(In)Security of e-voting

Pascal Lafourcade

UNIVERSITÉ Clermont Auvergne

LIMOS

Algotel 2021

Nowadays Security is Everywhere!

Outline

Security: Cryptography

Cryptography

Security: Cryptography

Cryptography

Primitives
RSA, Elgamal,
AES, DES, SHA-3...

Security: Cryptography

Cryptography

Primitives
RSA, Elgamal,
AES, DES, SHA-3...

Protocols
Distributed
Programs

Security: Cryptography for a Property

Primitives
RSA, Elgamal,
AES, DES, SHA-3...

Cryptography

Please Login

Username: Username

Password: ••••••

☐ Remember Password

Protocols
Distributed
Programs

Security: Cryptography for a Property in an Hostile Environment

Cryptography


Please Login

Username: Username

Password: ●●●●●●

☐ Remember Password

Primitives
RSA, Elgamal,
AES, DES, SHA-3...

Protocols
Distributed
Programs

Security:Cryptography for a Property in an Hostile Environment

Cryptography

Verification

Please Login

Username: Username

Password: ••••••

☐ Remember Password

Primitives
RSA, Elgamal,
AES, DES, SHA-3...

Protocols
Distributed
Programs

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

✓ Publish the protocol and wait until someone finds an attack.

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

- ✗ Publish the protocol and wait until someone finds an attack.
- ✓ Prove that there is no attack.

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

✗ Publish the protocol and wait until someone finds an attack.

✓ Prove that there is no attack.

Usual problems with proofs:

- proving is a difficult task,
- pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is a good one?

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

✗ Publish the protocol and wait until someone finds an attack.

✓ Prove that there is no attack.

Usual problems with proofs:

- proving is a difficult task,
- pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is a good one?

✓ Publish the proof and wait until someone finds a mistake.

Designing Secure Schemes is Difficult!

How can we be convinced that a protocol is a good one?

✗ Publish the protocol and wait until someone finds an attack.

✓ Prove that there is no attack.

Usual problems with proofs:

- proving is a difficult task,
- pencil-and-paper proofs are error-prone.

How can we be convinced that a proof is a good one?

✗ Publish the proof and wait until someone finds a mistake.

✓ Computer-Aided Security.

Why Verification is Useful !

Formal Security Verification Team

Formal Security Verification Team

Formal Security Verification Team

Formal Security Verification Team

Success Story of Verification in Security

1995

≥ 17

(Casper/FDR)

Success Story of Verification in Security

1995      ≥ 17 
(Casper/FDR)
2003: ProVerif certified email protocol (B. Blanchet et al)

Success Story of Verification in Security

1995

≥ 17

(Casper/FDR)

2003: ProVerif certified email protocol (B. Blanchet et al)

2005: Flaw in Kerberos 5.0 with MSR 3.0 (I. Cervesato et al)

Success Story of Verification in Security

1995  ≥ 17 

(Casper/FDR)

2003: ProVerif certified email protocol (B. Blanchet et al)

2005: Flaw in Kerberos 5.0 with MSR 3.0 (I. Cervesato et al)

AVISPA AVANTSSAR SATMC(A. Armando et al)

2008:
- Unknown Security flaw of Single Sign-On for Google Apps
- Proof of TLS using Proverif (Fournet et al)

Success Story of Verification in Security

1995  ≥ 17 

(Casper/FDR)

2003: ProVerif certified email protocol (B. Blanchet et al)

2005: Flaw in Kerberos 5.0 with MSR 3.0 (I. Cervesato et al)

AVISPA AVANTSSAR SATMC(A. Armando et al)

2008:
- Unknown Security flaw of Single Sign-On for Google Apps
- Proof of TLS using Proverif (Fournet et al)

2010: TOOl for cryptoKi ANalysis (G. Steel et al)

TOOKAN

Success Story of Verification in Security

1995  ≥ 17 

(Casper/FDR)

2003: ProVerif certified email protocol (B. Blanchet et al)

2005: Flaw in Kerberos 5.0 with MSR 3.0 (I. Cervesato et al)

AVISPA AVANTSSAR SATMC (A. Armando et al)

2008:
- Unknown Security flaw of Single Sign-On for Google Apps
- Proof of TLS using Proverif (Fournet et al)

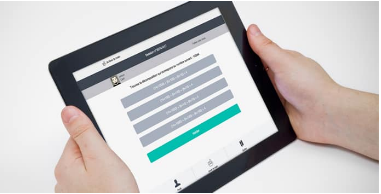
2010: TOOl for cryptoKi ANalysis (G. Steel et al) TOOKAN 

2019: UKano (L. Hirschi et al) 

Other Tools: Athena, Brutus, Certycrypt, CL-ATSE, Coprové, Cryptoverif, Easycrypt, Hermes, Murphy, OFMC, Scyther, TA4SP, Tamarin ...

Outline

E-Voting *vs* Traditional Voting



Vote électronique

Vote traditionnel

- \+ Accessibility
- \+ Reducing the abstention rate
- \+ Automatic counting
- \+ Less organisation costs

Two e-voting (1/2)

Offline

+ Efficient and fast counting
+ Vote in any voting station
- Trust the machines

Two e-voting (2/2)

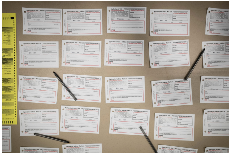
Online

- \+ Vote at home
- \+ Easy process
- \+ Less costs
- \- Possible influence

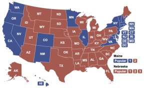
Voting Protocol Organisation

5 Phases

1. Registration
2. Validation
3. Vote
4. Counting
5. Verification



Security Requirements



Fairness

Eligibility

Universal Verifiability

Individual Verifiability

Secure e-voting protocol

Correctness

Privacy

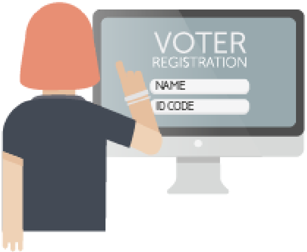
Coercion-Resistance

Robustness

Receipt-Freeness

Eligibility

Only the registered voters can vote

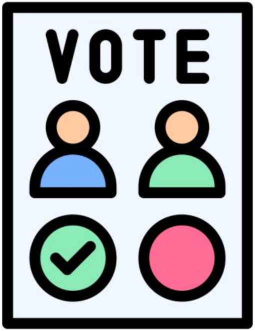


Prevent double voting

Robustness

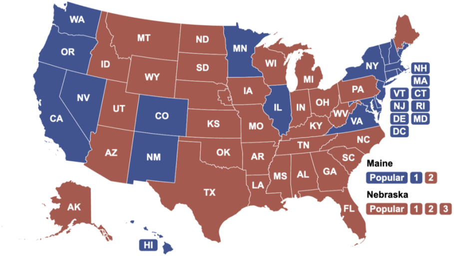
Tolerate a certain number of misbehaving voters

Correctness



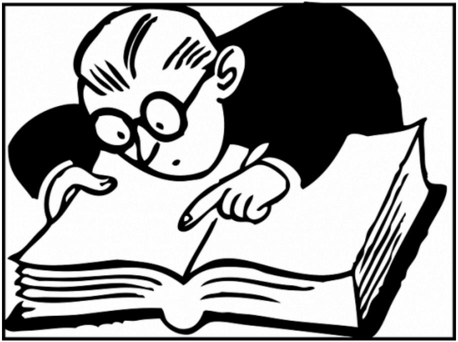
Results should be correct

Fairness



No preliminary results

Individual Verifiability



Each voter can check whether his vote was counted correctly

Universal Verifiability

Anybody can verify that the announced result corresponds to the sum of all votes

Anonymity

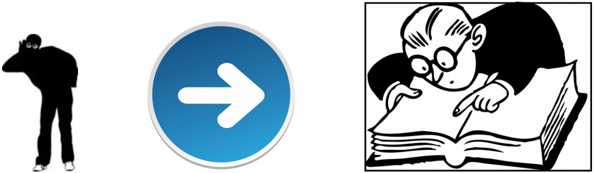
Privacy: unlinkability between the voter and his vote

Receipt-Freeness: A voter cannot construct a receipt

Corecion-Resistance: A coercer cannot be sure the voter followed his instructions

Privacy implies Individual Verifiability

2018 Cortier et al.



A system without Individual Verifiability cannot acheive privacy !

Dispute Resolution in Voting

In 2020, by David Basin, Sasa Radomirovic, Lara Schmid

Reduction Results: How many agents ?

- Security properties: **two** agents are sufficient.
  2004 by Hubert Comon-Lundh, Véronique Cortier
- When Are **Three Voters** Enough for Privacy Properties?
  2016 by Myrto Arapinis, Véronique Cortier, Steve Kremer

Outline

- Motivations
- Formal Methods
- e-voting
- Hierarchy of Privacy Notions
- Some Attacks
    - Sicilian
    - Vote Copy
    - Bulletin Board
    - Cryptographic Flaw
    - Clash
    - Machine Bugs
- Blockchain and vote
- Conclusion

State of the Art

Several Definitions for Privacy for e-voting protocols:

[DKR09,DKR10,MN06,BHM08,KT09,KSR10,LJP10,SC11,...]

But

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

State of the Art

Several Definitions for Privacy for e-voting protocols:

[DKR09,DKR10,MN06,BHM08,KT09,KSR10,LJP10,SC11,...]

But

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

OUR GOAL

Propose fine-grain definitions
to compare security levels of protocols

4 Dimensions for Privacy [DLL'12a, DLL'11]

Modeling in Applied π-Calculus

1. Communication between the attacker and the targeted voter

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

4 Dimensions for Privacy [DLL'12a, DLL'11]

Modeling in Applied π-Calculus

1. Communication between the attacker and the targeted voter

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

2. Intruder is controlling another voter:

Outsider (O)

Insider (I)

4 Dimensions for Privacy [DLL'12a, DLL'11]

Modeling in Applied π-Calculus

1. Communication between the attacker and the targeted voter

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 Dimensions for Privacy [DLL'12a, DLL'11]

Modeling in Applied π-Calculus

1. Communication between the attacker and the targeted voter



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 among the notions

$$\begin{tikzcd}
& CR^{O,FA,AB} \arrow[drr] \arrow[ddl] \arrow[dddd] & & & & & & CR^{I,FA,AB} \arrow[llllll] \arrow[dll] \arrow[ddl] \arrow[dddd] \\
& & & CR^{O,FA,EB} \arrow[ddl] \arrow[dddd] & & CR^{I,FA,EB} \arrow[ll] \arrow[ddl] \arrow[dddd] & & \\
*CR^{O,PO,AB} \arrow[drr] \arrow[ddddd] & & & & & & CR^{I,PO,AB} \arrow[llllll] \arrow[dll] \arrow[ddddd] & \\
& & CR^{O,PO,EB} \arrow[ddd] & & CR^{I,PO,EB} \arrow[ll] \arrow[ddd] & & & \\
& RF^{O,FA,AB} \arrow[drr] \arrow[dddl] \arrow[dddd] & & & & & & RF^{I,FA,AB} \arrow[llllll] \arrow[dll] \arrow[dddl] \arrow[dddd] \\
& & & RF^{O,FA,EB} \arrow[dl] \arrow[dddd] & & RF^{I,FA,EB} \arrow[ll] \arrow[dl] \arrow[dddd] & & \\
& & RF^{O,PO,EB} \arrow[dddd] & & RF^{I,PO,EB} \arrow[ll] \arrow[dddd] & & & \\
*RF^{O,PO,AB} \arrow[urr] \arrow[dddd] & & & & & & RF^{I,PO,AB} \arrow[ull] \arrow[llllll] \arrow[dddd] & \\
& VP^{O,FA,AB} \arrow[drr] \arrow[dddl] & & & & & & VP^{I,FA,AB} \arrow[llllll] \arrow[dll] \arrow[dddl] \\
& & & VP^{O,FA,EB} \arrow[dl] & & VP^{I,FA,EB} \arrow[ll] \arrow[dl] & & \\
& & VP^{O,PO,EB} & & VP^{I,PO,EB} \arrow[ll] & & & \\
*VP^{O,PO,AB} \arrow[urr] & & & & & & VP^{I,PO,AB} \arrow[ull] \arrow[llllll] &
\end{tikzcd}$$

Outline

- Motivations
- Formal Methods
- e-voting
- Hierarchy of Privacy Notions
- Some Attacks
    - Sicilian
    - Vote Copy
    - Bulletin Board
    - Cryptographic Flaw
    - Clash
    - Machine Bugs
- Blockchain and vote
- Conclusion

Sicilian Attack

Arlette
François
Emanuel
Marine
Jean-Luc
Arnaud
Ségolène
Jacques
Georges
Charles
Jean-Marie
Valérie

With 12 candidates, $>$ 479 millions possible combinations!

$> 2,000,000$ votes have been cast

helios
Trust the vote.

`https://vote.heliosvoting.org/`

Helios code is Open Source
Based on scientific papers
Use mixnet

Attacking and fixing Helios

By V. Cortier et al in 2010

Replaying a voter's ballot

- Alice votes A
- Bob votes B
- Charlie votes like Alice

This attack works on other protocols like Lee et al and Sako et al.

Belenios

`https://www.belenios.org/`

Belenios code is Open Source

Bulletin Board

- Fifty Shades of Ballot Privacy: Privacy against a Malicious Board, by Véronique Cortier, Joseph Lallemand, Bogdan Warinschi in 2020
- Fixing the Achilles Heel of E-Voting: The Bulletin Board by, Lucca Hirshi, Lara Schmid, David Basin in 2021

Russian Online Election

In 2019, Breaking the encryption scheme of the Moscow Internet voting system by P. Gaudry et al

- Elgamal key sizes are too small (CADO-NFS)
- Counting the number of votes cast for a candidate.

1994 Benaloh's Scheme

$$enc(a, pk_S) * enc(b, pk_S) = enc(a + b, pk_S)$$

Partial homomorphic are widely used in voting schemes

$$\prod enc(v_i, pk_S) = enc(\sum v_i, pk_S)$$

Original Benaloh's scheme is ambiguous

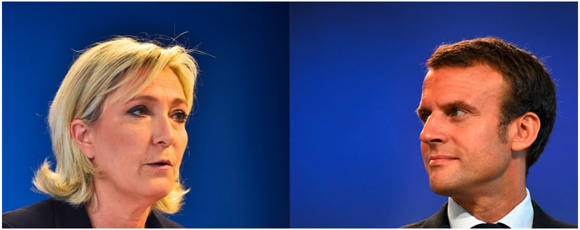
$$dec(enc(14, pk_S), sk_S) = 14 \mod 15 \text{ or } 14 \mod 5 = 4$$

Revisited Benaloh's encryption [FLA'11]

- Drawing false parameters: 33%
- Proposition of corrected version
- Proof using Kristian Gjosteen result.

Impact

Example with 15 voters



$\{0\}_{pk_S}$ $\{1\}_{pk_S}$

- $\prod enc(v_i, pk_S) = enc(\sum v_i, pk_S) = enc(14, pk_S)$
- Result can be either 14 or 4

Clash Attack on the verifiability of e-voting systems

By 2012 Kuesters et al.

YOU VOTED ON
May 19, 2015

YOUR I.D. NO. IS
002577744 D

→ VOTER ←

THIS IS YOUR RECEIPT

YOU VOTED ON
May 19, 2015

YOUR I.D. NO. IS
002577744 D

→ VOTER ←

THIS IS YOUR RECEIPT

Different voters with the same receipt

⇒ Authorities can manipulate the election without being detected

Attacks

- In 2007, Security Analysis of the Diebold AccuVote-TS Voting Machine by A. Feldman et al.
- In 2012, Attacking the Washington, D.C. Internet Voting System, by Scott Wolchok et al.
- In 2017 Voting Machine Hacking Village by Matt Blaze et al.

  - AVS WinVote DRE
  - Premier AccuVote TSx DRE
  - ES&S iVotronic DRE
  - PEB version 1.7c-PEB-S
  - Sequoia AVC Edge DRE
  - Diebold Express Poll 5000 electronic pollbook

With limited resources and information, they can be hacked.

Outline

Hyperledger Fabric

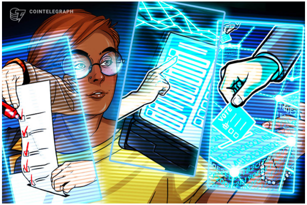
Ledger

- Public
- Infalsifiable
- Distributed

⇒ Verfiability !

DABSTERS

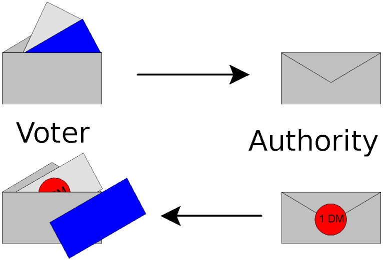
Distributed **A**uthorities using **B**lind **S**ignature
To **E**ffect **R**obust **S**ecurity in e-voting



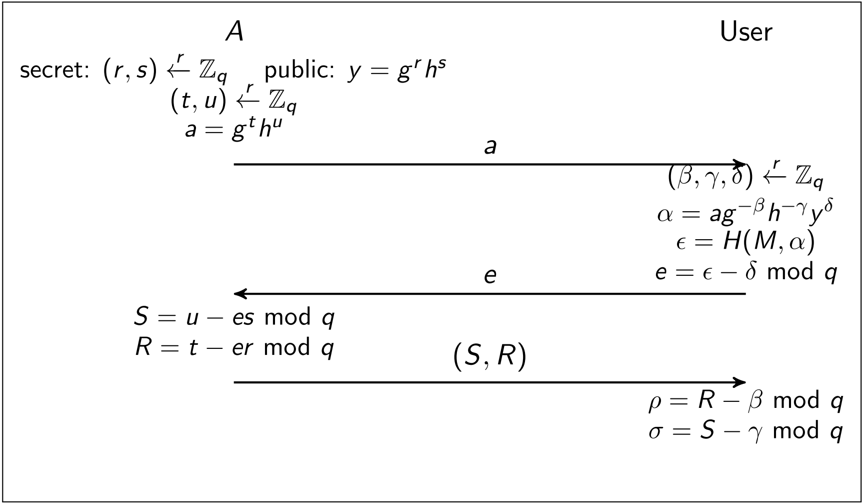
Ingredients

- BlindCons : BFT consensus + Blind Signtaure
- Shamir Secret Sharing
- Identity Based Encryption
- Eliptic Curve $P = k.Q$
- Pairing $e(aP, bQ) = e(P, Q)^{ab}$
- Hash Function

Okamoto-Schnorr Blind Signature



Okamoto-Schnorr Blind Signature



A

User

secret: $(r, s) \overset{r}{\leftarrow} \mathbb{Z}_q$ public: $y = g^r h^s$

$(t, u) \overset{r}{\leftarrow} \mathbb{Z}_q$

$a = g^t h^u$

$\xrightarrow{\quad a \quad}$

$(\beta, \gamma, \delta) \overset{r}{\leftarrow} \mathbb{Z}_q$

$\alpha = a g^{-\beta} h^{-\gamma} y^{\delta}$

$\epsilon = H(M, \alpha)$

$e = \epsilon - \delta \bmod q$

$\xleftarrow{\quad e \quad}$

$S = u - es \bmod q$

$R = t - er \bmod q$

$\xrightarrow{\quad (S, R) \quad}$

$\rho = R - \beta \bmod q$

$\sigma = S - \gamma \bmod q$

Participants

Registration Authorities

Voters

Counting Authorities

Ballot Structure

Counting authorithies shift with $offset = H(g)$ candidate names

Ballot Number BN			
Pseudo ID "C_j"	**Candidate Name** "nom_j"	**Choice**	**Conter-values** "$CV_{BN,nom_j,k}$"
0	Paul	☐	$CV_{BN,nom_0,0}$
1	Nico	☐	$CV_{BN,nom_1,1}$
2	Joel	☐	$CV_{BN,nom_2,2}$

$BN = \{g, D\}_{PK_A}$, g a generator and D random
$Q_{BN} = H(BN)$
S_k secret key of the Authority
$Q_{name_j} = H(name_j)$
$CV_{BN,namej,k} = e(Q_{name_j}, S_k \cdot Q_{BN})$

Phase 1: Registration

Register
to VOTE

$$Credential_V = S_M \cdot H(ID_V)$$

S_M a shared key between authorities

Phase 2: Validation

- Setup the blockchain
- Publish the list of voters signed by the authorities

Phase 3: Vote

Counting Authorities post on the blockchain encrypted ballots

Voter decrypts his own ballot

Ballot Number BN			
Pseudo ID "C_j"	**Candidate Name** "nom_j"	**Choice**	**Conter-values** "$CV_{BN,nom_j,k}$"
0	Paul	☐	$CV_{BN,nom_0,0}$
1	Nico	☐	$CV_{BN,nom_1,1}$
2	Joel	☐	$CV_{BN,nom_2,2}$

$Q_{BN} = H(BN)$

S_k secret key of the Authority

$Q_{name_j} = H(name_j)$

$CV_{BN,namej,k} = e(Q_{name_j}, S_k \cdot Q_{BN})$

Voter computes $Q_{C_j} = H(C_j)$ and with IBE $EncVote = \{BN\}_{Q_{C_j}}$

Uses $Credential_V$ to have this vote blindly signed

Publish his vote blindly signed

Phase 4: Counting

For each C_j candidate an authority decrypt the ballot to obtain BN

Find the corresponding offset and reconstruct the original bulletin

Count the voices for each candidate

Then write the final result

Publish also Counter-Values on the Blockchain

$$CV_{BN,namej,k} = e(Q_{name_j}, S_k \cdot Q_{BN})$$

and

$$\sigma_{k,name_j} = \sum_{i=1}^{l_j} S_k \cdot Q_{BN_i}$$

Phase 5: Verification

$$\begin{aligned}
\prod_{i=1}^{l} CV_{BN_i} &= \prod_{k=1}^{m}\prod_{j=1}^{m}\prod_{i=1}^{l_j} CV_{BN_{i,name_j},k} \\
&= \prod_{k=1}^{m}\prod_{j=1}^{m}\prod_{i=1}^{l_j} e(Q_{name_j}, S_k \cdot Q_{BN_i}) \\
&= \prod_{k=1}^{m}\prod_{j=1}^{m} e(Q_{name_j}, \sum_{i=1}^{l_j} S_k \cdot Q_{BN_i}) \\
&= \prod_{k=1}^{m}\prod_{j=1}^{m} e(Q_{name_j}, \sigma_{k,name_j})
\end{aligned}$$

Summary

DABSTERS in e-voting	
Eligibility	✓
Fairness	✓
Robustnsse	✓
Integrity	✓
Individual Verifiability	✓
Universal Verifiability	✓
Anonymity	✓
Receipt-Freeness	✓
Coercion Resistance	✗
Vote and Go	✓
Vote choice	Multiple

Formal Verification of DABSTERS

Properties	**Results**	**Time**
Vote Secrecy	✓	0.012 s
Authentification	✓	0.010 s
Vote Privacy	✓	0.024 s

Using Proverif

Outline

Motivations

Formal Methods

e-voting

Hierarchy of Privacy Notions

Some Attacks
- Sicilian
- Vote Copy
- Bulletin Board
- Cryptographic Flaw
- Clash
- Machine Bugs

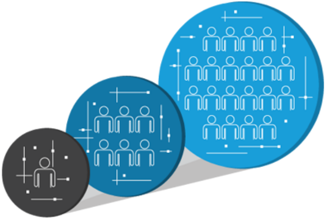
Blockchain and vote

Conclusion

Summary

- Voting is important for democracy
- Protocols must be open
- Design of voting protocols is not easy
- Formal Verification can help
- Proving all properties togheter is difficult

Future Work



- Scalability
- Human aspect are not yet taken into account
- End-to-end verification
- All properties in on tool !

Thank you for your attention.

Questions ?