Automated Analysis of Accountability

Alessandro Bruni, Rosario Giustolisi, Carsten Schürmann

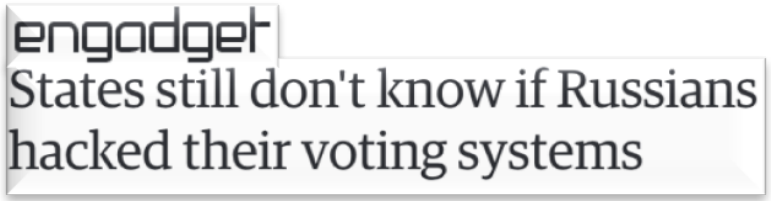
IT UNIVERSITY OF COPENHAGEN

Information Security Conference
23 November 2017
Ho Chi Mihn, Vietnam

Motivation

Motivation

State of the Union



Attacks **persist** $\rightarrow$ Paradigm **shift**:

- From attack avoidance to attack detection, with:
- Verifiability and Accountability, the topic of this presentation

Contributions

- We propose general definitions for verifiability and accountability that are amenable to automated verification in the symbolic model

- We validate the applicability of these definitions in three different case studies:
  1. a secure distributed exam protocol;
  2. the "Bingo Voting" scheme;
  3. Google's Certificate Transparency scheme.

Accountability

Generally ensures that

- The failure of a system's goal is detectable
- Misbehaving principals can be blamed

Stronger than verifiability

Accountability at present

- Individual verifiability
  - voting [SRKM10], auction [DHL13], exams [DGK+15], ...
- Universal verifiability
  - voting [KRS10], auction [DHL13], exams [DGK+15], ...
- Auditability
  - general definition [GFZN09]
- Non-repudiation
  - certified email protocols [BP06, AB03]
- Accountability
  - general definition [KTV10]

Automated Analysis of Accountability

IDEA: Specify the soundness and completeness conditions for (verifiability and) accountability tests that can be checked as reachability properties.

Definition (Protocol)

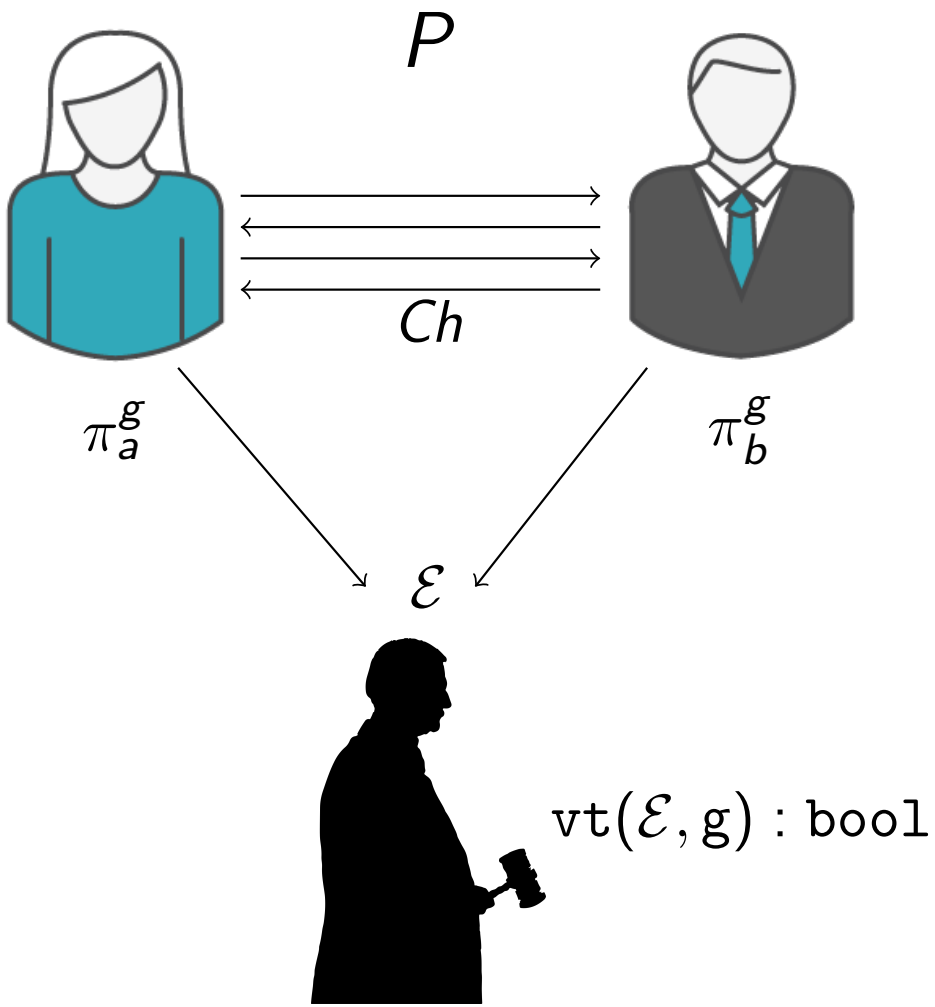
A *protocol* is a tuple $P = \langle \mathsf{Ch}, \mathsf{A}, \Pi, \mathsf{G} \rangle$ such that:

- $\mathsf{Ch} = \{ch_1, \ldots, ch_n\}$ is a set of *channels*;
- $\mathsf{A} = \{\alpha_1, \ldots, \alpha_n\}$ is a set of *principals*;
- Π is the set of *programs* run by the principals;
- G is the set of *goals* that the protocol aims to meet.

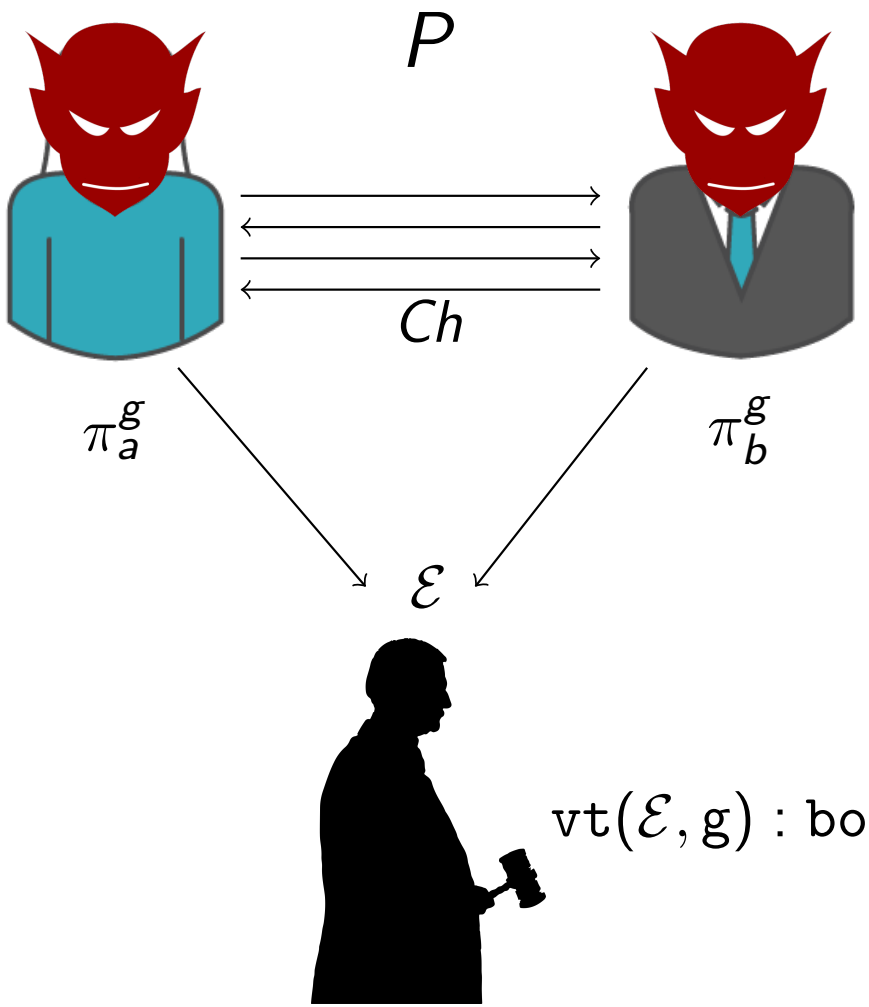
goal-convergent programs

- Π^G is the set of all tuples $\{\pi_{\alpha_i}\}_{\alpha_i \in \mathsf{A}}$
- $\Pi^g_{\alpha_i}$ is the set of α_i's goal-convergent programs

Verifiability



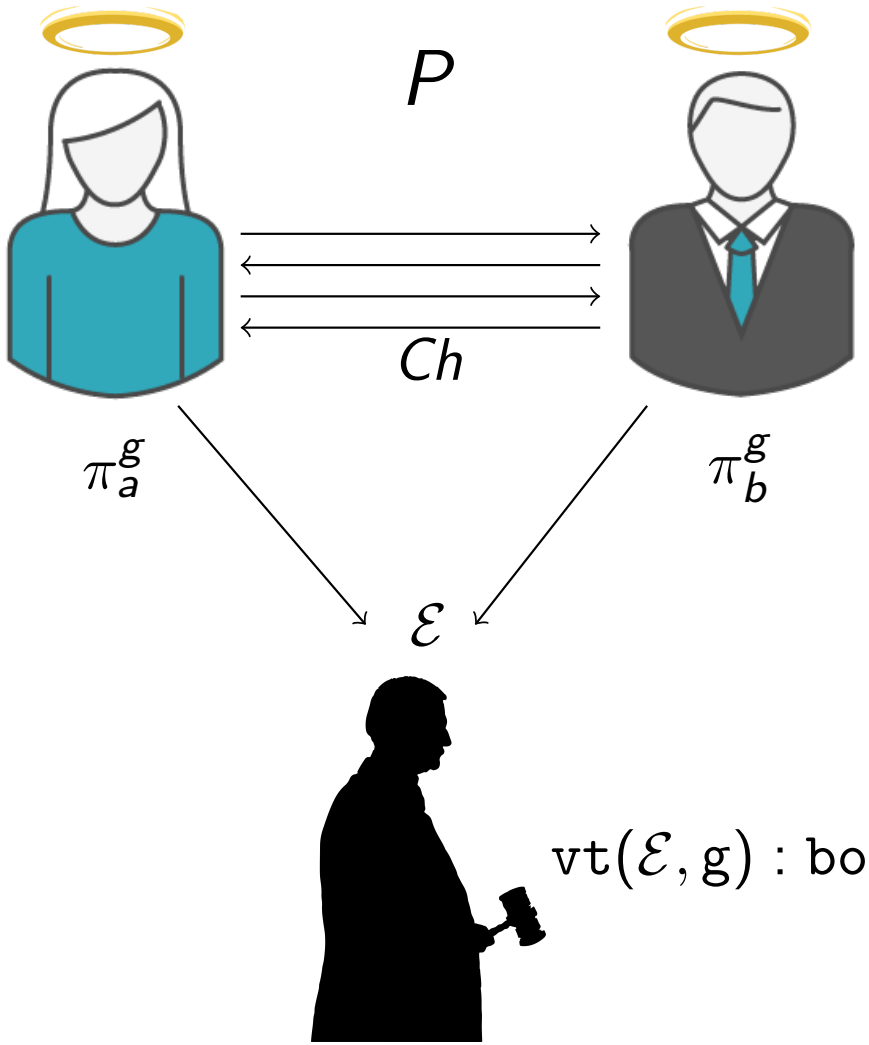
Verifiability



soundness

$\mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{true} \implies \mathrm{g}$ holds in $r(P)$

Verifiability



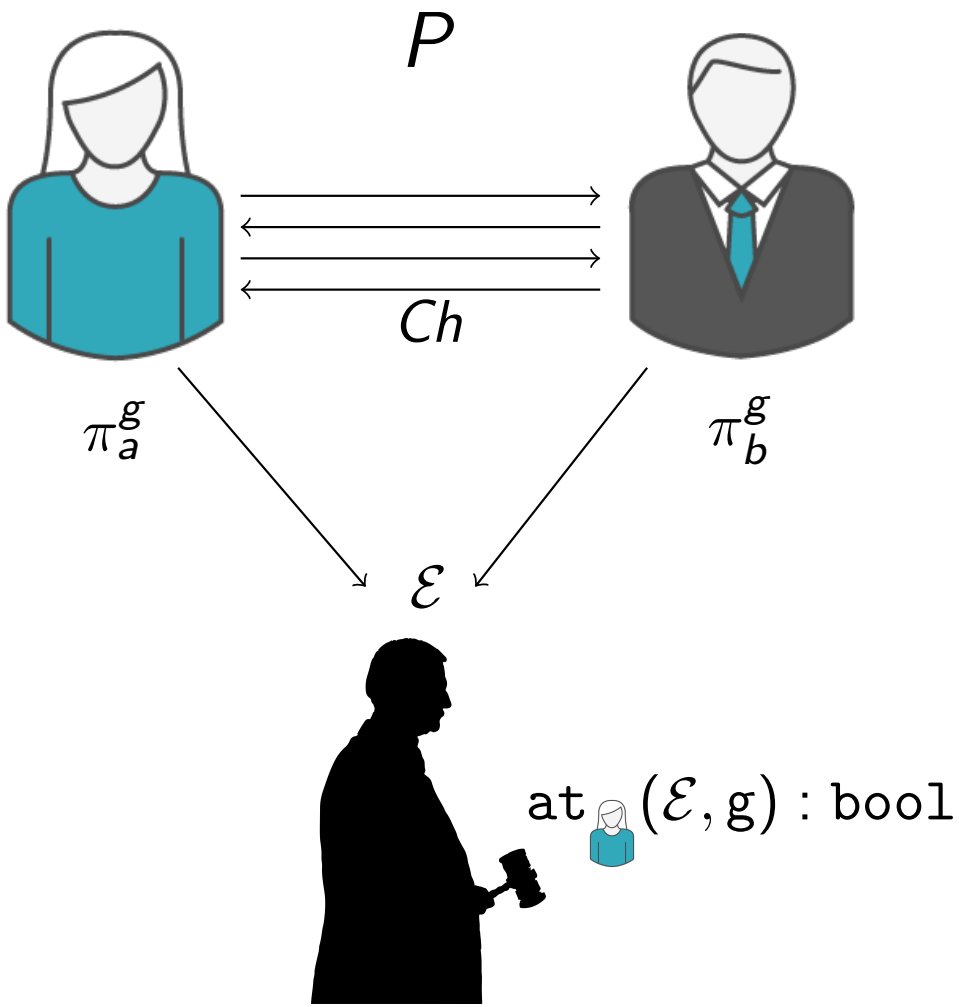
soundness

$\mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{true} \implies$ g holds in $r(P)$

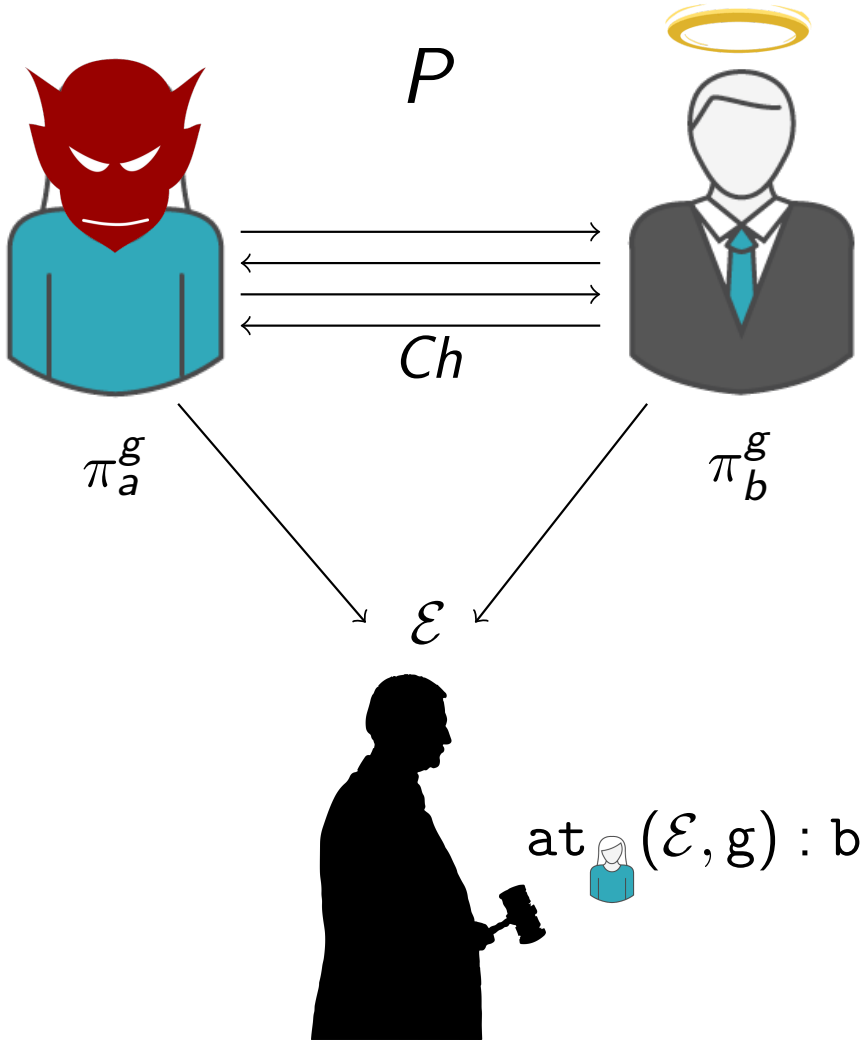
completeness

g holds in $r(P) \implies \mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{true}$

Accountability



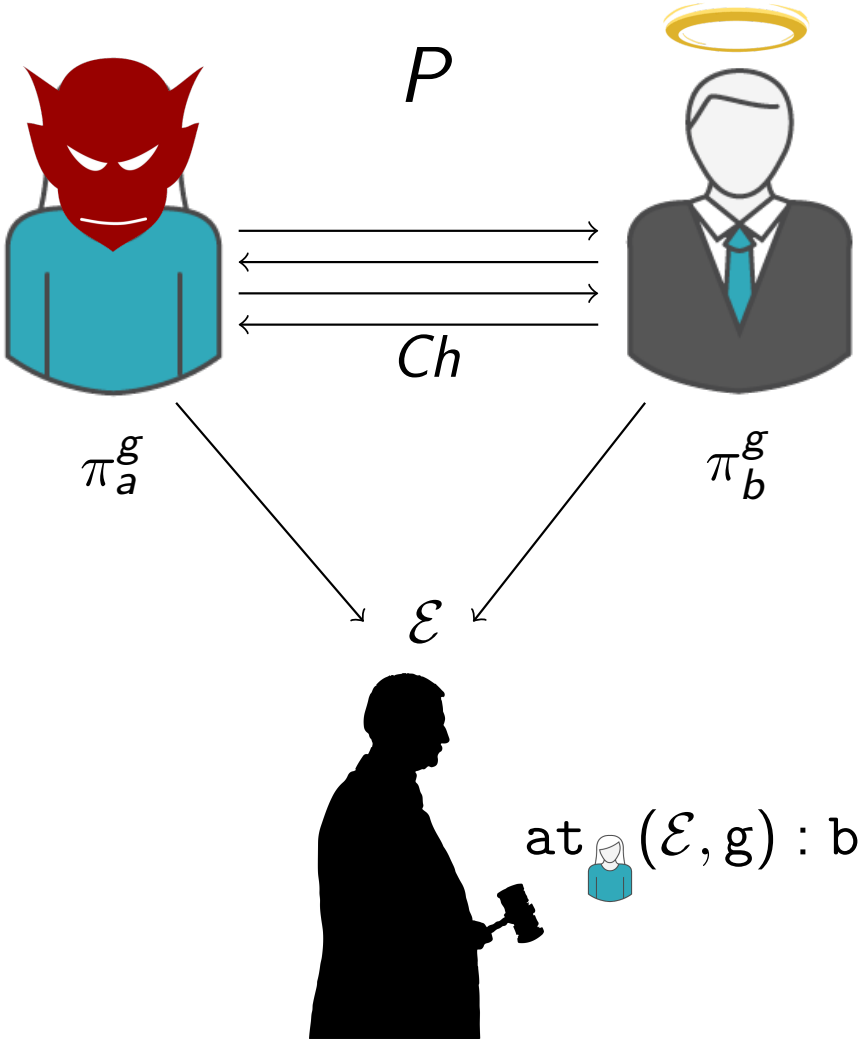
Accountability



soundness

$\mathtt{at}_{\text{👩}}(\mathcal{E}, \mathrm{g}) : \mathtt{false} \implies \pi_{\text{👩}} \in \Pi^{\mathrm{g}}_{\text{👩}}$

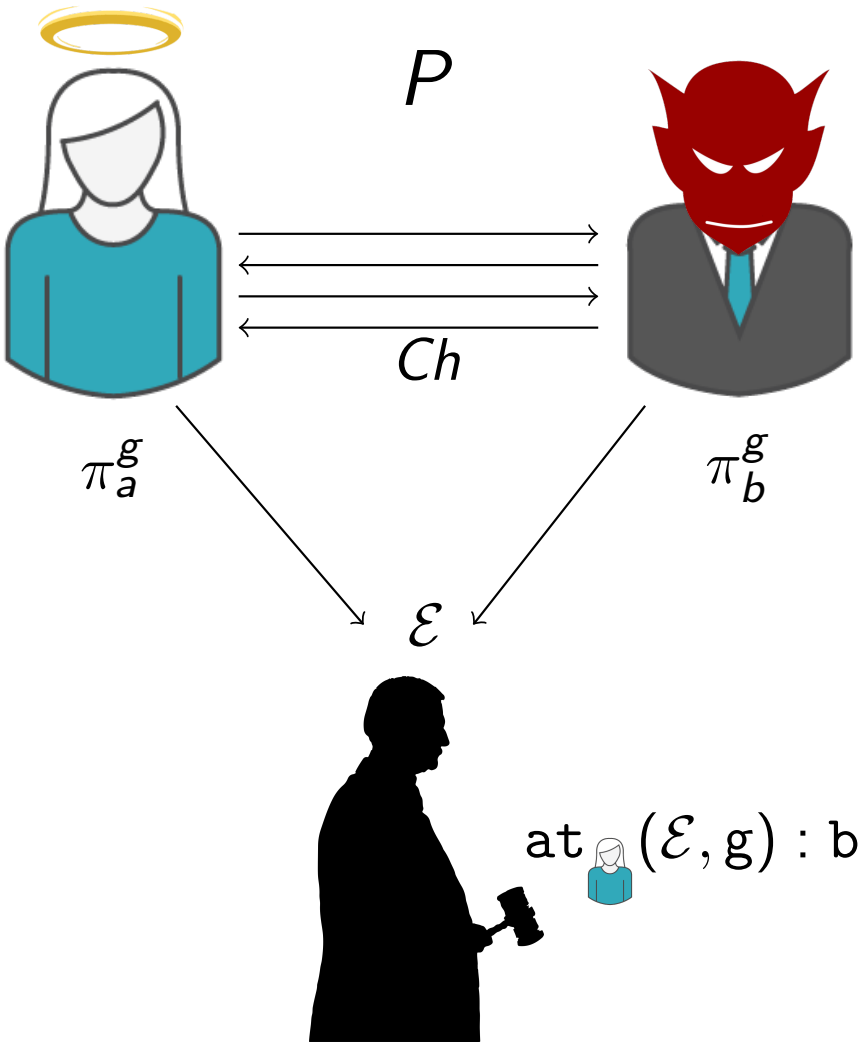
Accountability



soundness

$\mathtt{at}_{\text{}}(\mathcal{E}, \mathrm{g}) : \mathtt{false} \implies \mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{true}$

Accountability



$\mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{bool}$

soundness

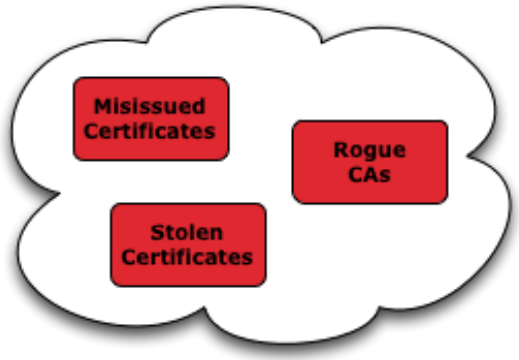
$\mathtt{at}_{\text{🧍}}(\mathcal{E}, \mathrm{g}) : \mathtt{false} \implies \mathtt{vt}(\mathcal{E}, \mathrm{g}) : \mathtt{true}$

completeness

$\pi_{\text{🧍}} \in \Pi^{\mathrm{g}}_{\text{🧍}} \implies \mathtt{at}_{\text{🧍}}(\mathcal{E}, \mathrm{g}) : \mathtt{false}$

Certificate Transparency

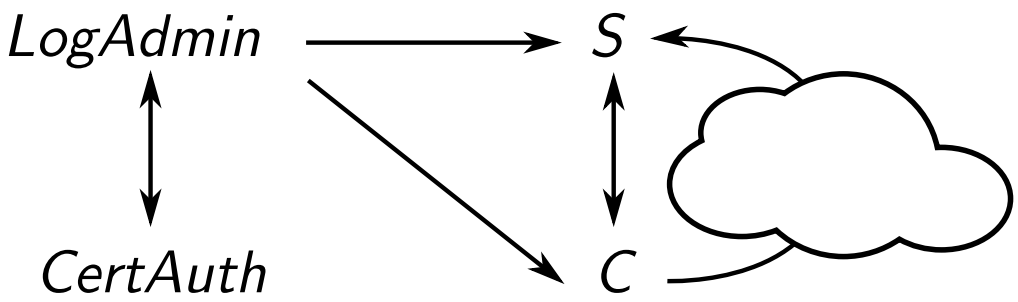
Problems with the traditional Public Key Infrastructures:



Certificate Transparency does not solve those issues, but **adds accountability** to the infrastructure

- Proposed by Ben Laurie at Google in 2012
- Basic idea: maintain a public log (by a Log Authority) of all issued certificates
- When a certificate is misused, the malicious agents are detected

Phases



Certificate issuance:

1. Server sends PK and a proof of identity to the CA
2. CA checks the identity and produces certificate, along with a promise of inclusion in a public log
3. Log authority includes the certificate

Client sessions

1. Client receives the certificate from Server
2. Checks that the certificate is valid, that it's included in a public log, and that the log is extending his previous history
3. Propagates this information to other clients

Accountability in CT

- We built a model of Certificate Transparency in AIF-ω, which allows for modeling and verification of stateful protocols
- Restricted to a *synchronous version* of the protocol, as CT is not accountable when the log is not updated

Accountability test for CertAuth

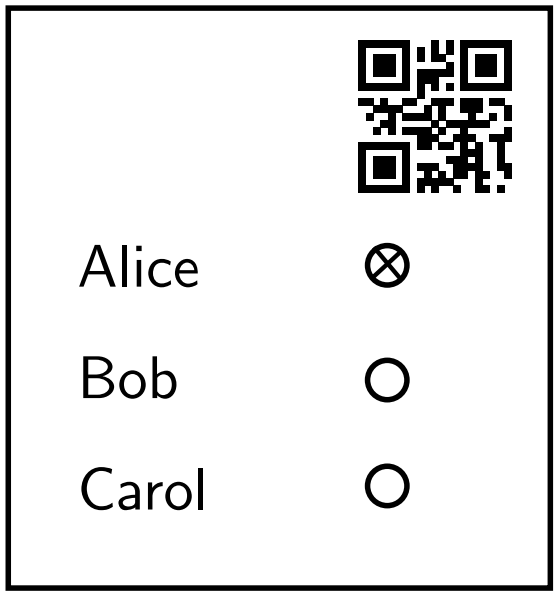
input $cert = sign_{CA}(PK, S, info)$ and $poi \stackrel{?}{=} proofOfID(PK', S')$
test $poi = \bot$ **or** $PK \neq PK'$ **or** $S \neq S'$

Accountability test for LogAdmin

input log_1 and log_2, two observed log histories
test $log_1 \preceq log_2$ **or** $log_2 \preceq log_1$

Result: accountability is sound and complete for both CA and LA

Bingo Voting

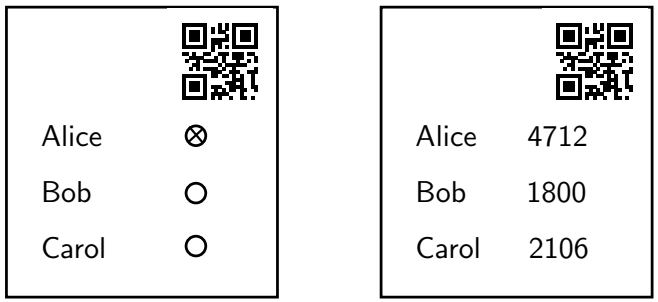


1. cryptographic voting scheme proposed by Bohli, Müller-Quade and Röhrich in 2007 [BMQR07]
2. uses trusted random number generator to provide individual verifiability
3. receipts to voters to check that their vote was counted correctly, but not enough information to reveal their vote

Bingo Voting explained

Setup

1. Produce commitment for each candidate to n dummy votes (random numbers)
2. Publish the commitment along with a ZKP of equal distribution of votes to candidates



Voting

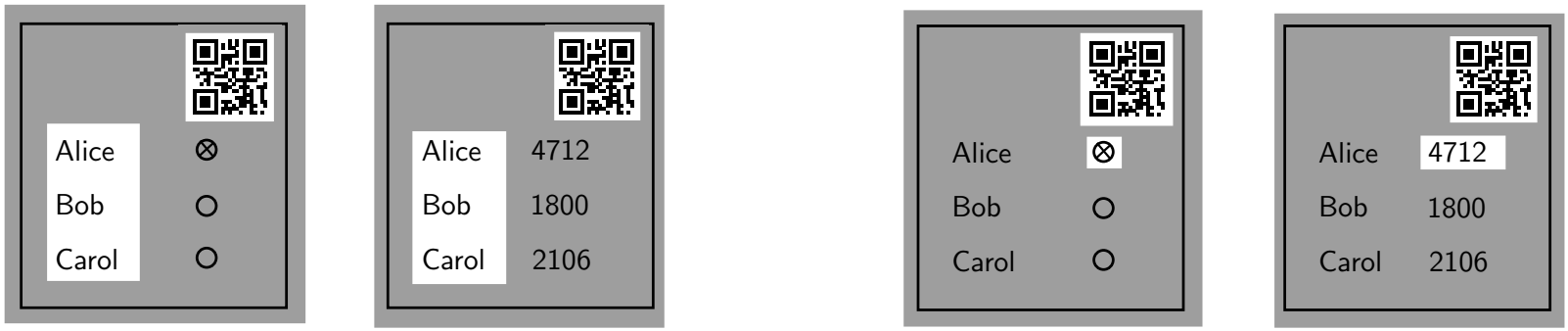
1. Scan voter choice, print random barcode on ballot
2. Produce receipt with a fresh random number for selected candidate, and a dummy vote all other candidates

Counting

1. Dummy votes are removed from the original pool
2. Each candidate keeps their remaining dummy votes

Dispute Resolution in Bingo Voting

- In the case of a dispute, the voter can put paper ballot and receipt inside privacy sleeves.
- Two types of sleeves:



- The voter can choose to reveal that either the candidates of the receipt are not in the same order of the ballot, or that the corresponding vote has been removed from the pool.

Accountability

- The accountability test for the Voting Authority corresponds to the Verifiability test:

Verifiability test

input $screen = r$, $paper = choice, barcode\_p$ and
$receipt = r1, r2, barcode\_r$
test $(choice = c1$ **and** $r = r1$ **and** $barcode\_p = barcode\_r)$ **or**
$(choice = c2$ **and** $r = r2$ **and** $barcode\_p = barcode\_r)$

- However, this test is sound but not complete: two colluding voters can collaborate to indict the voting authority
- The first votes and obtains the receipt
- The second swaps his receipt with the first voter's receipt so that the two bar-codes mismatch.

Exam

- Goal: evaluate candidates
- Submission: tests over some options
- Evaluation: marking algorithm that outputs a **ranking** of tests

- Entrance exam
- Bar examinations
- Skill tests
- Personnel selection
- Project proposals
- Public tenders
- Conference management systems

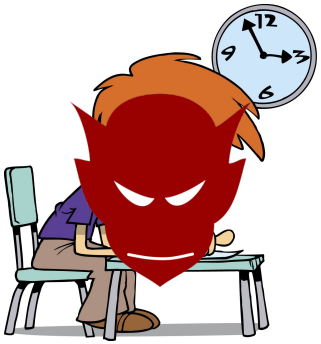
TOEFL iBT

Epso

CISCO CCNA

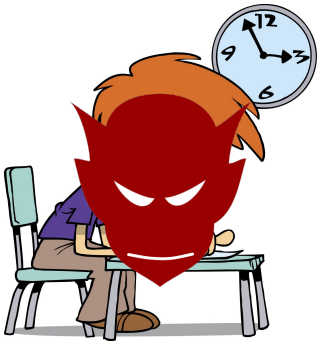
Threats

Threats



- Candidate cheating
- Corrupted exam authority
- Unfair examiners
- Outside attackers

Threats

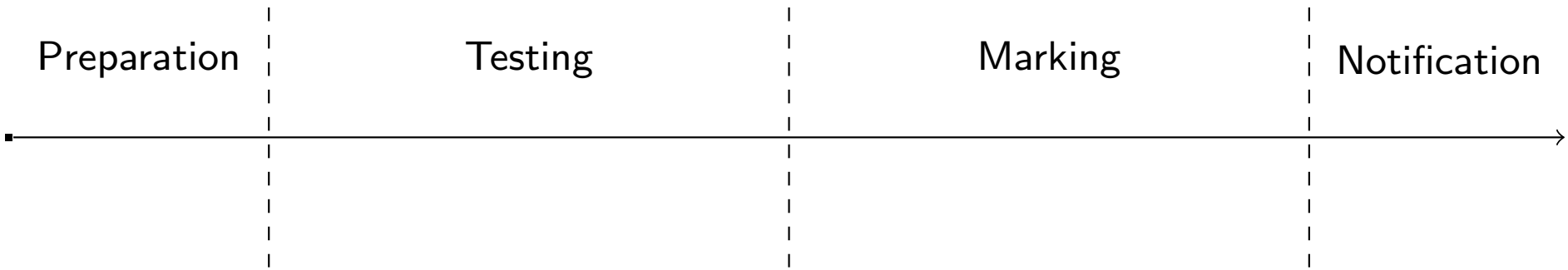


- Candidate cheating
- Corrupted exam authority
- Unfair examiners
- Outside attackers

Real Threats!

- Atlanta Public Schools scandal (2009)
- Turkish Public Personnel Selection Exam (2010)
- UK student visa tests fraud (2014)

Phases



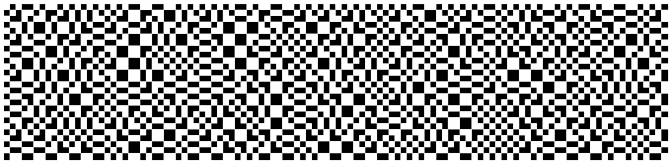
Goal (original)

- Ensure authentication and anonymity despite a corrupted authority
- Paper-and-pencil exam

WATA IV

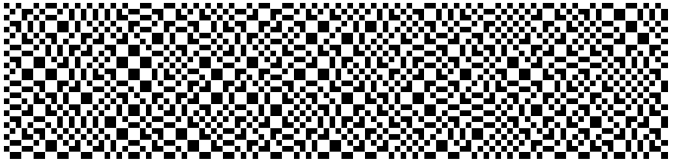
- This sheet must be printed in a transparency paper -

Examiner Transparency



Name: **John**
Surname: **Smith**
ENRL Number: **012/3456789**
Exam Date: **21/12/2014**

Candidate Paper



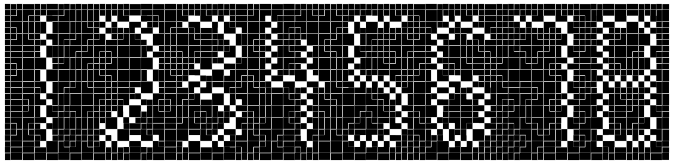
Name: **John**
Surname: **Smith**
ENRL Number: **012/3456789**
Exam Date: **21/12/2014**

Instruction: At examination venue, overlay this paper sheet with the Examiner Transparency. Then, write the token into dedicated test form.

- This sheet must be printed in a transparency paper -

Examiner Transparency

Candidate Paper



Name: **John**
Surname: **Smith**
ENRL Number: **012/3456789**
Exam Date: **21/12/2014**

Instruction: At examination venue, overlay this paper sheet with the Examiner Transparency. Then, write the token into dedicated test form.

Remove the need of a trusted party

Combine **oblivious transfer** and **visual cryptography**

- C and A jointly generate the pseudonym

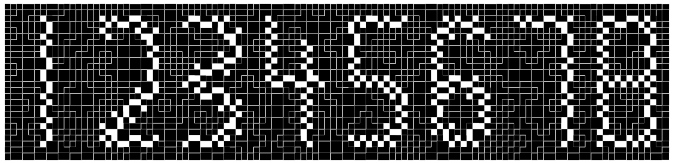
  1. C provides a **commitment** to an index into an array.
  2. A fills the array with a **secret permutation** of the characters.
  3. Only when the two secrets are brought **together** the selection of a character is determined.

Remove the need of a trusted party

- This sheet must be printed in a transparency paper -

Examiner Transparency

Candidate Paper



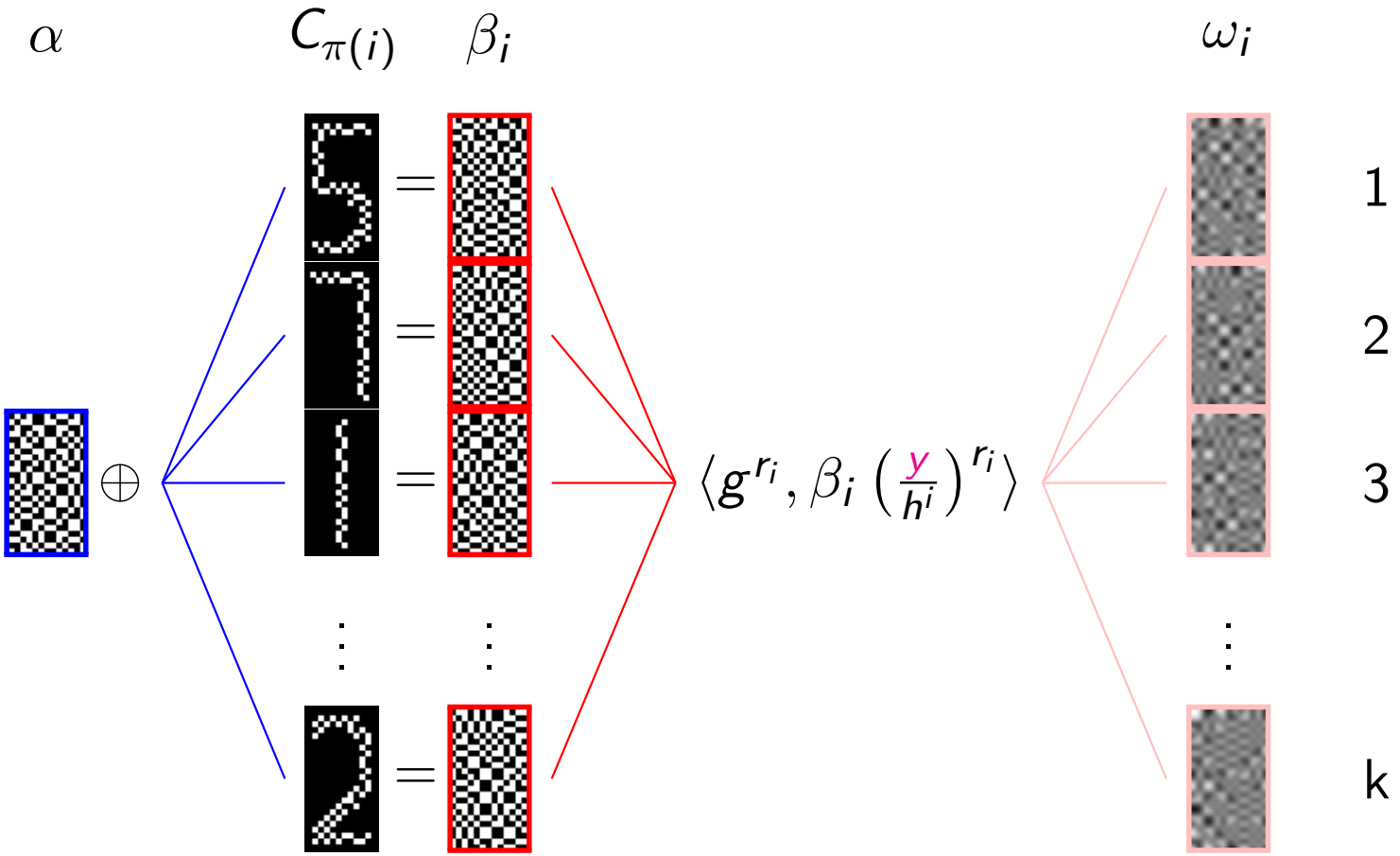
Name: **John**
Surname: **Smith**
ENRL Number: **012/3456789**
Exam Date: **21/12/2014**

Name: **John**
Surname: **Smith**
ENRL Number: **012/3456789**
Exam Date: **21/12/2014**

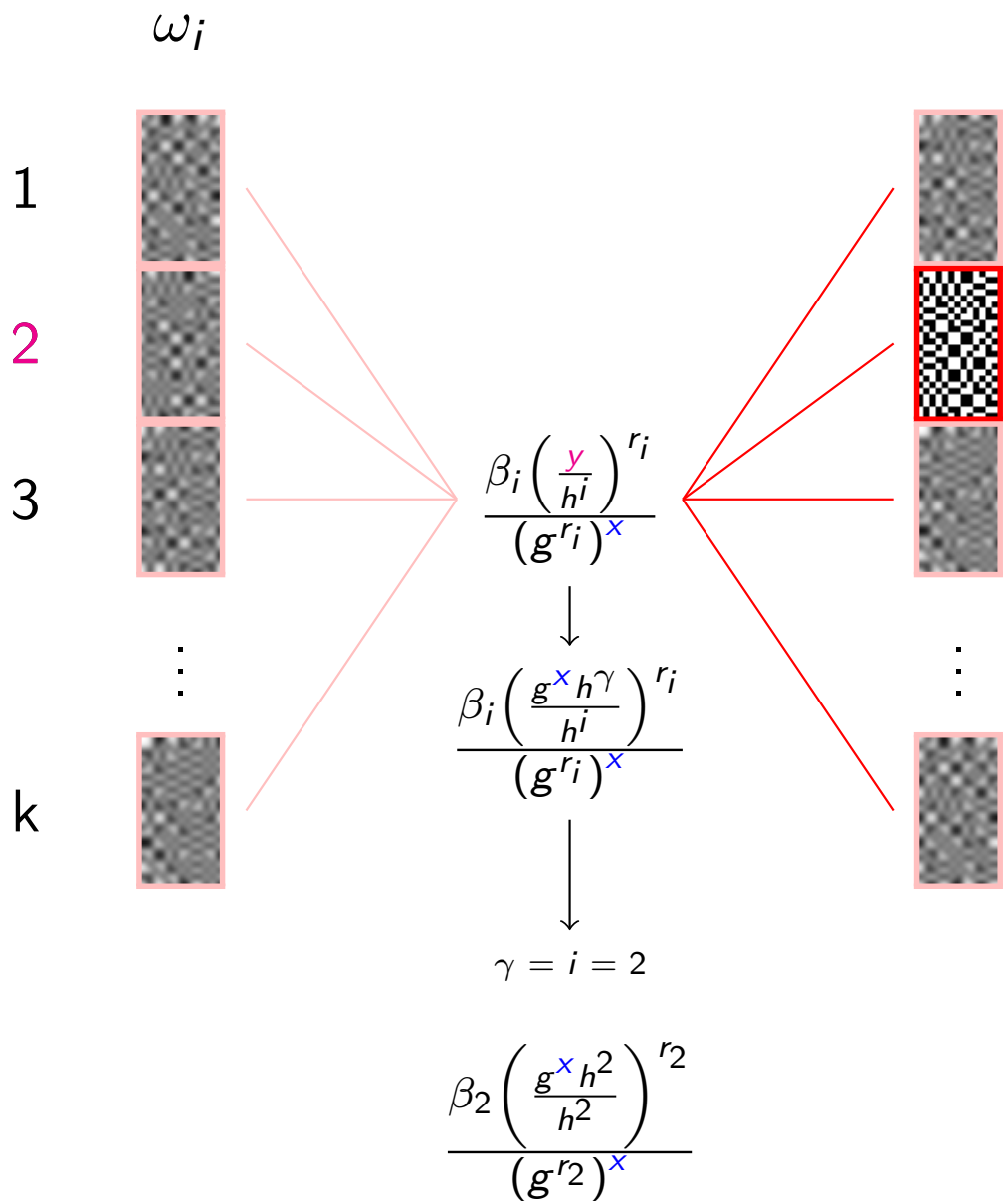
Instruction: At examination venue, overlay this paper sheet with the Examiner Transparency. Then, write the token into dedicated test form.

Remove the need of a trusted party

- A fills the array with a **secret permutation** of the characters.



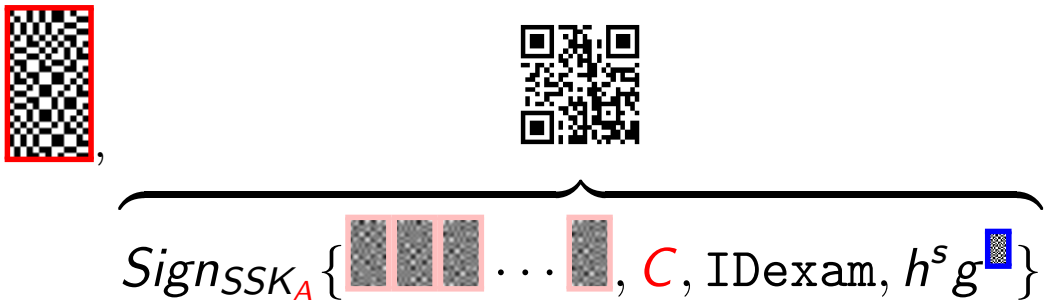
Remove the need of a trusted party



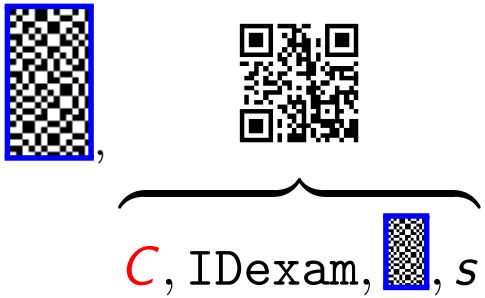
Remove the need of a trusted party

- Only when the two secrets are brought **together** the selection of a character is determined.

C prints

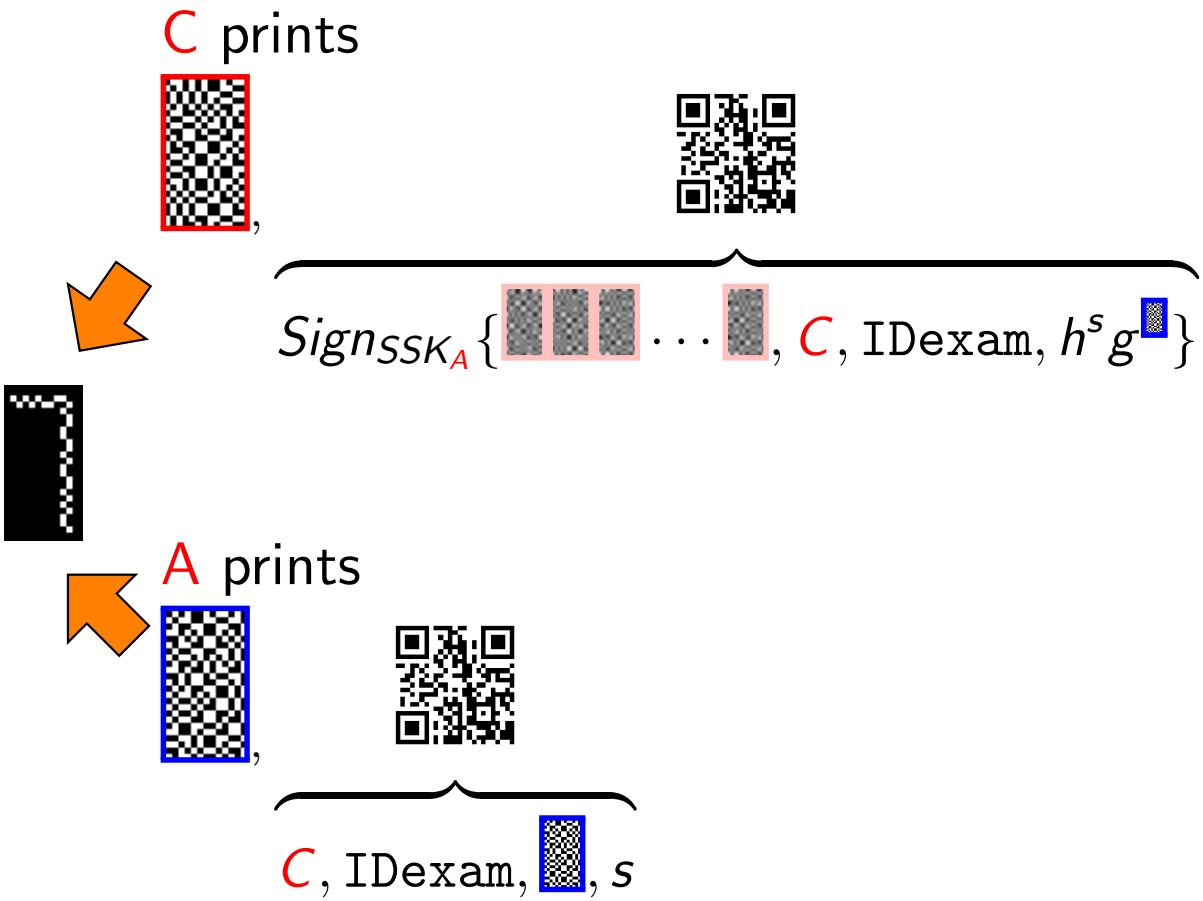


A prints



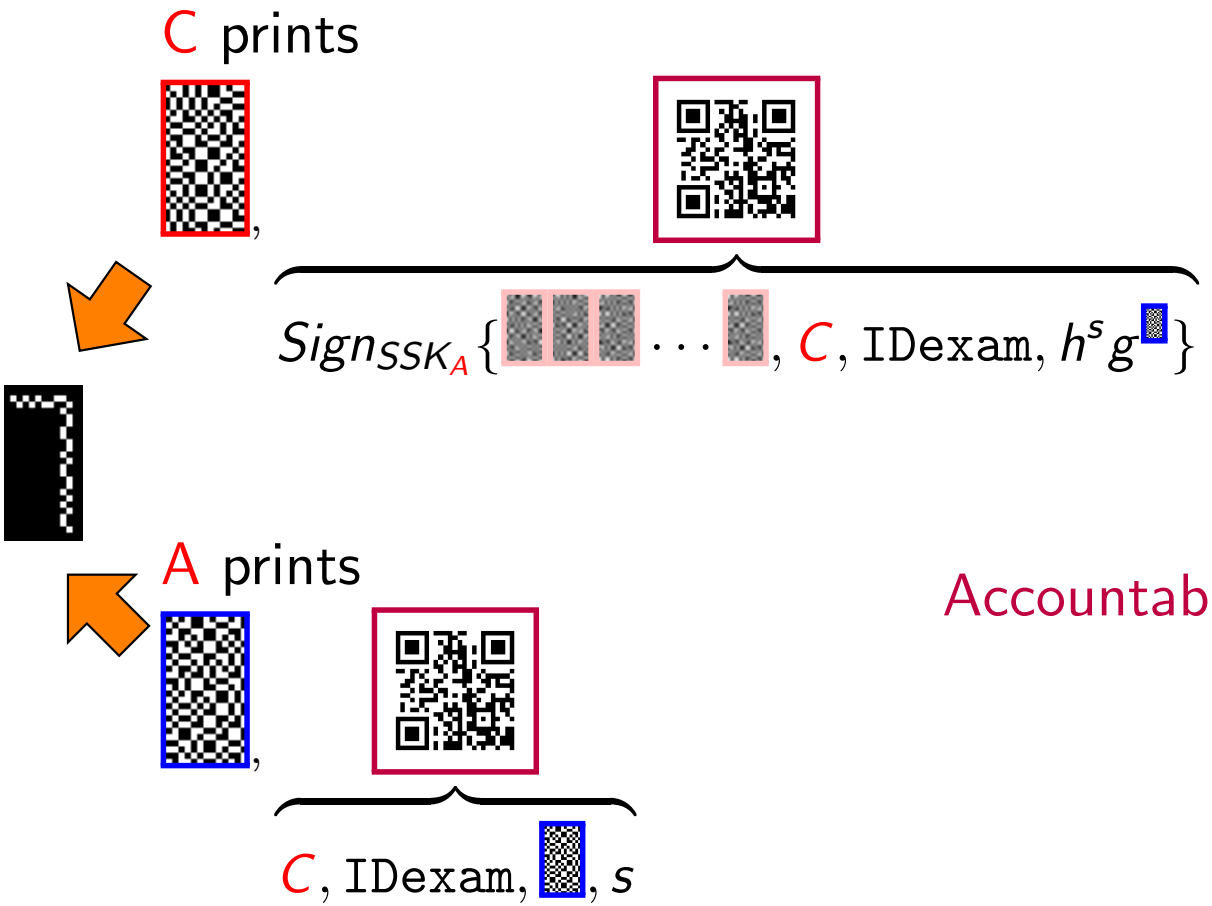
Remove the need of a trusted party

- Only when the two secrets are brought **together** the selection of a character is determined.



Remove the need of a trusted party

- Only when the two secrets are brought **together** the selection of a character is determined.



Accountability: dispute resolution

Exam Accountability

Algorithm 1: The accountability test for the Candidate

Data:

- $paper = \beta, c, I, sign1, sign2$ where
  - $sign1 = Sign_A\{com_A\}$.
  - $sign2 = Sign_A\{com_C, \Omega\}$.
- $transp = \alpha, a$.

if $sign1 = \perp$ **or** $sign2 = \perp$ **or** $com_c \neq commit(c, I)$ **or** $\beta \neq deobf(\Omega, c)$ **then**
  return `true`
else
  return `false`

Algorithm 2: The accountability test for the Administrator

Data:

- $paper = \beta, c, I, sign1, sign2$ where
  - $sign1 = Sign_A\{com_A\}$.
  - $sign2 = Sign_A\{com_C, \Omega\}$.
- $transp = \alpha, a$.

if $sign1 \neq \perp$ **and** $sign2 \neq \perp$ **and** $com_A \neq commit(a, \alpha)$ **then**
  return `true`
else
  return `false`

Results

Tool	Protocol	Goal	Verifiability	Accountability
AIF-ω	Google Certificate Transparency	Valid certificate	✓	✓
ProVerif	Bingo Voting	Cast as intended ballot	✓	×
ProVerif	Secure computer-based exam	Intelligible pseudonym	✓	✓

Conclusion

- Accountability is an essential property of complex protocols
- We propose a **framework to reason about accountability** in the symbolic model, and express **soundness** and **completeness** of accountability tests as *reachability properties*
- We show how to cast this framework into three relevant case studies:
  1. Certificate Transparency
  2. Bingo Voting
  3. Secure Exams
- A big advantage of our approach is that it can be used to construct automated proofs with current technology

Conclusion

- Accountability is an essential property of complex protocols
- We propose a **framework to reason about accountability** in the symbolic model, and express **soundness** and **completeness** of accountability tests as *reachability properties*
- We show how to cast this framework into three relevant case studies:
  1. Certificate Transparency
  2. Bingo Voting
  3. Secure Exams
- A big advantage of our approach is that it can be used to construct automated proofs with current technology

Thank you :)


Martin Abadi and Bruno Blanchet.
Computer-Assisted Verification of a Protocol for Certified Email, pages 316–335.
Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.


Josh Benaloh, Matthew Bernhard, J. Alex Halderman, Ronald L. Rivest, Peter Y. A. Ryan, Philip B. Stark, Vanessa Teague, Poorvi L. Vora, and Dan S. Wallach.
Public evidence from secret ballots.
CoRR, abs/1707.08619, 2017.


Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich.
Bingo Voting: Secure and Coercion-Free Voting Using a Trusted Random Number Generator, pages 111–124.
Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.


Giampaolo Bella and Lawrence C. Paulson.
Accountability protocols: Formalized and verified.
ACM Trans. Inf. Syst. Secur., (2):138–161, May 2006.


Jannik Dreier, Rosario Giustolisi, Ali Kassem, Pascal Lafourcade, and Gabriele Lenzini.
A framework for analyzing verifiability in traditional and electronic exams.
In *Information Security Practice and Experience 11th International Conference, ISPEC 2015, Beijing, China, May 5-8, 2015*, 2015.


J. Dreier, Jonker H., and P. Lafourcade.
Defining verifiability in e-auction protocols.
In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS'13)*, pages 547–552, Hangzhou, China, 2013. ACM.


N. Guts, C. Fournet, and F. Zappa Nardelli.
Reliable evidence: Auditability by typing.
In *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09)*, volume 5789 of *Lecture*

Notes in Computer Science, pages 168–183, Saint-Malo, France, 2009. Springer.


S. Kremer, M. Ryan, and B. Smyth.
Election verifiability in electronic voting protocols.
In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS'10)*, volume 6345 of *LNCS*, pages 389–404. Springer, 2010.


Ralf Küsters, Tomasz Truderung, and Andreas Vogt.
Accountability: Definition and relationship to verifiability.
In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 526–535, New York, NY, USA, 2010. ACM.


B. Smyth, M. Ryan, S. Kremer, and K. Mounira.
Towards automatic analysis of election verifiability properties.
In *Proceedings of the Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the*

Theory of Security (ARSPA-WITS'10), volume 6186 of *LNCS*, pages 146–163. Springer, 2010.