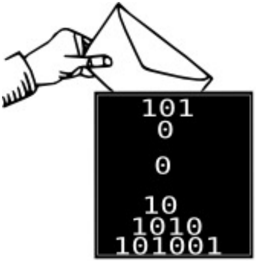
Securité et vérification fomrelle

Pascal Lafourcade

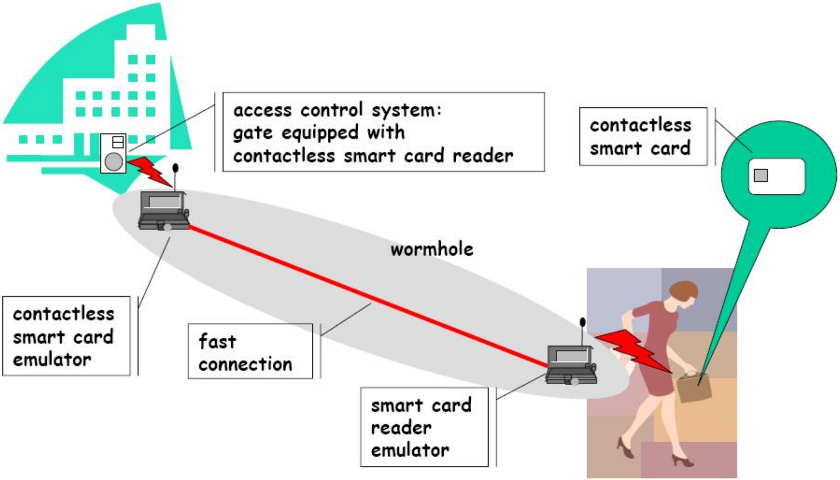
8 mars 2017

Nowadays Security is Everywhere!



Due to the succes of Computer Science.

Wormhole Attack



Paypal Attack

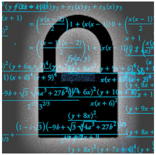
“Model-Based Vulnerability Testing of Payment Protocol Implementations”, Ghazi Maatoug, Frédéric Dadeau and Michael Rusinowitch, Hotspot 2014

Hacking Pacemakers:

Manufacturers are still not putting security first when designing implantable medical devices (2012)

Formal Verification Approaches

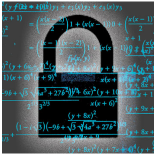
Designer



Attacker

Formal Verification Approaches

Designer

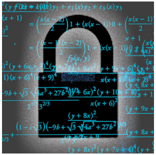


Attacker

Security Team

Formal Verification Approaches

Designer

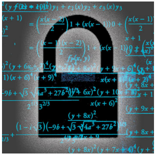


Attacker

Give a proof

Security Team

Formal Verification Approaches



Designer

Attacker

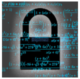
Give a proof

Find a flaw

Security Team

What is cryptography based security?

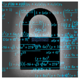
Cryptography:



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

What is cryptography based security?

Cryptography:



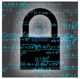
- Primitives: RSA, Elgamal, AES, DES, SHA-3 ...
- Protocols: Distributed Algorithms

Properties:

- Secrecy,
- Authentication,
- Privacy ...

What is cryptography based security?

Cryptography:



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

Properties:

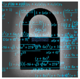
- Secrecy,
- Authentication,
- Privacy ...

Intruders:

- Passive
- Active
- CPA, CCA ...

What is cryptography based security?

Cryptography:



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

Properties:

- Secrecy,
- Authentication,
- Privacy ...

Intruders:

- Passive
- Active
- CPA, CCA ...

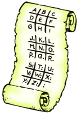
Designing **secure** cryptographic protocols is **difficult**

Security of Cryptographic Protocols

How can we be convinced that a protocols is secure?

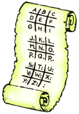
Security of Cryptographic Protocols

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.

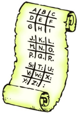
Security of Cryptographic Protocols

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

My Research Topics

- Formal analysis: e-exam, e-voting, e-eauction, SCADA
- Automatic analysis of cryptographic primitives
- WSN: Privacy, Secure Routing, Distance Bounding

Plan

Plan

Traditional security properties

- Common security properties are:
  - Confidentiality or Secrecy: No improper disclosure of information
  - Authentification: To be sure to talk with the right person. disclosure of information
  - Integrity: No improper modification of information
  - Availability: No improper impairment of functionality/service

Authentication

"On the Internet, nobody knows you're a dog."

Mechanisms for Authentication

KNOW	HAVE	ARE	DO
			
Passwords ID Questions Secret Images	Token (Smart) Card Phone	Face Iris Hand/Finger	Behavior Location Reputation

Other security properties

- Perfect Forward Secrecy (PFS) is s a property of key-agreement protocols that ensures that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future.
- Non-repudiation (also called accountability) is where one can establish responsibility for actions.
- Fairness is the fact there is no advantage to play one role in a protocol comparing with the other ones.
- Privacy

  Anonymity: secrecy of principal identities or communication relationships.

  Pseudonymity: anonymity plus link-ability.

  Data protection: personal data is only used in certain ways.

e-services :

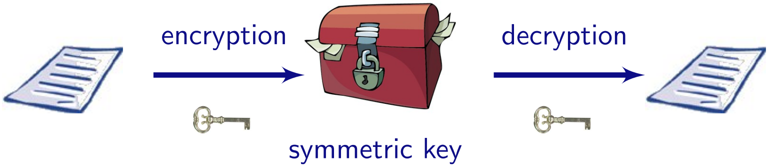
- e-voting
- e-auction
- e-examen
- e-reputation
- e-cash
- ...

Users except more properties and security with electronic services!

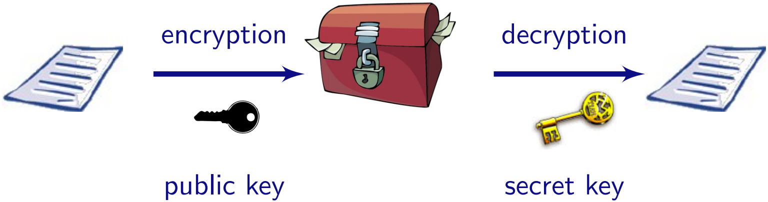
Plan

Symmetric vs Asymmetric Encryption

Symmetric Encryption (DES, AES)

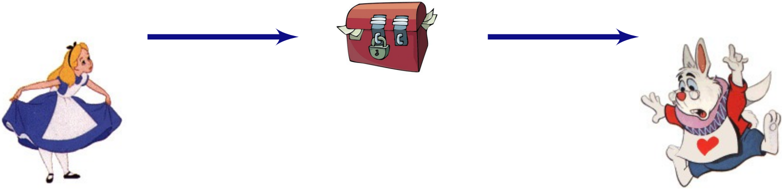


Asymmetric Encryption (RSA, Elgamal ...)

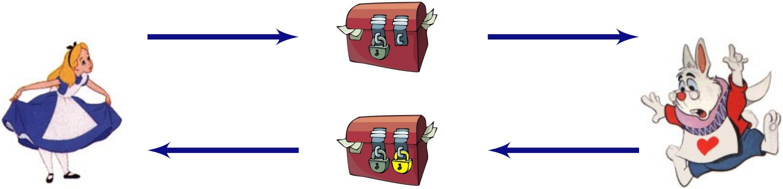


3-pass Shamir

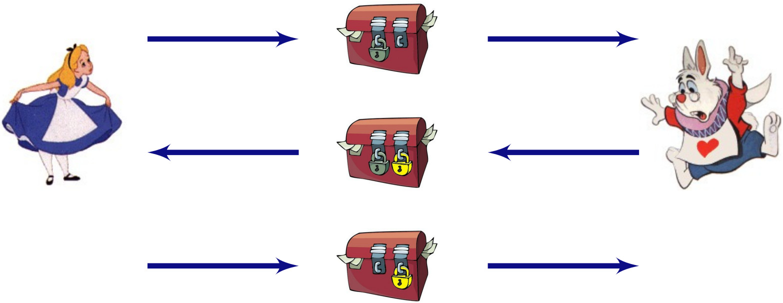
3-pass Shamir



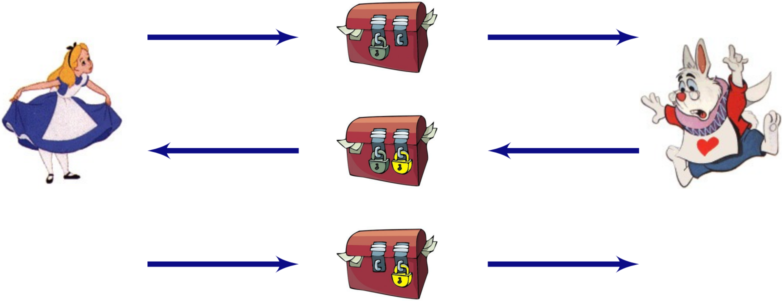
3-pass Shamir



3-pass Shamir



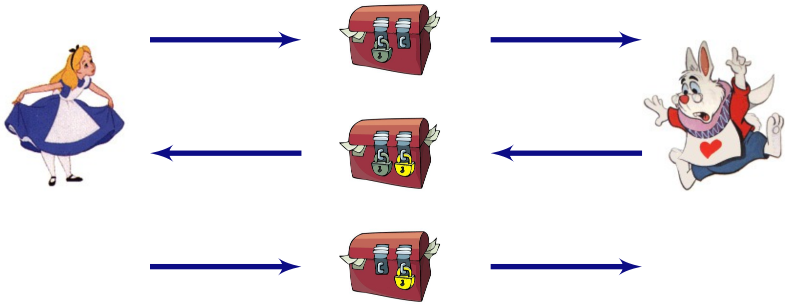
3-pass Shamir



Abstract Representation

$$1 \quad A \rightarrow B : \{m\}_{K_A}$$

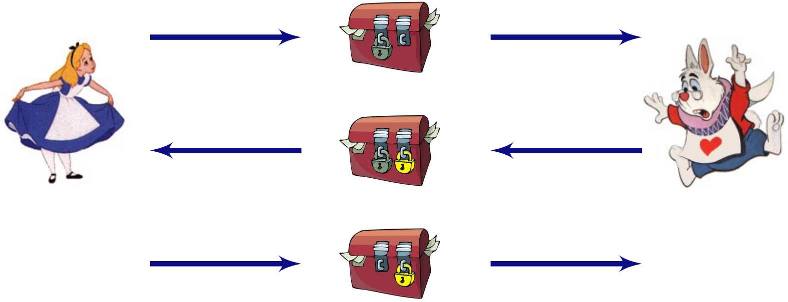
3-pass Shamir



Abstract Representation

$$
\begin{array}{lllll}
1 & A & \rightarrow & B & : \{m\}_{K_A} \\
2 & B & \rightarrow & A & : \{\{m\}_{K_A}\}_{K_B}
\end{array}
$$

3-pass Shamir

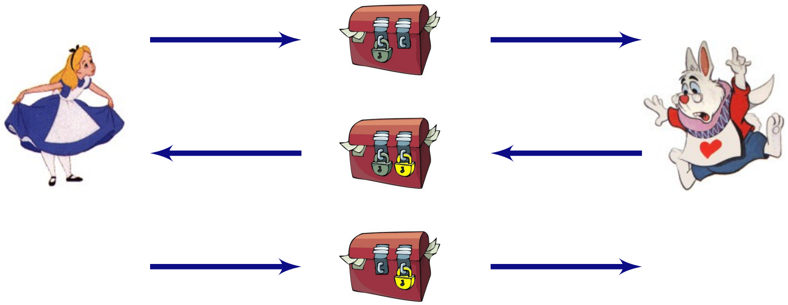


Abstract Representation

$$
\begin{array}{llcll}
1 & A & \rightarrow & B & : \{m\}_{K_A} \\
2 & B & \rightarrow & A & : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}
\end{array}
$$

Commutative Encryption

3-pass Shamir



Abstract Representation

$$
\begin{array}{lllll}
1 & A & \rightarrow & B & : \{m\}_{K_A} \\
2 & B & \rightarrow & A & : \{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A} \\
3 & A & \rightarrow & B & : \{m\}_{K_B}
\end{array}
$$

Commutative Encryption

Example

Needham Schroeder Key Echange 1976

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

- Use cryptography
- Small programs
- Distributed

Plan

Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$\{m\}_k = m \oplus k$

XOR Properties (ACUN)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ — **A**ssociativity
- $x \oplus y = y \oplus x$ — **C**ommutativity
- $x \oplus 0 = x$ — **U**nity
- $x \oplus x = 0$ — **N**ilpotency

Logical Attack on Shamir 3-Pass Protocol (I)

Perfect encryption one-time pad (Vernam Encryption)

$\{m\}_k = m \oplus k$

XOR Properties (ACUN)

- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ — **A**ssociativity
- $x \oplus y = y \oplus x$ — **C**ommutativity
- $x \oplus 0 = x$ — **U**nity
- $x \oplus x = 0$ — **N**ilpotency

Vernam encryption is a commutative encryption :

$$\{\{m\}_{K_A}\}_{K_I} = (m \oplus K_A) \oplus K_I = (m \oplus K_I) \oplus K_A = \{\{m\}_{K_I}\}_{K_A}$$

Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$\{m\}_k = m \oplus k$

Shamir 3-Pass Protocol

$$
\begin{array}{llcll}
1 & A & \rightarrow & B: & m \oplus K_A \\
2 & B & \rightarrow & A: & (m \oplus K_A) \oplus K_B \\
3 & A & \rightarrow & B: & m \oplus K_B
\end{array}
$$

Passive attacker :

$$m \oplus K_A \qquad m \oplus K_B \oplus K_A \qquad m \oplus K_B$$

Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

$\{m\}_k = m \oplus k$

Shamir 3-Pass Protocol

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & m \oplus K_A \\ 2 & B & \rightarrow & A: & (m \oplus K_A) \oplus K_B \\ 3 & A & \rightarrow & B: & m \oplus K_B \end{array}$$

Passive attacker :

$$m \oplus K_A \oplus m \oplus K_B \oplus K_A \oplus m \oplus K_B = m$$

Cryptography is not sufficient !

Example : Needham Schroeder Key Echange

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

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

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

Cryptography is not sufficient !

Example : Needham Schroeder Key Echange

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

Broken 17 years after, by G. Lowe

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

Cryptography is not sufficient !

Example : Needham Schroeder Key Echange

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

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

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

Broken 17 years after, by G. Lowe

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

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

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

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

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

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

Computer-Aided Security

Plan

Necessity of Tools

- Protocols are small recipes.
- Non trivial to design and understand.
- The number and size of new protocols.
- Out-pacing human ability to rigourously analyze them.

GOAL : A tool is finding flaws or establishing their correctness.

- completely automated,
- robust,
- expressive,
- and easily usable.

Existing Tools: AVISPA, Scyther, Proverif, Hermes, Casper/FDR, Murphi, NRL ...

Questions?

How can we find such attacks automatically?

- Models for Protocols
- Models for Properties
- Theories and Dedicated Techniques
- Tools
  - Automatic
  - Semi-automatic

Why is it difficult to verify such protocols?

- Messages: Size not bounded
- Nonces: Arbitrary number
- Intruder: Unlimited capabilities
- Instances: Unbounded numbers of principals
- Interleaving: Unlimited applications of the protocol.

Complexity

Complexity depends of intruder capabilities. In classical Dolev-Yao intruder model we (pair + encryption) we have the following results:

- Passive Intruder
  Problem is polynomial
- Bounded Number of sessions
  Problem is NP-complete
  Tools can verify 3-4 sessions: useful to finds flaws ! OFMC, Cl-Atse, SATMC, FDR, Athena...
- Unbounded Number of sessions
  Problem is in general undecidable
  Tools are corrects, but uncomplete (can find false attacks, can not terminate) Proverif, TA4SP, Scyther.

Which tool for what ?

	Proverif	Scyther	OFMC	Cl-atse	TA4SP	SAT-MC
Secrecy	X	X	X	X	X	X
Authentication	X	X	X	X	X	X
Equivalence Obs	X					
Bounded nb S		X	X	X	X	X
Unbounded nb S	X	X			X	
Xor	x		X	X		
DH	x		X	X		
Fast	X	X				
User friendly		X				

Success Story of Formal Verification

Tools based on different theories for several properties

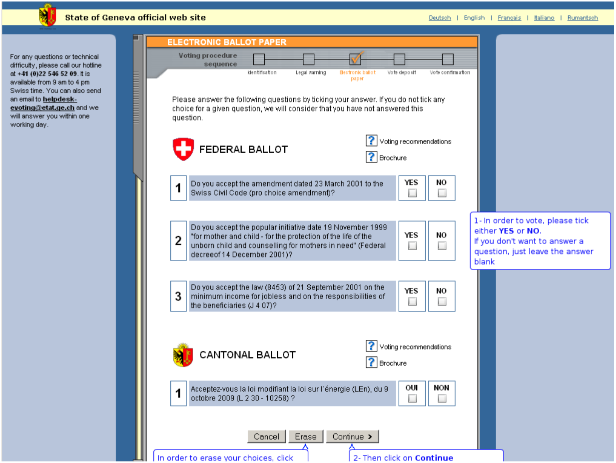
1995	Casper/FRD [Lowe]
2001	Proverif [Blanchet]
2003	Proof of certified email protocol with Proverif [AB]
	OFMC [BMV]
	Hermes [BLP]
	Flaw in Kerberos 5.0 with MSR 3.0 [BCJS]
2004	TA4SP [BHKO]
2005	SATMC [AC]
2006	CL-ATSE [Turuani]
2008	Scyther [Cremers]
	Flaw of Single Sign-On for Google Apps with SAT-MC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

Plan

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 [FPS'11, ICC'12 WS-SFCS,ESORICS'12]:

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

4 Dimensions for Privacy [FPS'11, ICC'12 WS-SFCS]

Modeling in Applied π-Calculus

1. **Communication btwn the attacker & the targeted voter**

   [DKR09]  

   Vote-Privacy (VP) Receipt-Freeness (RF) Coercion-Resistance (CR)

2. **Intruder is controlling another voter**

   Outsider (O)  Insider (I)

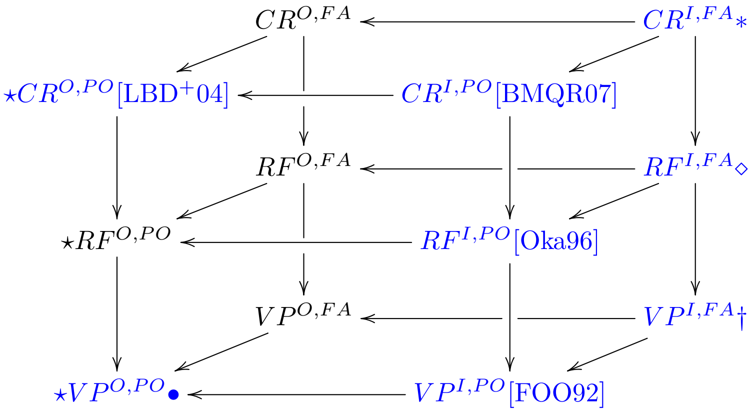
3. **Secure against Forced-Abstention**: (FA) or not (PO)

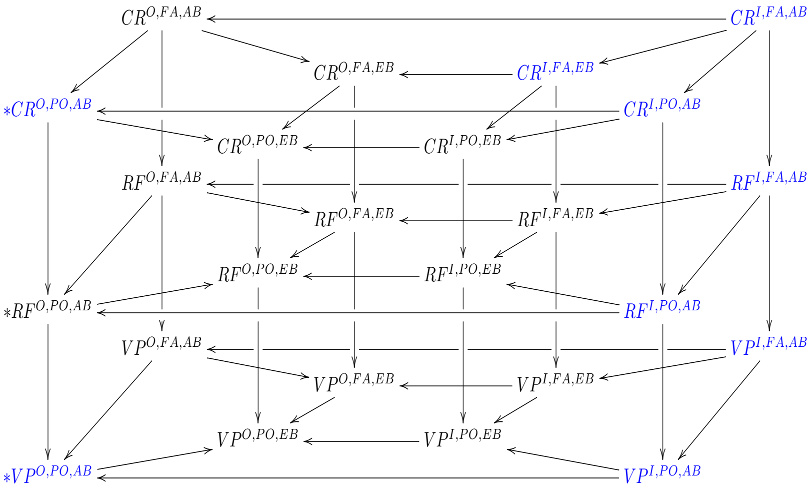
4. **Honest voters behavior**:

   $\exists$  $\forall$

Relations without ∃ and ∀ [FPS'11, ICC'12 WS-SFCS]



All relations among the notions [FPS'11, ICC'12 WS-SFCS]



Privacy for Weighted Votes [ESORICS'12]

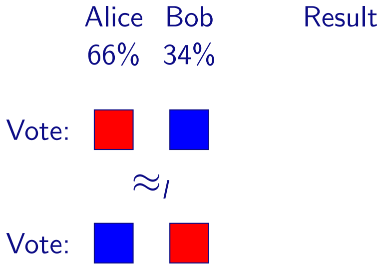
Alice Bob Result

Vote: (red) (blue)

$\approx_l$

Vote: (blue) (red)

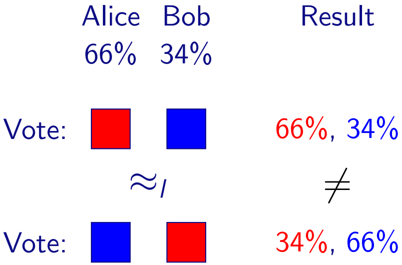
Privacy for Weighted Votes [ESORICS'12]



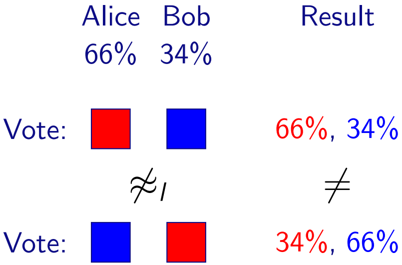
Privacy for Weighted Votes [ESORICS'12]

	Alice	Bob	Result
	66%	34%	
Vote:			66%, 34%
	$\approx_l$		
Vote:			34%, 66%

Privacy for Weighted Votes [ESORICS'12]

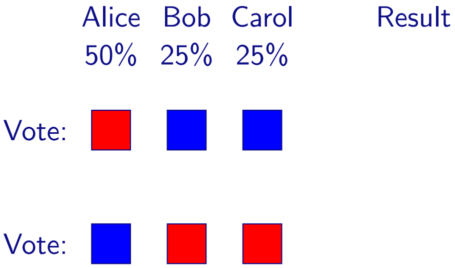


Privacy for Weighted Votes [ESORICS'12]



Privacy for Weighted Votes [ESORICS'12]

Still: Some privacy is possible!



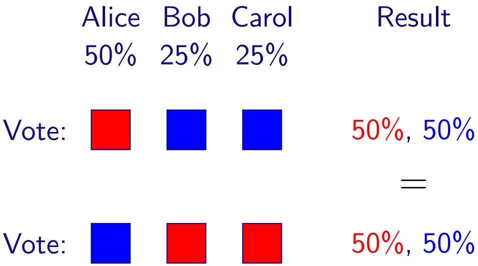
Privacy for Weighted Votes [ESORICS'12]

Still: Some privacy is possible!

	Alice	Bob	Carol	Result
	50%	25%	25%	
Vote:	■ (red)	■ (blue)	■ (blue)	50%, 50%
Vote:	■ (blue)	■ (red)	■ (red)	50%, 50%

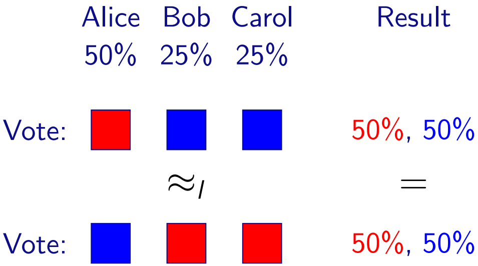
Privacy for Weighted Votes [ESORICS'12]

Still: Some privacy is possible!



Privacy for Weighted Votes [ESORICS'12]

Still: Some privacy is possible!



Vote-Privacy (VP) for weighted votes [ESORICS'12]

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

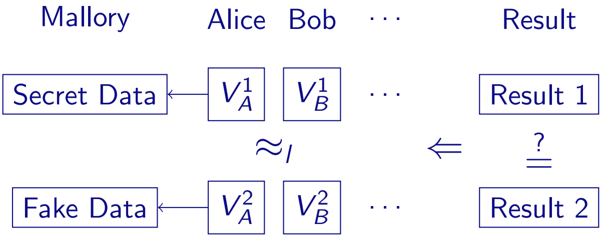
Single-Voter Receipt Freeness (SRF) [ESORICS'12]

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) [ESORICS'12]

Mallory		Alice	Bob	$\cdots$		Result
Secret Data	$\longleftarrow$	V_A^1	V_B^1	$\cdots$		Result 1
		$\approx_l$			$\Leftarrow$	$\stackrel{?}{=}$
Fake Data	$\longleftarrow$	V_A^2	V_B^2	$\cdots$		Result 2

Single-Voter Receipt Freeness (SRF) [ESORICS'12]



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

Multi-Voter Receipt Freeness (MRF) [ESORICS'12]

...

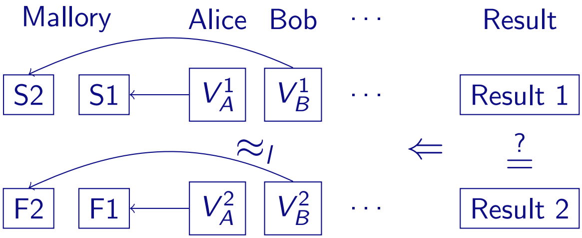
Mallory | Alice | Bob | $\cdots$ | Result

S1 $\leftarrow$ V_A^1 | V_B^1 | $\cdots$ | Result 1

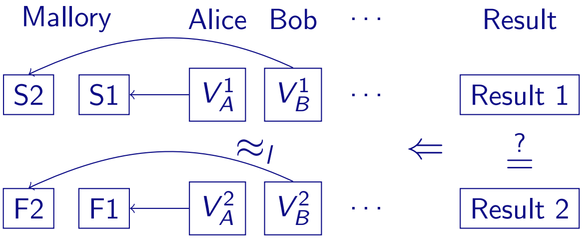
$\approx_l$ $\Leftarrow$ $\stackrel{?}{=}$

F1 $\leftarrow$ V_A^2 | V_B^2 | $\cdots$ | Result 2

Multi-Voter Receipt Freeness (MRF) [ESORICS'12]

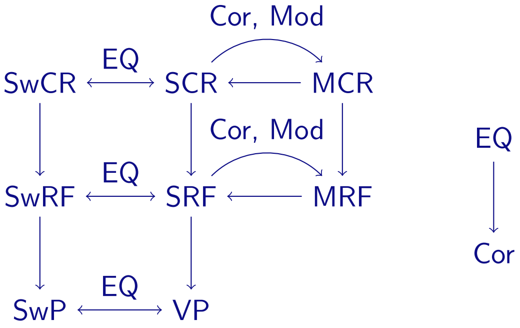


Multi-Voter Receipt Freeness (MRF) [ESORICS'12]



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

One Coerced Voter is enough! [ESORICS'12]



Unique decomposition of processes in the applied π-calculus.

Plan

Plan

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 ebay

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

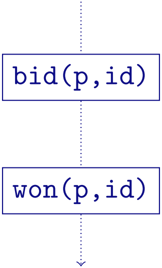
Events [POST'13]

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]

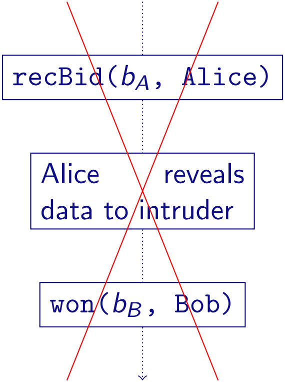
On every trace:



Non-Cancellation [POST'13]

Alice Bob

Bid b_A $>$ b_B



Strong Noninterference & Weak Noninterference [POST'13]

Definition (Strong Noninterference (SN))

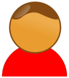
An auction protocol ensures *Strong Noninterference (SN)* if for any two auction processes AP_A and AP_B that halt at the end of the bidding phase (i.e. where we remove all code after the last `recBid` event) we have $AP_A \approx_l AP_B$.

Definition (Weak Noninterference (WN))

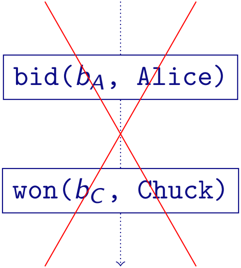
Like Strong Noninterference, but we consider only processes with the same bidders.

Highest Price Wins [POST'13]

Alice (honest)	Chuck (corrupted)

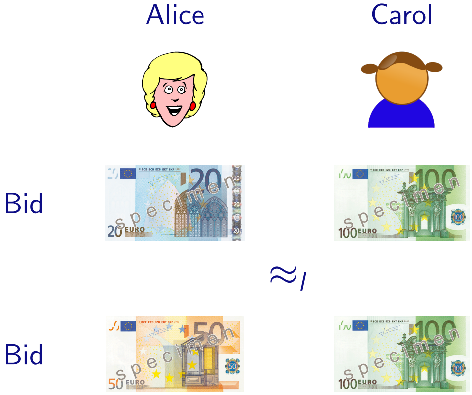


Bid b_A $>$ b_C



Strong Bidding-Price Secrecy (SBPS) [D10]

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.

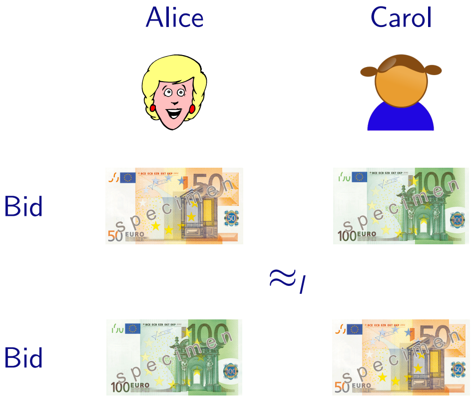
	Alice	Carol
		
Bid		

$$\approx_l$$

Bid		

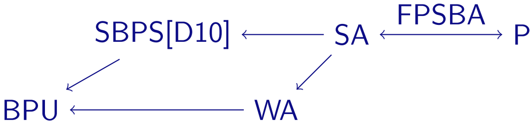
Weak Anonymity (WA) [POST'13]

Unlinkability, but also for the winner.

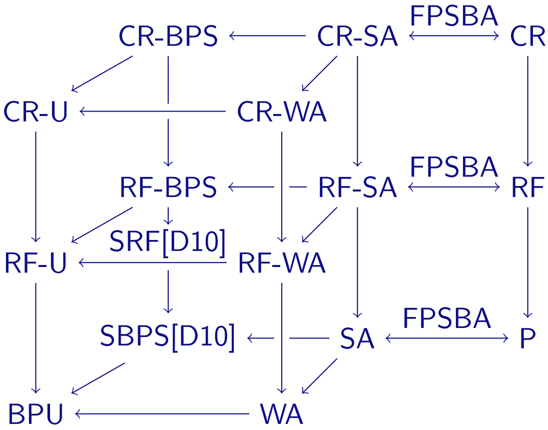


e-Auctions: Hierarchy of Privacy Notions [POST'13]

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



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



Protocol by Curtis et al.[C07]: Registration [POST'13]

Main idea: a registration authority (RA) distributes pseudonyms, which are then used for bidding.

Bidder

Registration Authority

Protocol by Curtis et al.[C07]: Registration [POST'13]

Main idea: a registration authority (RA) distributes pseudonyms, which are then used for bidding.

Bidder

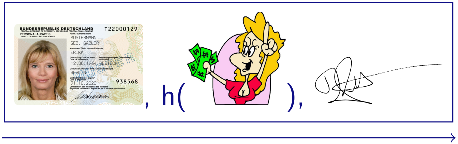
Registration Authority

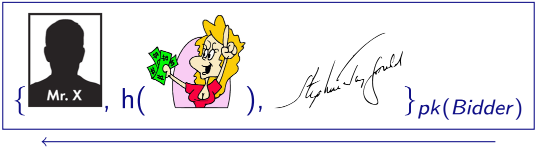
, h(), 

Protocol by Curtis et al.[C07]: Registration [POST'13]

Main idea: a registration authority (RA) distributes pseudonyms, which are then used for bidding.

Bidder | Registration Authority





Bidding [POST'13]

The bidder uses his pseudonym to submit his bids.

Bidder

Registration Authority

Bidding [POST'13]

The bidder uses his pseudonym to submit his bids.

Bidder

Registration Authority

, {

}$_{pk(Auctioneer)}$,

Bidding [POST'13]

The bidder uses his pseudonym to submit his bids.

Bidder → Registration Authority

, {  }$_{pk(Auctioneer)}$, 

→

{  }$_{pk(Auctioneer)}$, 

←

Bidding Cont'd [POST'13]

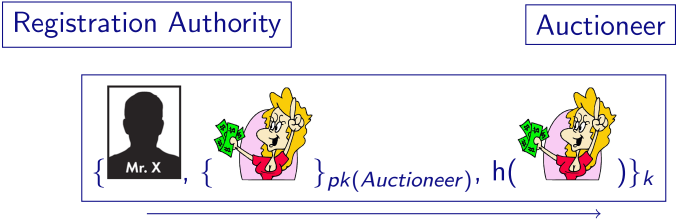
The Registration Authority forwards the bids to the auctioneer, encrypted using a symmetric key k, which is revealed at the end.

Registration Authority

Auctioneer

Bidding Cont'd [POST'13]

The Registration Authority forwards the bids to the auctioneer, encrypted using a symmetric key k, which is revealed at the end.



Bidding Cont'd [POST'13]

The Registration Authority forwards the bids to the auctioneer, encrypted using a symmetric key k, which is revealed at the end.

Registration Authority → Auctioneer

{

, {$\}_{pk(Auctioneer)}$, h($)\}_k$

k, n

Completion [POST'13]

The auctioneer decrypts the bids using k and his secret key $sk(Auctioneer)$, and announces the winning pseudonym.

Registration Authority

Auctioneer

Completion [POST'13]

The auctioneer decrypts the bids using k and his secret key $sk(Auctioneer)$, and announces the winning pseudonym.

Analysis [POST'13]

Formal analysis using ProVerif:

- **Non-Repudiation:** ✗ attack, the messages from the RA to the auctioneer are not authenticated - anybody can impersonate the RA
- **Non-Cancellation:** ✗ same attack
- **Highest Price Wins:** ✗ same attack
- **Weak Noninterference:** (✓) OK if first message (hash of bid) is encrypted.
- **Privacy:** (✓) Weak Anonymity if first message is encrypted and synchronization is added

 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)

Verification Test [ASIACCS'13]

Definition (Verification Test)

Efficient terminating algorithm: Data $\rightarrow$ Bool

- Input : data visible to a participant
- Output : Boolean value.

The protocol model [ASIACCS'13]

Definition (Auction protocol)

($\mathcal{B}$, S, $\mathbb{L}$, *getPrice*, *isReg*, *win*, *winBid*) where

- $\mathcal{B}$ is the set of bidders and S is the seller,
- $\mathbb{L}$ is a list of all submitted bids,
- $getPrice\colon EBid \mapsto Bid$
- $isReg\colon EBid \mapsto Bool$
- $win\colon List(Bid) \mapsto Index$
- *winBid* is a variable of the index of the winning bid at the end.

Verifiability for First-Price Auctions [ASIACCS'13]

Definition (Verifiability - 1st-Price Auctions)

($\mathcal{B}$, S, $\mathbb{L}$, *getPrice*, *isReg*, *win*, *winBid*) ensures *Verifiability* if the following Verification Tests rv_s, rv_w, ov_l, ov_w, ov_s are sound:

1. **Registration and Integrity Verifiability (RV)**:
   - $rv_s = true \rightarrow \forall b \in \mathbb{L}$: $isReg(b) = true$
   - $rv_w = true \rightarrow winBid \in Indices(\mathbb{L})$
2. **Outcome Verifiability (OV)**:
   - $ov_l = true \rightarrow myBid \neq win(getPrice(\mathbb{L}))$
   - $ov_w = true \rightarrow myBid = win(getPrice(\mathbb{L}))$
   - $ov_s = true \rightarrow winBid = win(getPrice(\mathbb{L}))$

And complete:

- If all participants follow the protocol correctly, the above tests succeed ($\Longleftarrow$).

Simple Example [ASIACCS'13]

1. All bidders publish their bids on a bulletin board[1].
2. At the end the auctioneer announces the winner.

Verification tests:

- ov_l, ov_w & ov_s: everybody can compute the winner on the public list of unencrypted bids
- rv_w: anyone can test if the winning bid is published
- rv_s: no sound test possible.

[1] not encrypted and not signed

Simple Example [ASIACCS'13]

1. All bidders publish their bids on a bulletin board[1].
2. At the end the auctioneer announces the winner.

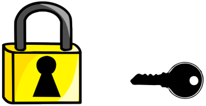
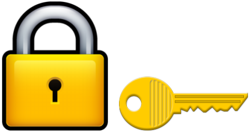
Verification tests:

- ov_l, ov_w & ov_s: everybody can compute the winner on the public list of unencrypted bids
- rv_w: anyone can test if the winning bid is published
- rv_s: no sound test possible. Solution: add signatures

[1] not encrypted and not signed

Protocol by Sako [S00][ASIACCS’13]

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

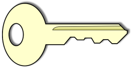
- Price 10 €: 
- Price 5 €: 
- Price 1 €: 

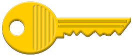
Set up [ASIACCS'13]

A public constant c

Bulletin Board

Authorities





Bidding Phase [ASIACCS'13]

Select a Price

- For 5 €:

- For 1 €:

Bidding Cont'd [ASIACCS'13]

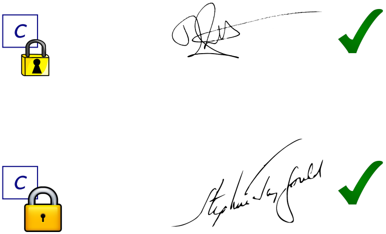
The signed bids are published on the bulletin board:

Bid Opening [ASIACCS'13]

1. The signatures are checked.

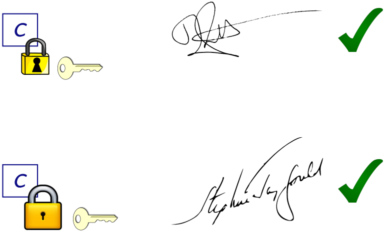
Bid Opening [ASIACCS'13]

1. The signatures are checked.



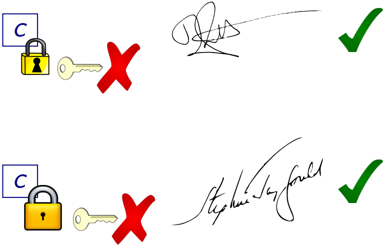
Bid Opening [ASIACCS'13]

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



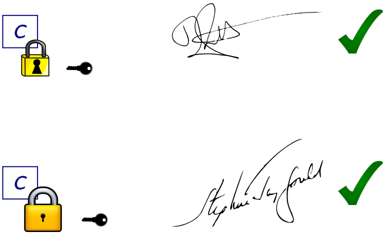
Bid Opening [ASIACCS'13]

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



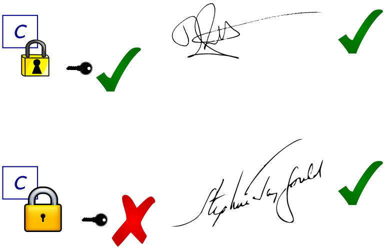
Bid Opening [ASIACCS'13]

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.

Analysis [ASIACCS'13]

The verification tests are sound and complete, proof using ProVerif and CryptoVerif.
Necessary hypotheses (CryptoVerif):

- A UF-CMA signature scheme
- A correct encryption scheme with the following properties:
  - A function `pkey` that computes the public key given the secret key.
  - Either of two private keys giving the same public key can be used to decrypt correctly.

Plan

Bitcoin : monnaie électronique

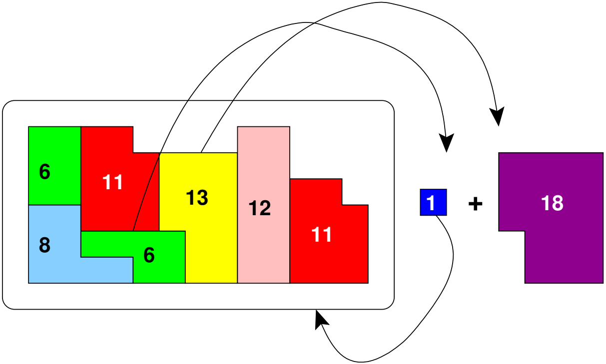
Créée en 2008 par Satoshi Nakamoto (1 BTC $\approx$ 945 euros)

1	BTC = 1 Bitcoin	
0,01	BTC = 1 cBTC	= 1 centiBitcoin (ou bitcent)
0,001	BTC = 1 mBTC	= 1 milliBitcoin
0,000 001	BTC = 1 μBTC	= 1 microBitcoin
0,000 000 01	BTC = 1 Satoshi	

Taux de change du bitcoin

Payer 18 BTC avec des pièces

Nous acceptons
Les **bitcoins**



Clef symétrique

Exemples

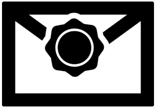
- DES
- AES

Chiffrement à clef publique

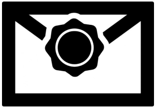
Exemples

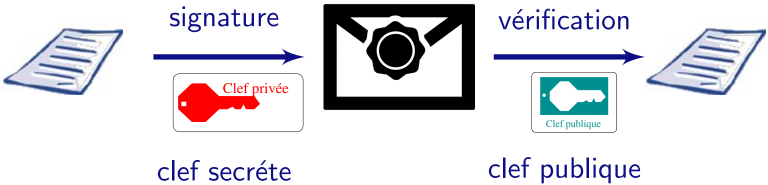
- RSA : $c = m^e \mod n$
- ElGamal : $c \equiv (g^r, h^r \cdot m)$

Signature



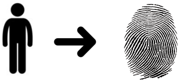
Signature



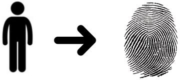


RSA: $m^d \mod n$

Fonction de Hachage (RIPEMD-160, SHA-256, SHA-3)

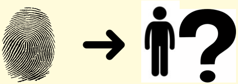


Fonction de Hachage (RIPEMD-160, SHA-256, SHA-3)

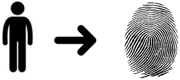


Propriétés de résitance

- Pré-image

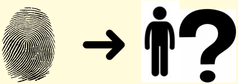


Fonction de Hachage (RIPEMD-160, SHA-256, SHA-3)

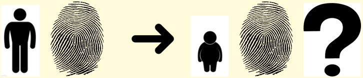


Propriétés de résitance

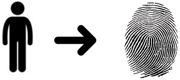
- Pré-image



- Seconde Pré-image

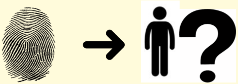


Fonction de Hachage (RIPEMD-160, SHA-256, SHA-3)

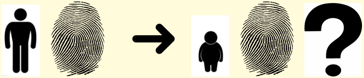


Propriétés de résitance

- Pré-image



- Seconde Pré-image



- Collision

Propriétés d'une monnaie électronique

- Non-Falsifiable (Unforgeable)

- Eviter la double dépense & identifiation fraudeur & "présemption d'innocence"

- Respect de la vie privée :
  - Anonymat faible : non identification d'un acheteur
  - Anonymat fort : non traçabilité d'un acheteur

Bitcoins : caractéristiques

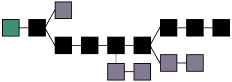
- Le nombre total de bitcoins est **fini**

21 millions BTC

- Les transactions utilisent des **PKI**
- Numéro de compte :

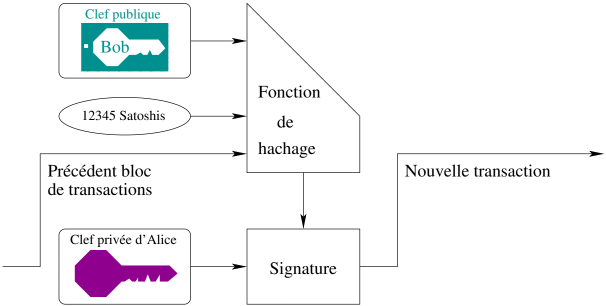
$$\text{RIPEMD-160}(\text{SHA-256}(\text{ECDSA}_{pub}))$$

- Toutes les transactions sont **publiques**
- **Blockchain** : un système pair-à-pair qui garantit la validité des transactions

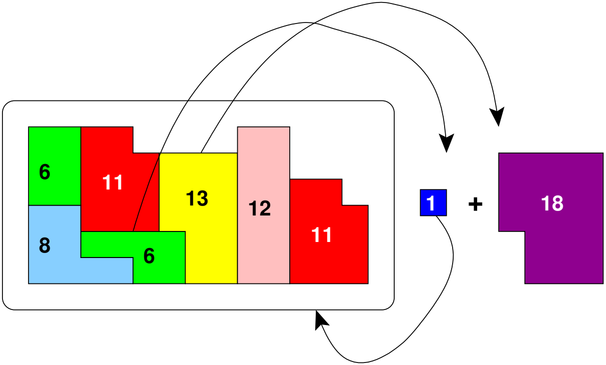


Comment faire une transaction?

Alice donne 12345 Satochis ($\approx 5c$) à Bob.



Payer 18 BTC avec des pièces

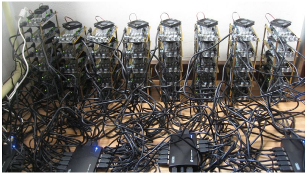


- Seuls des bitcoins possédés peuvent être dépensés

Miner des Bitcoins

Miner des Bitcoins

Les "*mineurs*" valident les transactions contre des bitcoins



Miner des Bitcoins

- Valider = résoudre un **objectif de hachage**
- Récompense initiale 50 BTC pour une validation
- Divisée par 2 tous les 210000 validations

$$\sum_{i=0}^{32} \frac{50}{2^i} \times 210\,000 = 21 \text{ millions BTC}$$

Miner : Objectif de hachage

Cible = 00000000000000000254845fa930deac4086b3e3bce21147e93f463b206d8076

Trouver une nombre n tel que

$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, n)) = x < Cible$$

Avoir un 0 plus de au début de x

Miner : Objectif de hachage

Cible = 00000000000000000254845fa930deac4086b3e3bce21147e93f463b206d8076

Trouver une nombre n tel que

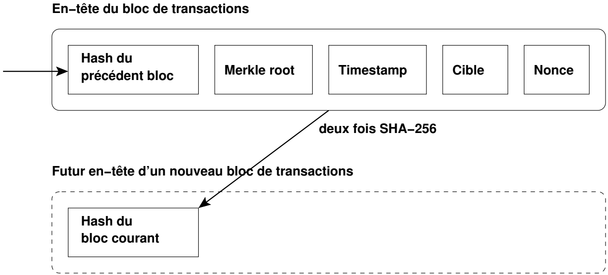
$$\text{SHA-256}(\text{SHA-256}(\text{Transactions}, n)) = x < \mathit{Cible}$$

Avoir un 0 plus de au début de x

Stratégie : brute force

Tester toutes les valeurs possibles de n

Miner : Proof of work



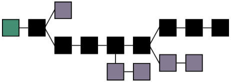
Avoir un zéro de plus au début

SHA-256(SHA-256(en-tête de bloc))

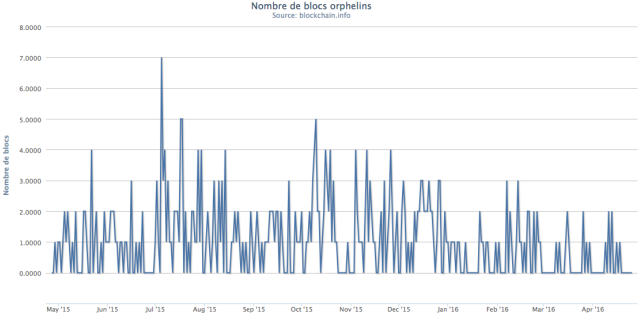
- les transactions passées (95 Go)
- les transactions à valider
- les secondes depuis 01/01/1970
- un nonce
- etc ...

Miner = Validation des transactions

Cible: 00000000000000000254845fa930deac4086b3e3bce21147e93f463b206d8076



- La chaîne la plus longue persiste (attaque 51 %)
- Validation toutes les 10 minutes (6 confirmations)



Autres crypto-monnaies

Classification des Altcoins

1. "Pourris coins"
2. Clônes de Bitcoin
3. Minage plus utiles, moins énergivores
4. Non-basés sur la preuve de travail
    - Proof of Stake (Peercoin)
    - Proof of Retreivability (Permacoin)
    - Proof of Capacity (Burstcoin)
    - Proof of Space (SpaceMint)

Bitcoin : Crypto-monnaie dématérialisée décentralisée

- Preuve de travail = Objectif de Hachage
- Création de la monnaie = récompense aux mineurs
- Miner = difficile + energivore



Bitcoin : Crypto-monnaie dématérialisée décentralisée

- Preuve de travail = Objectif de Hachage
- Création de la monnaie = récompense aux mineurs
- Miner = difficile + energivore

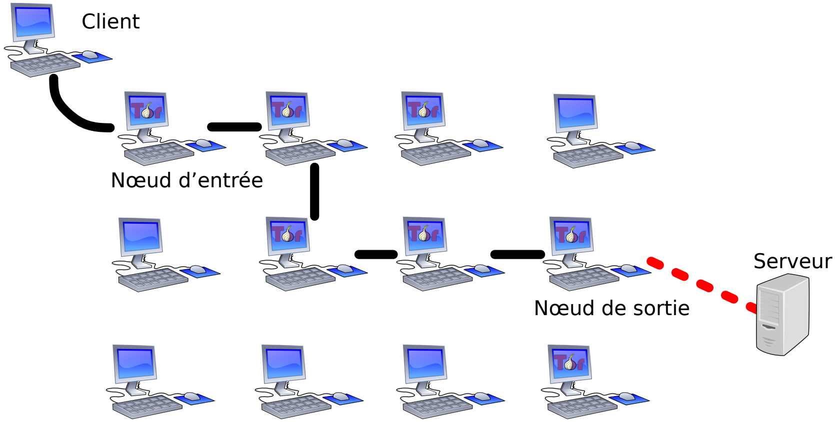


- Perte ou vol de la clef secrète = irréversible
- Monnaie anonyme et traçable



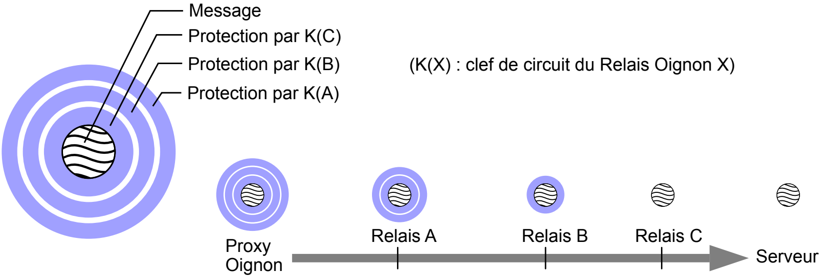
Plan

Application : The Onion Router (TOR)



https://www.torproject.org

Application : Tor



Plan

Merci pour votre attention.

Questions ?