Comparison of Cryptographic Verification Tools Dealing with Algebraic Properties

Pascal Lafourcade, Vanessa Terrade, Sylvain Vigier

Comparison of Cryptographic Verification Tools Dealing with Algebraic Properties

Pascal Lafourcade, Vanessa Terrade, and Sylvain Vigier *

Université Grenoble 1, CNRS,Verimag, FRANCE
firstname.lastname@imag.fr

Abstract. Recently Kuesters et al proposed two new methods using ProVerif for analyzing cryptographic protocols with Exclusive-Or and Diffie-Hellman properties. Some tools, for instance CL-Atse and OFMC, are able to deal with Exclusive-Or and Diffie-Hellman. In this article we compare time efficiency of these tools verifying some protocols of the litterature that are designed with such algebraic properties.

1 Introduction

Use of cryptographic primitives for encoding secret data is not sufficient to ensure security. For example even with encrypted data, there exist security flaws: an intruder is able to discover information that should remain secret between two participants. As a consequence, an important activity of research on this topic leads to the development of different tools for verifying cryptographic protocols. One of the main properties required by cryptographic protocols is the *secrecy*, which means that secret data generated by an honest agent should not be learnt by an intruder. Another important property is the *authentication*, which means that every party can authenticate the party with whom they are executing the protocol.

Over the last decades many automatic tools based on formal analysis techniques have been presented for verifying cryptographic protocols [2, 8, 11, 14, 18, 29, 31, 33, 40, 20]. All these tools use the so-called Dolev-Yao intruder model. This modelling of the adversary offers a good level of abstraction and allows such tools to perform a formal analysis. All these works use the *perfect encryption hypothesis*, which means that the only way to decrypt a cipher text is to know the inverse key. This hypothesis abstracts the cryptography in order to detect a "logical flaw" due to all possible interleaving of different executions of the protocol. For relaxing this assumption, many works have been done for verifying protocols under some equational theories, like Exclusive-Or. In [16] we list protocols using

* This work was supported by ANR SeSur SCALP, SFINCS, AVOTE.

algebraic properties by conception and protocols that are safe without considering any algebraic property but that are flawed if we consider an algebraic property. All these examples are evidences that relaxing the perfect encryption hypothesis by considering equational theories is an important issue in security. In order to achieve this goal some tools have been developed for considering some algebraic properties [7, 8, 20, 25, 26, 44].

Contribution: We compare tools which are able to deal with algebraic properties for the verification of cryptographic protocols. More specifically we look at the Exclusive-Or property and the so-called Diffie-Hellman property, since these are the most frequently used. In order to verify cryptographic protocols using such properties we use CL-Atse and OFMC two tools of the platform Avispa. Both CL-Atse and OFMC analyze protocols with such algebraic properties for a bounded number of sessions. On the other hand ProVerif can verify cryptographic protocols for an unbounded number of session, but is not able to deal with such properties. But, recently Kuesters et al proposed in [25, 26] two methods for transforming a protocol description using Exclusive-Or and Diffie-Hellman property into a ProVerif input file. Hence we use these two "translators" in order to compare the following three tools: CL-Atse, OFMC and ProVerif. Our main contribution is to compare some protocols presented in [15] which are using Exclusive-Or and Diffie-Hellman. We also compare the tools using a more complex e-auction protocol [22]. We have chosen this protocol because it is longer than protocols in [15] and uses Exclusive-Or.

We check secrecy and authentication properties for most of the protocols under study. Moreover for the e-auction protocol, we analyze the non-repudiation property, meaning that a participant cannot claim that he never did something. The non-repudiation is a property that is often involved in e-commerce protocols, because the seller or the bank usually do not want a customer to be allowed to deny a transaction. For modelling the property of non-repudiation we follow the approach of L. Vigneron et al [24, 37] which expresses non-repudiation in term of authentication. Using this method we are able to check this property with the three tools in presence of algebraic properties.

State of the art: Some work exists on comparing the performance of different security protocol analysis tools. In [32] C. Meadows compares the approach used by NRL [31] and the one used by G. Lowe in FDR [35] on the Needham-Schroeder example [34]. The two tools are shown to be complementary even though NRL is considerably slower. In [4], a large set

of protocols is verified using the AVISS tool and timing results are given. In [45], a similar test is performed using the successor of AVISS, called the Avispa tool suite [2]. As the AVISS/Avispa tools consist of respectively three and four back-end tools, these tests effectively work as a comparison between the back end tools. No conclusions about the relative results are drawn in these articles. A qualitative comparison between Avispa and Hermes [11] is presented in [23]. This test leads to some generic advice for users of these two tools. It is not based on actual testing, but rather on conceptual differences between the modeling approaches of the tools. In [13], a number of protocol analysis tools are compared with respect to their ability to find a particular set of attacks.

Recently in [17], we proposed a fair comparison of the following tools: Casper/FDR, ProVerif, Scyther and Avispa. We obtain for the first time an "efficiency ranking" of such tools. In this work we only looked at some protocols without any algebraic property. Here we continue our investigations with algebraic properties.

Outline In the next Section we describe briefly the different tools used. In Section 3, for each analyzed protocol we present a short description and eventual existing or new attacks found by the tools. Finally in the last Section, we conclude by discussing our results obtained in Section 4.

2 Tools

In this section, we present the three tools used for our comparison. We have chosen these tools because we have already compared them in [17] on a set of protocols without algebraic properties and because they are dealing with the two main algebraic properties used in cryptographic protocols: Exclusive-Or and Diffie-Hellman.

Avispa [2] (V: 1.1) Automated Validation of Iternet Security Protocols and Applications consists of the following four tools:

- CL-Atse [43] developed by M. Turuani
- Sat-MC [3] created by A. Armando et al
- OFMC [6] designed by S. Mödersheim
- and TA4SP [10] proposed by Y. Boichut

All these tools take the same input language called HLPSL (High Level Protocol Specification Language). We describe a little bit more the two tools OFMC and CL-Atse which deal with selected algebraic properties. CL-Atse: (V: 2.2-5) Constraint-Logic-based Attack Searcher applies

constraint solving with simplification heuristics and redundancy elimination techniques [43]. OFMC (V: 2006/02/13) On-the-Fly Model-Checker employs symbolic techniques to perform protocol falsification as well as bounded analysis, by exploring the state space in a demand-driven way [6].

ProVerif [8, 9] (V: 1.16) developed by B. Blanchet analyzes an unbounded number of sessions by using over-approximation and represents protocols by Horn clauses. ProVerif accepts two kinds of input files: Horn clauses and a subset of the Pi-calculus. The tool uses an abstraction of fresh nonce generation, enabling it performs unbounded verification for a class of protocols. It can handle many different cryptographic primitives (shared and public-key cryptographic, hash functions...) and an unbounded number of sessions of the protocol. In ProVerif it is possible to model any equational theory, but the tool might not terminate. It is the case with Exclusive-Or or Diffie-Hellman exponentiation, however commutativity of the exponentiation alone is supported by ProVerif. It allows B. Blanchet's tool to verify protocols that only use this particular algebraic property of the exponentiation.

Recently R. Küster and T. Truderung proposed two new "tools" XOR-ProVerif [27] and DH-ProVerif [25]. These small programs transform a protocol using Exclusive-Or and Diffie-Hellman mechanism into a protocol in Horn clauses compatible with ProVerif. The input files for XOR-ProVerif and DH-ProVerif are Prolog files. This allows us to compare ProVerif with CL-Atse and OFMC for protocols using algebraic properties of Exclusive-Or and Diffie-Hellman.

3 Results

In this section, we present the results obtained by implementing the selected protocols in OFMC, CL-Atse and ProVerif. For our experiments we used a PC DELL E4500 Intel dual Core 2.2 Ghz with 2 GB of RAM. Due to obvious reasons all complete descriptions of protocols are not given here, but for each protocol we try to present the minimum in order to clearly explain the results given by the tools. In [28], we propose a short description and an implementation of each analyzed protocols.

We first present protocols dealing with Exclusive-Or and after the ones using Diffie-Hellman. All time measurements are given in Figure 1 and 2. In the ProVerif column of these tables we mention two numbers, the first one corresponds to the time of the transformation done by Kuesters et al's tool and the second one is the verification time given by

ProVerif. All our experiments are accessible at this address http://www-verimag.imag.fr/∼plafourc/FAST09.tar.gz

Notations: we denote principals by $A, B, S...$, messages by M_i, nonces generated by A, B by N_A,N_B, public keys by PK_A, $PK_B...$, symmetric keys between A and B by K_{AB}, fresh symmetric keys by K_A, a prime number by P, a primitive root by G. The Exclusive-Or is denoted by $A \oplus B$. The exponentiation of G by the nonce N_A modulo P is denoted by $G^{N_A} \bmod P$.

3.1 Bull's Authentication Protocol

This protocol [12, 36] aims at establishing fresh session keys between a fixed number of participants and a server. The protocol is recursive and uses one key for each pair of agents adjacent in the chain. In our modelling, the protocol is initiated by A and then goes through B and C before reaching S. At the end, new session keys K_{AB} and K_{BC} are established. Each key K_{XY} should be known exactly by X and Y (and also S), even if other participants are malicious.

Results: The checked property is the secrecy of K_{AB} between A and B and the secrecy of K_{BC} between B and C. We first notice that OFMC is slower than CL-Atse. The analysis using XOR-ProVerif crashes after more that one hour and the size of partial output produced is more that 400 MB. This corresponds to the fact that the algorithm proposed by Kuesters et al is exponential in the number of variables used in Exclusive-Or and the number of constants used in the protocol. This point demonstrates clearly the limit of the approach using XOR-ProVerif.

So we restrict the cases considered in the XOR-ProVerif file in order that the analysis end. It means that guided by the already known attack we fix some variables by the name of the principal according to the attack. This allows XOR-ProVerif to generate in 5 seconds the input file for ProVerif. In this setting ProVerif found the same attack as Avispa tools in $5 + 12 = 17$ seconds.

The attack found is the same as the one presented in the survey [16, 36]. I is part of the protocol since he plays the third role, and the different steps of the protocol take place normally. Yet, at the end of the protocol, I is able to retrieve the key shared by A and B. Indeed, he had with step 4 $K_{AB} \oplus h(N_B, K_{BS})$ and $K_{BI} \oplus h(N_B, K_{BS})$ with which he can compute $K_{AB} \oplus K_{BI}$. Moreover, he had also obtained K_{BI}, as it is the aim of the

protocol, and with $K_{AB} \oplus K_{BI}$, he can obviously retrieve K_{AB}, that should be shared only by A and B. We can see in this attack that the intruder uses the property of the Exclusive-Or: $X \oplus X = 0$. Indeed, it permits him to eliminate the term $h(N_B, K_{BS})$ in order to find $K_{AB} \oplus K_{BI}$. We propose a new version of the protocol, to prevent the third participant from using this property.

New Protocol: In this new version of the protocol, the idea was to introduce a second nonce created by B, which permits to avoid the dual use of a unique nonce on which lays on the attack. Here, S uses the first nonce N_{B1} to encrypt the key K_{AB} and the second nonce N_{B2} to encrypt K_{BC}. As a result, the attack consisting in computing two parts intended for B to find $K_{AB} \oplus K_{BI}$ is no more valid.

The analysis of this second version with the back end OFMC did not end after more that 20 hours of computation, while the analysis with CL-Atse gave no attack after 1h23. This result shows that small modification can drastically change the time of verification. We can also notice that in this case and as we have observed in the case without algebraic considerations, OFMC is slower than CL-Atse. Finally XOR-ProVerif also crashes with this new protocol, but if we fix again some variables in XOR-ProVerif the transformation ends in 17 seconds and the total analysis takes 2 minutes and 15 seconds.

3.2 Wired Equivalent Privacy Protocol

The Wired Equivalent Privacy protocol (WEP) [1], is used to protect data during wireless transmission. To encrypt a message M for X, A applies the operator $\oplus$ to $RC4(v, K_{AX})$ and $[M, C(M)]$, where $RC4$ is a public algorithm using v an initial vector and a symmetric key K_{AX}, and C is an integrity checksum function. For decrypting the received message, X computes $RC4(v, K_{AX})$ and after applying Exclusive-Or, he obtains $[M, C(M)]$ and can verify that the checksum is correct.

Results: The property verified was the secrecy of M_2 between A and B. All the tools found quickly the same following attack:

0.1. A $\longrightarrow$ B : $v, ([M_1, C(M_1)] \oplus RC4(v, K_{AB}))$
0.2. A $\longrightarrow$ I : $v, ([M_1, C(M_1)] \oplus RC4(v, K_{AI}))$
1. A $\longrightarrow$ I : $v, ([M_2, C(M_2)] \oplus RC4(v, K_{AB}))$

First of all, A sends the same message M_1 to B and I in steps 0.1 and 0.2. I is able to determine $RC4(v, K_{AI})$. Then, by computing $([M_1, C(M_1)] \oplus RC4(v, K_{AB})) \oplus ([M_1, C(M_1)] \oplus RC4(v, K_{AI})) \oplus RC4(v, K_{AI})$, I can reach $RC4(v, K_{AB})$ and consequently, he can access to every following messages intended for B. Indeed, in step 1, the intruder intercepts $([M_2, C(M_2)] \oplus RC4(v, K_{AB}))$ and he can compute $([M_2, C(M_2)] \oplus RC4(v, K_{AB})) \oplus RC4(v, K_{AB}))$, which is equal to $[M_2, C(M_2)]$.

We have implemented a new version of this protocol, based on the changing of the initial vector at every message sent. Like for the "Bull's Authentication Protocol", it has permitted to prevent the intruder from retrieving $RC4(v, K_{AB})$ with the Exclusive-Or property, and to ensure secrecy for the messages reserved to B. Avispa tools and ProVerif have indeed considered the new protocol safe in less than 1 second.

3.3 Gong's Mutual Authentication Protocol

The protocol [21] aims at providing mutual authentication and distributing a fresh secret key K. It makes use of a trusted server S with which each of the two agents A and B shares a secret password (P_A and P_B respectively). As an alternative to encryption algorithms, this protocol uses the one-way functions f_1,f_2,f_3 and g. The principal B can obtain, using the properties of Exclusive-Or, the triple (K, H_A, H_B) from the message that he receives at step 3, and check it by computing $g(K, H_A, H_B, P_B)$. Knowing P_A and after receiving N_S, A uses the functions f_1,f_2 and f_3 to get K, H_A and H_B. Hence, she can verify the message H_B sent by B at step 4 and sending the message H_A to B in order to prove her identity.

Results: Here, we checked the secrecy of the key created. It was declared safe by CL-Atse and OFMC. This time OFMC with 19 seconds is faster than CL-Atse with one minute and 34 seconds. For ProVerif, we have no result since converting the XOR-ProVerif file to horn clauses returns "out of global stack". Here also the number of variables used in Exclusive-Or is important comparing to the other protocols, which could explain the crash of the tool.

3.4 TMN

This is a symmetric key distribution protocol [30, 42]. Indeed, here, the key K_B is given to A. For each session, the server verifies that the keys K_A and K_B have not been used in previous sessions.

Results: The same attack which is described below was found with ProVerif and Avispa in less than one second. There exists another attack presented in the survey (or see [39]) based on a property of encryption. It used the fact that $\{X\}_{PK_S} * \{Y\}_{PK_S} = \{X * Y\}_{PK_S}$. Yet, the tools used does not seem to be able to find this attack, since they can not take into account such property.

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

In the first step, A starts a normal session with B. In the second step, I intercepts the message sent by S and then, in step 3, he impersonates B and sends his own symmetric key to the server. Finally, the intruder intercepts B and $K_I \oplus K_A$ and as he knows K_I, he can find K_A by computing $(K_I \oplus K_A) \oplus K_I$. Finally, I can transmit $B, K_I \oplus K_A$ to A.

3.5 Salary Sum

This simple protocol [38] allows a group of people to compute the sum of their salaries without anyone declaring his own salary to the others. For the sake of simplicity, the protocol is only considered for four principals A, B, C and D.

Results: We verified the secrecy of all salaries, and we found different attacks, depending on the tool used. This protocol uses addition, in order to make the comparison we replace addition by Exclusive-Or in the analyzed version.

The new attack found with Avispa tools is based on the fact that I plays both the role of C and D:

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

In this attack, B believes he is doing the protocol with I in the third position while C believes he is doing it with I in the fourth position. Indeed, in step 2, B sends $N_A \oplus S_A \oplus S_B$ encrypted with PK_I and in step 4, C sends $N_A \oplus S_A \oplus S_B \oplus S_C$ encrypted with PK_I too. Consequently, by subtracting these two numbers, I can obviously reach S_C, which should have remain secret.

Note that the first implementation ends with XOR-ProVerif but this times ProVerif does not terminate after more than six hours. Here we can see the limitations of ProVerif. It is well known that for some protocols ProVerif does not terminates. On the other side for the first time on this example we can clearly observe that OFMC is much more better than CL-Atse.

We also change a little bit the modeling by fixing some agent in the XOR-ProVerif input file. With this new version ProVerif terminates and finds an attack in less than $1+11 = 12$ seconds. The attack described below is very similar to the attack of the survey. S_A, S_B, S_C, S_D are numbers representing salaries.

1. A $\longrightarrow$ I : $A, \{N_A \oplus S_A\}_{PK_I}$
2. I(D) $\longrightarrow$ A : $D, \{N_A \oplus S_A\}_{PK_A}$
3. A $\longrightarrow$ I : S_A

Here, I is the second participant of the protocol. Indeed, A sends him $N_A \oplus S_A$ as expected. Then, contrary to the normal protocol, I impersonates D and sends to A directly what he has received: $N_A \oplus S_A$. A, believing it comes from D, applies again the Exclusive-Or with N_A, and consequently sends S_A to I, instead of the sum of all the salaries. In the attack presented in the survey, I adds $S_I \oplus S_I \oplus S_I$ to $N_A \oplus S_A$ before sending it to A. Then, A sends him $S_A \oplus S_I \oplus S_I \oplus S_I$ and since he knows S_I, he can retrieve S_A. The principles of the two attacks are the same, but the second version permits to prevent A from realizing it is his own salary he received at step 3.

3.6 E Auction

The H-T Liaw, W-S Juang and C-K Lin's protocol [22] is an improvement of the Subramanian's protocol. It satisfies the requirements for electronic auction like anonymity, privacy... but also adds the properties of non-repudiation, untraceability, auditability, one time registration and unlinkability.

Results: We check the secrecy, the authentication and the non repudiation with the three tools and all find it secure in less than 1 second. This protocol is composed of 13 exchanges of messages, but contains only two Exclusive-Or operations. This confirms the theory, meaning that the complexity for verifying protocol with Exclusive-Or increases exponentially with the number of Exclusive-Or operations.

3.7 Diffie-Hellman Key Exchange Protocol

In the protocol presented in [19], the initiator A first chooses a prime number P and a primitive root G of the group $\mathbb{Z}/P\mathbb{Z}$. He sends them with the exponentiation of G by a fresh number N_A and the responder does the same with a fresh number N_B. At the end, they share a common key which is the exponentiation of G by N_A and N_B.This protocol has to guarantee the secrecy of the fresh key.

Results: The implementation of the protocol realized is the simplified version of the one presented above. Indeed, in the first step of the protocol, A sends to B only G^{N_A}, and we consider that P and G were known by everybody. ProVerif and Avispa give us in less than one second the following authentication attack:
1. A $\longrightarrow$ I : $P, G, (G^{N_A})$ mod P
2. I(A) $\longrightarrow$ B : P, X_1, X_2
3. B $\longrightarrow$ I(A) : $(X_1^{N_B})$ mod P

Here, we can see that B (but also A) has no means to check authentication on the three numbers received. In the first step, I intercepts the numbers sent by A. Then, he can send his own numbers P, X_1, X_2 to B, and B, believing they come from A, sends her $(X_1^{N_B})$ mod P but I intercepts it. Yet, this attack is considered dangerous only if I has access to the key created during the protocol by B. To do so, I can send G for both X_1 and X_2. Indeed, the key created by B will be (G^{N_B}) mod P, that is to say the same number that he sends at step 3. Another way to obtain the key created by B is to send G for X_1 and (G^{N_I}) mod P for X_2. Here, the key of B will be $({G^{N_I}}^{N_B})$ mod P, and I can compute it since he knows (G^{N_B}) mod P, obtained in step 3. Note that the same attacks are applicable also for A.

3.8 IKA

The Initial Key Agreement [5] or also called the GDH.2 protocol, uses the same idea as the Diffie-Hellman protocol but it is extended to an unlimited number of agents. It aims at establishing a group key between a fixed number of participants X_1,...,X_n. Each agent has a secret nonce N_i and at the end of the protocol, the key shared between all principals is the exponentiation of G (primitive root of the group $\mathbb{Z}/P\mathbb{Z}$ where P is a prime number) by all the participant's nonces.

Results: We found an attack with CL-Atse and OFMC in less than 2s. This attack is similar to the attack on the Diffie-Hellman protocol. For DH-ProVerif this protocol was already studied by Kuesters et al in [25]. As it is mentioned in [25] a naive modelling of this protocol produces an input file for ProVerif, which does not terminate. The situation is similar as the situation we found in the Salary Sum with Exclusive-Or. The authors "used a technique inspired by the one sometimes used in the process calculus mode for ProVerif when encoding phases". Hence they proposed two versions: one safe version for one session (second line in the table of Figure 2) and one unsafe version for two sessions in order to find the well-known attack (first line in the table of Figure 2). We adapt the two versions given by the authors for 4 principals to 3 principals, in order to analyze the same configuration as the one used in the Avispa tools. We observe that ProVerif is a little bit slower that the other tools, due to the translation performed by XOR-ProVerif.

Name	Avispa		ProVerif
Properties studied	OFMC	CL-Atse	XOR-ProVerif
Bull [12]	UNSAFE Survey secrecy attack 0.08 s	UNSAFE Survey secrecy attack 0.08 s	No result XOR-ProVerif Does not end
Bull v2	The analysis Does not end time search: 20 h	SAFE 1 h 10 min	No result XOR-ProVerif Does not end
WEP [1]	UNSAFE Survey secrecy attack 0.01 s	UNSAFE Survey secrecy attack less than 0.01 s	UNSAFE Survey secrecy attack less than 1 s
WEP v2	SAFE 0.01 s	SAFE less than 0.01 s	SAFE less than 1 s
Gong [21]	SAFE 19 s	SAFE 1 min 34 s	No result Does not end
Salary Sum [38]	UNSAFE New secrecy attack 0.45 s	UNSAFE New secrecy attack 11 min 16 s	UNSAFE Survey secrecy attack Does not end
TMN [30, 42]	UNSAFE New secrecy attack 0.04 s	UNSAFE New secrecy attack less than 0.01 s	UNSAFE New secrecy attack less than 1 s
E-Auction [22]	SAFE less than 1s	SAFE 0.59 s	SAFE less than 1 s

Fig. 1. Results for protocols using XOR

Name	Avispa		ProVerif
Properties studied	OFMC	CL-Atse	DH-ProVerif
D.H [19]	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 [5]	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

Fig. 2. Results for protocols using DH

4 Conclusion and Discussion

In this paper we have compared time efficiency of cryptographic verification tools using Exclusive-Or and Diffie-Hellman properties. In Figure 1 and 2 we sum up the results obtained with the different tools for the protocols under study. Globally, we found the same attacks with OFMC, CL-Atse, and XOR-ProVerif or DH-ProVerif. Most of the time these attacks were identical to those of the survey [16] except for Salary Sum and TMN. These exceptions are normal since for the first one we change the addition into Exclusive-Or and for the second one any tool can deal with the homomorphism property used in the attack presented in the survey.

For the Exclusive-Or property, it seems that when 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 difference between the Bull's protocol and the E-auction's protocol shows clearly that the number of Exclusive-Or used in a protocol is the parameter which increases verification time. This confirms that complexity is exponential in the number of Exclusive-Or. This also explains the failure of XOR-ProVerif in this situation. On the other hand, if the number of variables and constants is not too large ProVerif is very efficient and faster that Avispa tools. Finally, for some protocols, such as modified version of the Salary Sum for ProVerif or the improved version of Bull's protocol for OFMC the tools were not able to end the analysis in a limited period of time.

For Diffie-Hellman property, 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.

Indeed, the use of variables with Exclusive-Or or "exponentiation" seems to increase rapidly the search time of the tools, especially for XOR-ProVerif, but also for OFMC and CL-Atse.

Future work: Recently a new version of OFMC has been proposed in the project AVANTSSAR. A new version of TA4SP is also announced on the website of the author, this new version deals with some algebraic properties including Exclusive-Or and Diffie-Hellman. In the future we plan to check the same protocols with this new version of OFMC and also to include the new version of TA4SP in our study. Moreover we would like to see if the new OFMC is more efficient than its older version. We also would like to include the tool Maude NPA [20] in our analysis This tool uses rewriting techniques for proving security of cryptographic protocols in presence of equational theories. Finally we project to continue this preliminary analysis in a fair way as we did in [17].

References

1. IEEE 802.11 Local and Metropolitan Area Networks: Wireless LAN Medium Acess Control (MAC) and Physical (PHY) Specifications, 1999.
2. A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P.-C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, Michael R., J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In *Proc. of CAV'2005*, LNCS 3576, pages 281–285. Springer, 2005.
3. A. Armando and L. Compagna. An optimized intruder model for SAT-based model-checking of security protocols. In A. Armando and L. Viganò, editors, *ENTCS*, volume 125, pages 91–108. Elsevier Science Publishers, March 2005.
4. Alessandro Armando, David A. Basin, Mehdi Bouallagui, Yannick Chevalier, Luca Compagna, Sebastian Mödersheim, Michaël Rusinowitch, Mathieu Turuani, Luca Viganò, and Laurent Vigneron. The aviss security protocol analysis tool. In *CAV '02: Proceedings of the 14th International Conference on Computer Aided Verification*, pages 349–353, London, UK, 2002. Springer-Verlag.
5. G. Ateniese, M. Steiner, and G. Tsudik. New multiparty authentication services and key agreement protocols. *IEEE Journal of Selected Areas in Communications*, 18(4):628–639, 2000.
6. D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In *Proc. of ESORICS'03*, volume 2808 of *LNCS*, pages 253–270. Springer-Verlag, 2003.
7. David A. Basin, Sebastian Mödersheim, and Luca Viganò. Ofmc: A symbolic model checker for security protocols. *Int. J. Inf. Sec.*, 4(3):181–208, 2005.
8. B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proc. CSFW'01*, pages 82–96. IEEE Comp. Soc. Press, 2001.
9. B. Blanchet. *Cryptographic Protocol Verifier User Manual*, 2004.
10. Y. Boichut, P.-C. Héam, O. Kouchnarenko, and F. Oehl. Improvements on the Genet and Klay technique to automatically verify security protocols. In *Proc. AVIS'04*, April 2004.

11. L. Bozga, Y. Lakhnech, and M. Perin. HERMES: An Automatic Tool for Verification of Secrecy in Security Protocols. In *Computer Aided Verification*, 2003.
12. J. Bull and D. J. Otway. The authentication protocol. Technical Report DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03, Defence Research Agency, 1997.
13. M. Cheminod, I. Cibrario Bertolotti, L. Durante, R. Sisto, and A. Valenzano. Experimental comparison of automatic tools for the formal analysis of cryptographic protocols. In *DepCoS-RELCOMEX 2007, June 14-16, 2007, Szklarska Poreba, Poland*, pages 153–160. IEEE Computer Society, 2007.
14. R. Corin and S. Etalle. An improved constraint-based system for the verification of security protocols. In *Proc. 9th International Static Analysis Symposium (SAS)*, volume 2477 of *LNCS*, pages 326–341. Springer, Sep 2002.
15. V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.
16. Veronique Cortier, Stephanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.
17. Cas J.F. Cremers, Pascal Lafourcade, and Philippe Nadeau. Comparing state spaces in automatic protocol analysis. In *Formal to Practical Security*, volume 5458/2009 of *Lecture Notes in Computer Science*, pages 70–94. Springer Berlin / Heidelberg, 2009.
18. C.J.F. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc.*, volume 5123/2008 of *Lecture Notes in Computer Science*, pages 414–418. Springer, 2008.
19. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Society*, 22(6):644–654, 1976.
20. Santiago Escobar, Catherine Meadows, and José Meseguer. Maude-npa: Cryptographic protocol analysis modulo equational properties. In Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri, editors, *FOSAD*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer, 2007.
21. L. Gong. Using one-way functions for authentication. *SIGCOMM Computer Communication*, 19(5):8–11, 1989.
22. LIAW Horng-Twu, JUANG Wen-Shenq, and LIN Chi-Kai. An electronic online bidding auction protocol with both security and efficiency. *Applied mathematics and computation*, 174:1487–1497, 2008.
23. M. Hussain and D. Seret. A comparative study of security protocols validation tools: HERMES vs. AVISPA. In *InProc. ICACT'06*, volume 1, pages 303– 308, 2006.
24. Francis Klay and Laurent Vigneron. Automatic methods for analyzing nonrepudiation protocols with an active intruder. In Pierpaolo Degano, Joshua D. Guttman, and Fabio Martinelli, editors, *Formal Aspects in Security and Trust*, volume 5491 of *Lecture Notes in Computer Science*, pages 192–209. Springer, 2008.
25. R. Küsters and T. Truderung. Using ProVerif to Analyze Protocols with Diffie-Hellman Exponentiation. In *Proceedings of the 22nd IEEE Computer Security Foundations Symposium (CSF 2009)*, pages 157–171. IEEE Computer Society, 2009.
26. Ralf Küsters and Tomasz Truderung. Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM Conference on Computer and Communications Security*, pages 129–138. ACM, 2008.

27. Ralf Küsters and Tomasz Truderung. Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. In *ACM Conference on Computer and Communications Security*, pages 129–138, 2008.
28. Pascal Lafourcade, Vanessa Terrade, and Sylvain Vigier. Comparison of cryptographic verification tools dealing with algebraic properties. Technical Report TR-2009-16, Verimag, October 2009.
29. G. Lowe. Casper: a compiler for the analysis of security protocols. *J. Comput. Secur.*, 6(1-2):53–84, 1998.
30. G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. *IEEE Transactions on Software Engineering*, 23(10):659–669, 1997.
31. C. Meadows. Language generation and verification in the NRL protocol analyzer. In *Proc. CSFW'96*, pages 48–62. IEEE Comp. Soc. Press, 1996.
32. Catherine Meadows. Analyzing the needham-schroeder public-key protocol: A comparison of two approaches. In Elisa Bertino, Helmut Kurth, Giancarlo Martella, and Emilio Montolivo, editors, *ESORICS*, volume 1146 of *Lecture Notes in Computer Science*, pages 351–364. Springer, 1996.
33. J.C. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using Murphi. In *IEEE Symposium on Security and Privacy*, May 1997.
34. R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communication of the ACM*, 21(12):993–999, 1978.
35. A. W. Roscoe. *Model-checking CSP*. Prentice Hall, 1994.
36. Peter Y. A. Ryan and Steve A. Schneider. An attack on a recursive authentication protocol. a cautionary tale. *Inf. Process. Lett.*, 65(1):7–10, 1998.
37. Judson Santos Santiago and Laurent Vigneron. Study for Automatically Analysing Non-repudiation. In *1er Colloque sur les Risques et la Sécurité d'Internet et des Systèmes- CRiSIS 2005*, pages 157–171, 10 2005.
38. B. Schneier. *Applied Cryptography*. Wiley, second edition, 1996.
39. G.J. Simmons. Cryptoanalysis and protocol failures. *Communications of the ACM*, 37(11):56–65, 1994.
40. D. Song, S. Berezin, and A. Perrig. Athena: A novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 9(1/2):47–74, 2001.
41. M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman key distribution extended to groups. In *Proc. 3rd ACM Conference on Computer and Communications in Security, (CCS'96)*, pages 31–37. ACM Press, 1996.
42. M. Tatebayashi, N. Matsuzaki, and D. B. Newman. Key distribution protocol for digital mobile communication systems. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)*, volume 435 of *LNCS*, pages 324–333, Santa Barbara (California, USA), 1989. Springer-Verlag.
43. M. Turuani. The CL-Atse protocol analyser. In *Proc. RTA'06*, LNCS, August 2006.
44. Mathieu Turuani. The CL-Atse Protocol Analyser. In Frank Pfenning, editor, *17th International Conference on Term Rewriting and Applications - RTA 2006 Lecture Notes in Computer Science*, volume 4098 of *Lecture Notes in Computer Science*, pages 277–286. Springer, 08 2006.
45. L. Viganò. Automated security protocol analysis with the AVISPA tool. *ENTCS*, 155:61–86, 2006.

Appendix

A Diffie-Hellman Key Exchange Protocol

Author(s) and Reference(s):
W. Diffie and M. Hellman (1978) [19]

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

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

B Bull's Authentication Protocol

Author(s) and Reference(s):
J. Bull (1997) [12]

Notations:
A, B, C, S: principals
K_{AB},K_{BC}: fresh symmetric keys
N_A,N_B,N_C : nonces
K_{AS},K_{AB},K_{AC}: symmetric keys
h: hash function (message, symmetric key $\rightarrow$ message)

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: $h([C, S, N_C, X_B], K_{CS}), [C, S, N_C, X_B]$

Protocol scheme:
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 : $A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}}$

B.1 The new version

Notations:

X_B: $h([B, C, N_{B1}, N_{B2}, X_A], K_{BS}), [B, C, N_{B1}, N_{B2}, X_A]$

Protocol scheme:

1. A $\longrightarrow$ B : X_A
2. B $\longrightarrow$ C : X_B
3. C $\longrightarrow$ S : X_C
4. S $\longrightarrow$ C : $A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}},$
$B, A, K_{AB} \oplus h(N_{B1}, K_{BS}), \{B, A, N_{B1}\}_{K_{AB}},$
$B, C, K_{BC} \oplus h(N_{B2}, K_{BS}), \{B, C, N_{B2}\}_{K_{BC}},$
$C, B, K_{BC} \oplus h(N_C, K_{CS}), \{C, B, N_C\}_{K_{BC}}$
5. C $\longrightarrow$ B : $A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}},$
$B, A, K_{AB} \oplus h(N_{B1}, K_{BS}), \{B, A, N_{B1}\}_{K_{AB}},$
$B, C, K_{BC} \oplus h(N_{B2}, K_{BS}), \{B, C, N_{B2}\}_{K_{BC}}$
6. B $\longrightarrow$ A : $A, B, K_{AB} \oplus h(N_A, K_{AS}), \{A, B, N_A\}_{K_{AB}}$

C Wired Equivalent Privacy Protocol

Author(s) and Reference(s):

[1]

Notations:

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 → message)

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

Note that the protocol consists normally only in the step 1 given above. But since the attack we would like to retrieve is based on an other message knew by the intruder, we have added the step 0.

D Gong's Mutual Authentication Protocol

Author(s) and Reference(s):
L. Gong (1989) [21]

Notations:
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)

Protocol scheme:
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

E Salary Sum

Author(s) and Reference(s):
B.Schneier [38]

Notations:
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)

Protocol scheme:
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 : $S_A + S_B + S_C + S_D$

F TMN

Author(s) and Reference(s):
M. Tatebayashi, N. Matsuzaki, and D.B Newman (1989) [30] [42]
Notations:
A, B, S : principals
K_A,K_B: fresh symmetric keys
PK_S: public key of the server

Protocol scheme:
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 : $B, K_B \oplus K_A$

G IKA

Author(s) and Reference(s):
M. Steiner, G. Tsudik, and M. Waidner (1996) [41, 5]

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

Protocol scheme:
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 : $(G^{N_B})^{N_C}, (G^{N_A})^{N_C}$

H E-auction

Author(s) and Reference(s):
H-T Liaw, W-S Juang and C-K Lin [22]

Notations:
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.

Protocol scheme:
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 : $\{Auction's\ product\ information, selling\ price, order\}^{1/d}[M_2]$,
$H(M_2, order, product)$

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 : $\{M_2, B_{id}, price, paid, product\}^d$