How to trust digital applications?

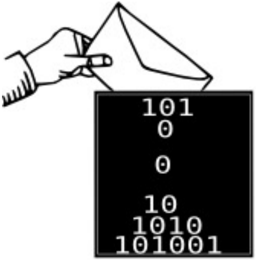
Formal methods can help you

Pascal Lafourcade

7th November, 2013

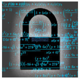
Digital Confidence Seminar

Nowadays Security is Everywhere!



What is cryptography based security?

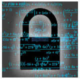
Cryptographic Protocols:



- Primitives: RSA, Elgamal, AES, DES, SHA-3 ...
- Distributed Algorithms

What is cryptography based security?

Cryptographic Protocols:



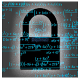
- Primitives: RSA, Elgamal, AES, DES, SHA-3 ...
- Distributed Algorithms

Properties:

- Secrecy,
- Authentication,
- Privacy ...

What is cryptography based security?

Cryptographic Protocols:



- Primitives: RSA, Elgamal, AES, DES, SHA-3 ...
- Distributed Algorithms

Properties:

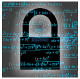
- Secrecy,
- Authentication,
- Privacy ...

Intruders:

- Passive
- Active
- CPA, CCA ...

What is cryptography based security?

Cryptographic Protocols:



- Primitives: RSA, Elgamal, AES, DES, SHA-3 ...
- Distributed Algorithms

Properties:

- Secrecy,
- Authentication,
- Privacy ...

Intruders:

- Passive
- Active
- CPA, CCA ...

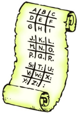
Designing **secure** cryptographic protocols is **difficult**

Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

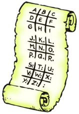
Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

- Prove that there is no attack under some assumptions.

Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

- Prove that there is no attack under some assumptions.
  - proving is a difficult task,
  - pencil-and-paper proofs are error-prone.

Security of Cryptographic Protocols

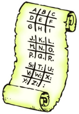
How can we be convinced that a protocols is secure?

- Prove that there is no attack under some assumptions.
  - proving is a difficult task,
  - pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is correct?

Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?



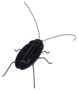


- Prove that there is no attack under some assumptions.
  - proving is a difficult task,
  - pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is correct?







Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

- Prove that there is no attack under some assumptions.
  - proving is a difficult task,
  - pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is correct?

Computer-Aided Security.

Formal Verification Approaches

Designer

Attacker

Formal Verification Approaches

Designer

Attacker

Security Team

Formal Verification Approaches

Designer

Attacker

Give a proof

Security Team

Formal Verification Approaches

Designer

Attacker

Give a proof

Find a flaw

Security Team

Back to 1995

     ≥ 17  (FDR)

Back to 1995

     ≥ 17  (FDR)

- **Cryptography:** Perfect Encryption hypothesis
- **Property:** Secrecy, Authentication
- **Intruder:**
  - Active
  - Controlling the network
  - Several sessions

Needham Schroeder Protocol

Needham-Schroeder Key Exchange Protocol

$$A \rightarrow B : \{A, N_A\}_{Pub(B)}$$

$$B \rightarrow A : \{N_A, N_B\}_{Pub(A)}$$

$$A \rightarrow B : \{N_B\}_{Pub(B)}$$

Needham Schroeder Protocol

Needham-Schroeder Key Exchange Protocol

$$A \rightarrow B : \{A, N_A\}_{Pub(B)}$$

$$B \rightarrow A : \{N_A, N_B\}_{Pub(A)}$$

$$A \rightarrow B : \{N_B\}_{Pub(B)}$$

Attack by G. Lowe 

$$A \rightarrow I : \{A, N_A\}_{Pub(I)}$$

$$A \leftarrow I : \{N_A, N_B\}_{Pub(A)}$$

$$A \rightarrow I : \{N_B\}_{Pub(I)}$$

$$I \rightarrow B : \{A, N_A\}_{Pub(B)}$$

$$I \leftarrow B : \{N_A, N_B\}_{Pub(A)}$$

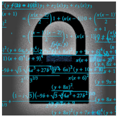
$$I \rightarrow B : \{N_B\}_{Pub(B)}$$

Success Story of Symbolic Verification

Tools based on different theories for several properties

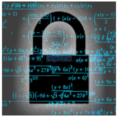
1995	Casper/FRD [Lowe]
2001	Proverif [Blanchet]
2003	Proof of certified email protocol with Proverif [AB]
	OFMC [BMV]
	Hermes [BLP]
	Flaw in Kerberos 5.0 with MSR 3.0 [BCJS]
2004	TA4SP [BHKO]
2005	SATMC [AC]
2006	CL-ATSE [Turuani]
2008	Scyther [Cremers]
	Flaw of Single Sign-On for Google Apps with SAT-MC [ACCCT]
	Proof of TLS using Proverif [BFCZ]
2010	TOOKAN [DDS] using SAT-MC for API
2012	Tamarin [BCM]

Main Contributions:

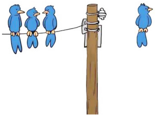


- Hoare Logics for **cryptography**
  - Asymmetric Encryptions
  - Encryption Modes
  - Message Authentication Codes

Main Contributions:

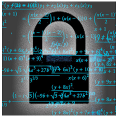


- Hoare Logics for **cryptography**
  - Asymmetric Encryptions
  - Encryption Modes
  - Message Authentication Codes

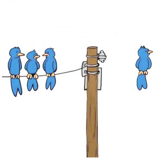


- **Intruder** models and algorithms for WSN
  - Neighbourhood Discovery Protocols
  - Independent Intruders
  - Routing Algorithms

Main Contributions:



- Hoare Logics for **cryptography**
  - Asymmetric Encryptions
  - Encryption Modes
  - Message Authentication Codes



- **Intruder** models and algorithms for WSN
  - Neighbourhood Discovery Protocols
  - Independent Intruders
  - Routing Algorithms

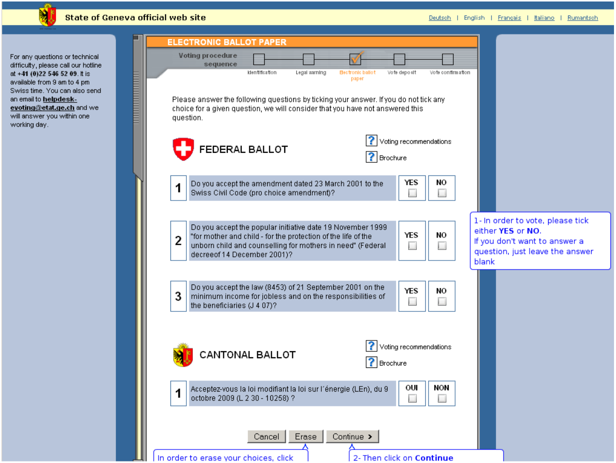
- **Security and Privacy Properties for**
  - E-voting
  - E-auction
  - E-examn

Outline

Internet voting

Available in

- Estonia
- France
- Switzerland
- . . .



Security Properties of E-Voting Protocols

Fairness

Individual Verifiability

Eligibility

Universal Verifiability

Correctness

Privacy

Receipt-Freeness

Robustness

Coercion-Resistance

Security Properties of E-Voting Protocols

Fairness

Individual Verifiability

Eligibility

Universal Verifiability

Correctness

Privacy

Receipt-Freeness

Robustness

Coercion-Resistance

Motivation

Existing several models for Privacy, but they

- designed for a specific type of protocol
- often cannot be applied to other protocols

Motivation

Existing several models for Privacy, but they

- designed for a specific type of protocol
- often cannot be applied to other protocols

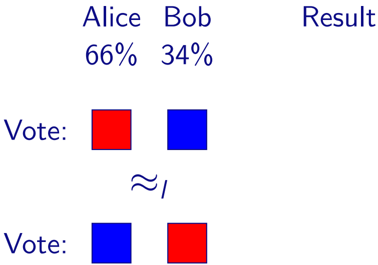
Our Contributions:

- Define **fine-grained** Privacy definitions to **compare** protocols
- Analyze **weighted votes** protocols
- **One coercer is enough**

Privacy for Weighted Votes [DLL'12b]

	Alice	Bob	Result
Vote:	red	blue	
	$\approx_l$		
Vote:	blue	red	

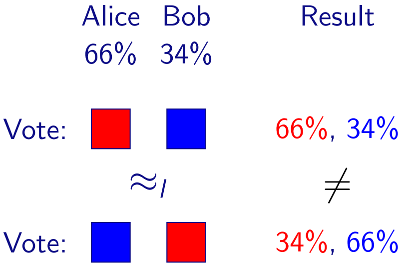
Privacy for Weighted Votes [DLL'12b]



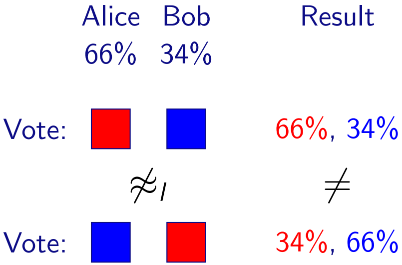
Privacy for Weighted Votes [DLL'12b]

	Alice	Bob	Result
	66%	34%	
Vote:	■ (red)	■ (blue)	66%, 34%
	$\approx_l$		
Vote:	■ (blue)	■ (red)	34%, 66%

Privacy for Weighted Votes [DLL'12b]



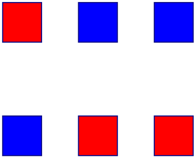
Privacy for Weighted Votes [DLL'12b]



Privacy for Weighted Votes [DLL'12b]

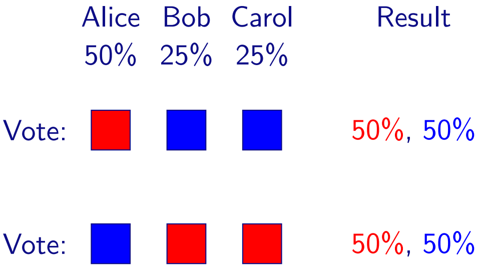
Still: Some privacy is possible!

	Alice	Bob	Carol	Result
	50%	25%	25%	
Vote:				
Vote:				



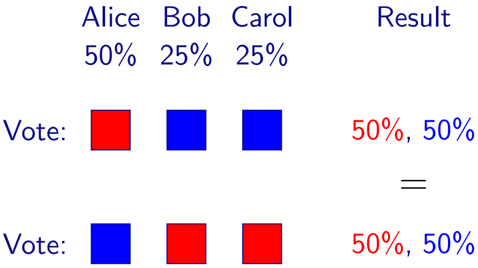
Privacy for Weighted Votes [DLL'12b]

Still: Some privacy is possible!



Privacy for Weighted Votes [DLL'12b]

Still: Some privacy is possible!



Privacy for Weighted Votes [DLL'12b]

Still: Some privacy is possible!

	Alice	Bob	Carol	Result
	50%	25%	25%	
Vote:				50%, 50%
		$\approx_l$		=
Vote:				50%, 50%

Definition of Vote-Privacy (VP) for weighted votes

Idea: Two instances with the same result should be bi-similar

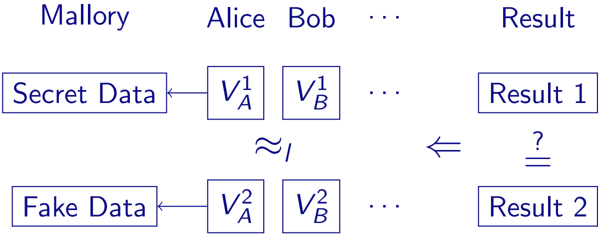
	Alice	Bob	$\cdots$		Result
Vote 1:	V_A^1	V_B^1	$\cdots$		Result 1
	$\approx_l$			$\Leftarrow$	$\stackrel{?}{=}$
Vote 2:	V_A^2	V_A^2	$\cdots$		Result 2

Single-Voter Receipt Freeness (SRF)

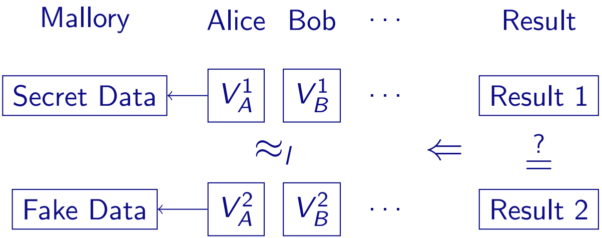
Mallory

Alice	Bob	$\cdots$		Result
V_A^1	V_B^1	$\cdots$		Result 1
$\approx_l$			$\Leftarrow$	$\stackrel{?}{=}$
V_A^2	V_B^2	$\cdots$		Result 2

Single-Voter Receipt Freeness (SRF)



Single-Voter Receipt Freeness (SRF)

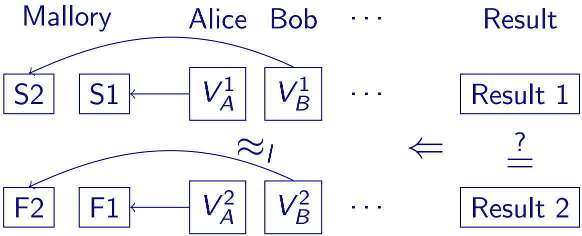


If a protocol respects (EQ), then (SRF) and (SwRF) are equivalent.

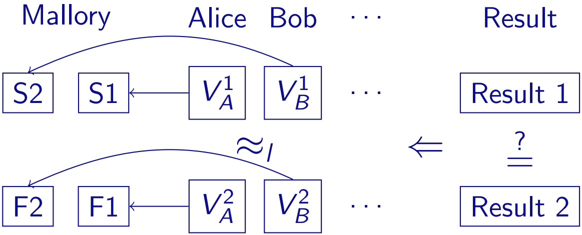
Multi-Voter Receipt Freeness (MRF)

Mallory		Alice	Bob	$\cdots$		Result
S1	$\leftarrow$	V_A^1	V_B^1	$\cdots$		Result 1
		$\approx_l$			$\Leftarrow$	$\stackrel{?}{=}$
F1	$\leftarrow$	V_A^2	V_B^2	$\cdots$		Result 2

Multi-Voter Receipt Freeness (MRF)

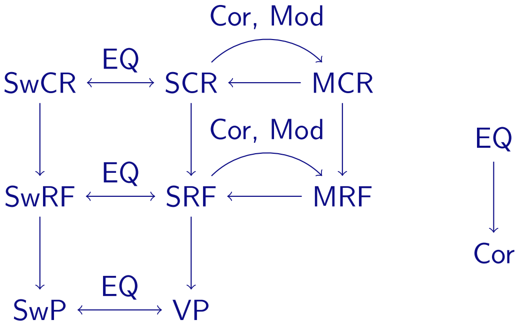


Multi-Voter Receipt Freeness (MRF)



(MRF) implies (SRF) and (MCR) implies (SCR).

One Coerced Voter is enough!



Unique decomposition of processes in the applied π-calculus.

Outline

e-Auctions

quicksales.com.au

Sotheby's

GUN BROKER.COM
The Bidding Starts Now

AutoBidsOnline.com
Don't Request a Quote, Set Your Price!

ricardo.ch

ebay

mercado Libre

hood.de

WineCommune Buy and Sell Fine Wine - Online!

Competing parties

Bidders/Buyers

+

Seller

Auctioneer

Several e-Auctions

Many possible (complex) mechanisms:

- Sealed Bid
- English: open ascending price auction.
- Dutch: tulips market.
- First Price
- Second Price (Vickrey auction)
- ...

e-Auctions: Security Requirements

[POST'13, ASIACCS'13]

Fairness

Verifiability

Non-Repudiation

Non-Cancellation

Security Requirements

Secrecy of Bidding Price

Receipt-Freeness

Coercion-Resistance

Anonymity of Bidders

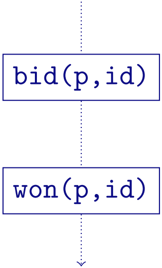
Events [POST'13]

To express our properties, we use the following events:

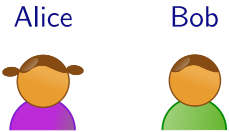
- `bid(p,id)`: a bidder `id` bids the price `p`
- `recBid(p,id)`: a bid at price `p` by bidder `id` is recorded by the auctioneer/bulletin board/etc.
- `won(p,id)`: a bidder `id` wins the auction at price `p`

Non-Repudiation [POST'13]

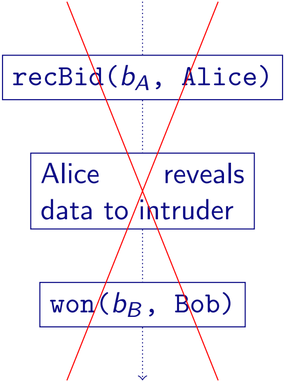
On every trace:



Non-Cancellation [POST'13]



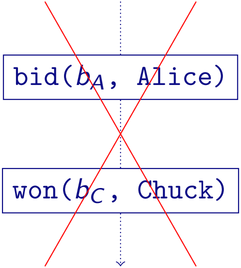
Bid b_A > b_B



Highest Price Wins [POST'13]

Alice (honest) | Chuck (corrupted)

Bid b_A > b_C



Strong Noninterference & Weak Noninterference [POST'13]

Definition (Strong Noninterference (SN))

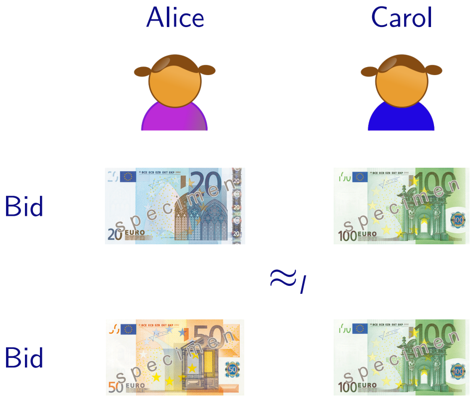
An auction protocol ensures *Strong Noninterference (SN)* if for any two auction processes AP_A and AP_B that halt at the end of the bidding phase (i.e. where we remove all code after the last `recBid` event) we have $AP_A \approx_l AP_B$.

Definition (Weak Noninterference (WN))

Like Strong Noninterference, but we consider only processes with the same bidders.

Strong Bidding-Price Secrecy (SBPS) [DJP10]

Main idea: Observational equivalence between two situations.



Bidding-Price Unlinkability (BPU) [POST'13]

The list of bids can be public, but must be unlinkable to the bidders.

Alice | Bob | Carol

Bid

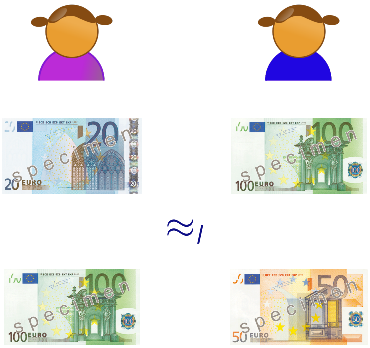
$\approx_l$

Bid

Strong Anonymity (SA) [POST'13]

The winner may stay anonymous.

Alice Carol



Bid

$\approx_l$

Bid

Weak Anonymity (WA) [POST'13]

Unlinkability, but also for the winner.

Alice

Carol

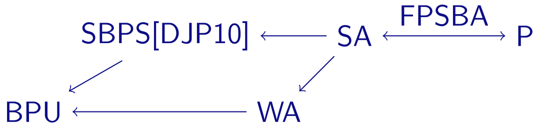
Bid

$\approx_l$

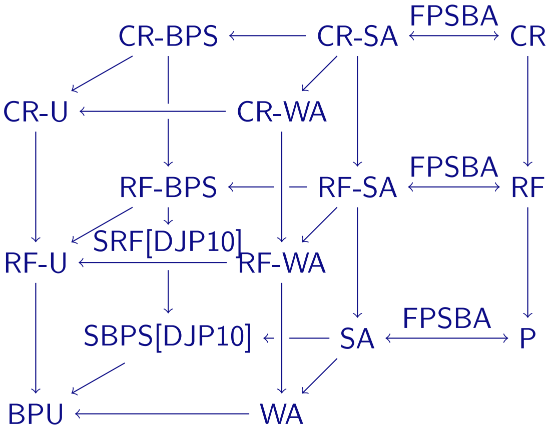
Bid

e-Auctions: Hierarchy of Privacy Notions [POST'13]

e-Auctions: Hierarchy of Privacy Notions [POST'13]



e-Auctions: Hierarchy of Privacy Notions [POST'13]



Motivation: Three different perspectives [ASIACCS'13]

- A losing bidder:

- A winning bidder:

- The seller:

Registration and Integrity Verifiability [ASIACCS'13]

- Origination of all 

and  ($rv_{submitted}$)
- Integrity of 

and  (rv_w)

The losing bidder verifies that he actually lost [ASIACCS'13]

- 

$<$  (ov_l)

The winning bidder checks [ASIACCS'13]

- Correction of the computation of  i.e.

  $myBid =$  (ov_w)

The seller verifies [ASIACCS'13]

- $b_{win} =$ 
- Correction of the computation of  (os_w)

Verifiability Framework

Registration and Integrity Verifiability (RV)

- Anyone can verify that all bids on the list were submitted by registered bidders
- Anyone can verify that the winning bid is one of the submitted bids

Outcome Verifiability (OV)

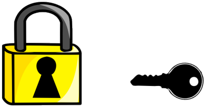
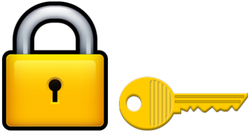
- A losing bidder can verify that his bid was not the winning bid
- A winning bidder can verify that his bid was the winning bid
- The seller can verify that the winning bid is actually the highest submitted bid

Example Sako 2000

Protocol by Sako[ASIACCS'13]

Each price corresponds to a pair of public and private keys.

- Price 10 €: 

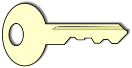
- Price 5 €: 
- Price 1 €: 

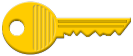
Set up [ASIACCS'13]

A public constant c

Bulletin Board

Authorities





Bidding Phase [ASIACCS'13]

Select a Price

- For 5 €:

- For 1 €:

Bidding Cont'd [ASIACCS'13]

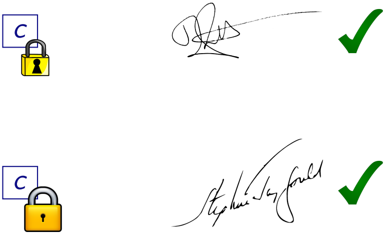
The signed bids are published on the bulletin board:

Bid Opening [ASIACCS'13]

1. The signatures are checked.

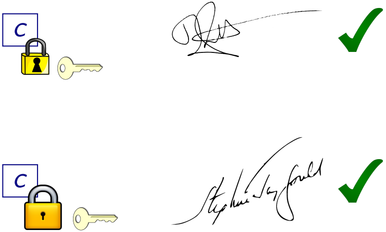
Bid Opening [ASIACCS'13]

1. The signatures are checked.



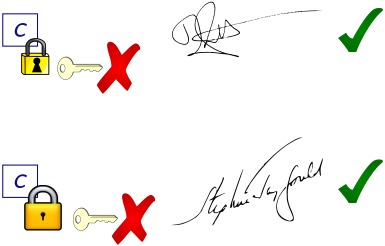
Bid Opening [ASIACCS'13]

1. The signatures are checked.
2. The bids are decrypted using the first private key.



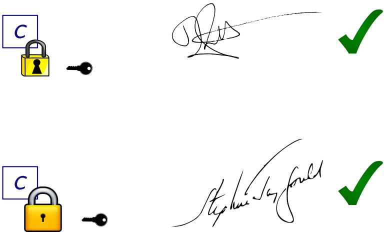
Bid Opening [ASIACCS'13]

1. The signatures are checked.
2. The bids are decrypted using the first private key.



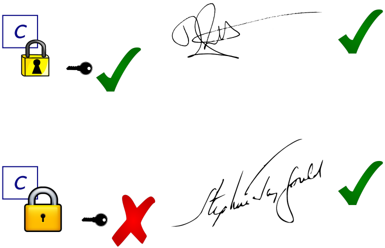
Bid Opening [ASIACCS'13]

1. The signatures are checked.
2. The bids are decrypted using the first private key.
3. If the decryption is correct, a winner is found. Otherwise use next key.



Bid Opening [ASIACCS'13]

1. The signatures are checked.
2. The bids are decrypted using the first private key.
3. If the decryption is correct, a winner is found. Otherwise use next key.



Registration Verification [ASIACCS'13]

1. rv_s: Anybody can verify the signatures.

2. rv_w: Anybody can check if the announced winning bid was published on the bulletin board.

Registration Verification [ASIACCS'13]

1. rv_s: Anybody can verify the signatures.

2. rv_w: Anybody can check if the announced winning bid was published on the bulletin board.

Outcome Verification (ov_l,ov_w,ov_s) [ASIACCS'13]

1. The authorities publish the used private keys, here keys 1  and 2  .
2. To verify the result, the parties check if the private keys correspond to the public keys:

3. They repeat the same decryptions as the authorities.

Outcome Verification (ov_l,ov_w,ov_s) [ASIACCS'13]

1. The authorities publish the used private keys, here keys 1  and 2  .
2. To verify the result, the parties check if the private keys correspond to the public keys:

3. They repeat the same decryptions as the authorities.

Case studies

	Brandt	Curtis et al.	Sako
Non-Repudiation	✗	✗	✓
Non-Cancellation	✗	✗	✓
Higest Price Wins	✗	✗	✓
Weak Noninterference	✓	✓	✓
Privacy	✗	(WA)	(SBPS)
Verifiability	✗	✗	✓

Automatic analysis using ProVerif
Computational Proof of Verifiability for Sako's protocol using CryptoVerif

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

- Difficult to understand for a *"normal"* (non-expert) user

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

- Difficult to understand for a *"normal"* (non-expert) user
- **Idea**: Use *physical* properties to ensure verifiability

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

- Difficult to understand for a *"normal"* (non-expert) user
- **Idea**: Use *physical* properties to ensure verifiability

Two protocols:

- Cardako: A *cardboard* version of Sako's protocol

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

- Difficult to understand for a *"normal"* (non-expert) user
- **Idea**: Use *physical* properties to ensure verifiability

Two protocols:

- Cardako: A *cardboard* version of Sako's protocol
- Woodako: A *wooden* box implementation of Sako's protocol

True Bidder-Verifiability: Motivation

Verifiability often heavily relies on complex cryptography:

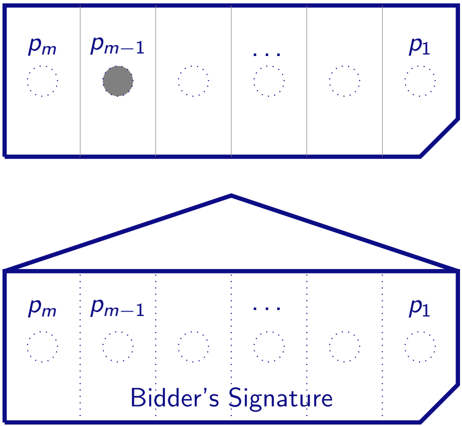
- Difficult to understand for a *"normal"* (non-expert) user
- **Idea**: Use *physical* properties to ensure verifiability

Two protocols:

- Cardako: A *cardboard* version of Sako's protocol
- Woodako: A *wooden* box implementation of Sako's protocol

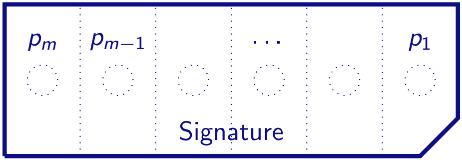
Goal: "Prove Verifiability to your Grandmother and Proverif!"

Cardako: The Protocol



Cardako: The Protocol Cont'd

All the envelopes are swapped between bidders.



Joint determination of the winner

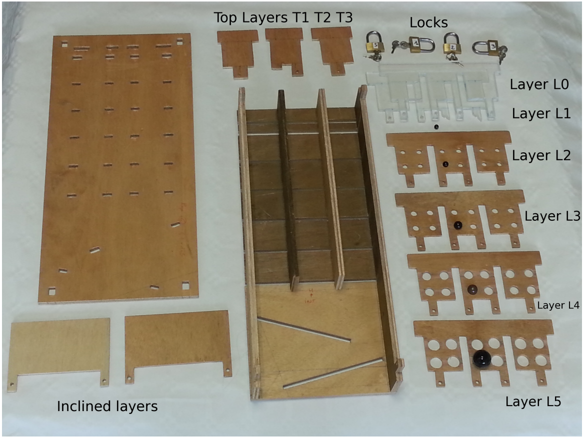
- starting with the highest possible price
- If this succeeds, a bid for this price was found
  - The signature allows the identification of the winner
- If this fails for all bids
  - repeat the procedure for the second price, etc.

Cardako: Formal Analysis Cont'd

Results:

- **Non-Repudiation:** ✓
- **Non-Cancellation:** ✓
- **Weak Non-Interference:** ✓
- **Highest Price Wins:** ✓
- **Verifiability:** ✓
- **Privacy:**
  - Dishonest bidders: open envelopes ✗
  - Honest bidders ✓

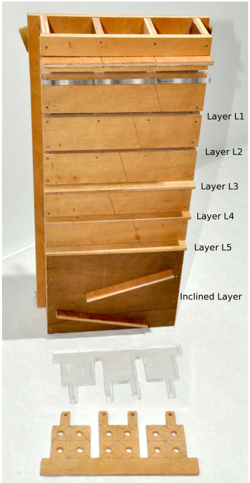
Woodako Protocol: Box



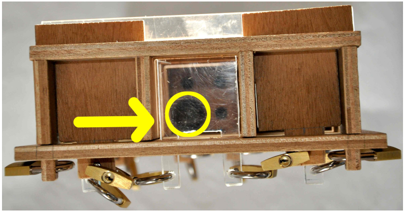
Woodako Protocol: Inside

Woodako Protocol: Setup

The Woodako box after two prices have been tested



Bidder verifiability (i.e. view from top)



Seller verifiability (i.e. view from bottom)

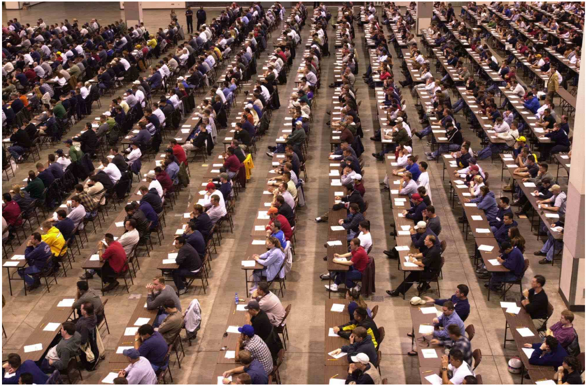
Formal Verification

Results:

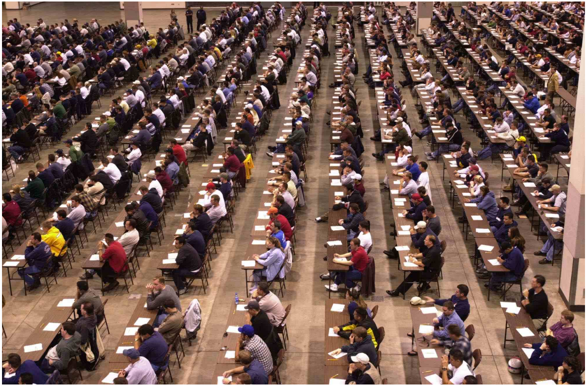
- **Non-Repudiation:** ✓
- **Non-Cancellation:** ✓
- **Weak Non-Interference:** ✓
- **Highest Price Wins:** ✓
- **Verifiability:** ✓
- **Privacy:**
  - Dishonest bidders ✓
  - Honest bidders ✓

Outline

E-exam



E-exam



Information technology for the assessment of knowledge and skills.

Educational assessment

coursera

UDACITY

edX

IELTS™
English for International Opportunity

CISCO™
CCNA

TOEFL® iBT

E-exam : Players

Candidates

Examination Authorities

Examiners

E-exam : Players

Candidates

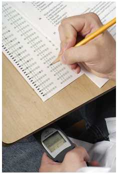
Examination Authorities

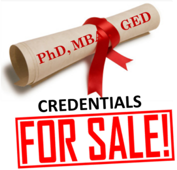
Examiners

4 Phases

1. Registration
2. Examination
3. Marking
4. Notification

Threats



- Candidate cheating
- Bribed examiners
- Untrusted exam authority
- etc. . .

Existing E-exam protocols

- Huszti et al.
- Castella-Roca et al.
- WATA
- NEMO-SCAN

Security Properties

8 properties in 2 categories

- Authentication
- Privacy

Authentication Properties

- Candidate eligibility
- Form authorship during examination (copy - candidate)
- Form authenticity during marking
- Mark authenticity during notification

Pricacy Properties

- Question indistinguishability
- Anonymous marking (link between mark and candidate)
- Anonymous examiners
- Mark priviacy

Application: Huszti's Protocol

"A Secure Electronic Exam System" uses Several phasis

1. Setup
2. Candidate Registration
3. Examinaer Registration
4. Examination
5. Marking
6. Notification

Huszti's Protocol

Setup

1 - EA publishes g and $h = g^s$

2 - $Committee \rightarrow_{priv} EA$:
$\{question, \{question\}_{SSK_{committee}}, time_{x1}\}_{PK_{MIX}}$

Candidate Registration

3 - EA checks C's eligibility, and calculates $\tilde{p} = (PK_C)^s$

4 - $EA \rightarrow NET : \{\tilde{p}, g_C\}$

5- NET calculates $p' = \tilde{p}^{\Gamma}$, and $r = g_C^{\Gamma}$, and stores $time_{nt}$

6 - $NET \rightarrow C : \{p', r\}$

7 - C calculates $p = r^{SK_C}$

8 - $EA \longleftrightarrow C : ZKP_{eq}((p, p'), (g, h))$ $//C$'s pseudonym: (r, p, p')

Huszti's Protocol

Examiner Registration

9 - EA checks E's eligibility, and calculates $\tilde{q} = (PK_E)^s$

10 - $EA \rightarrow E : \{\tilde{q}, g_E\}$

11 - E calculates $q' = \tilde{q}^{\alpha}$, $t = g_E^{\alpha}$, and $q = t^{SK_E}$

12 - $EA \longleftrightarrow E : ZKP_{eq}((q, q'), (g, h))$ 13 - $E \rightarrow EA : \{t, q, q', h\}$

14 - EA checks $q^s = q'$

15 - $E \longleftrightarrow EA : ZKP_{sec}(SK_E)$

16 - EA stores $\{ID_E, PK_E\}_{PK_{MIX}}, h$

Examination

17 - $C \rightarrow EA : \{r, p, p', h\}$

18 - EA checks $p^s = p'$

19 - $C \longleftrightarrow EA : ZKP_{sec}(SK_C)$

20 - $EA \rightarrow C : \{question, \{question\}_{SSK_{committee}}, time_{x1}\}_{PK_{MIX}}$

21 - $C \rightarrow EA : \{r, p, \{answer\}_{PK_{MIX}}, time_{x2}\}$

22 - $EA \rightarrow C : Hash(r, p, p', h, trans_C, question, time_{x1}, time_{x2}$
$\{answer\}_{PK_{MIX}})$

Huszti's Protocol

Marking

23 - $EA \rightarrow E : \{answer\}_{PK_{MIX}}$ // Note that EA stored $\{ID_E, PK_E\}_{PK_{MIX}}, h)$

24 - $E \rightarrow EA :$ $\{mark, Hash(mark, answer), [Hash(mark, answer)]^{SK_E}, verzkp, t, q\}$

25 - $E \longleftrightarrow EA :$ $ZKP_{eq}(Hash(mark, answer), [Hash(mark, answer)]^{SK_E}), (t, q))$

Notification

26 - $EA \rightarrow NET : \{p'\}$ //Note that $r = g_C^{\Gamma}$, $p = PK_C^{\Gamma}$, $p' = g_C^{\Gamma s}$

27 - NET calculates $p' = \tilde{p}^{\Gamma}$

28 - $NET \rightarrow EA : \{p', \tilde{p}\}$

29 - EA publishes $mark, Hash(mark, answer),$ $[Hash(mark, answer)]^{SK_E}, verzkp$

Formal Verification with Proverif

Property	Result
Candidate Eligibility	×
Form Authorship	×
Form Authenticity	×
Mark Authenticity	×
Question Indistinguishability	✓
Anonymous Marking	×
Anonymous Examiner	×
Mark Privacy	×

Outline

Summary

1. Protocol
2. Properties
3. Intruder Model

Summary

1. Protocol
2. Properties
3. Intruder Model

- E-voting
- E-auction
- E-exam

Each application has his own specificity !

3 Lessons to Learn

1. Design a secure protocol is not an easy task.

3 Lessons to Learn

1. Design a secure protocol is not an easy task.
2. Using cryptographic is a good idea but not enough.

3 Lessons to Learn

1. Design a secure protocol is not an easy task.
2. Using cryptographic is a good idea but not enough.
3. Always prove your protocol using fromal methods !

Thanks for your attention

Questions ?

Jean-Louis Lanet : Virus dans une carte mythe ou (proche) réalité ?

Tous les supports de communications connectés par un réseau subissent des attaques. Nous sommes habitués à devoir protéger nos ordinateurs mais aussi depuis peu les téléphones portables et les tablettes sont aussi sujets à des attaques. Récemment un chercheur allemand a réussi à faire exécuter un code arbitraire dans une carte SIM via l'opérateur de télécommunication. Nous présenterons comment un élément aussi sécurisé que la carte à puce pourrait tre sensible à telles attaques.

Guillaume Vernat, Coffreo : La confiance numérique vue du côté de l'utilisateur.

12th December 2013 at 14h00 Amphi B.
http://confiance-numerique.clermont-universite.fr/