Comparison of Cryptographic Verification Tools Dealing with Algebraic Properties

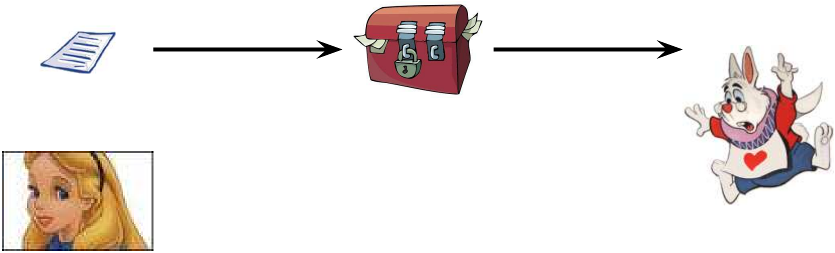
Pascal LAFOURCADE, Vanessa Terrade & Sylvain Vigier

Université Joseph Fourier, VERIMAG

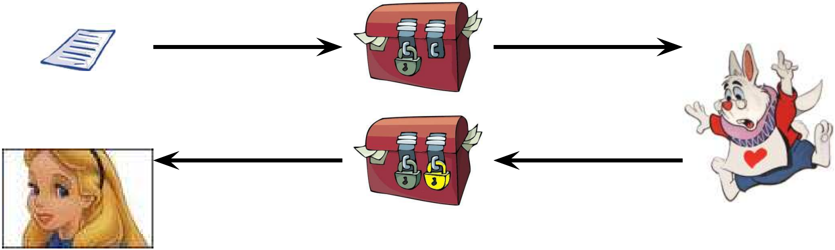
6th September 2009 Eindhoven

Workshop on Formal Aspects in Security and Trust

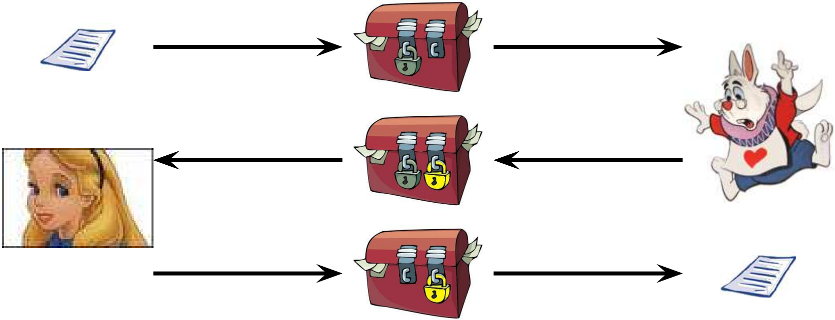
Basic Example :



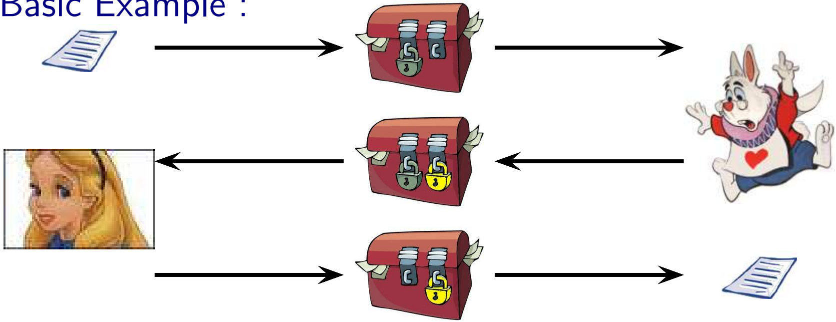
Basic Example :



Basic Example :



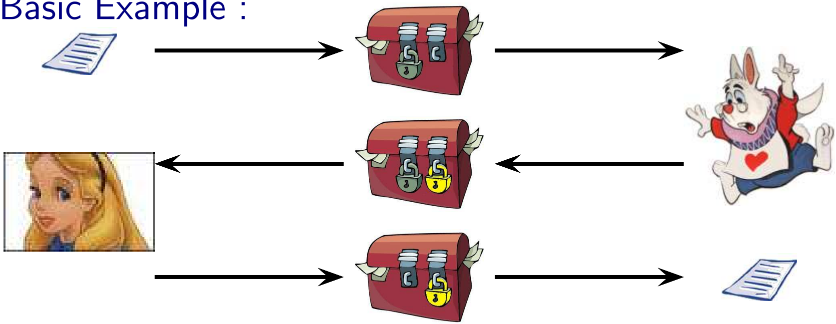
Basic Example :



Shamir 3-Pass Protocol

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

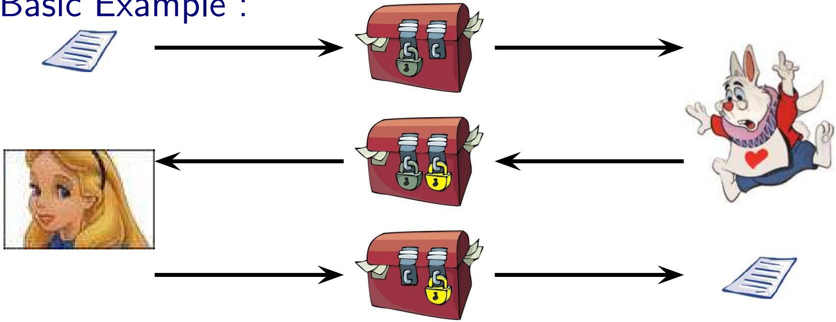
Basic Example :



Shamir 3-Pass Protocol

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

Basic Example :

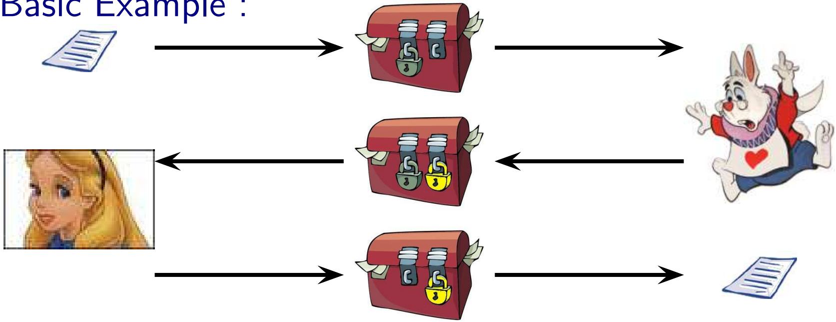


Shamir 3-Pass Protocol

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

Commutative Encryption

Basic Example :



Shamir 3-Pass Protocol

1	A	$\rightarrow$	B	:	$\{m\}_{K_A}$	Commutative
2	B	$\rightarrow$	A	:	$\{\{m\}_{K_A}\}_{K_B} = \{\{m\}_{K_B}\}_{K_A}$	Encryption
3	A	$\rightarrow$	B	:	$\{m\}_{K_B}$	

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

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

Passive attacker :

$m \oplus K_A \quad m \oplus K_B \oplus K_A \quad m \oplus K_B$

Logical Attack on Shamir 3-Pass Protocol (II)

Perfect encryption one-time pad (Vernam Encryption)

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

Shamir 3-Pass Protocol

1	A	$\rightarrow$	B :	$m \oplus K_A$
2	B	$\rightarrow$	A :	$(m \oplus K_A) \oplus K_B$
3	A	$\rightarrow$	B :	$m \oplus K_B$

Passive attacker :

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

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.

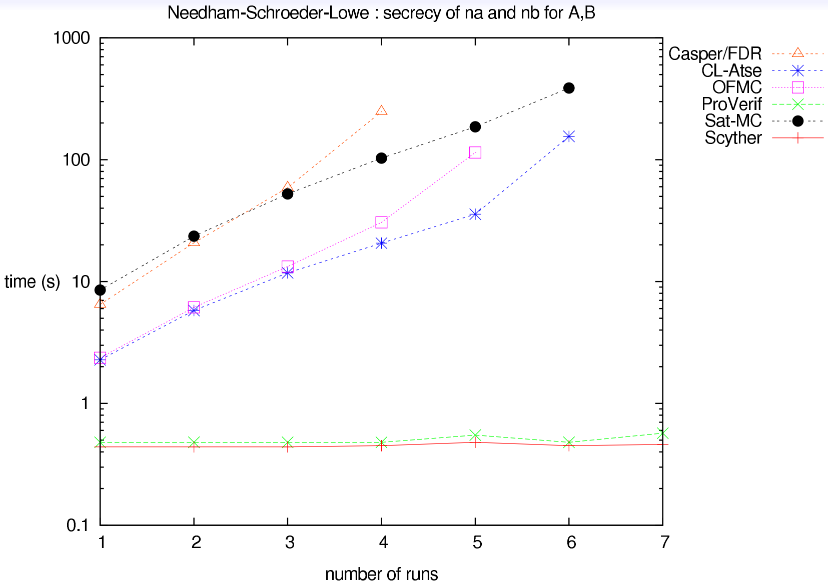
- completely automated,
- robust,
- expressive,
- and easily usable.

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

Comparison of Tools Dealing with Algebraic Properties ?

State of the art

- **Compariosn of NRL qnd Casper**.
  C. Meadows "Analyzing the needham-schroeder public-key protocol: A comparison of two approaches". In ESORICS 96
- **Time performence comparison of AVISPA Tools**
  L. Vigano "Automated Security Protocol Analysis With the AVISPA Tool" ENTCS 2006.
- **Usability comparison between AVISPA and HERMES**
  M. Hussain and D. Seret "A Comparative study of Security Protocols Validation Tools: HERMES vs. AVISPA". ICACT'06.
- **Comparison on the ability to find some attacks.**
  M. Cheminod, I. C. Bertolotti, L. Durante, R. Sisto, and A. Valenzano. "Experimental comparison of automatic tools for the formal analysis of cryptographic protocols". DepCoSRELCOMEX 2007.
- **Time efficiency comparison of: AVISPA, Proverif, Scyther, Casper/FDR**
  Comparing State Spaces in Automatic Security Protocol Verification" C. Cremers and P. Lafourcade. (AVoCS'07)


Needham-Schroeder-Lowe : secrecy of na and nb for A,B
time (s)
1000
100
10
1
0.1
1
2
3
4
5
6
7
number of runs
Casper/FDR
CL-Atse
OFMC
ProVerif
Sat-MC
Scyther

Outline

Tools

Protocol
- using Exclusive-Or
- using Diffie-Hellman

Conclusion & Perspective

Outline

Tools Dealing with Exclusive-Or and Diffie-Hellman

- **Avispa**:
  - OFMC: On-the-fly Model-Checker employs several symbolic techniques to explore the state space in a demand-driven way.
  - CL-Atse: Constraint-Logic-based Attack Searcher applies constraint solving with simplification heuristics and redundancy elimination techniques.
- **Proverif**: Analyses unbounded number of session using over-approximation with Horn Clauses.
  - XOR-ProVerif and DH-ProVerif: are two tools developed by Kuesters et al for analyzing cryptographic protocols with Exclusive-Or and Diffie-Hellman properties, using ProVerif

PC DELL E4500 Intel dual Core 2.2 Ghz with 2 GB of RAM.

Outline

Notations:

- $A, B, S \ldots$: principals
- messages M_i: messages
- N_A,N_B: nonces
- PK_A, PK_B: public keys
- K_{AB}: symmetric keys
- a prime number by P,
- a primitive root by G.
- Exclusive-Or is denoted by $A \oplus B$
- the exponentiation of G by the nonce N_A is denoted by G^{N_A}.

We use protocols from " Survey of Algebraic Properties Used in Cryptographic Protocols", V. Cortier, S. Delaune and P. Lafourcade.

Wired Equivalent Privacy Protocol: WEP

A, B: principals
X: any principal (B or the intruder)
M_1,M_2: messages
K_{AB}: symmetric key
$RC4$: function modeling the RC4 algorithm (message,symmetric key $\rightarrow$ message)
v: initial vector used with RC4 (a constant)
C: intregrity checksum (message $\rightarrow$ message)

0. A $\longrightarrow$ X : $v, ([M_1, C(M_1)] \oplus RC4(v, K_{AX}))$
1. A $\longrightarrow$ B : $v, ([M_2, C(M_2)] \oplus RC4(v, K_{AB}))$

WEP

Survey attack

- OFMC 0.01 s
- CL-Atse less than 0.01 s
- XOR-ProVerif less than 1 s

Same time for corrected version.

M. Tatebayashi, N. Matsuzaki, and D.B Newman (1989)

A, B, S : principals
K_A,K_B: fresh symmetric keys
PK_S: public key of the server

1. A $\longrightarrow$ S : $B, \{K_A\}_{PK_S}$
2. S $\longrightarrow$ B : A
3. B $\longrightarrow$ S : $A, \{K_B\}_{PK_S}$
4. S $\longrightarrow$ A : $B, K_B \oplus K_A$

TMN

UNSAFE, new attack

1. $A \longrightarrow S: \quad B, \{K_A\}_{PK_S}$
2. $S \longrightarrow I: \quad A$
3. $I(B) \longrightarrow S: \quad A, \{K_I\}_{PK_S}$
4. $S \longrightarrow I: \quad B, K_I \oplus K_A$

Hence I deduces K_A,
but not the survey attack based on
$\{X\}_{PK_S} * \{Y\}_{PK_S} = \{X * Y\}_{PK_S}$.

- OFMC less one second
- CL-Atse less one second
- XOR-ProVerif: less one second

H-T Liaw, W-S Juang and C-K Lin

A : the auctioneer
B : the bidder
T : the third party
K : the bank
d : the auctioneer's public key
t : the third party's public key
e : the bank's public key
c : the bidder's public key
$1/pk$: the corresponding private key to the public key pk.
B_{info} :bidder's information.
r : bidder's random number.
w, x, y, z : third party's random number.
B_{id} : bidder's specific number.

H-T Liaw, W-S Juang and C-K Lin

1. A $\longrightarrow$ everybody :
$\{Auction's\ product\ information, list\ of\ recognized\ third\ parties\}^{1/d}[M_1]$
2. B $\longrightarrow$ T : $\{B_{info}, c, r, Auction\ product\ information\}^t$
3. T $\longrightarrow$ Web : $M_1, H(r), H(w), H(x), H(y), H(z)$
4. T $\longrightarrow$ B : $\{Auction's\ product\ information, r, B_{id}\}^c$
5. T $\longrightarrow$ K : $\{M_1, B_{id}, payment, deposit, y\}^e$
6. K $\longrightarrow$ B : $\{M_1, B_{id}, deposit\ deducting\ certification, y\}^c$
7. B $\longrightarrow$ T :
$\{M_1, B_{id}, deposit\ deducting\ certification, price, y, r\}^f$
8. T $\longrightarrow$ B : $\{M_1, B_{id}, order, price, r\}^c$
9. T $\longrightarrow$ A : $\{M_1, order, maximum\ price\ offered, z\}^d$
10. A $\longrightarrow$ Web :
$\{Auction's\ product\ information, selling\ price, order\}^{1/d}[M_2], H(M_2, order,$
11. T $\longrightarrow$ K : $\{M_2, B_{id}, price, x, z \oplus w, paid\}^e$
12. K $\longrightarrow$ A : $\{M_2, B_{id}, price, z \oplus w, paid\}^d$
13. A $\longrightarrow$ B : $\{M_2, B_{id}, price, paid, product\}^d$

E-auction

SAFE

- OFMC less than 1 s
- CL-Atse less than 1 s
- XOR-ProVerif less than 1 s

J. Bull (1997)

X_A: $h([A, B, N_A], K_{AS}), [A, B, N_A]$
X_B: $h([B, C, N_B, X_A], K_{BS}), [B, C, N_B, X_A]$
X_C: $h([C, S, N_C, X_B], K_{CS}), [C, S, N_C, X_B]$
1. A $\longrightarrow$ B : X_A
2. B $\longrightarrow$ C : X_B
3. C $\longrightarrow$ S : X_C
4. S $\longrightarrow$ C :
$A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}}, B, A, K_{AB} \oplus h(N_B, K_{BS}), \{B, A, N_B\}_{K_{AB}}, B, C, K_{BC} \oplus h(N_B, K_{BS}), \{B, C, N_B\}_{K_{BC}}, C, B, K_{BC} \oplus h(N_C, K_{CS}), \{C, B, N_C\}_{K_{BC}}$
5. C $\longrightarrow$ B :
$A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}}, B, A, K_{AB} \oplus h(N_B, K_{BS}), \{B, A, N_B\}_{K_{AB}}, B, C, K_{BC} \oplus h(N_B, K_{BS}), \{B, C, N_B\}_{K_{BC}}$
6. B $\longrightarrow$ A : $A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}}$

Result on Bull

Survey attack found

- OFMC 0,08 s
- CL-Atse 0,08 s
- XOR-ProVerif CRASH

Analysis

- XOR-ProVerif crashes after more that one hour and 400 MB. Why?

Result on Bull

Survey attack found

- OFMC 0,08 s
- CL-Atse 0,08 s
- XOR-ProVerif CRASH

Analysis

- XOR-ProVerif crashes after more that one hour and 400 MB. Why?
  Due to the exponential algorithm proposed by Kuesters in the number of variables used in Exclusive-Or and the number of constants used in the protocol.
- New version: Attack found in $5 + 12 = 17$ seconds.

Corrected Version of Bull

- OFMC Does not end after 20h
- CL-Atse 1h10 s
- XOR-ProVerif CRASH

OFMC is slower than CL-Atse.

Salary Sum

A, B, C, D : principals
PK_A, PK_B, PK_C, PK_D : public keys
N_A : nonce
S_A, S_B, S_C, S_D: numbers (salaries)

1. A $\longrightarrow$ B : $A, \{N_A + S_A\}_{PK_B}$
2. B $\longrightarrow$ C : $B, \{N_A + S_A + S_B\}_{PK_C}$
3. C $\longrightarrow$ D : $C, \{N_A + S_A + S_B + S_C\}_{PK_D}$
4. D $\longrightarrow$ A : $D, \{N_A + S_A + S_B + S_C + S_D\}_{PK_A}$
5. A $\longrightarrow$ B,C,D : $S_A + S_B + S_C + S_D$

Salary Sum

UNSAFE, new attack

1. $\mathsf{A} \longrightarrow \mathsf{B}: \quad A, \{N_A \oplus S_A\}_{PK_B}$
2. $\mathsf{B} \longrightarrow \mathsf{I}: \quad B, \{N_A \oplus S_A \oplus S_B\}_{PK_I}$
3. $\mathsf{I(B)} \longrightarrow \mathsf{C}: \quad B, \{N_A \oplus S_A \oplus S_B\}_{PK_C}$
4. $\mathsf{C} \longrightarrow \mathsf{I}: \quad C, \{N_A \oplus S_A \oplus S_B \oplus S_C\}_{PK_I}$

Hence I deduces S_C

- OFMC 0,45 s
- CL-Atse 11 min 16 s
- XOR-ProVerif: ProVerif does not end after 6h
- new version : attack in 1 s + 11 s = 12 s

Gong's Mutual Authentication Protocol (1989)

A, B, S : principals
N_A, N_B, N_S : fresh numbers
P_A, P_B : Passwords
K : fresh symmetric key ($K = f_1(N_S, N_A, B, P_A)$)
H_A, H_B : message ($H_A = f_2(N_S, N_A, B, P_A)$ and
$H_B = f_3(N_S, N_A, B, P_A)$)
f_1, f_2, f_3, g : hash functions (message,message,message,message
$\longrightarrow$ message)
1. A $\longrightarrow$ B : A, B, N_A
2. B $\longrightarrow$ S : A, B, N_A, N_B
3. S $\longrightarrow$ B : $N_S, f_1(N_S, N_B, A, P_B) \oplus K, f_2(N_S, N_B, A, P_B) \oplus H_A, f_3(N_S, N_B, A, P_B) \oplus H_B, g(K, H_A, H_B, P_B)$
4. B $\longrightarrow$ A : N_S, H_B
5. A $\longrightarrow$ B : H_A

Gong

SAFE

- OFMC 19 s
- CL-Atse 1 min 34 s
- XOR-ProVerif Does not end
  (“out of global stack” for the conversion)

Exclusive-Or Summary

Tools / Protocols	Avispa		ProVerif
	OFMC	CL-Atse	XOR-ProVerif
Bull	UNSAFE Survey attack 0.08 s	UNSAFE Survey attack 0.08 s	No result XOR-ProVerif Does not end (3s + 5s)
Bull v2	The analysis Does not end time search: 20 h	SAFE 1 h 10 min	No result XOR-ProVerif Does not end (13s + 2min 4s)
WEP	UNSAFE Survey attack 0.01 s	UNSAFE Survey attack less than 0.01 s	UNSAFE Survey attack less than 1 s
WEP v2	SAFE 0.01 s	SAFE less than 0.01 s	SAFE less than 1 s
Gong	SAFE 19 s	SAFE 1 min 34 s	No result Does not end (Out of global stack)
Salary Sum	UNSAFE New attack 0.45 s	UNSAFE New attack 11 min 16 s	UNSAFE New attack Proverif Does not end
TMN	UNSAFE New attack 0.04 s	UNSAFE New attack less than 0.01 s	UNSAFE New attack less than 1 s
EAuction	SAFE less than 1s	SAFE 0.59 s	SAFE less than 1 s

W. Diffie and M. Hellman (1978)

A, B: principals
P:prime number
G:primitive root
N_A,N_B: nonces

1. A $\longrightarrow$ B : $P, G, (G^{N_A}) mod P$
2. B $\longrightarrow$ A : $(G^{N_B}) mod P$

Diffie Hellmann

UNSAFE

- OFMC less than 1 s
- CL-Atse less than 1 s
- XOR-ProVerif less than 1 s

M. Steiner, G. Tsudik, and M. Waidner (1996) IKA

A, B, C : principals
N_A, N_B, N_C : nonces
G : primitive root

1. A $\longrightarrow$ B : G, G^{N_A}
2. B $\longrightarrow$ C : $G^{N_B}, G^{N_A}, (G^{N_A})^{N_B}$
3. C $\longrightarrow$ A,B : $(G^{N_B})^{N_C}, (G^{N_A})^{N_C}$

IKA

UNSAFE

- OFMC less than 1 s
- CL-Atse less than 1 s
- XOR-ProVerif 3s + 1s = 4s

Diffie-Hellman Summary

Tools / Protocols	Avispa		ProVerif
	OFMC	CL-Atse	DH-ProVerif
D.H	UNSAFE Survey authentication attack 0.01 s	UNSAFE Survey authentication attack less than 0.01 s	UNSAFE Survey authentication attack less than 1 s
IKA	UNSAFE Survey authentication and secrecy attack less than 0.01 s	UNSAFE Survey authentication and secrecy attack less than 0.01 s	UNSAFE 1s+2min 33s SAFE 3s + 1s

Outline

Conclusion

- Usually same attacks with OFMC, CL-Atse, and XOR-ProVerif or DH-ProVerif.
- Attack most of the time identical to those of the survey (except for Salary Sum and TMN)

Conclusion for Exclusive-Or

- OFMC terminates it is globally faster that CL-Atse.
- But for protocols using a large number of Exclusive-Or operations, *e.g.* for instance in the Bull's protocol, OFMC does not terminates whereas CL-Atse does.
- the number of Exclusive-Or used in a protocol is the parameter which increases verification time.
- If the number of variables and constants is not too large ProVerif is very efficient and faster that Avispa tools.

Conclusion for Diffie-Hellman

All protocols were analyzed quickly by all the tools.
This confirms the polynomial complexity of DH-ProVerif and the fact that this equational theory is less complex than Exclusive-Or.

Conclusion

- Automatic verification is necessary.
- Tool are very helpful for design and verification.
- Use your favorite tool.
- Modeling of a protocol is quite tricky.
- Know the limitations of the tool and what you are checking.

Next

- Others Protocols
- Others properties
- Others Tools: Maude NPA, TA4SP, new OFMC (Open source Fixedpoint Model-Checker v.2009)

First Results

- New OFMC change only few seconds our results
- TA4SP is “slow” and often return “UNCONCLUSIVE”
- Maud is slower than all the other dedicated tools

Thank you for your attention

Questions ?