مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (39) العدد (3) 2017
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (39) No. (3) 2017

التحقق من البروتوكولات الأمنية باستخدام الطرق الرسمية

الدكتور محمد منير قلاش*
الدكتور عبد الكريم السالم**
الدكتور عبد الرزاق بدوية***

(تاريخ الإيداع 11 / 4 / 2017. قُبِل للنشر في 2/ 5 / 2017)

□ ملخّص □

توجد العديد من الطرق الرسمية المعتمدة Formal Methods لاختبار البروتوكولات الأمنية وكشف كونها آمنة أم لا. أهمها: أفيسبا Avispa، كاسبر Casper، بروفيرف ProVerif، سايثر Scyther. لقد تم التطرق سابقاً إلى تنفيذ مقارنات باستخدام طريقتين فقط من الطرق المذكورة (ProVerif, Scyther).

تم في هذا البحث التحقق من البروتوكولات الأمنية والقيام بتنفيذ مقارنة بين الطرق الأربعة المذكورة من حيث نفسها البارامترات التي استخدمت في تنفيذ المقارنة بين الطريقتين سابقاً: أسلوب العمل، لغة البرمجة المستخدمة، واجهة المستخدم، أسلوب الإدخال، وطريقة إظهار النتائج. وتقديم خيارات للمستخدم باختيار الطريقة المناسبة حسب البارامتر المطلوب.

تم تنفيذ الاختبار على ستة من البروتوكولات الأمنية المختلفة وهي: بروتوكول التحقق كاو شاو Kao Chow Authentication Protocol، بروتوكول 3-د الآمن 3-D Secure، بروتوكول ندهام-شرودر للمفتاح العمومي Needham-Schroeder Public Key Protocol، بروتوكول تبادل المفاتيح دفي-هلمان Diffie-Hellman key exchange، - بروتوكول اندرو سكيور Andrew Secure RPC Protocol، وبروتوكول مصادقة مصافحة التحدي Challenge Handshake Authentication Protocol.

الكلمات المفتاحية: الطرق الرسمية، الأمن، بروتوكول أمني، نمذجة البروتوكول، عمليات التفحص، أدوات التحليل.

* دكتور- كلية الهندسة-جامعة القلمون الخاصة
**أستاذ- كلية الهندسة الميكانيكية والكهربائية- جامعة البعث
***أستاذ- كلية الهندسة الميكانيكية والكهربائية-جامعة دمشق

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (39) العدد (3) 2017
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (39) No. (3) 2017

Verification of Security Protocols Using Formal Methods

Dr. Mohammad Muneer Kallash*
Dr. Abdulkarim Assalem**
Prof. Abdelrazak Badawieh***

(Received 11 / 4 / 2017. Accepted 2 / 5 / 2017)

□ ABSTRACT □

There are many of Formal Methods for testing security protocols detecting being safe or not. Including Avispa, Casper, ProVerif, Scyther. Previously a comparisons using two of mentioned methods (ProVerif, Scyther).

In this, research a comparison between the four mentioned methods in terms of the same used parameters in the previous comparison: working style, the modeling language, user interface, input, and output. As a result, the user provided with options to choose the appropriate method depending on the desired parameter.

Six different of security protocols have been tested and finally the results have been compared; these protocols are Kao Chow Authentication Protocol, 3-D Secure Protocol, Needham-Schroeder Public Key Protocol, Diffie–Hellman key exchange, Andrew Secure RPC Protocol, and Challenge Handshake Authentication Protocol

Keywords: Formal Methods, Security, Security Protocol, Protocol Modelling, Checking Process, Analysis Toolkit.

***Engineering Faculty- University of Kalamoon**
****Mechanical & Electrical- Faculty Al-Baath University**
*****Mechanical & Electrical Faculty- Damascus University**

مقدمة:

البروتوكول هو مجموعة من التعليمات والتي تتبع قواعد محددة لإنشاء اتصال ما، أما بروتوكول الأمن فهو بروتوكول اتصال عادي يضمن أن الرسالة المرسلة لا تتعرض للهجوم باستخدام آليات تشفير محددة ، وعلى الرغم من وجود آليات مختلفة للتشفير إلا أن البروتوكول قد لا يكون أمناً بالضرورة [7]، وبالتالي يشكل أمن المعلومات والخصوصية وسرية هوية المتعاملين في الشبكة هاجساً حقيقياً لدى الباحثين مما يدفعهم للعمل على ضمان الأنظمة المعلوماتية والتي ترتبط بخصائص عديدة هي: الخصوصية Confidentiality، المرجعية Integrity، والأصالة Authentication.

وفقاً ل يانغ ورفاقه Yang et al [26] عادة ما يتم اختبار البروتوكولات الأمنية والتحقق منها بإحدى طريقتين:

1- الأمن المبرهن provable security والتي يتم من خلالها اثبات مدى أمان البروتوكول من خلال اختبار المستوى الأعظم للسرية الممكن الوصول إليه [13].

2 - الطرق الرسمية formal methods: وهي نوع التقنيات المعتمد على الرياضيات حيث يتم نمذجة المواصفات للتطوير والتحقق من سرية وأمان البروتوكولات مما يدعم موثوقية ومتانة التصميم [19]، وبالتالي فإن الطرق الرسمية تسمح بإنشاء تقنيات مؤتمتة وأدوات تقوم بالتحقق من أن البروتوكولات المراد تنصيبها تحترم معايير الأمان[2].

في هذا البحث وفي الجزء الأول منه تم استعراض بعض الأدوات الرسمية Formal methods tools المستخدمة في اختبار البروتوكولات الأمنية وتحديد أربعة منها لتجريبها، ومن ثم يستعرض البروتوكولات الأمنية المختبرة مع نتائج الاختبارات عليها. في الجزء الثالث تم تحليل ومقارنة خصائص الأدوات الأربعة المستخدمة في الاختبارات، وختاماً استخلاص النتائج.

أهمية البحث وأهدافه:

هدف البحث:

يهدف هذا البحث إلى لدراسة وتحليل أداء عدد من أدوات تحليل البروتوكولات بالطرق الرسمية وتحديد مزايا كل منها من خلال إجراء اختبارات الأمان على عدد من البروتوكولات الأمنية المستخدمة، واقتراح أفضل أسلوب لاختبار البروتوكولات الأمنية.

أدوات التحليل للطرق الرسمية:

يوجد العديد من الطرق الرسمية المعتمدة المبنية على أدوات التحليل الرسمية تم انتقاء أربعة منها واختبارها في هذا البحث وهي:

1- كاسبر/اف دي آر Casper/FDR 2- أفيسبا AVISPA

3- بروفيرف ProVerif 4- سايزر Scyther

1- كاسبر/اف دي آر Casper/FDR: [1, 9, 16, 19, 20] يعد الكاسبر CASPER A Compiler for the Analysis of Security Protocols من أهم أدوات اختبار البروتوكولات الأمنية يعتمد على حسابات

التفاضل والتكامل، تم تطويره في جامعة أكسفورد Oxford، يستخدم لغة CSP لتوصيف البروتوكولات للتسهيل والتقليل من الأخطاء يتم توصيف البرتوكول بلغة ال FDR، إن عملية التحقق في CSP بسيطة

تبدأ عن طريق تحديد البروتوكول كمجموعة عمليات متفاعلة في CSP، بعد ذلك يتم تطوير نموذج للشبكة وتحديد المهاجم بالإضافة إلى صياغة خواصه عن طريق المعادلات وتحليل إمكانيات الوصول. بعد ذلك وباستخدام مدقق نموذج متكامل، يتم البحث عن الثغرات التي تؤدي إلى فشل البروتوكول.

2- أفيسبا AVISPA: [10, 19, 22, 24] إن الأفيسبا AVISPA (Automated Validation of Applications Internet Security Protocols and) مبادرة متعددة الأطراف، بهدف تطوير تقنية جديدة وأداة لتحليل بروتوكولات أمن المعلومات التي تدعم الجيل الجديد من تطبيقات الانترنت، ونتيجة لهذا المشروع، تم إنشاء أداة أفيسبا AVISPA كوسيلة للتحقق التلقائي من مستوى أمان البروتوكولات الأمنية، هذه الأداة تدمج الطرق مختلفة النهج لتحليل الأمن، بدءً من نموذج فحص التقنيات لتحليل تزوير البروتوكول، إلى أساليب التحقق الرمزية على أساس التحقق بشكل مجرد ، السمة الرئيسية لهذه الأداة هي أدوات التحليل المقدمة. حيث تتألف من أربع أدوات – يوضح الشكل (1) بنية الأفيسبا وأدواتها– يتم فيها تشفير البروتوكول بلغة الـ HLPSL high-level specification language:

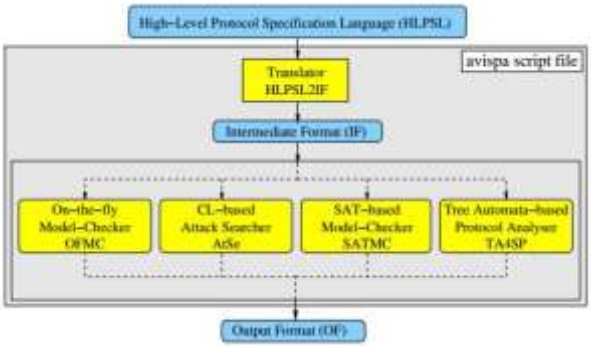
1- الأداة الأولى: CL-Atse Constraint-Logic-based Attack Searcher تستخدم منطق الهجوم مع التقييد حيث يطبق محددات للحل والتبسيط مع تقنيات إلغاء التكرار

2- الأداة الثانية OFMC On-the-Fly Model-Checker تقوم بتوظيف تقنيات الترميز لفحص أداء اختراق البروتوكول فضلا عن التحليل المحدود، من خلال استكشاف فضاء الحالة بطريقة مبنية على الحاجة، قدمت هذه الأداة عدد من التحسينات من خلال تخفيض الترتيب جزئياً.

3- الأداة الثالثة SAT-MC SAT-based Model-Checker تبني معادلات تشفير مقترحة لكل الآثار المحتملة على البروتوكول، تستخدم خوارزمية حل من نوع SAT.

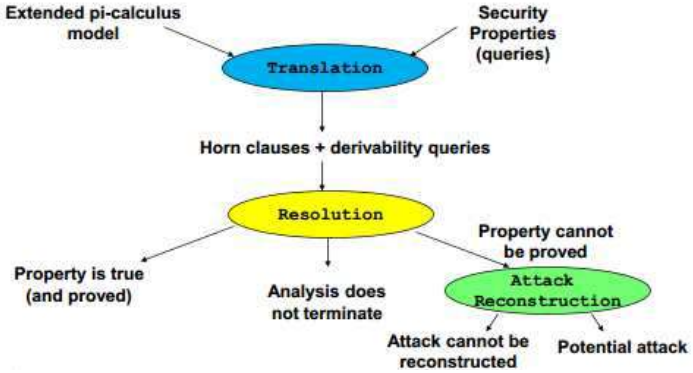
4- الأداة الرابعة TA4SP Tree Automata for Security Protocols تعتمد على أسلوب التقريب التلقائي للتحليل لمعرفة الاختراق تستخدم اللغات الشجرية العادية وتقوم بإعادة كتابة البروتوكول لكي تقوم بتقديم قيم فروق تقريبية.

إن الأدوات الثلاث الأولى تقوم بأخذ السيناريو المعماري من الـ HLPSL للبروتوكول وتقوم بتحليله ككل. أما الأداة الأخيرة فتقوم بأخذ السيناريو المعماري من الـ HLPSL أيضاً ولكن تقوم بالتحقق من فضاء الحالة والتي تحدد عدد مرات التكرار وبما أنها تعمل على التقدير الزائد فقد تجد أخطار هجمات زائفة.



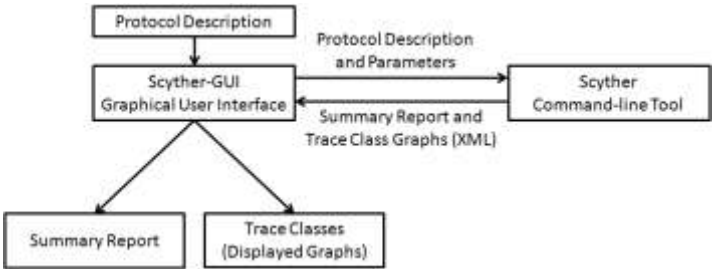
الشكل (1) بنية الـ أفيسبا [19]

3- بروفيرف ProVerif:[3, 11, 14, 19, 22] إن أداة الـ ProVerif هي أداة لتحليل البرتوكولات باستخدام تقنيات فاي للتفاضل والتكامل fi calculus وتوسعاتها من نظريات المعادلات والتوابع والتي يمكن بواسطتها تمثيل معاملات التشفير، تقوم بتحليل عدد غير محدود من البروتوكولات اعتماداً على خصائص جمل هورن Horn clauses. يتم التحقق من سرية الرسائل عن طريق اضافة جملة للمخترق في كل رسالة ثم يقوم بمحاولة مهاجمتها ويقوم بمحاولة الاستدلال على البيانات من خلال جمل هورن. يوضح الشكل (2) بنية البروفيرف.



الشكل (2) بنية الـ بروفيرف [14]

4- سايزر Scyther: [5, 15, 19, 26] تعد الـ Scyther واحدةً من أكثر أدوات الاختبار حداثة طورها كرمرز Cremers في جامعة ايدنهون للتقنية Technology Eindhoven University of وهي ذات واجهة رسومية يتم تنفيذ تحليل البروتوكولات الأمنية فيها بواسطة ضغطة زر واحدة، تتضمن هذه الأداة سطر أوامر يكتب الأوامر فيه بلغة البايثون Python، في هذه الأداة يتم أخذ وصف البروتوكول والبارامترات الأخرى كدخل أما الخرج فيقدم تقرير ملخص ويعرض مخطط لكل هجوم ويوضح الشكل (3) آلية عمل هذه الأداة.



الشكل (3) آلية عمل الـ Scyther [19]

المنهجية:

كما ذكر سابقاً تعتمد هذه الدراسة على اجراء الاختبارات ومقارنة أداء أدوات الاختبار الرسمية للبروتوكولات الأمنية من خلال تطبيقها على عدد من البروتوكولات الأمنية، حيث تم اختيار عدد من البروتوكولات في صيغتها الأولى لكي يتم تطبيق الاختبارات عليها، كون هذه البروتوكولات غالباً ما تم اكتشاف ثغرات أمنية فيها رغم أمنها في الظاهر مما يظهر قوة الأدوات المختبرة في كشف الثغرات الأمنية. حيث تم مراجعة العديد من الأوراق والاطلاع على نتائج تحليلها للبروتوكول باستخدام طرق رسمية مختلفة ومن ثم تم إعادة إجراء الاختبارات باستخدام بعض هذه الطرق. وبناء على ما سبق فقد تم دراسة الأدبيات المنشورة في هذا المجال واعتماد الاختبار التي نفذتها بالأداتين ProVerif

وScyther وتم إجراء الاختبارات مرة أخرى لنفس البروتوكولات التي تم اختبارها بواسطة الأداتين المذكورتين مرة أخرى بواسطة Avispa وCasper/FDR.

البروتوكولات المختبرة Analyzed and tested Protocols:

لقد تم اختيار ستة بروتوكولات وهي:

1- بروتوكول كاو شاو للتحقق Kao Chow Authentication Protocol.

2- بروتوكول 3-د الآمن 3-D Secure Protocol

3-بروتوكول ندهم-شرودر للمفتاح العمومي Needham-Schroeder Public Key Protocol

4- بروتوكول تبادل المفاتيح دفي-هلمان Diffie-Hellman key exchange

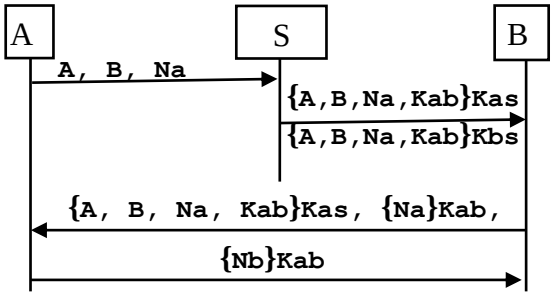
5- بروتوكول Andrew Secure RPC Protocol

6- بروتوكول مصادقة مصافحة التحدي Challenge Handshake Authentication Protocol

1- بروتوكول كاو شاو للتحقق Kao Chow Authentication Protocol:

هو بروتوكول تحقق مشترك مع بروتوكول توزيع المفاتيح يهدف للحصول على تحقق قوي مع تبادل رسائل منخفض. وهنا يستخدم طرف ثالث موثوق به S لتوليد مفتاح الجلسة K وتوزيعه إلى طرفي الاتصال المستخدمين A, B والذين يملكان مفتاحين للتشفير مع الطرف الثالث Kas, Kbs ليستخدماه في تشفير الاتصال فيما بينهما [12]. ويوضح الشكل (4) مخطط وشيفرة هذا البروتوكول.

1. A -> S : A, B, Na
2. S -> B : {A, B, Na, Kab}Kas, {A, B, Na, Kab}Kbs
3. B -> A : {A, B, Na, Kab}Kas, {Na}Kab, Nb
4. A -> B : {Nb}Kab



الشكل (4) مخطط وشيفرة Kao Chow Authentication Protocol

1- في هذا البروتوكول يقوم الطرف A بإرسال رسالة تتضمن عبارة إلى موزع المفاتيح S وتعريف عن نفسه ويطلب مفتاح للجلسة بينه وبين الطرف B.

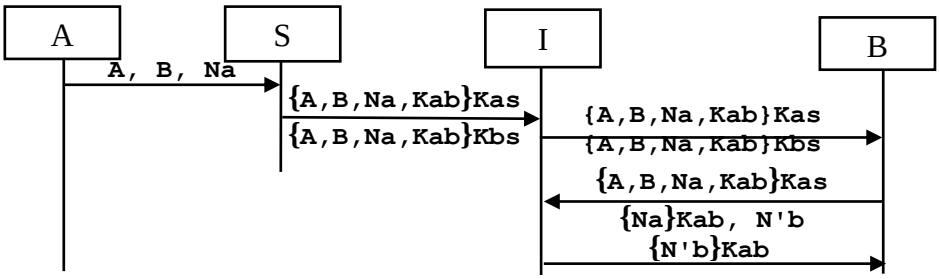
2- يقوم المخدم بإرسال رسالة إلى B تتضمن شقين الأول هو رسالة مشفرة بالمفتاح الخاص بينه وبين A إلى تتضمن مفتاح الجلسة الجديد بالإضافة إلى رسالة A الأصلية {A, B, Na, Kab}Kas والشق الثاني هو نفس رسالة A إلى S مشفرة بمفتاح B بينه وبين S {A, B, Na, Kab}Kbs.

3- يقوم B بإرسال رسالة إلى A تتضمن الجزء الذي أرسله S والمتعلق به {A, B, Na, Kab}Kas وعبارة A المرسلة إلى S مشفرة بمفتاح الجلسة {Na}Kab بالإضافة إلى عبارة من B هي Nb.

4- بعدها يقوم A بالإرسال إلى B عبارة B مشفرة بمقتاح الجلسة {Nb}Kab.

إن مشكلة هذا البروتوكول هي في أنه قد يواجه قيام المخترق بأخذ دور المخدم S ومخاطبة B على أنه S إذ يقوم بعد الخطوة الثانية وبدلاً من السماح لرسالة S بالوصول إلى B مباشرة يقوم هو بالحصول على الرسالة وإعادة

ارسالها إلى B وبالتالي يحصل التزوير في الرسالة وانتحال شخصية S وبعد أن يرد B يقوم المهاجم بانتحال شخصية A واستلام الرسالة من B وبالتالي يتم إجراء المحادثة. ويوضح الشكل(5) كيفية حدوث الهجوم على هذا البروتوكول.



الشكل (5) كيفية حدوث الهجوم على بروتوكول Kao Chow

اختبار Kao Chow Authentication Protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(1):

الجدول (1) نتائج الاختبار

	sender side	receiver side	initiator side
Scyther	Attack	Attack	No Attack
ProVerif	Attack	Attack	Attack
Casper/FDR	Attack	Attack	No Attack
Avispa	Attack	Attack	Attack

1- عندما تم اختبار هذا البروتوكول في Scyther وجد إمكانية للاختراق من كلا طرفي الاتصال المرسل والمستقبل، إن مفتاح الجلسة كان مؤمناً من الطرف A ولكنه كان مكشوفاً في طرف المستقبل وهذا سبب اعتبار إمكانية الاختراق على كلا طرفي الاتصال، وعندما تم إعادة كتابة الشفرة من طرف المحرض تم اعتبار أنه لا يوجد هجوم.

2- عندما تم اختبار البروتوكول في ProVerif فإنه تم الحصول على نتائج مشابهة والتي هي إمكانية تحديد مفتاح الجلسة من طرف المستقبل إضافة إلى أن مفتاح الجلسة لم يكن أمناً من طرف المحرض.

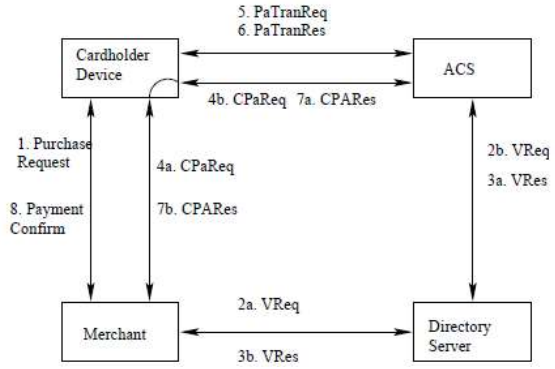
3- في Casper وجد إمكانية للاختراق من كلا طرفي الاتصال المرسل والمستقبل، إن مفتاح الجلسة كان مؤمناً من الطرف A ولكنه كان مكشوفاً في طرف المستقبل وهذا سبب اعتبار إمكانية الاختراق على كلا طرفي الاتصال، ولم يتم اكتشاف أي اختراق من طرف المحرض.

4- في Avispa فإنه تم الحصول على نتائج مشابهة لنتائج ProVerif من كشف مفتاح الجلسة من طرف المستقبل والمحرض.

2- بروتوكول 3-د الآمن 3-D Secure:

هو بروتوكول يعتمد على XML حيث يستخدم كطبقة أمنية إضافية للتحويلات المعتمدة على البطاقات البنكية، قامت شركة فيزا Visa بتطويره بغية تحسين أمن المعلومات عند اجراء عمليات الدفع عن طريق الإنترنت. تم تقديمه لعملاء فيزا ثم قامت ماستر كارد MasterCard باعتماده.

قامت فيزا/سكيوركود Visa/Secure Code والتي ستقوم بإنشاء طريق لإعادة التوجيه إلى موقع المصرف الذي أصدر البطاقة ليتم السماح بالعملية. كل مصدر بطاقة سيكون قادرا على التعامل مع أي طريقة للتحقق، إن أكثر هذه الطرق شيوعاً هي كلمة السر. لذا وللشراء عن طريق الانترنت بطريقة فعالة توجب وجود كلمة سر مربوطة مع البطاقة. بالإضافة إلى هذا فإن بروتوكول فيزا يتطلب من البنك التحقق من الصفحة مباشرة وبالتالي فإن البنك يصبح هو المسؤول عن التسريبات الأمنية [18, 25]، ويوضح الشكل (6) مخطط هذا البروتوكول.



الشكل (6) مخطط البروتوكول 3-D Secure [18]

1. C -> M : {pan, expiry}{keyMC} -- VRreq
2. M -> DS : {pan, macqbin, mid, mpasswd}{keyDSM}
2a. DS -> ACS :{pan,macqbin,mid,mpasswd} {keyACSDS} -- VRres
3. ACS -> DS : {panyes,acctid,url,proto} {keyACSDS}
3a. DS -> M : {panyes,acctid,url,proto}{keyDSM} -- CPAreq
4. M -> C: {{macqbin,mid,mname,murl,xid,pdate,pamt, expiry,acctid}{SK(M)}% pareq }{keyMC}
4a. C -> ACS : { pareq % {macqbin,mid,mname, murl, xid,pdate,pamt,expiry,acctid} {SK(M)} }{keyACSC} -- PATranReq
5. ACS -> C : {mname,pamt,pdate,panshort, expiry}{keyACSC} -- PATranRes
6. C -> ACS : {password}{keyACSC} -- CPARes
7. ACS -> C: {{macqbin,mid,xid,pdate,pamt,panshort, datetime,transtatus,cavv,eci, cavvalg}{SK(ACS)} % pares }{keyACSC}
7a. C -> M : { pares % {macqbin,mid,xid,pdate, pamt,panshort,datetime,transtatus,cavv, eci,cavvalg}{SK(ACS)} }{keyMC} -- Payment Confirm
8. M -> C : { {transtatus}{keyMC} }{SK(M)}

شيفرة البروتوكول 3D Secure [18]

اختبار 3-D secure protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(2):

الجدول (2) نتائج اختبار 3-D secure protocol

	customer side	merchant side	bank side
Scyther	No Attack	No Attack	Attack
ProVerif	No Attack	No Attack	No Attack
Casper/FDR	No Attack	Attack	No Attack
Avispa	No Attack	No Attack	No Attack

1- عند تحليل هذا البروتوكول بواسطة Scyther لم يكتشف أي اختراق من طرف الزبون أو التاجر ولكن الهجوم من طرف البنك، حيث أن مفاتيح الجلسة كانت سرية من طرفي الزبون والتاجر ولكن كشف المفاتيح من طرف البنك، وهنا يظهر بعض قصور هذه الأداة بسبب عدم قدرة هذه الأداة على المقارنة متحولين بنفس الوقت.

2- بتطبيق الاختبار بواسطة ProVerif وجد أن البروتوكول آمن تماماً ولا يحصل أي اختراق سواء بسبب كلمة السر أو المفاتيح.

3- في Casper وجد أن البروتوكول آمن بشكل كامل ولكن يمكن أن يقوم التاجر بعمليات استنساخ عمليات التحقق.

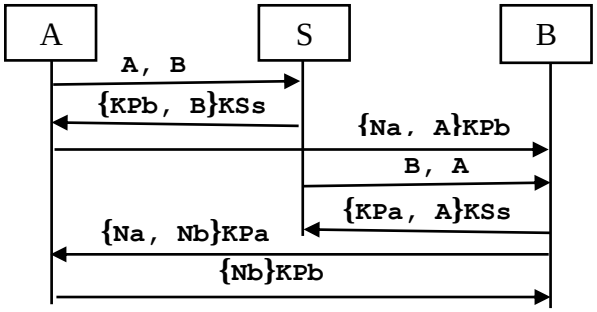
4- في Avispa كان البروتوكول آمناً بشكل كامل أيضاً.

3- بروتوكول ندهم-شرودر للمفتاح العمومي Needham-Schroeder Public Key Protocol:

إن بروتوكول ندهم-شرودر للمفتاح العمومي يعتمد على تشفير المفتاح العمومي، حيث يتم المصادقة المتبادلة بين الأطراف على الشبكة باستخدام المفتاح العمومي حيث يفترض أن كل طرف يعرف المفتاح العمومي للأطراف الأخرى، عادة ما يمثل A و B كأطراف متصلة [17]. وفيما يلي شرح لهذا البروتوكول وآلية عمله:

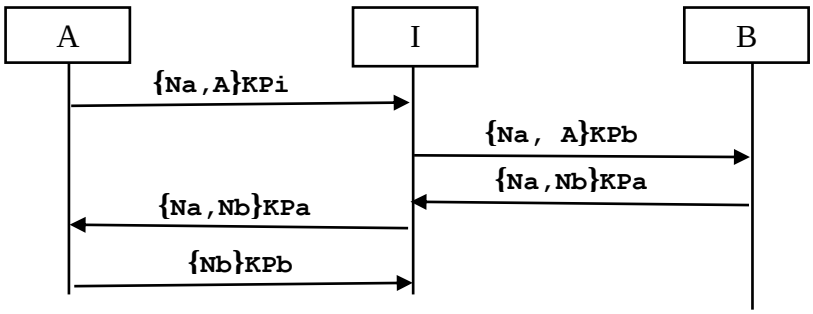
يقوم الطرف A بإعلام المخدم عن طرفي الاتصال A, B يقوم المخدم S بإرسال رسالة إلى A فيها المفتاح العمومي الخاص ب B وعنوان B مشفرة بالمفتاح الخصوص الخاص بـ S وبالتالي يقوم A بإرسال رسالة إلى B تحوي عبارة مع عنوانه مشفرة بالمفتاح العمومي لـ B عندها يطلب B من S المفتاح العمومي لـ A فيقوم بإرساله له، في المرحلة السادسة يقوم B بإرسال رسالة تحوي على عبارة من B مع العبارة التي أرسلها A مشفرة بالمفتاح العمومي الخاص بـ A يعيد A إلى B العبارة التي أرسلها مشفرة بمفتاحه العمومي ويوضح الشكل (7) مخطط وشيفرة هذا البروتوكول.

1. A -> S : A,B
2. S -> A : {KPb, B}KSs
3. A -> B : {Na, A}KPb
4. B -> S : B,A
5. S -> B : {KPa, A}KSs
6. B -> A : {Na, Nb}KPa
7. A -> B : {Nb}KPb



الشكل (7) مخطط وشيفرة Needham–Schroeder Public Key Protocol

إن مشكلة هذا البروتوكول هي في أنه قد يواجه قيام المخترق بأخذ دور المخدم S ومخاطبة A, B على أنه S إذ يقوم بعد الخطوة الثانية وبدلاً من السماح لرسالة S بالوصول إلى B مباشرة يقوم هو بالحصول على الرسالة وإعادة ارسالها إلى B وبالتالي يحصل التزوير في الرسالة وانتحال شخصية S وبعد أن يرد B يقوم المهاجم بانتحال شخصية A واستلام الرسالة من B وبالتالي يتم إجراء المحادثة ويقوم الـ I بإعطاء الأطراف مفاتيحه الخاصة. ويوضح الشكل (8) كيفية الهجوم عليه.



الشكل (8) الهجوم على Needham–Schroeder Public Key Protocol

اختبار Needham–Schroeder Public Key Protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(3):

1- باختبار هذا البروتوكول بواسطة Scyther وجد أنه يوجد هناك إمكانية لاختراقه من طرف المستقبل وبالتالي يمكن الحصول على كلا العبارتين من قبل المخترق.

2- مع تطبيق الأداة ProVerif ولعدد غير محدد من المرات فإنه قد تم الحصول على نتائج مشابهة لما تم الحصول عليه في Scyther.

3- بتطبيق Casper نجد إن هذا البروتوكول قابل للاختراق عند المستقبل.

4- نفس النتائج يتم الحصول عليها عند اختبار هذا البروتوكول باستخدام Avispa.

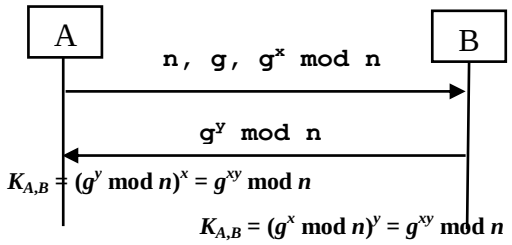
الجدول (3) نتائج الاختبار Needham–Schroeder Public Key Protocol

	sender side	receiver side	synchronization and agreement
Scyther	No Attack	Attack	Attack
ProVerif	No Attack	Attack	Attack
Casper/FDR	Attack	Attack	Attack
Avispa	Attack	Attack	Attack

4- بروتوكول تبادل المفاتيح دفي-هلمان Diffie–Hellman key exchange:

هو برتوكول تشفير يسمح لفريقين أن يتبادلا مفتاح التشفير على قناة غير آمنة بدون أن يكون لهما معرفة مسبقة ببعضهما البعض. هذا المفتاح يستخدم لتشفير الاتصالات المستقبلية بين الطرفين باستخدام مفتاح متماثل. مبدأه كالتالي يوجد نوعان من الأعداد الأولية المعروفة الرقم الأولي p و g. كل طرف يختار رقم عشوائي أقل من p ويحتفظ به لنفسه وباستخدام خصائص تابع الموديولار في الرياضيات modular arithmetic وبناء عليه يتم احتساب المفتاح ومن ثم تبادله بين الأطراف على قناة غير آمنة، الشكل (9) يوضح مخطط وشيفرة هذا البروتوكول[8].

1. A -> B : n, g, g^x mod n
2. B -> A : g^y mod n



الشكل (9) آلية عمل وشيفرة Diffie–Hellman key exchange

يختار A عدد x ويبقيه لنفسه ويختار B عدد y ويبقيه لنفسه ومن ثم يرسل A إلى B رسالة تحتوي على عددين أولين n و g بعد ذلك يقوم A بتوليد $g^x \bmod n$ ويرسله إلى B محققاً ويقوم B بتوليد $g^y \bmod n$ ويرسله لـ A وبالتالي فإن المفتاح يكون من طرف A:

$$K_{A,B} = (g^y \bmod n)^x = g^{xy} \bmod n$$

ومن طرف B:

$$K_{A,B} = (g^x \bmod n)^y = g^{xy} \bmod n$$

اختبار Diffie Hellman key exchange protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(4):

1- لا يمكن محاكاة هذا البروتوكول بواسطة Scyther بسبب أن هذه الأداة لا يمكنها محاكاة عمليات الرفع إلى قوة.

2- في ProVerif نجد أن المفاتيح المولدة غير آمنة لا من طرف المصدر ولا من طرف المستقبل.

3- في Casper عندما يتم تحليل البروتوكول تم العثور على امكانية اختراق واضحة بحيث أنه لا يمكن استخدام الآس كعمليات مصادقة.

4- نفس النتائج تم الحصول عليها في Avispa حيث أن هذا البرتوكول غير آمن ومعرض للاختراق.

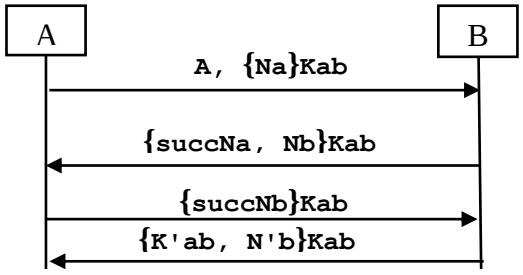
الجدول (4) نتائج اختبار Diffie–Hellman key exchange protocol

	Initiator side	Receiver side
Scyther	Can't be modeled	Can't be modeled
ProVerif	Attack	Attack
Casper/FDR	Attack	Attack
Avispa	Attack	Attack

5- بروتوكول أندرو لتأمين الـ Andrew Secure RPC Protocol RPC:

يعمل هذا البروتوكول على توزيع مفتاح خاص بكل جلسة عمل جديدة بين الأطراف المتصلة A و B ويجب على هذا البروتوكول ضمان سرية المفتاح الرئيسي المشترك الجديد k. يجب على قيمة k أن تكون معروفة فقط من الأطراف المتصلة A وB. كما يتوجب على البرتوكول أن يضمن صحة وسلامة k في الرسالة التي تم انشاؤها من قبل A في نفس الجلسة. والرسالة الأخيرة تحوي على N'b والذي يمكن استخدامه مستقبلاً كرقم مصافحة، الشكل (10) يوضح مخطط وشيفرة هذا البروتوكول[21].

1. A -> B : A, {Na}Kab
2. B -> A : {succNa, Nb}Kab
3. A -> B : {succNb}Kab
4. B -> A : {K'ab, N'b}Kab



الشكل (10) مخطط وشيفرة Andrew Secure RPC Protocol

إن مشكلة هذا البروتوكول هي في الخطوة الرابعة منه لا يحصل A يدل على أن هذه الرسالة هي من B وبالتالي يمكن للمخترق I أن يقوم بإعادة إرساله لاحقاً في جلسات أخرى مما يعطي ايحاء إلى B بأنه قد حصل على اتصال من A.

اختبار Andrew Secure RPC Protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(5):

1- عندما يتم اختبار هذا البروتوكول في Scyther يتم اكتشاف أن هذا البروتوكول غير آمن ومعرض للأختراق في طرف المصدر وأن العبارة يمكن الحصول عليها في طرف المستقبل وأن الهجوم يكون بشكل أساسي عن طريق كشف مفتاح الجلسة. وبسبب ذلك لايمكن للطرف A أن يعرف فيما إذا كانت الرسالة من قبل B أو من قبل المخترق حيث يمكن للمخترق استخدام المفتاح الأقدم مع A، وبالتالي هناك اختراق أكيد واحد على الأقل.

2- عند اختبار البروتوكول بواسطة ProVerif نحصل على نتائج مسبقة والتي هي أن مفتاح الجلسة ليس آمناً من طرف المصدر، ولكنه في المقابل آمن من طرف المستقبل.

3- في Casper فإن هذا البروتوكول هو غير آمن بشكل كامل لا من طرف المصدر ولا من طرف المستقبل حيث يمكن للمخترق الحصول على مفتاح الجلسة وبالتالي حدوث اختراق.

4- نتيجة مماثلة لما تم استخلاصه في Casper يمكن الحصول عليها في Avispa وهي أن البروتوكول غير آمن.

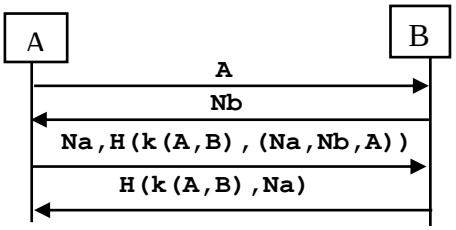
الجدول (5) نتائج اختبار Andrew Secure RPC Protocol

	Initiator side	Receiver side
Scyther	Attack	Attack
ProVerif	Attack	No Attack
Casper/FDR	Attack	Attack
Avispa	Attack	Attack

6- بروتوكول مصادقة مصافحة التحدي Challenge Handshake Authentication (CHAP) Protocol:

يستخدم البروتوكول المصافحة للتحقق ثلاثية الطرق بشكل دوري للتحقق من هوية الطرف الآخر. يتم ذلك عن طريق بناء رابط اولي، ومن الممكن تكرارها في أي وقت بعد إنشاء الرابط. ما أن ينشأ هذا الرابط فإن المتحقق يرسل رسالة التحدي إلى الطرف الآخر. يقوم الطرف الآخر بالاستجابة بقيمة محسوبة باستخدام التشفير احادي الاتجاه one-way hash، بعد ذلك يقوم المتحقق بفحص الاستجابة ويقارنها بالقيمة المتوقعة لديه، وفيما إذا كانت القيمتين متطابقتين فإن المتحقق يأخذ علماً بذلك، وفيما عدا ذلك فإنه يجب انهاء الاتصال [23].

1. A -> B : A
2. B -> A : Nb
3. A -> B : Na,H(k(A,B),(Na,Nb,A))
4. B -> A : H(k(A,B),Na)



الشكل (11) مخطط شيفرة Challenge Handshake Authentication Protocol

يرسل الطرف A رسالة إلى B تتضمن اسمه ثم يقوم B بالرد بواسطة عبارة ما يرد عليه B برسالة تحوي عبارة من A ونتيجة تابع التشفير احادي الاتجاه بواسطة شيفرة مسبقة التعريف لقيمة عبارة B بالإضافة إلى قيمة عبارة A مع اسم A، يرد B بنتيجة تابع التشفير احادي الاتجاه لقيمة A.

اختبار Challenge handshake authentication protocol:

قام الباحثون في هذا البحث باختبار هذا البرتوكول بواسطة أداتي Casper و Avispa ومن ثم مقارنة نتائج الأختبار مع نتائج اختبار دالال ورفاقه Dalal et al [6] التي تم تنفيذها بواسطة Scyther و ProVerif وفيما يلي نتائج الاختبارات، والتي يمكن تلخيصها في الجدول(6):

الجدول (6) نتائج اختبار Challenge Handshake Authentication Protocol

	Initiator side	Server side
Scyther	Attack	No Attack
ProVerif	No Attack	No Attack
Casper/FDR	No Attack	No Attack
Avispa	No Attack	No Attack

1- إن الاختبار بواسطة Scyther اظهر أن مفتاح التشفير المتماثل آمن على كلا الطرفين ولكن هناك اختراق في الهجوم التزامن والهجوم المتوافق على الطرف A وليس على المخدم وذلك بسبب كون الرسالة الأولى من المخدم غير مشفرة.

2- عند اجراء الاختبار بواسطة ProVerif فإن النتيجة هي أن المفتاح المتماثل لا يمكن اختراقه على كلا طرفي الاتصال وبالتالي فإن البروتوكول آمن.

3- نفس نتائج ProVerif نحصل عليها عند الاختبار بواسطة Casper والبروتوكول آمن.

4- نفس نتائج ProVerif نحصل عليها عند الاختبار بواسطةِ Avispa والبروتوكول آمن.

تقييم الطرق الرسمية المستخدمة:

تم في هذا البحث استخلاص مقارنة بين مزايا كل أداة من الأدوات ووضع جدول يلخص مزايا كل من هذه الأدوات كما في الجدول (7):

الجدول (7) مقارنة مزايا الطرق الرسمية

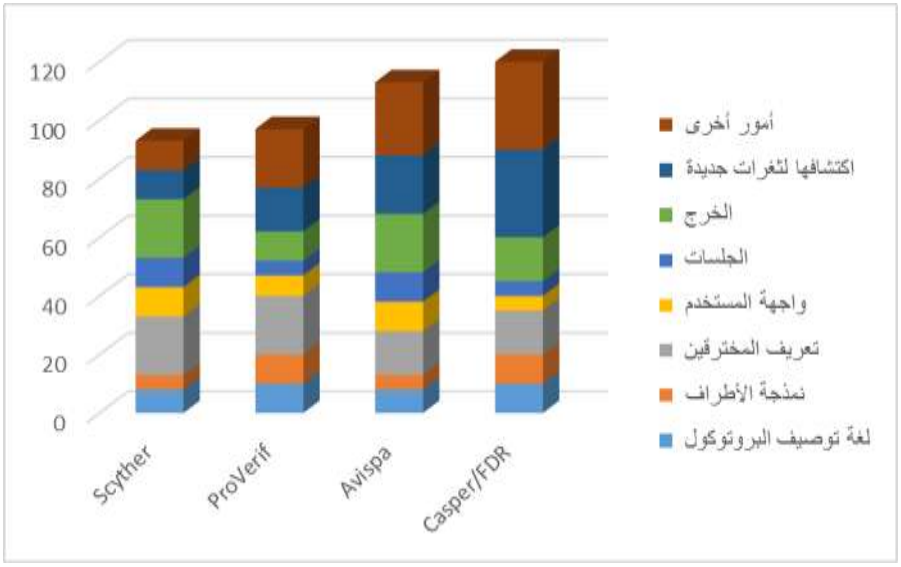
	Scyther	ProVerif	Avispa	Casper/FDR
لغة توصيف البروتوكول	SPDL	Horn clauses or pi calculus	HLSPL	CSP
نمذجة الأطراف	كأدوار	كعمليات	كأدوار	كعمليات
دخل	- لا داعي لتعريف المهاجمين	- لا حاجة لتحديد المخترقين	- المخترقون المتوقعون لا يحتاجون للتوصيف كعملاء	- يجب تحديد عدد العملاء - يجب تحديد المهاجمين وقوتهم
واجهة المستخدم	رسومية	نصية	رسومية	نصية
الجلسات	محدد أو غير محدد	غير محدد	محدد أو غير محدد	محدد
الخرج	- عندما يتم توليد الهجوم فإنه يتم توليد مخطط تتبع مع تفسير ذاتي - يتم توليد جميع الهجمات ومتابعة أثارها	- تولد ثلاثة حالات البروتوكول موثوق أو غير موثوق بالإضافة إلى توليد مخطط تتبع للهجوم.	- يتم الفحص لعمق محدد. - ملاحقة آثار الهجوم - إظهار جميع الآثار. - يتم توليد آثار الهجوم مع شرح ذاتي لكل أثر.	- للحصول على الخرج يتطلب وقت طويل نسبياً

أمور أخرى	– تقوم الأداة وبشكل آلي بفحص جميع المتغيرات الممكنة ولا حاجة لتعريف أي منها – لا يمكن فحص تساوي المتغيرات المختلفة – لا يوجد أي تعريف لمفهوم القنوات – لا يمكنها توصيف جميع البروتوكولات	– يتم التحقق فقط من الهجمات المحددة في البرنامج. – يتم بحث أي أثر ممكن لهجوم عندما يتوقف البروتوكول – يجب تحديد قنوات الاتصال	– يتم التحقق من سرية جميع المتغيرات الممكنة وفق التقدير الخاص للأداة، دون طب صريح بذلك – يتم متابعة كل أثر ممكن.	– يتم التحقق فقط من الهجمات المحددة في البرنامج. – يجب تحديد قنوات الاتصال – من الممكن تعديل قوة المهاجم على أي قناة من قنوات.

وبناء على المقارنة أعلاه فقد وضع جدول يعطي لكل طريقة من هذه الطرق علامة في كل موضوع من المواضيع المذكورة كما هو موضح في الجدول(8) وتم نقله إلى مخطط بياني(12).

الجدول (8) تقييم الطرق الرسمية الأربعة

	علامة التقييم الأعظمية	Scyther	ProVerif	Avispa	Casper/FDR
لغة توصيف البروتوكول	10	8	10	8	10
نمذجة الأطراف	10	5	10	5	10
تعريف المخترقين	20	20	20	15	15
واجهة المستخدم	10	10	7	10	5
الجلسات	10	10	5	10	5
الخرج	20	20	10	20	15
اكتشافها لثغرات جديدة	30	10	15	20	30
أمور أخرى	30	10	20	25	30
المجموع	140	93	97	113	120



الشكل (12) تقييم الطرق الرسمية الأربعة

وبالتالي يمكننا أن نلاحظ أن الطرق الرسمية للتحقق من البروتوكولات الأمنية هي مجال بحثي مهم وواعد فإن هذه الأدوات ولدى القيام بتطبيقها لاختبار ستة بروتوكولات أظهرت خلافات في النتائج فـ Scyther تتمكن من تحليل أحد البروتوكولات ونفس الأداة في مرة أخرى أظهرت وجود اختراق في حين باقي الأدوات ونفس لـ Casper فقد تمكنت من كشف عدة عيوب في البروتوكولات لم تكشفها الأدوات الأخرى. باستخدام هذه الطرق من الممكن كشف الثغرات الموجودة في البروتوكولات الأمنية مما يساهم في تصحيح هذه الثغرات، كما أن كشف الثغرات لا يعني بالضرورة أن هذه البروتوكولات أصبحت آمنة 100%.

الاستنتاجات والتوصيات:

أولاً: أظهرت نتائج اختبار عدد من البروتوكولات الأمنية بواسطة مجموعة من طرق التحقق الرسمية، ما يلي:

1- اعتبار البروتوكولات التالية غير آمنة:

- Kao Chow Authentication Protocol
- Needham–Schroeder Public Key Protocol
- Diffie–Hellman key exchange
- Andrew Secure RPC Protocol

2- اعتبار البروتوكولات التالية آمنة:

- 3-D Secure
- Challenge Handshake Authentication Protocol

ثانياً: يقترح لضمان أمن البروتوكول بشكل حتمي القيام باختبار البروتوكول بواسطة طرق مختلفة.

ثالثاً: يتضح حسب المخطط (12) والجدول (8) بأن طريقة كاسبر Casper هي أفضل الطرق من حيث الأمان تليها طريقة أفيسبا Avispa.

الخاتمة:

مما سبق أعلاه نجد أن الطرق الرسمية المستخدمة للتحقق من البروتوكولات الأمنية تؤدي جميعاً نفس الغرض المطلوب وهو التحقق من مدى أمان البروتوكول، ولكنها تختلف واحدة عن أخرى عن طريق التحقق من البروتوكول بأسلوب التحقق ومدى سهولة التعامل. في هذا البحث تم اثبات أن أفضل طريقة هي طريقة كاسبر Casper هي أفضل الطرق من حيث الأمان تليها طريقة أفيسبا Avispa.

أعمال مستقبلية:

- يقترح اختبار عدد أكبر من البروتوكولات مستقبلاً لزيادة الثقة في الأدوات والبروتوكولات.
- يقترح اختبار البروتوكولات مستقبلاً بطرق جديدة.

المراجع:

[1] ALSHEHRI, A. A. *NFC mobile coupon protocols: developing, formal security modelling and analysis, and addressing relay attack* (Doctoral dissertation, University of Surrey). (2015).

[2] BIONDI, F., & LEGAY, A. Security and Privacy of Protocols and Software with Formal Methods. In *International Symposium on Leveraging Applications of Formal Methods* (pp. 883-892). Springer International Publishing. (2016).

[3] BLANCHET, B., SMYTH, B., & CHEVAL, V. (2015). ProVerif 1.90: *Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*. *URL:* http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf online last accessed on 22-7-2016

[4] BUTLER, R. W. (2001). What is Formal Methods?. *NASA LaRC Formal Methods Program*. https://shemesh.larc.nasa.gov/fm/fm-what.html, online last accessed on 20-7-2016

[5] CREMERS, C. The Scyther tool: Automatic verification of security protocols. (2009).

[6] DALAL, N., SHAH, J., HISARIA, K., & JINWALA, D. A comparative analysis of tools for verification of security protocols. *Int'l J. of Communications, Network and System Sciences*, *3*(10), 779. (2010).

[7] DENNING, D. E., & SACCO, G. M. *Timestamps in key distribution protocols. Communications of the ACM*, *24*(8), (1981). 533-536.

[8] DIFFIE, W., & HELLMAN, M. *New directions in cryptography. IEEE transactions on Information Theory*, *22*(6), (1976). 644-654.

[9] DOLEV, D., & YAO, A. *On the security of public key protocols. IEEE Transactions on information theory*, *29*(2), (1983). 198-208.

[10] HENZL, M., & HANACEK, P. A *Security Formal Verification Method for Protocols Using Cryptographic Contactless Smart Cards*. Radioengineering. (2016).

[11] HIRSCHI, L., BAELDE, D., & DELAUNE, S. *A Method for Verifying Privacy-Type Properties: The Unbounded Case.* (2016).

[12] KAO, I., & CHOW, R. *An efficient and secure authentication protocol using uncertified keys. ACM SIGOPS Operating Systems Review*, *29*(3), (1995). 14-21.

[13] KOBLITZ, N., & MENEZES, A. J. *Another look at" provable security". Journal of Cryptology*, *20*(1), (2007). 3-37.

[14] KODRA, S., *Protocol verification with Proverif*. University of Tartu. (2015)

[15] KURKOWSKI, M., KOZAKIEWICZ, A., & SIEDLECKA-LAMCH, O. (2017). Some Remarks on Security Protocols Verification Tools. In *Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology–ISAT 2016–Part II* (pp. 65-75). Springer International Publishing.

[16] LOWE, G. Casper: *A compiler for the analysis of security protocols*. Journal of computer security, 6(1, 2), (1998). 53-84.

[17] NEEDHAM, R. M., & SCHROEDER, M. D. *Using encryption for authentication in large networks of computers. Communications of the ACM*, *21*(12), (1978). 993-999.

[18] PASUPATHINATHAN, V., PIEPRZYK, J., WANG, H., & CHO, J. Y. (2006, January). Formal analysis of card-based payment systems in mobile devices. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54* (pp. 213-220). Australian Computer Society, Inc..

[19] PATEL, R., BORISANIYA, B., PATEL, A., PATEL, D., RAJARAJAN, M., & ZISMAN, A. Comparative analysis of formal model checking tools for security protocol verification. In *International Conference on Network Security and Applications* (pp. 152-163). Springer Berlin Heidelberg. (2010).

[20] RAMEZANI, K., SITHIRASENAN, E., & SU, K. *Formal Security Analysis of EAP-ERP Using Casper*. *IEEE Access*, *4*, (2016). 383-396.

[21] SATYANARAYANAN, M. *Integrating security in a large distributed system. ACM Transactions on Computer Systems (TOCS)*, *7*(3), (1989). 247-280.

[22] SHINDE, A. H., & UMBARKAR, A. J. *Analysis of Cryptographic Protocols AKI, ARPKI and OPT using ProVerif and AVISPA.* International Journal of Computer Network and Information Security, 8(3), (2016). 34.

[23] SIMPSON, W. A. (1996). *PPP challenge handshake authentication protocol (CHAP).* http://www.ietf.org/rfc/rfc1994.txt, On-line last accessed on 23/8/ 2016

[24] VIGANÒ, L. *Automated security protocol analysis with the AVISPA tool*. Electronic Notes in Theoretical Computer Science, 155, (2006). 61-86.

[25] VISA, (2011) "*Verified by visa acquirer and merchant implementation guide."* https://usa.visa.com/dam/VCOM/download/merchants/verified-by-visa-acquirer-merchant-implementation-guide.pdf, On-line last accessed 23-5-2016.

[26] YANG, H., OLESHCHUK, V., & PRINZ, A. *Verifying Group Authentication Protocols by Scyther.* Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 7(2), (2016).3-19.