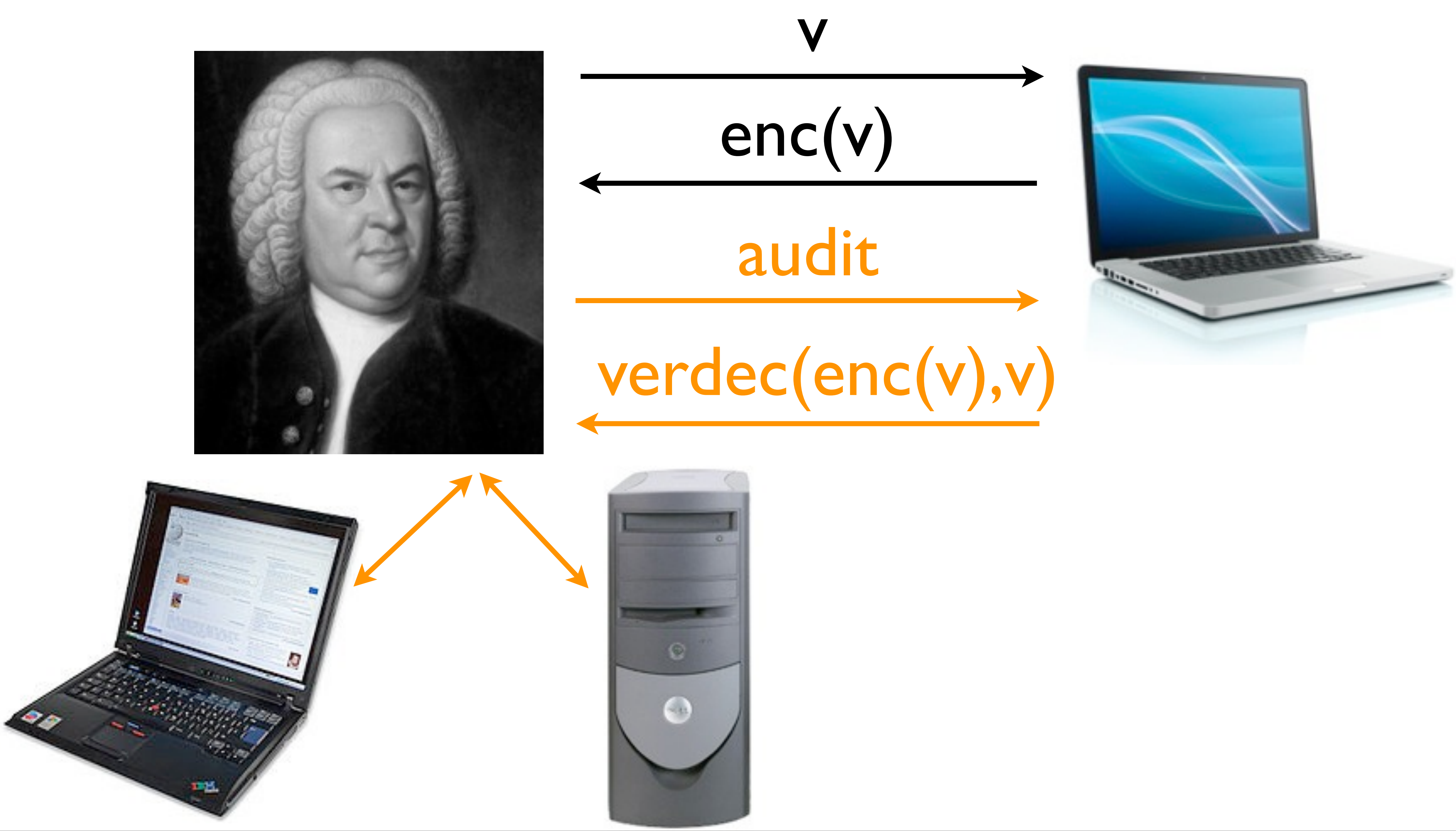
Cast NAND Audit

[Benaloh 2006]



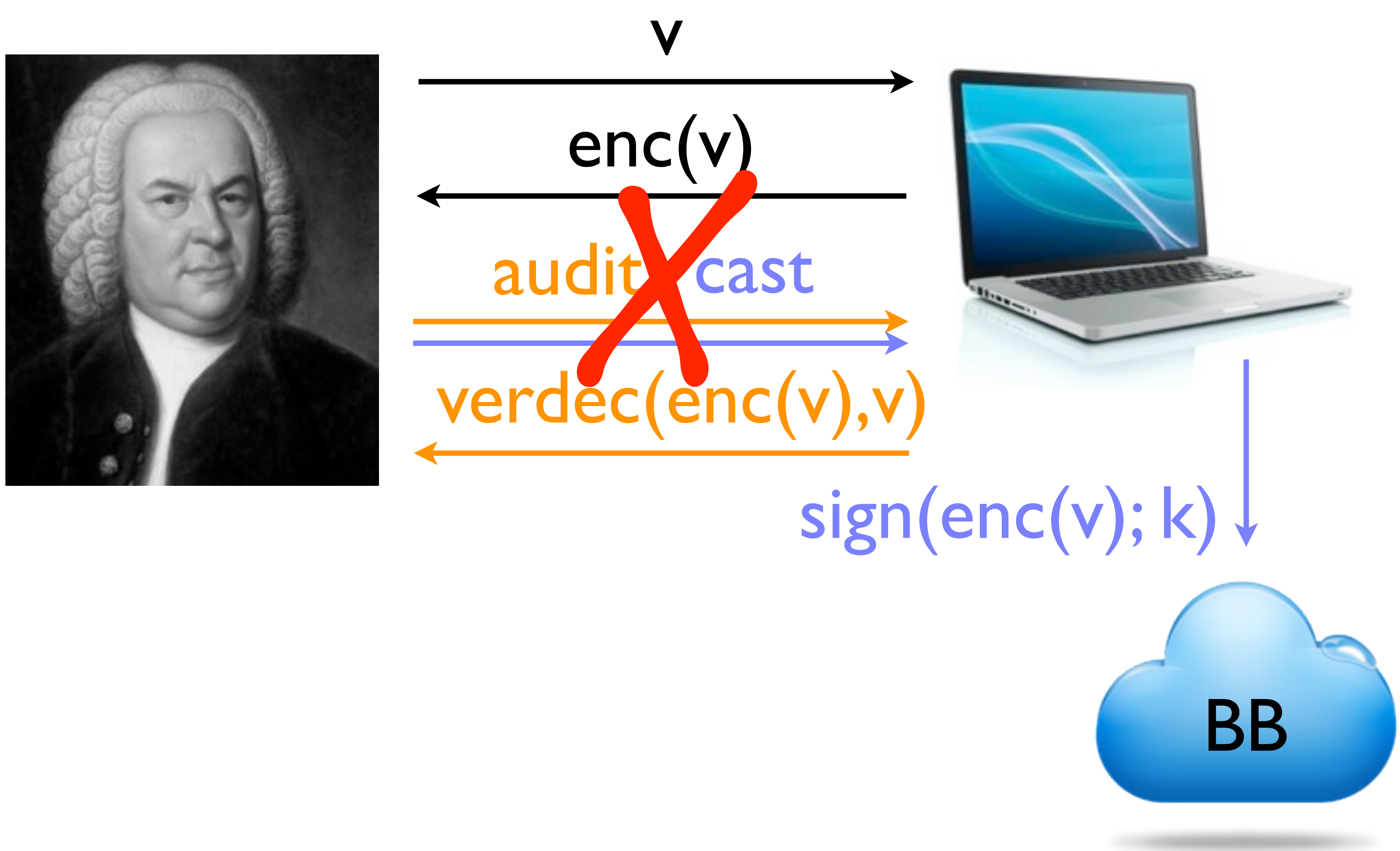
Cast NAND Audit

[Benaloh 2006]



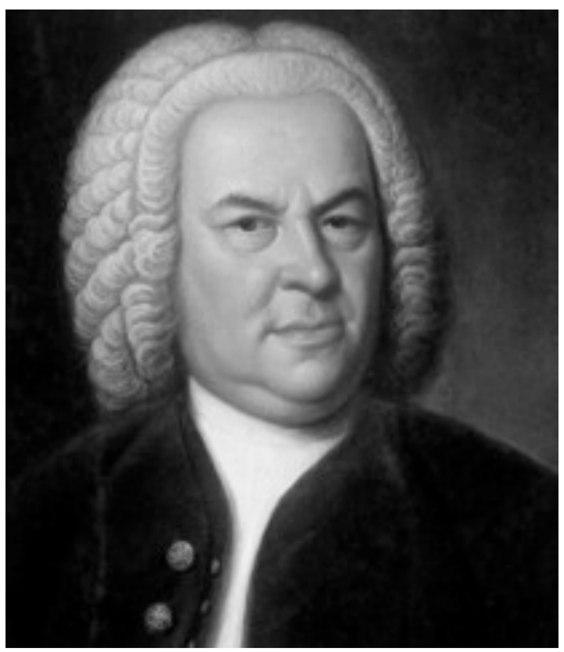
Cast NAND Audit

[Benaloh 2006]



Cast NAND Audit

[Benaloh 2006]



v

enc(v)

audit ~~/~~ cast

verdec(enc(v),v)

sign(enc(v); k)

...could prove how you voted

Proofs for People

[Neff 2004, Adida and Neff 06]

Used in VoteHere (Neff)

Proofs for People

[Neff 2004, Adida and Neff 06]

Prover

Verifier

Proofs for People

[Neff 2004, Adida and Neff 06]

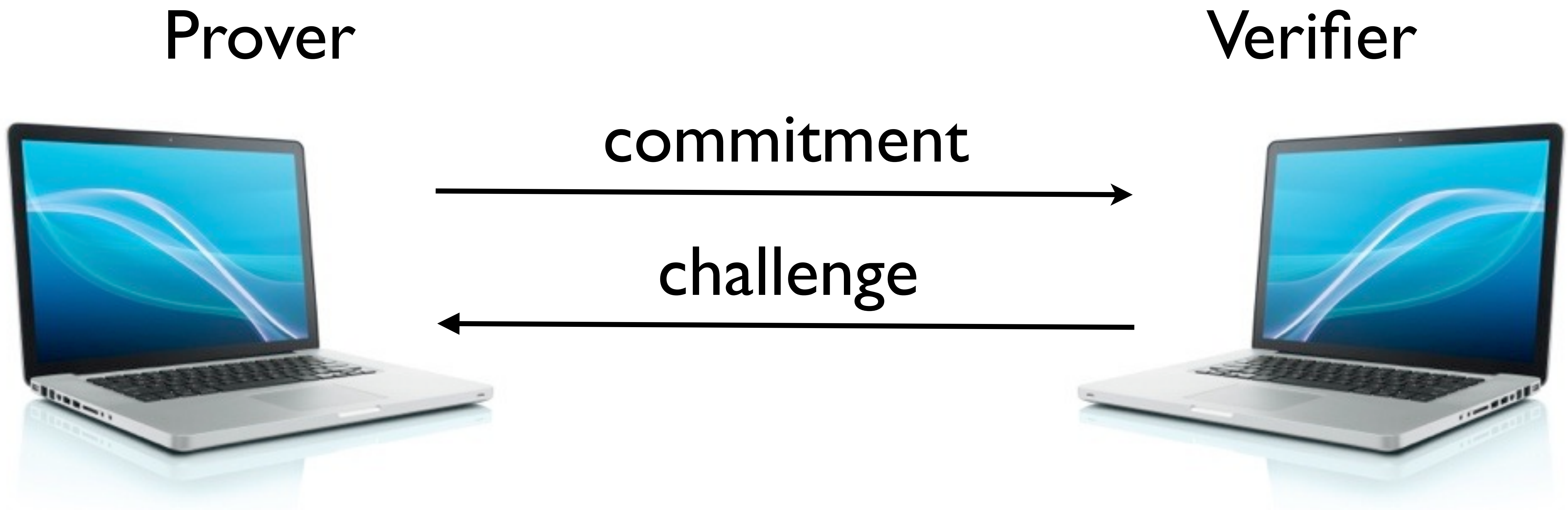
Prover

Verifier

commitment

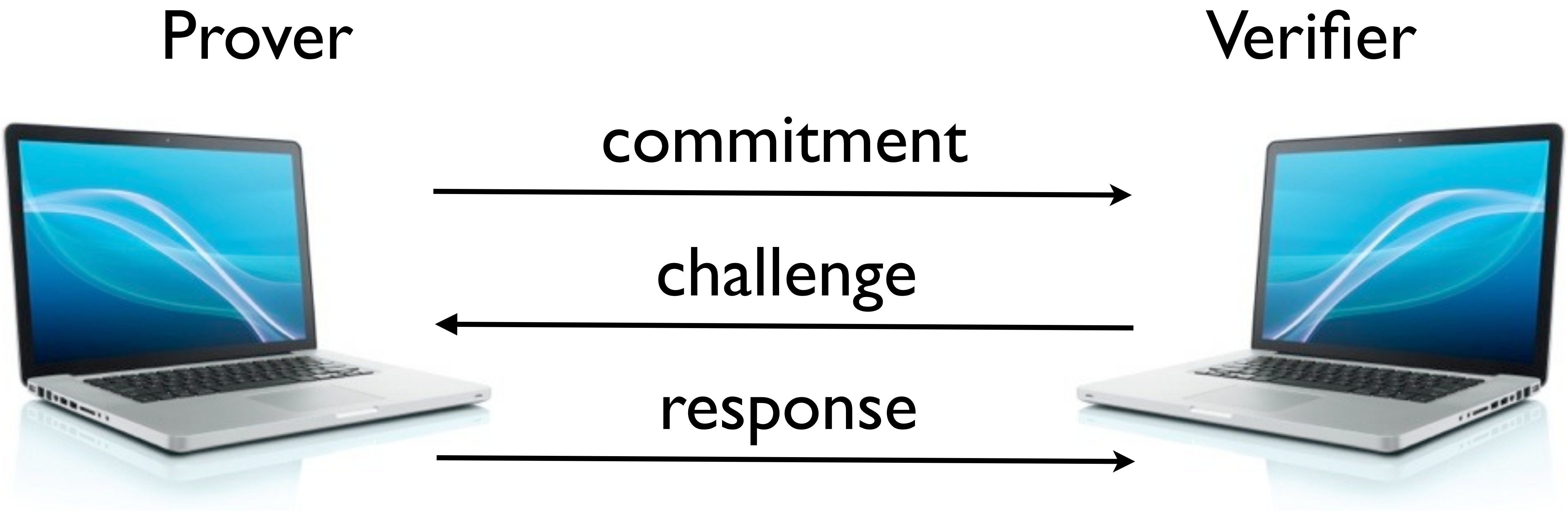
Proofs for People

[Neff 2004, Adida and Neff 06]



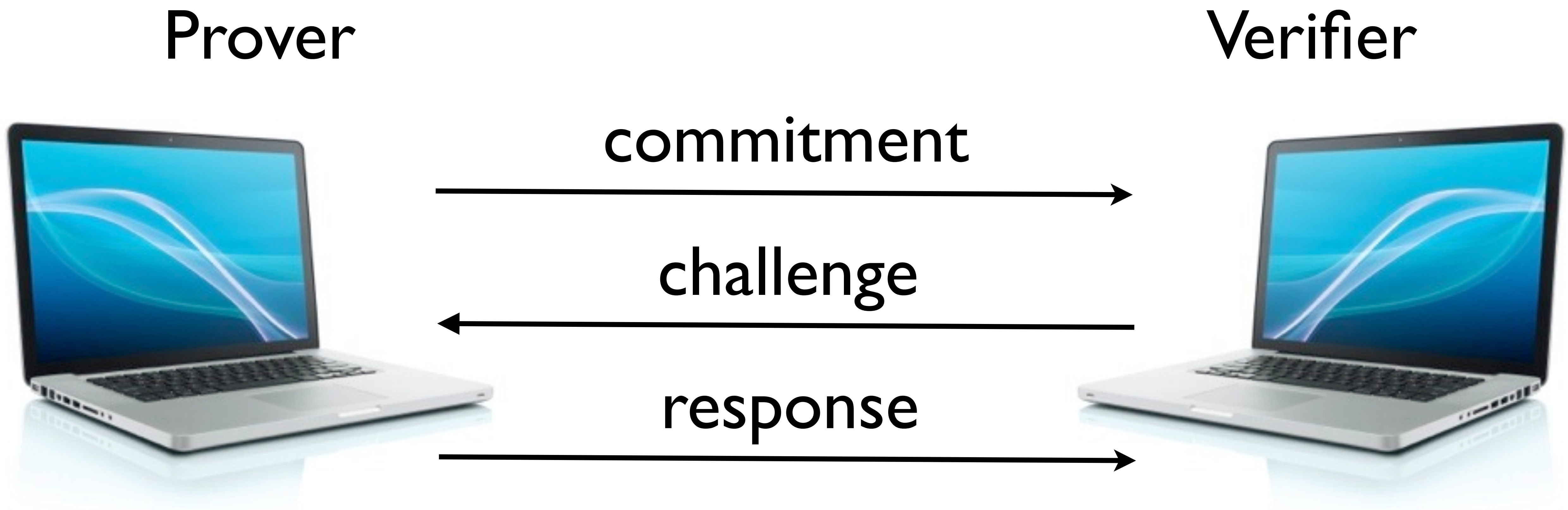
Proofs for People

[Neff 2004, Adida and Neff 06]



Proofs for People

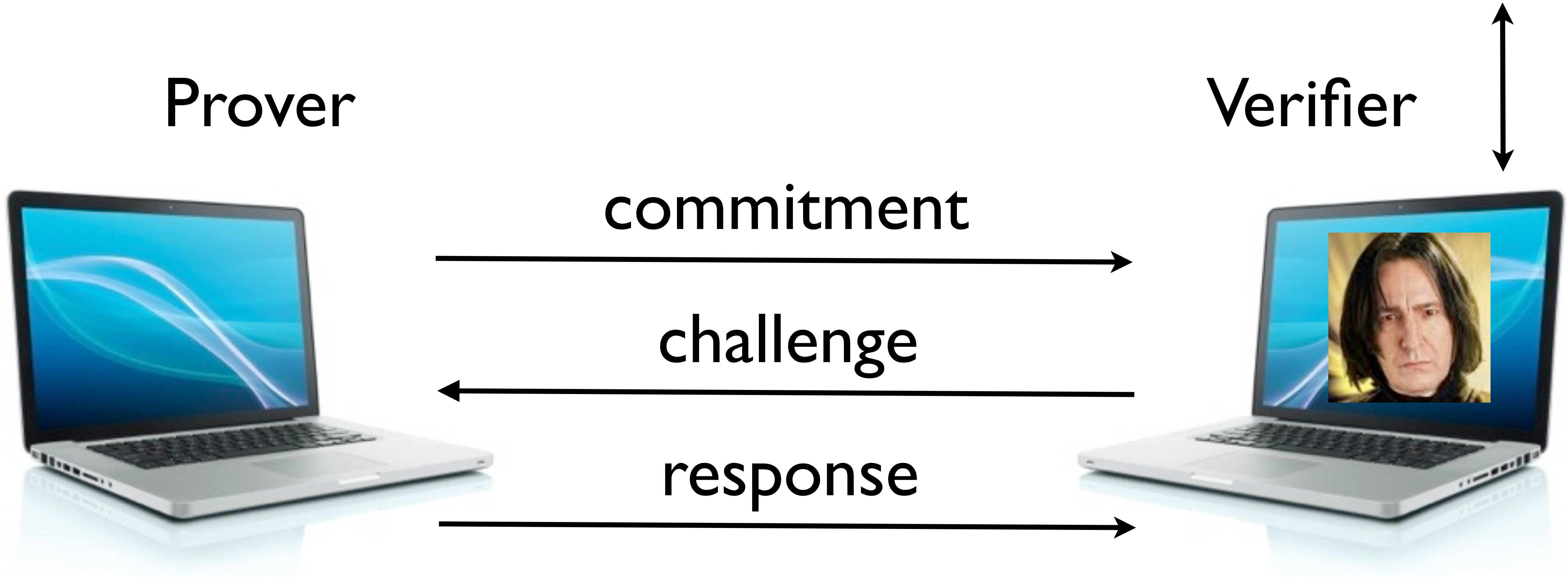
[Neff 2004, Adida and Neff 06]



Problem: people must trust machines

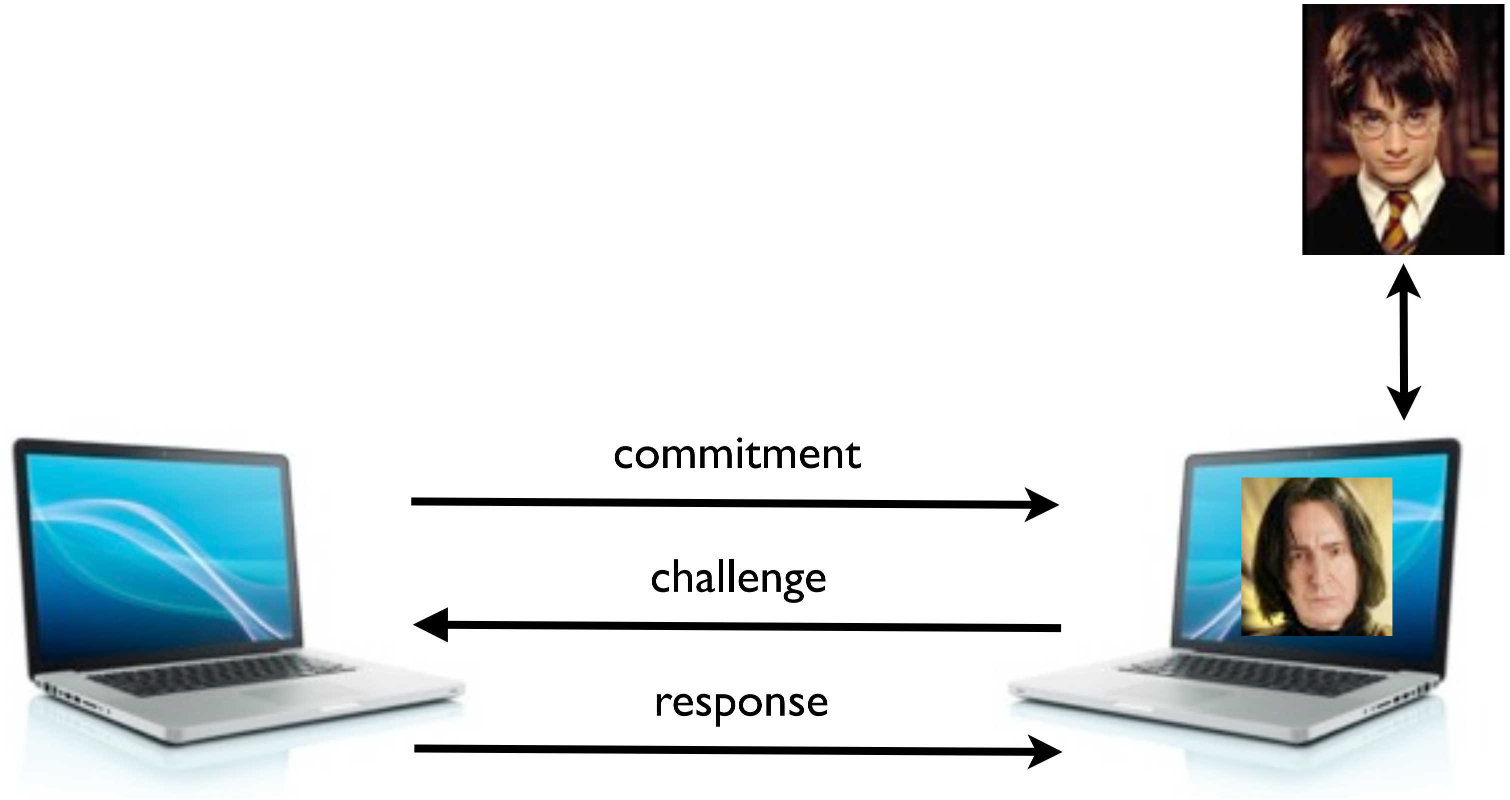
Proofs for People

[Neff 2004, Adida and Neff 06]



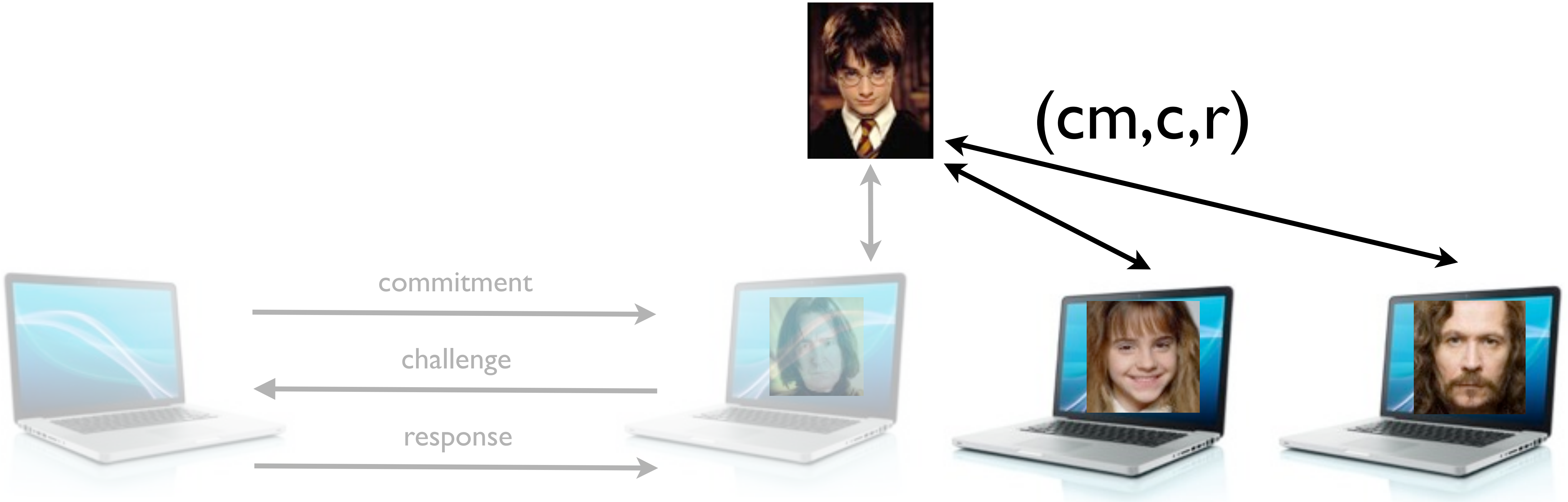
Problem: people must trust machines

Proofs for People

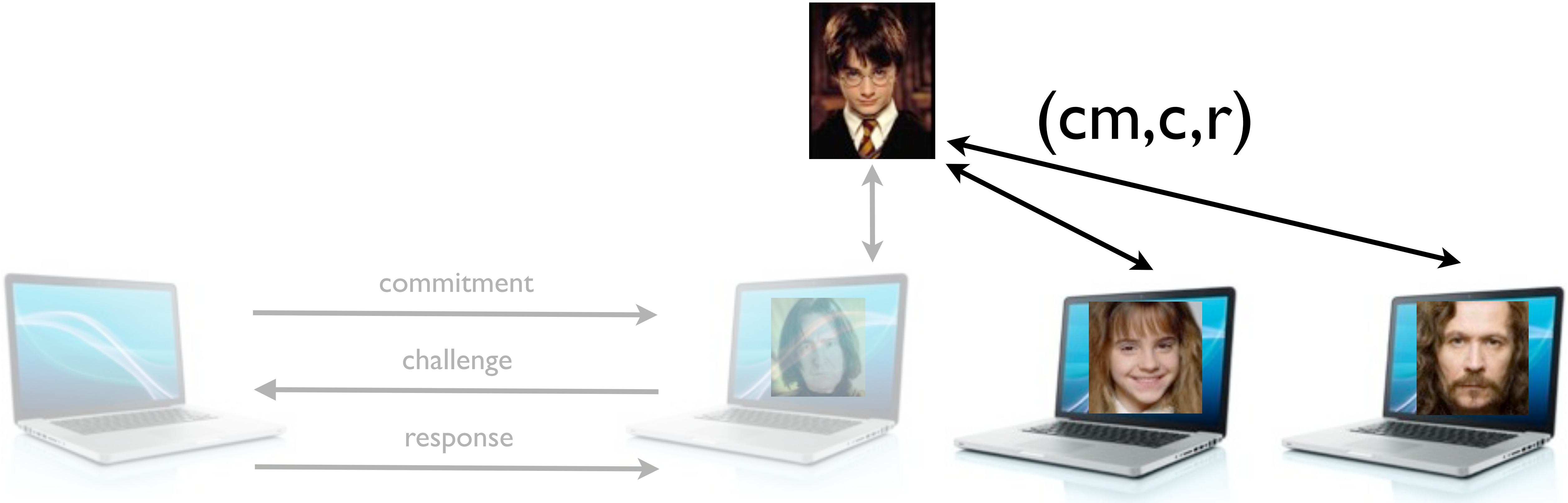


Proofs for People

Proofs for People

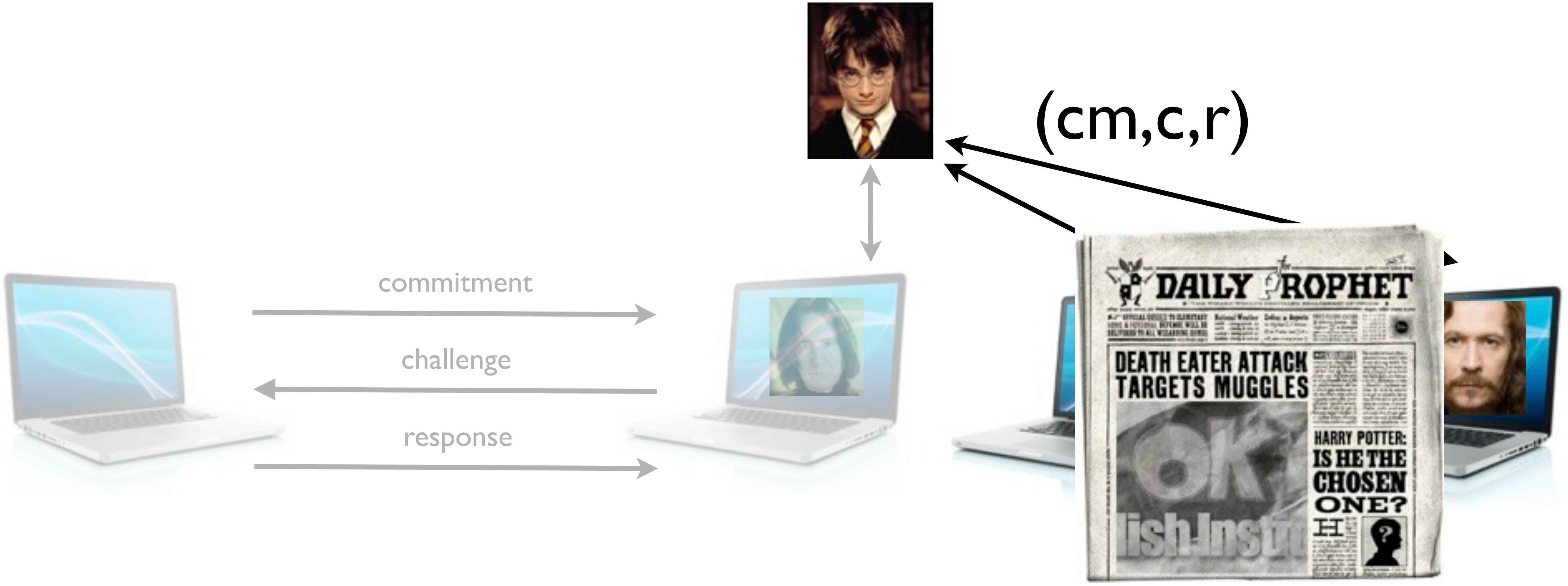


Proofs for People



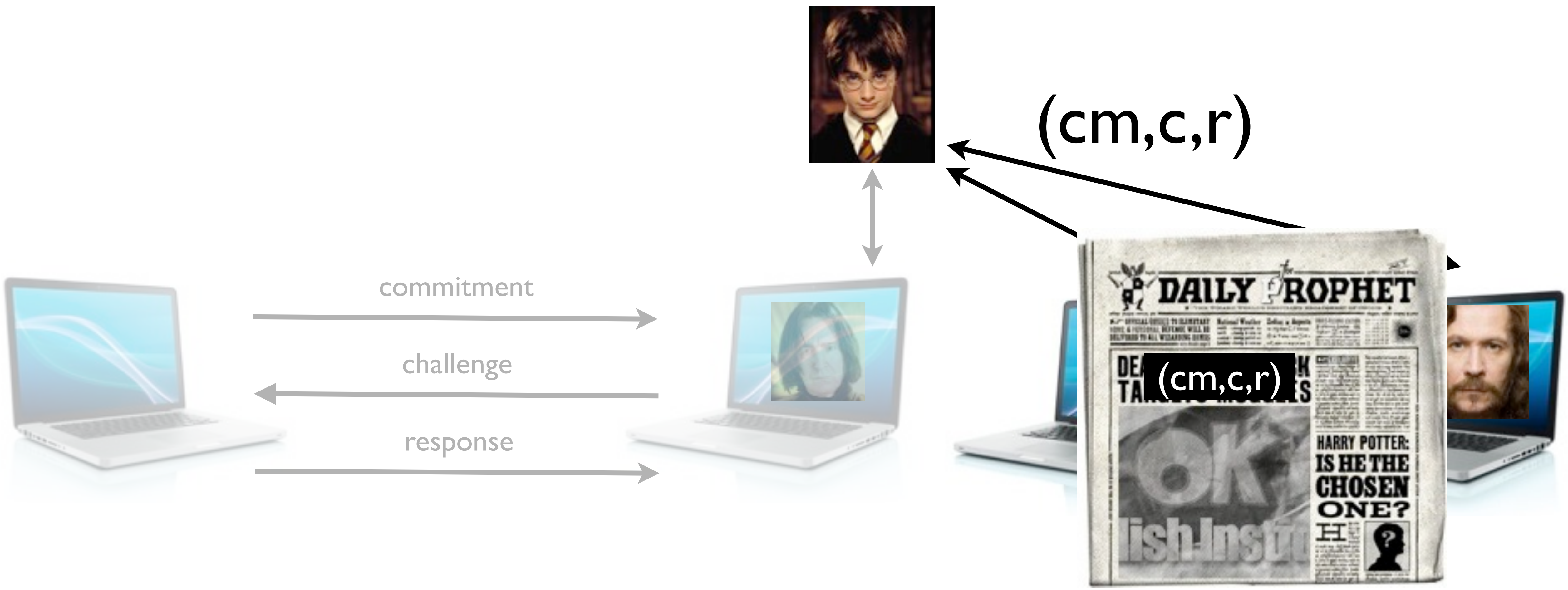
Repetition establishes truth without revealing secret

Proofs for People



Repetition establishes truth without revealing secret

Proofs for People



Repetition establishes truth without revealing secret

MarkPledge

[Neff 2004]

MarkPledge

[Neff 2004]

Ron

MarkPledge

[Neff 2004]

Ron

Ron: enc(1)
Draco: enc(0)

MarkPledge

[Neff 2004]

Ron

Ron: enc(1)
Draco: enc(0)

RECEIPT

MarkPledge

[Neff 2004]

Ron

Ron: enc(1)
Draco: enc(0)

RECEIPT

Ron: enc(1)
Draco: enc(0)

MarkPledge

[Neff 2004]

Ron

Ron: enc(1)
Draco: enc(0)

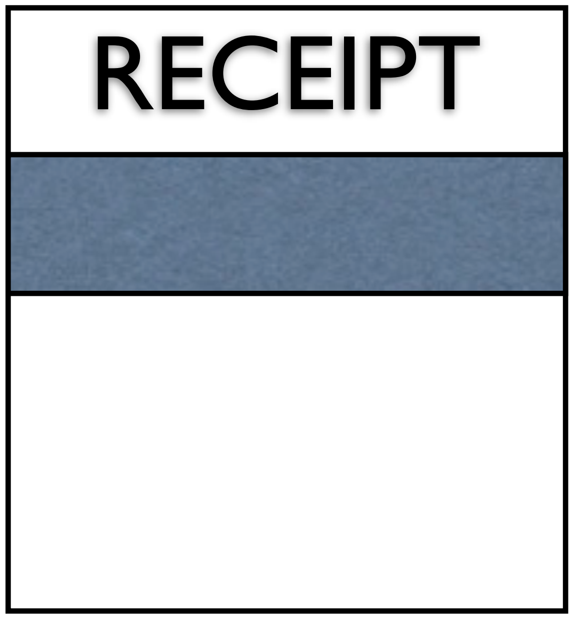
RECEIPT

MarkPledge

[Neff 2004]

Ron

Ron: enc(1)
Draco: enc(0)



MarkPledge

[Neff 2004]

Ron

7rtb

Ron: enc(1)
Draco: enc(0)

RECEIPT

MarkPledge

[Neff 2004]

Ron

7rtb

Ron: enc(1)
Draco: enc(0)

RECEIPT

Challenge: 7rtb

MarkPledge

[Neff 2004]

Ron

7rtb

Ron: enc(1)
Draco: enc(0)

RECEIPT

Challenge: 7rtb

Ron: 3m14
Draco: 0c8d

MarkPledge

[Neff 2004]

Ron

7rtb

Ron: enc(1)
Draco: enc(0)

RECEIPT

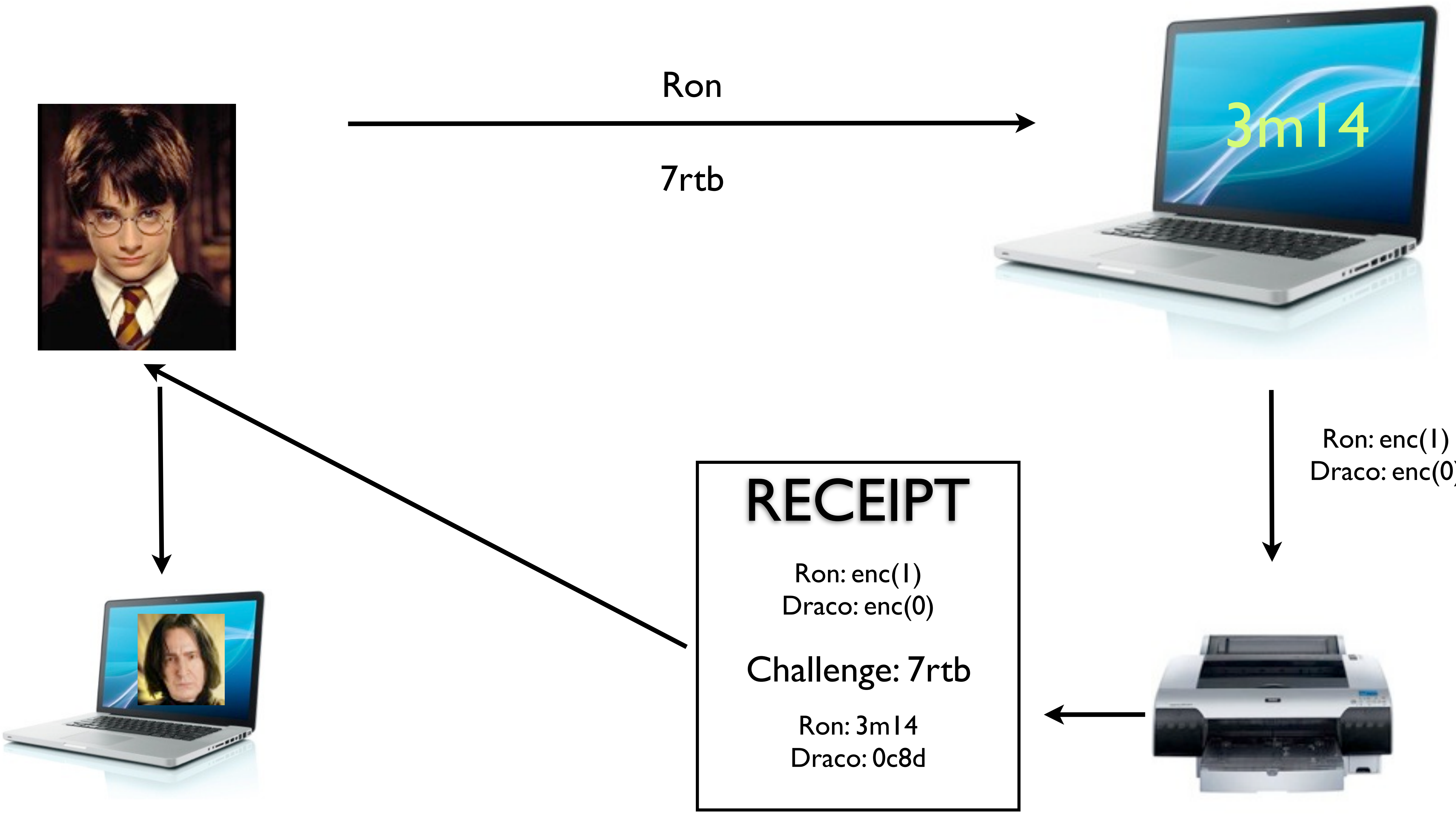
Ron: enc(1)
Draco: enc(0)

Challenge: 7rtb

Ron: 3m14
Draco: 0c8d

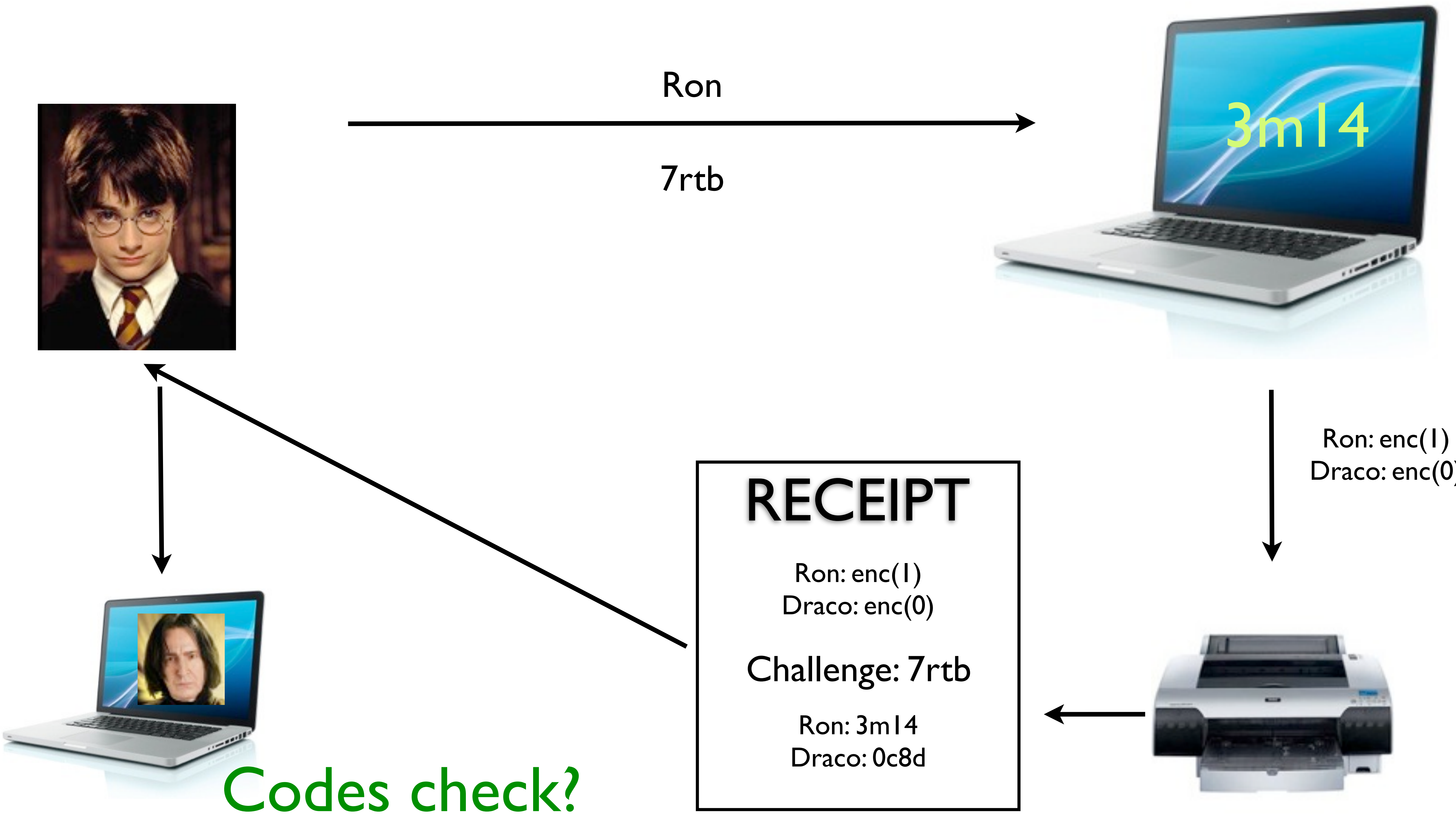
MarkPledge

[Neff 2004]



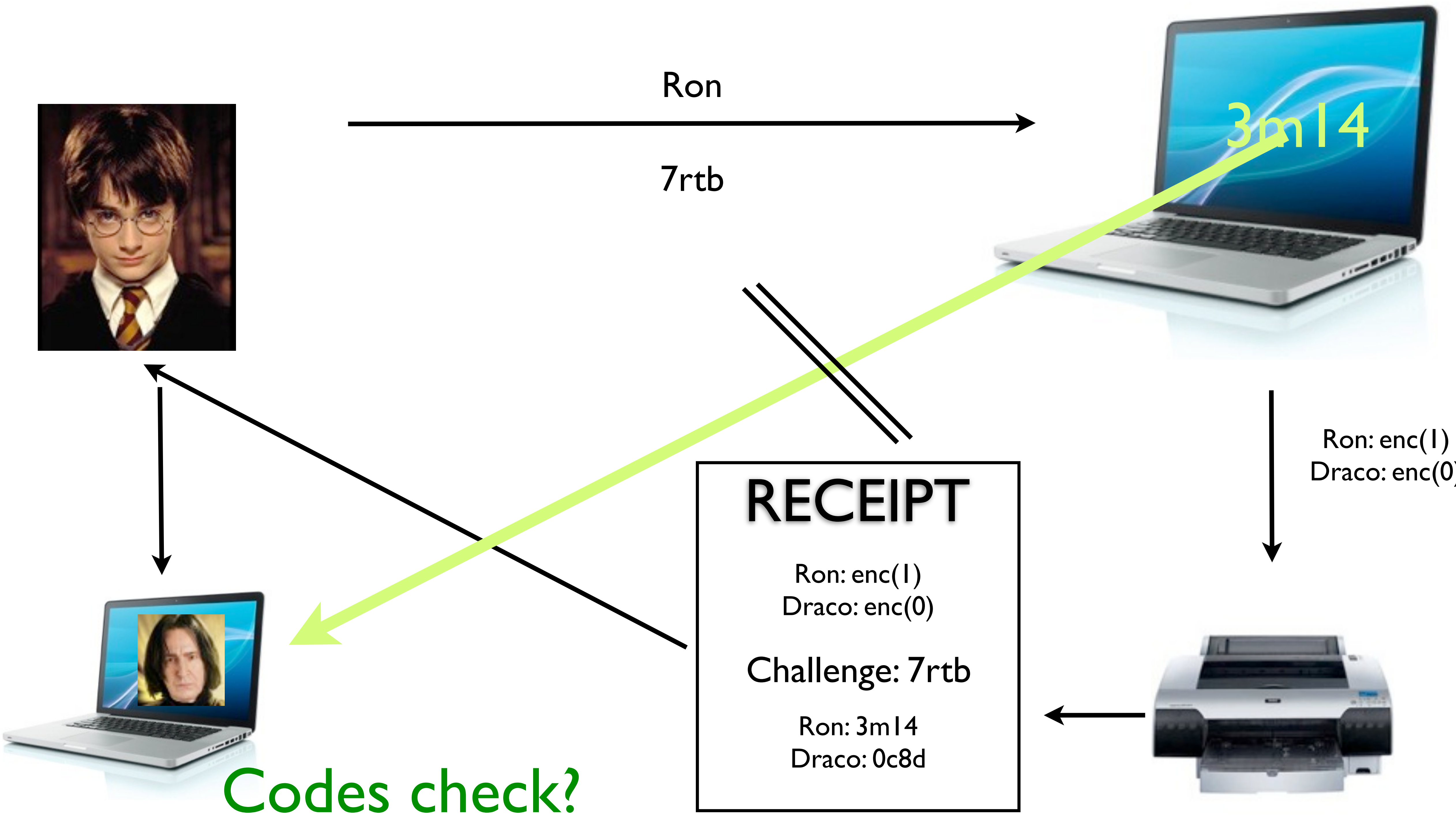
MarkPledge

[Neff 2004]



MarkPledge

[Neff 2004]



Verification Tasks

- Cast as intended
- Recorded as cast
- Counted as recorded

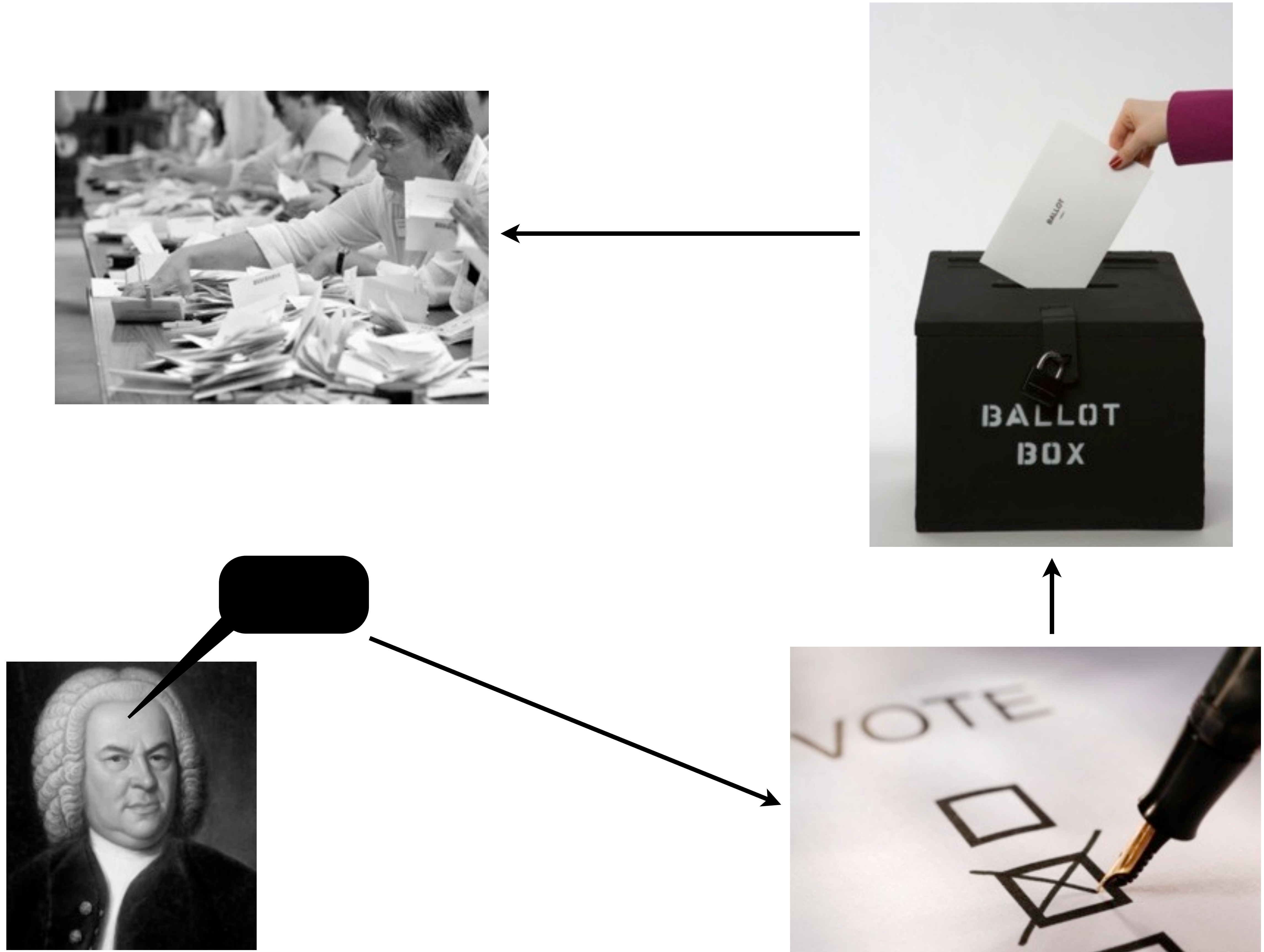
Accomplishments

- Voting machine learns vote
- Voting machine doesn't learn voter identity
- Voter is convinced of correctness of encryption
- Machine doesn't have to be trusted

Accomplishments

- Voting machine learns vote
- Voting machine doesn't learn voter identity
- Voter is convinced of correctness of encryption
- Machine doesn't have to be trusted

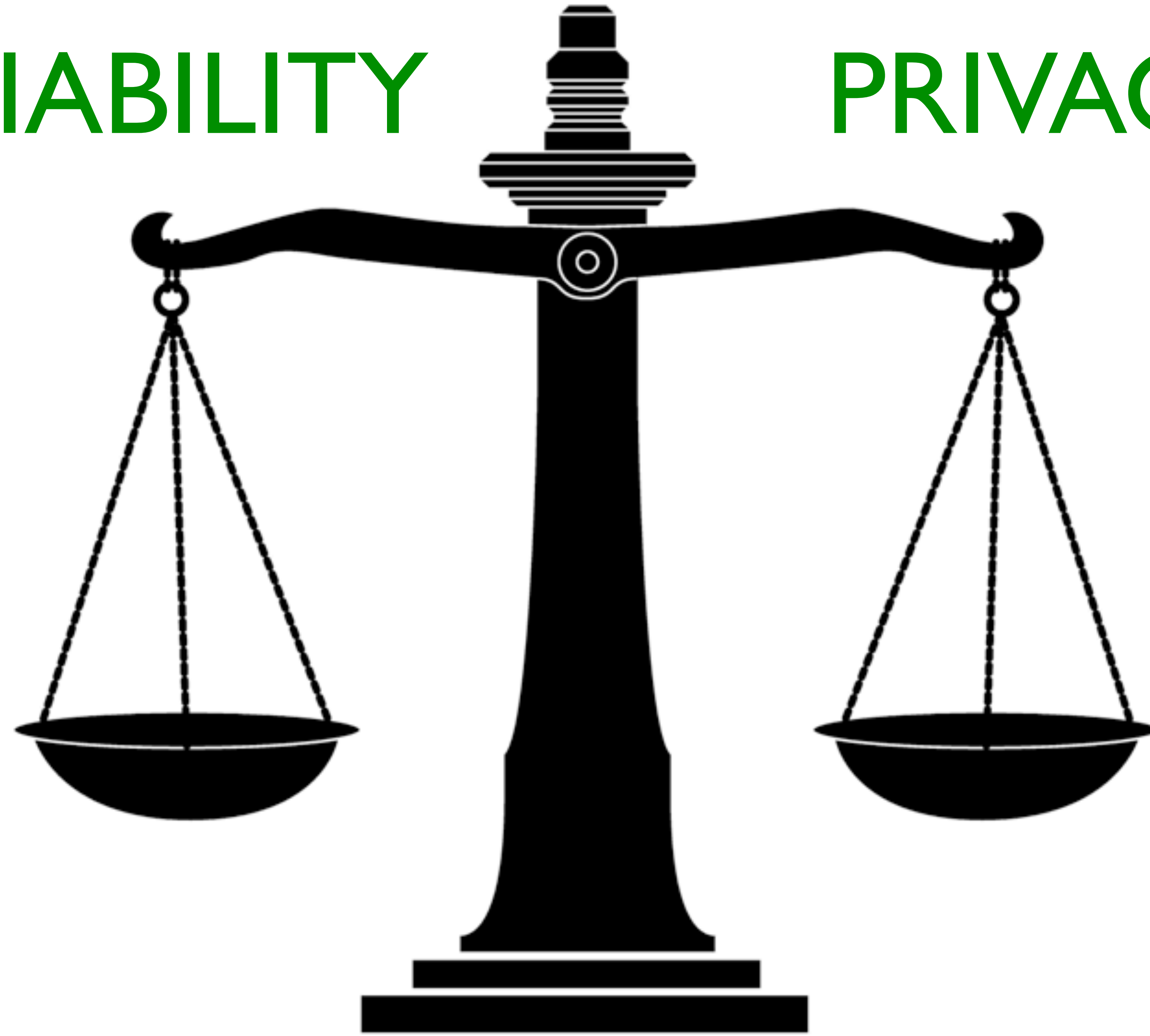
(voter can't be coerced or sell vote)


BALLOT
BOX
VOTE

VERIFIABILITY
in electronic voting

- Formal definitions
- Counted as recorded
- Recorded as intended

VERIFIABILITY



PRIVACY

VERIFIABILITY
in electronic voting

Michael Clarkson
George Washington University

International Summer School on Secure Voting
July 16, 2012