دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۲، جلد ۲۱، شماره ۱ (۱۱۲-۱۳۶)
https://monadi.isc.org.ir

مروری بر روش‌های تحلیل و اثبات امنیت پروتکل‌های امنیتی

سید محمد دخیل علیان*[۱]، معصومه صفخانی[۲] و فاطمه پیرمرادیان[۱]

[۱] دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران
[۲] دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجائی، تهران، ایران

اطلاعات مقاله

تاریخچه مقاله:
تاریخ دریافت: ۲۷ آذر ۱۴۰۲
تاریخ پذیرش: ۲۱ اسفند ۱۴۰۲
انتشار آنلاین: ۷ فروردین ۱۴۰۳

کلمات کلیدی:
پروتکل‌های احراز اصالت
ابزار تحلیل امنیت پروویریف
ابزار تحلیل امنیت تامارین
ابزار تحلیل امنیت اویسپا
ابزار تحلیل امنیت سایتر
منطق BAN
منطق GNY

نوع مقاله: مروری

چکیده

ارائه تمامی خدمات از راه دور مستلزم احراز اصالت متقابل طرفین شرکت‌کننده است. چارچوبی که این احراز اصالت به وسیله آن انجام می‌شود، پروتکل‌های احراز اصالت نام دارد. به عبارتی، پروتکل رمزنگاری یا رمزنگاشتی یک الگوریتم رمزنگاری توزیع شده است که بین حداقل دو یا چند هستار با یک هدف مشخص تعاملاتی را برقرار می‌نماید. در واقع، این پروتکل‌ها کانال‌های امن و ناامنی برای ارتباط بین طرفین شرکت‌کننده در پروتکل فراهم نموده‌اند. معمولاً از کانال‌های امن جهت ثبت‌نام و از کانال‌های ناامن جهت احراز اصالت متقابل استفاده می‌شود. کاربر بعد از ثبت‌نام در سرور و تأیید اصالت آن توسط سرور می‌تواند از خدماتی که سرور ارائه می‌دهد بهره‌مند شود. پروتکل‌های احراز اصالت بسیاری در زمینه‌هایی مانند مراقبت پزشکی الکترونیکی، اینترنت اشیاء، محاسبات ابری و غیره ارائه شده است. حریم خصوصی و گمنامی کاربران در این طرح‌ها، بزرگ‌ترین چالش در پیاده‌سازی بستر جهت بهره‌مندی خدمات از راه دور است. به دلیل اینکه احراز اصالت کاربران در بستر ناامن اینترنت اتفاق می‌افتد، پس نسبت به تمامی حملات اینترنتی موجود می‌تواند آسیب‌پذیر باشد. به طور کلی دو روش جهت تحلیل و اثبات امنیت پروتکل‌های احراز اصالت وجود دارد. روش صوری و روش غیرصوری. روش غیرصوری که مبتنی بر استدلال‌های شهودی، خلاقیت تحلیلگر و مفاهیم ریاضی است، سعی و تلاش در جهت یافتن خطاها و اثبات امنیت دارد. درحالی‌که روش صوری که به دو صورت دستی و خودکار انجام می‌شود، از انواع منطق‌های ریاضی و ابزارهای تحلیل امنیت خودکار استفاده نموده است. روش دستی با استفاده از مدل‌های ریاضی مانند مدل پیشگوی تصادفی و منطق‌های ریاضی مانند منطق BAN، منطق GNY و غیره و روش خودکار با استفاده از ابزارهای اویسپا، سایتر، پروویریف، تامارین و غیره انجام شده است. در واقع روش‌های اثبات و تحلیل امنیت پروتکل‌های امنیتی به دو دسته کلی مبتنی بر اثبات قضیه و وارسی مدل تقسیم شده‌اند، که در این مقاله جزئیات هرکدام از این روش‌های اثبات و تحلیل امنیت، تحلیل امنیت پروتکل ECCPWS با برخی از این روش‌ها و در نهایت مقایسه این روش‌ها با یکدیگر از لحاظ نقاط قوت، نقاط ضعف و غیره بیان شده است. در این مقاله، روش‌های مبتنی بر وارسی مدل و سپس روش‌های مبتنی بر اثبات قضیه شرح داده می‌شود.

۱ مقدمه

توسعه و پیشرفت فناوری موجب تسریع در انجام امور و حل بسیاری از مشکلات شده است. فناوری که به خوبی توانسته تعاملی بین انسان‌ها و اشیاء برقرار کند، اینترنت اشیاء نام دارد. پیشرفت‌های مداوم فناوری

*نویسنده مسئول
آدرس‌های رایانامه: Mdalian@iut.ac.ir (سید محمد دخیل علیان)، Safkhani@sru.ac.ir (معصومه صفخانی)، f.pirmoradian@ec.iut.ac.ir (فاطمه پیرمرادیان)

مبتنی بر اینترنت اشیاء و فناوری بی‌سیم زندگی انسان‌ها را متحول کرده است. گستره کاربرد اینترنت اشیاء در حوزه تجاری روز به روز در حال پیشرفت و اضافه شدن است. از جمله کاربردهای آن می‌توان به حوزه‌ی سلامت و پزشکی، حمل و نقل، مدیریت زیرساخت و مدیریت انرژی اشاره نمود. بنابراین، اطلاعات حساس ارسالی که بر روی کانال ارسال می‌شود، می‌تواند در برابر حملات مختلف آسیب‌پذیر باشد. از این رو، امنیت در فناوری مبتنی بر اینترنت اشیاء همواره یک چالش بسیار بزرگ و جدی بوده است. با گسترش حملات سایبری، باید انتظار داشته باشیم که کاربران از راه دور اقدامات سختگیرانه‌ای را برای حفظ اطلاعات خود انجام دهند. از این رو، طراحی پروتکل‌های امنیتی احراز اصالت سبک‌وزن امن با کمترین هزینه‌های محاسباتی و ارتباطی به یک چالش اصلی تبدیل شده است. استفاده از ابزارها، روش‌های اثبات و تحلیل امنیت خودکار به دلیل اهمیت تحلیل امنیتی پروتکل‌های احراز اصالت ارائه‌شده امری بسیار ضروری است. در این مقاله به معرفی انواع روش‌های اثبات و تحلیل صوری پروتکل‌های امنیتی پرداخته شده است. هم‌چنین، در صورتی که ابزار مورد نظر، مدل امنیتی در نظر گرفته شده برای پروتکل مطرح‌شده را امن اعلام نکند، باید تدابیر امنیتی مناسبی برای پروتکل در نظر گرفته شود. در هر کدام از این ابزارها، پروتکل با استفاده از فرمان‌های مختلف اجرا شده و خروجی مورد تحلیل و بررسی قرار خواهد گرفت. به طور کلی، روش‌های اثبات و تحلیل امنیت پروتکل‌های امنیتی به دو دسته‌ی کلی مبتنی بر اثبات قضیه و وارسی مدل تقسیم شده‌اند، که جزئیات هرکدام از این روش‌های اثبات امنیت در این مقاله توضیح داده شده است. لازم به ذکر است، اکثر ابزارهای درستی‌یابی پروتکل‌های امنیتی، مبتنی بر وارسی مدل هستند. در اینجا، ابتدا روش‌های مبتنی بر وارسی مدل و سپس روش‌های مبتنی بر اثبات قضیه شرح داده می‌شود [۱، ۲].

۲ معرفی مدل‌های امنیتی مهاجم

در پروتکل‌های امنیتی قابلیت‌ها و توانایی‌های مهاجم با استفاده از چندین مدل معرفی شده‌اند که تعدادی از این مدل‌های امنیتی در ادامه توضیح داده خواهند شد [۳-۶]. در این مدل‌ها، طرفین شرکت‌کننده در پروتکل به صورت $P_1, \ldots, P_n$ نشان داده می‌شوند که با هم ارتباط برقرار می‌نمایند. به عبارتی، مدل‌های مهاجم می‌تواند از دیدگاه‌های مختلف اعم از قدرت و توانایی‌های مهاجم و هم‌چنین داخلی و یا خارجی بودن مهاجم دسته‌بندی گردد [۱].

۱.۲ مدل مهاجم خارجی

در این مدل، مهاجم کنترل کاملی سرتاسر شبکه دارد، ولی او یک مهاجم خارجی است، یعنی هویت ثبت‌شده‌ی مجازی داخل سامانه و هیچ کلید خصوصی بلند مدتی ندارد.

۲.۲ مدل مهاجم داخلی

در این مدل، مهاجم هویت ثبت‌شده و قانونی داخل سامانه دارد و دارای کلید خصوصی بلند مدت نیز است. لازم به ذکر است، یک مهاجم داخلی به کانال امن دسترسی دارد و به عنوان مثال می‌تواند پیام‌های مبادله‌شده در کانال امن بین دو هستار دیگر را به دست آورد.

۳.۲ مدل مهاجم دالو-یائو

مدل مهاجم دالو-یائو می‌تواند به دو صورت داخلی و خارجی دسته‌بندی شود. در این مدل، مهاجم توانایی شنود، تغییر، حذف پیام‌های مبادله‌شده، دستکاری پیام‌ها، ساخت پیام‌های جدید و تزریق پیام را بین هر دو شرکت‌کننده در طول ارتباطات دارد. در این مدل، مهاجم به برخی از کلیدهای خصوصی بلندمدت نیز دسترسی دارد. در واقع، این مدل یک شرکت‌کننده مخرب و یا مهاجمی را مدل می‌کند که هویت برخی از شرکت‌کننده‌ها را جعل نموده است. اکثر ابزارهای تحلیل امنیت مبتنی بر این مدل هستند [۳-۶].

۴.۲ مدل مهاجم بازیگر تسخیرشده

مدل مهاجم تسخیرشده به دشمن کنترل کامل شبکه و همچنین توانایی یادگیری کلیدهای بلندمدت عامل احراز اصالت را می‌دهد. این مدل مهاجم جهت مدلسازی حمله جعل هویت با کلید تسخیرشده مورد استفاده قرار می‌گیرد [۶-۸].

۵.۲ مدل‌های مهاجم AF و AFC

مدل‌های مهاجم AF و AFC متناظر با مهاجمی هستند که قادر به یادگیری همه کلیدهای خصوصی بلندمدت شرکت‌کننده‌ها بعد از یک جلسه است. این مدل برای تحلیل رازمانی پیش سوی کامل استفاده می‌شود. در مقایسه با مدل AF، مزیت مدل AFC این است که دشمن می‌تواند به کلیدهای خصوصی بلندمدت بعد از جلساتی که حتی فعالانه در آن‌ها شرکت نکرده است دسترسی یابد [۴-۶].

۶.۲ مدل‌های مهاجم BR و BPR

بلار و روگوی مدل امنیتی BR را برای پروتکل‌های برقراری کلید پیشنهاد دادند. مدل BR متناظر با یک مهاجم داخلی است. در این مدل، مهاجم می‌تواند کلیدهای جلسات را به دست آورد و در نتیجه حملات کلید معلوم را مدلسازی نماید. هم‌چنین، مدل BPR اصلاح‌شده و تعمیم‌یافته مدل BR است که به منظور اجرای پروتکل‌های تبادل کلید رمزگذاری‌شده و احراز اصالت استفاده می‌شود [۴-۶].

۷.۲ مدل مهاجم CK

کانتی و کروسزیک یک مدل مهاجم مبتنی بر مدل BR به نام CK2001 را معرفی نمودند. مدل CK2001 یک مدل امنیتی مشهور است که با نام

مدل امنیتی CK شناخته شده است. در مدل مهاجم CK، هدف تضمین این است که نشت اطلاعات مخصوص یک جلسه (مانند کلید جلسه یا اطلاعات موقت) هیچ تأثیری بر امنیت مقادیر مخفی سایر جلسات نداشته باشد. در مدل مهاجم CK، مهاجم می‌تواند تمام اطلاعات داخلی جلسه (شامل اطلاعات موقت) به جز کلیدهای بلندمدت را دریافت کند. همچنین، در این مدل، مهاجم می‌تواند حافظه داخلی یک هستار که شامل کلیدهای مخفی بلندمدت از قبیل کلیدهای خصوصی، کلیدهای به‌اشتراک گذاشته‌شده در جلسات مختلف و اطلاعات مخصوص جلسه است را یاد بگیرد [۴–۱۱].

۸.۲ مدل مهاجم eCK

لاماکچیا، لاتر و میتیاگین، مدل CK را با بیان درخواست افشای کلیدهای موقت مخصوص یک جلسه گسترش و تعمیم دادند و آن را با مدل مهاجم تعمیم‌یافته eCK نشان می‌دهند. در واقع، این مدل مهاجمی را نشان می‌دهد که می‌تواند به طور کامل شبکه را کنترل، هر پیامی را استراق سمع و پیام دلخواه خود را ارسال کند. در مدل eCK، مهاجم می‌تواند تنها مقادیر موقت و تصادفی تولیدشده در هر جلسه را برای اینکه کلید خصوصی بلندمدت مالک آن به خطر نیفتد، فاش کند. مدل eCK بازی‌ای را توصیف می‌کند که در این بازی مهاجم با عواملی که پروتکل را اجرا می‌کنند نیز تعامل دارد. مهاجم اگر بتواند یک کلید جلسه واقعی را از یک رشته بیت تصادفی تمایز و تشخیص دهد، برنده بازی است. همچنین، مهاجم می‌تواند به موارد زیر دست یابد [۵–۱۴]:

- کلیدهای خصوصی بلندمدت طرفین شرکت‌کننده
- کلیدهای جلسه طرفین شرکت‌کننده
- مقادیر تصادفی که توسط عوامل شرکت‌کننده در پروتکل تولید می‌شود.

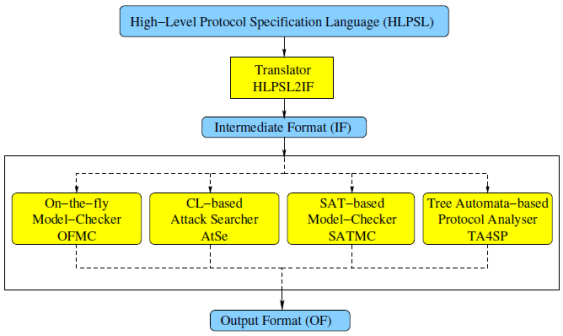
لازم به ذکر است، دشمن می‌تواند با حملات کانال جانبی و یا استفاده از مولدهای اعداد تصادفی ضعیف یا خراب توسط عوامل شرکت‌کننده به این مقادیر تصادفی دست یابد.

۳ ابزار تحلیل امنیت صوری خودکار اویسپا

ابزار اویسپا ابزاری دکمه‌ای جهت تحلیل امنیتی صوری خودکار پروتکل‌های رمزنگاری است. این ابزار نشان می‌دهد که آیا پروتکل احراز اصالت در برابر حملات مختلف، امن است و یا ناامن. معماری این ابزار در شکل ۱ نشان داده شده است [۱].

در ابزار اویسپا پروتکل به زبان سطح بالای HLPSL نوشته می‌شود. کد HLPSL با استفاده از مترجم HLPSL2IF به قالب میانی IF ترجمه می‌شود و سپس به عنوان ورودی به هر یک از چهار بررسی‌کننده وارد می‌شود. IF یک زبان سطح پایین است که مدل را جهت تحلیل با استفاده از چهار بررسی‌کننده آماده می‌کند. سپس IF قالب خروجی OF را ایجاد می‌نماید. بخش‌های مختلف OF در زیر توصیف شده است [۹].

- DETAILS: دلیل اینکه چرا پروتکل آزمایش‌شده امن است، دارای



شکل ۱. معماری ابزار اویسپا [۹]

حمله است و یا چرا تحلیل‌ها بی‌نتیجه‌اند، بیان می‌شود.

- SUMMARY: بیان می‌کند که آیا پروتکل ناامن، امن و یا تحلیل‌ها بی‌نتیجه است.
- GOAL: اهداف تحلیل انجام‌شده با ابزار اویسپا توضیح داده می‌شود.
- BACK-END: نام بررسی‌کننده‌های استفاده‌شده برای تحلیل‌ها که شامل OFMC، CL-AtSe، SATMC و TA4SP است، بیان می‌گردد.
- PROTOCOL: مشخصات HLPSL پروتکل در IF توضیح داده می‌شود.
- VULNERABILITY: آسیب‌پذیری و نظرات مرتبط در مورد امنیت پروتکل بیان می‌شود.

در این ابزار، چهار بررسی‌کننده به نام‌های OFMC، CL-AtSe، SATMC و TA4SP وجود دارد که در ذیل توضیح داده شده‌اند [۹]:

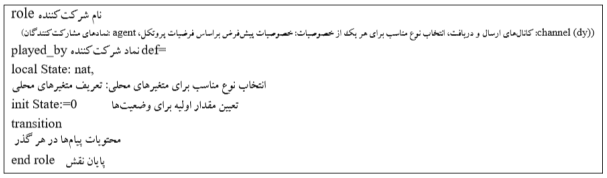
- بررسی‌کننده OFMC: این بررسی‌کننده با جست‌وجو در سامانه گذار توصیف‌شده با مشخصات IF به تأیید و یا عدم تأیید پروتکل‌ها می‌پردازد.
- بررسی‌کننده CL-AtSe: این بررسی‌کننده به تأیید و یا عدم تأیید پروتکل‌ها برای تعداد جلسات محدود پرداخته و از خصوصیات جبری عملگرهای رمزنگاری و چندین نوع بهینه‌سازی جهت کاهش و حذف افزونگی‌ها در اجرای پروتکل استفاده می‌کند. هم‌چنین در این بررسی‌کننده پیام‌های مهاجم با استفاده از متغیرها ذخیره می‌شوند.
- بررسی‌کننده SATMC: در این بررسی‌کننده، مجموعه‌ی حالت‌های نشان‌دهنده نقض ویژگی‌های امنیتی و رابط گذار مشخص‌شده با IF، یک دستور را تولید می‌کنند، سپس، این دستور به یک حل‌کننده SAT اعمال و در نتیجه هر برای مدل یافت‌شده یک حمله نشان داده می‌شود.
- بررسی کننده TA4SP: این بررسی‌کننده دانش مهاجم را از طریق یک ساختار درختی نشان می‌دهد.

۱.۳ زبان سطح بالای HLPSL

برای تحلیل پروتکل‌های امنیتی با استفاده از این ابزار، از زبان HLPSL استفاده می‌شود. این زبان مبتنی بر نقش است. از این رو، پروتکل امنیتی مبتنی بر تعریف نقش شرکت‌کنندگان پروتکل نوشته می‌شود. نقش‌ها مطابق خصوصیات جدول ۱ اقدام به برقراری ارتباط با یکدیگر می‌کنند.

جدول ۱. مؤلفه‌های مورد استفاده در زبان HLPSL [۸]

خصوصیات	کاربرد
agent	تعریف شرکت‌کننده
text	تعریف مقادیر یک‌بار مصرف
symmetric-key	کلید متقارن
public-key	کلید عمومی
hash-func	تابع چکیده‌ساز
nat	اعداد
bool	مقدار منطقی شامل true و false
protocol-id	تعریف برچسب



شکل ۲. تعریف نقش در محیط SPAN [۹]

این زبان جهت اخذ تأییدیه امنیتی پروتکل، از مدل مهاجم دالو-یائو در برابر حملات تکرار و فرددرمیانه استفاده می‌کند. از این رو، تمامی اطلاعات از طریق کانال دالو-یائو که یک کانال کاملاً ناامن است، ارسال می‌شود. پس، مهاجم به راحتی قادر به استراق‌سمع و تغییر پیام‌های ارسال‌شده است. در نتیجه، باید حفظ محرمانگی و عدم تغییر پیام‌های ردوبدل‌شده، عدم جعل هویت طرفین شرکت‌کننده در پروتکل و حفظ همزمانی مورد توجه قرار بگیرد [۹].

۲.۳ پیاده‌سازی پروتکل با استفاده از محیط SPAN

به منظور پیاده‌سازی و تحلیل امنیت یک پروتکل امنیتی با استفاده از ابزار اویسپا، لازم است که نقش‌های موجود در این محیط گام به‌گام مدل شوند [۹]:

- تعریف نقش شرکت‌کنندگان موجود در پروتکل
- تعیین اهداف امنیتی
- تعیین نقش نشست‌های پروتکل
- تعیین نقش محیط
- اجرای پروتکل

پیام‌های پروتکل بین طرفین شرکت‌کننده در پروتکل جابه‌جا می‌شوند. از این رو مهاجم می‌تواند به راحتی محتوای این پیام‌ها را دستکاری کند. قالب کلی تعریف نقش‌ها در پروتکل، مطابق با خصوصیات و فرضیات موجود در شکل ۲ نشان داده شده است.

```
goal
secrecy_of          شناسه ثابت استفاده شده برای مخفی بودن
authentication_on   شناسه ثابت استفاده شده برای احرازاصالت
end goal
```

شکل ۳. تعریف اهداف امنیتی در محیط SPAN [۹]

۳.۳ تعیین اهداف امنیتی در ابزار اویسپا

تمامی اهداف امنیتی از قبیل محرمانگی و احراز اصالت که در پروتکل وجود دارند، باید در قالب نقش هدف تعیین شوند. معمولاً در تحلیل امنیت پروتکل‌های احراز اصالت دو هدف امنیتی به کار برده می‌شود [۱، ۹]:

- احراز اصالت و تأیید هویت متقابل: به منظور حاصل شدن اطمینان از این‌که مهاجم از پیام‌های قبلی مجدداً استفاده نکرده باشد و برای اینکه تأیید هویت متقابل بین عوامل فرستنده و گیرنده صورت پذیرد، از عبارت‌های witness و request استفاده می‌شود.
- محرمانگی اطلاعات: این هدف با استفاده از عبارت secrecy_of مشخص می‌گردد.

جهت به دست آوردن تمامی خصوصیات و اهداف امنیتی از شناسه‌های ثابتی به صورت شکل ۳ استفاده می‌شود.

۴.۳ اجرای پروتکل‌های امنیتی با استفاده از ابزار اویسپا

به منظور اجرای پروتکل، باید نقش محیط که مجموعه‌ی تمامی نشست‌های پروتکل است، به‌صورت environment() فراخوانی شود. در محیط گرافیکی SPAN، با انتخاب گزینه‌های OFMC، CL-AtSe، SATMC و TA4SP نوع بررسی‌کننده تعیین می‌شود. بعد از انتخاب هر یک از این بررسی‌کننده‌ها دکمه اجرا را فشار داده، تا خروجی نشان داده شود. این خروجی تعیین‌کننده امن و یا ناامن بودن پروتکل بوده و با عبارت‌های SAFE و UNSAFE مشخص می‌شود. هم‌چنین در صورتی که پروتکل نسبت به دو حمله تکرار و فرددرمیانه آسیب‌پذیر باشد، شیوه نفوذ مهاجم و انجام حمله مشخص می‌شود [۹].

۵.۳ پروتکل ECCPWS

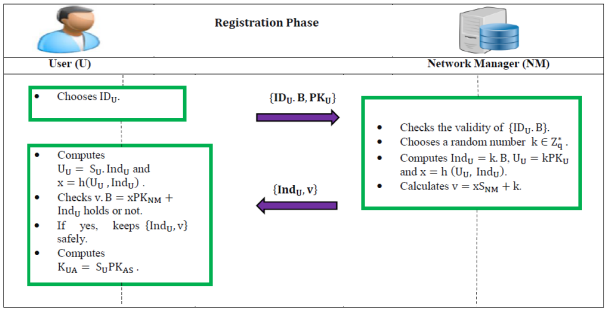
پروتکل ECCPWS شامل سه شرکت‌کننده به نام‌های NM، U، AS و هم‌چنین سه مرحله به نام‌های مرحله مقداردهی اولیه، ثبت‌نام و احراز هویت است، که در شکل‌های ۴ و ۵ به ترتیب، نشان داده شده است. تمامی علائم مورد استفاده در جدول ۲ بیان شده است.

۶.۳ تحلیل امنیتی صوری پروتکل ECCPWS با استفاده از ابزار اویسپا

در پیاده‌سازی طرح ECCPWS به زبان HLPSL، اطلاعات از مشخصات پروتکل به دست می‌آید. در HLPSL، نقش‌های اصلی و نقش‌های ترکیبی وجود دارد. نقش‌های اصلی، شرکت‌کنندگان مختلف را در پروتکل نشان

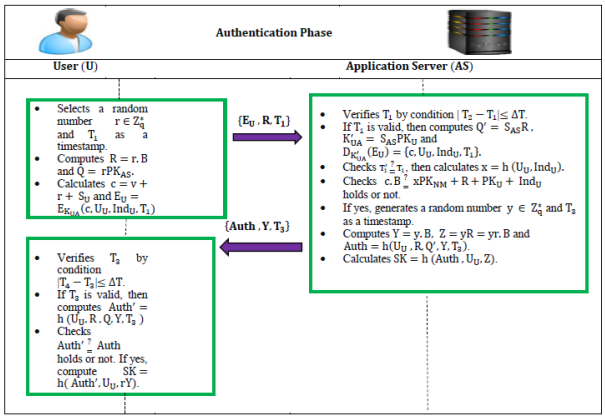
جدول ۲. علائم مورد استفاده در پروتکل ECCPWS

علائم	توضیح
NM	مدیر شبکه
AS	سرور کاربرد
U	کاربر
p	عدد اول در میدان متناهی
E	خم بیضوی با مرتبه p
G_E	گروه جمعی با مرتبه q
B	مولد گروه G_E
$h(\cdot)$	تابع چکیده‌ساز
$\mathbb{Z}_q$	اعداد صحیح $(0, 1, \ldots, q-1)$ در میدان متناهی
$\mathbb{Z}_q^*$	$\mathbb{Z}_q - \{0\}$
ID_U	شناسه کاربر U_i
SK	کلید جلسه
ΔT	حداکثر تأخیر قابل قبول ارتباط
K_{UA}	کلید مخفی به اشتراک گذاشته‌شده بین کاربر U و AS
(S_{AS}, PK_{AS})	زوج کلید خصوصی/عمومی AS
(S_{NM}, PK_{NM})	زوج کلید خصوصی/عمومی NM
(S_U, PK_U)	زوج کلید خصوصی/عمومی U
T_1, T_2, T_3, T_4	مهرهای زمانی جاری
$(\cdot)$	عملیات ضرب نقطه‌ای خم بیضوی

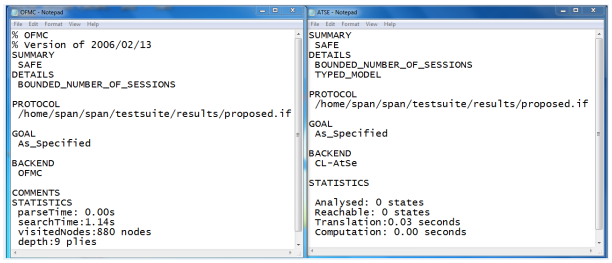


شکل ۴. مرحله ثبت‌نام پروتکل ECCPWS [۱]

می‌دهند (یعنی، نقش‌های U، NM و AS در پروتکل ECCPWS). نقش‌های ترکیبی شامل جلسه، هدف و محیط هستند، که سناریوهای مختلف از نقش‌های اصلی را نشان می‌دهند. مشخصات نقش‌های اصلی U، NM و AS پروتکل ECCPWS به زبان HLPSL به ترتیب در پیوست نشان داده شده است. هم‌چنین مشخصات نقش‌های ترکیبی جلسه، محیط و هدف در پیوست نشان داده شده است. نقش‌های محیط، هدف و جلسه شامل دانش مهاجم، مقادیر ثابت مورد استفاده، اهداف، مشخصات جلسات و همه مقادیر شرکت‌کنندگان است. نتایج تحلیل امنیتی پروتکل



شکل ۵. مرحله احراز هویت پروتکل ECCPWS [۱]



شکل ۶. نتایج ارزیابی طرح ECCPWS در ابزار اویسپا با استفاده از بررسی‌کننده‌های OFMC و ATSE [۱]

ECCPWS در شکل ۶ نشان داده شده .است نتایج به دست آمده، تضمین می‌کند که طرح مورد نظر در برابر حملات مختلف از قبیل حملات تکرار و فرددرمیانه امن است [۱].

۴ ابزار تحلیل امنیت صوری خودکار سایتر

ابزار سایتر، یک ابزار صوری خودکار برای تحلیل پروتکل‌های امنیتی برای تشخیص حملات است. در این ابزار، پروتکل‌ها مبتنی بر تعریف نقش هستند و زبان آن SPDL است. امنیت در این ابزار با بهره‌برداری از ادعاهای امنیتی بررسی می‌شود. هم‌چنین، نمودار سناریوی حمله برای هر حمله مطابق با هر ادعا تولید می‌شود. این ابزار یک واسط کاربر گرافیکی مطابق شکل ۷ ارائه می‌نماید. حتی، حالتی را که حمله‌ای به پروتکل امنیتی وارد نمی‌شود، مشخص می‌کند [۱].

سایتر یک ابزار برای ارزیابی صوری پروتکل‌های رمزنگاری مطابق با فرضیات رمزنگاری کامل است. یعنی، هر تابع رمزنگاری که مورد استفاده قرار می‌گیرد، امن در نظر گرفته می‌شود. هم‌چنین، مهاجم اگر کلید رمزگشایی را در اختیار داشته باشد، می‌تواند پیام‌های مبادله‌شده را رمزگشایی کند. ابزار سایتر امکانات گرافیکی مناسبی را به ما ارائه می‌دهد. دو فرض بسیار مهم در مدل این ابزار مطرح است. اولین فرض این است که، مدل رمزنگاری آن جعبه سیاه بوده و دومین فرض این است که پیام‌های ارسال‌شده به صورت انتزاعی هستند، یعنی مهاجم یا می‌تواند کل پیام را بفهمد و یا از پیام ارسالی چیزی را متوجه نشود. [۱۰–۱۶].

۱.۴ رخدادهای دریافت و ارسال

رخدادهای recv و send به ترتیب دریافت و ارسال پیام را نشان می‌دهند. توجه داشته باشید، در بسیاری از فایل‌های توصیف پروتکل ممکن است کلمه کلیدی read نیز استفاده شود که این مورد منسوخ شده است و می‌تواند با recv جایگزین شود. در اکثر موارد، هر رخداد send یک رخداد متناظر recv خواهد داشت. چنین تناظری با دادن برچسب یکسان به چنین رخدادهایی که با یک زیرنویس معلوم می‌شوند معین می‌گردد. اگر یک رخداد send یا recv هیچ رخداد متناظری نداشته باشد، سایتر یک هشدار در خروجی نشان می‌دهد. جهت نادیده گرفتن این هشدار، برچسب می‌تواند با علامت ! نشان داده شود. به طور مثال خواهیم داشت [۷، ۱۰]:

send_1!(I,I,Leak To Adversary)

۲.۴ ویژگی‌های امنیتی به عنوان ادعای رخدادها

ویژگی‌های امنیتی بخش ضروری از یک پروتکل امنیتی هستند و یک پروتکل امنیتی نباید بدون در نظرگرفتن ویژگی‌های دقیقی که قرار است رعایت کند، طراحی شود. ویژگی‌های امنیتی این ابزار با معرفی نقش رخداد به‌نام ادعای رخداد بیان می‌شوند. به طور مثال، محرمانه بودن یک عبارت بدین معنی است که اگر یک شرکت‌کننده با شرکت‌کنندگان دیگر ارتباط برقرار کند، عبارت مورد نظر باید در هر حالتی مخفی بماند. بنابراین، در این ابزار ادعاهای امنیتی زیادی از قبیل Alive، Nisynch، secret، weakagree و غیره وجود دارد که برخی از آن‌ها در ذیل به اختصار توضیح داده می‌شوند و خواننده علاقه‌مند برای جزئیات بیشتر می‌تواند به [۷، ۱۰] مراجعه نماید.

محرمانگی محرمانگی بیان می‌کند که اطلاعات خاص نباید توسط مهاجم فاش شود، حتی اگر این اطلاعات از طریق یک شبکه نامعتبر ارسال شود. ادعای رخداد محرمانگی به صورت claim(R;secret;S) نوشته و در نقش R اجرا می‌شود. این رخداد نشان می‌دهد که نقش R ادعا می‌کند که مقدار S باید به عنوان یک مقدار مخفی و ناشناخته برای مهاجم در نظر گرفته شود [۷، ۱۰].

برخط بودن یک پروتکل برای آغازگر A برخط بودن شرکت‌کننده دیگر B را تضمین می‌کند، اگر A (که به عنوان آغازگر عمل می‌کند) یک اجرای پروتکل را که ظاهراً قبلاً با پاسخ‌دهنده B اجرا کرده است، کامل کند. توجه داشته باشید که B ممکن است لزوماً باور نداشته باشد که او پروتکل را با A اجرا کرده است. هم‌چنین، B ممکن نیست پروتکل را اخیراً اجراکرده باشد [۷، ۱۰].

توافق ضعیف یک پروتکل برای آغازگر A توافق ضعیف با دیگر شرکت‌کننده B را تضمین می‌کند، اگر A یک اجرای پروتکل را ظاهراً با پاسخ‌دهنده B کامل کند و ظاهراً B قبلاً با A پروتکل را اجرا کرده است. توجه داشته باشید که B ممکن است لزوماً به عنوان پاسخ‌دهنده عمل نکرده باشد. بسیاری از پروتکل‌ها برخط بودن را تضمین می‌کنند ولی برای تضمین توافق ضعیف با شکست مواجه می‌شوند. بنابراین، A باور دارد که او پروتکل را با B اجرا کرده است، ولی B معتقد نیست که او پروتکل را با A اجرا کرده است و B باوردارد که او پروتکل را با مهاجم اجرا کرده است [۷، ۱۰].

توافق غیر یک به یک یک پروتکل این ادعا را برای آغازگر A، پاسخ‌دهنده B و مجموعه‌ای از اطلاعات ds (که ds مجموعه‌ای از متغیرهای آزاد است که در توصیف پروتکل ظاهر می‌شود) تضمین می‌کند، اگر A (به عنوان آغازگر عمل می‌کند) یک اجرای پروتکل را ظاهراً با پاسخ‌دهنده B کامل کند و B به عنوان پاسخ‌دهنده ظاهراً قبلاً با A پروتکل را اجرا کرده است و هم‌چنین دو شرکت‌کننده بر روی مقادیر داده متناظر در ds توافق کرده‌اند. لازم به ذکر است، این مورد تضمین نمی‌کند که یک رابطه یک‌به‌یک بین اجرای A و B وجود دارد. A ممکن است باور کند که او دو اجرا را کامل کرده است، درحالی‌که B فقط در یک اجرا شرکت کرده است [۷، ۱۰].

توافق یک به یک هنگامی که بخواهیم بیان کنیم که یک رابطه یک‌به‌یک بین دو اجرای شرکت‌کنندگان وجود دارد از اصطلاح توافق یک به یک و یا توافق استفاده می‌شود. این ارتباط یک‌به‌یک ممکن است در پروتکل‌های مالی زیاد مطرح باشد. یک پروتکل برای آغازگر A ادعای امنیتی توافق را با پاسخ‌دهنده B بر روی مجموعه‌ای از اطلاعات ds تضمین می‌کند، اگر A (به عنوان آغازگر عمل می‌کند) یک اجرای پروتکل را ظاهراً با پاسخ‌دهنده B تکمیل کند، B به عنوان پاسخ‌دهنده قبلاً پروتکل را ظاهراً با A اجرا کرده است و دو شرکت‌کننده بر روی مقادیر داده مربوط به همه‌ی متغیرها در ds توافق کرده‌اند و هر اجرای A متناظر با یک اجرای منحصربه‌فرد B است. پروتکل‌های محدودی وجود دارند که به توافق غیر یک به یک دست می‌یابند، ولی به توافق یک به یک دست نمی‌یابند. شرکت‌کننده A می‌پندارد که B سعی دارد که دو جلسه را با او ایجاد کند، درحالی‌که B فقط در تلاش برای ایجاد یک اجرای واحد است [۷، ۱۰].

تازگی و جدید بودن معنی «اخیر» به موقعیت و شرایط بستگی دارد.گاهی اوقات ما آن را طول مدت اجرای A و گاهی اوقات آن را به معنای حداکثر t واحد زمانی قبل از اینکه A اجرای خود را کامل کند، در نظر می‌گیریم. ضمناً مقدار t، زمان لازم برای احراز اصالت نیز نامیده می‌شود و مطمئناً مقدار t وابسته به پیاده‌سازی خواهد بود. اصطلاحات برخط بودن اخیر، توافق ضعیف اخیر، توافق غیر یک به یک اخیر و توافق اخیر برای اشاره به اخیر بودن اجرای B است. برای مثال، پروتکل یک مرحله‌ای زیر را در نظر بگیرید، که k_ab یک کلید مشترک بین A و B است.

Message 1. $A \rightarrow B : \{A, k\}_{k_{ab}}$

این پروتکل به B هیچ ضمانتی برای تازگی نمی‌دهد. زیرا پیام شامل اطلاعاتی نیست که B بداند تازه و جدید است. این پروتکل را می‌توان جهت دستیابی به توافق غیریک به یک با اضافه‌کردن یک مهرزمانی به پیام اصلاح کرد [۷، ۱۰].

همزمان‌سازی برای تعریف همزمان‌سازی یا همگام‌سازی، به سازوکاری نیاز است که مشخص کند که کدام عوامل نقش‌ها را اجرا می‌کنند. پروتکل می‌تواند شامل تعدادی نقش باشد که چندین مرتبه اجرا شود. بنابراین، فعالیت‌های مشخصی در یک پروتکل به نقش‌ها اختصاص داده می‌شود. به طور مثال، در بازی تئاتر، در اجراهای مختلف، یک بازیگر می‌تواند نقش‌های مختلفی را ایفا کند. همچنین، نقش یکسانی را می‌توان برای اجراهای مختلف بازی توسط بازیگران مختلف به عهده گرفت [۷، ۱۰].

دردسترس‌پذیری زمانی‌که یک ادعا در شرح پروتکل نوشته می‌شود، ابزار سایتر بررسی خواهدکردکه آیا اصلاً می‌توان به این ادعا دست یافت یا نه. اگر و تنها اگر اثر و ردپایی وجود داشته باشد که این ادعا در آن رخ دهد، این ادعا صحیح است. همچنین، دردسترس‌پذیری می‌تواند برای بررسی اینکه آیا خطای آشکاری در توصیف پروتکل وجود ندارد، مفید باشد [۷، ۱۰].

۳.۴ پروتکل‌های کمکی در ابزار سایتر

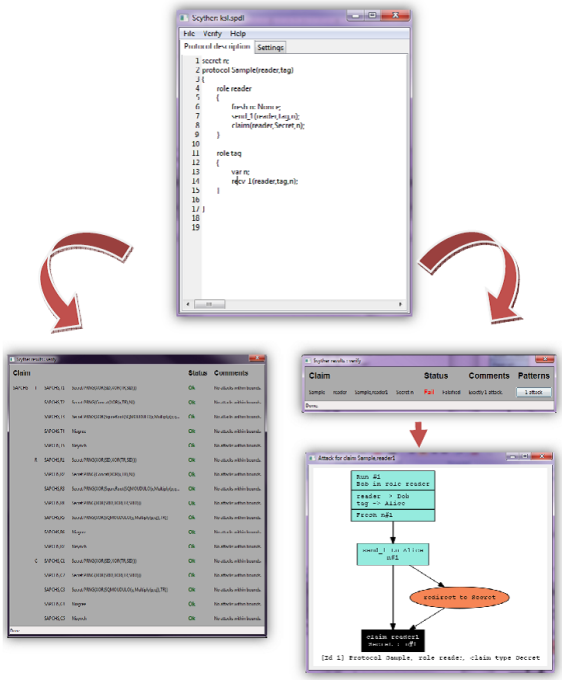
ممکن است قبل از نام پروتکل علامت «@» اضافه شود. این هیچ تأثیری بر روی مدل پروتکل و نتیجه تجزیه و تحلیل‌ها ندارد. با استفاده از این، پروتکل را به عنوان یک «پروتکل کمکی» علامت‌گذاری می‌کنیم. چنین پروتکل‌هایی اغلب برای مدلسازی قابلیت‌های بیشتر دشمن استفاده می‌شوند [۱۰].

۴.۴ اجرای پروتکل‌های امنیتی در ابزار سایتر

بعد از مشخص نمودن اهداف امنیتی و نقش‌ها، با اجرای دستور ارزیابی تحلیل پروتکل مورد نظر آغاز می‌شود. مشاهده می‌شود که خروجی این ابزار مانند ابزار اویسپا شامل دو حالت است: حالت اول، زمانی است که حمله‌ای بر علیه پروتکل مورد نظر شناسایی شود، که سناریوی گرافیکی حمله شناسایی‌شده نیز مشخص می‌شود. حالت دوم زمانی است که پروتکل مورد نظر توسط این ابزار، امن تشخیص داده شود و ادعاهای امنیتی آن تأیید گردند [۱۰–۱۷].

۵.۴ حملات چند پروتکلی در ابزار سایتر

روش‌های صوری و ابزارهای متناظر برای تجزیه و تحلیل پروتکل‌های امنیتی در سال‌های اخیر توسعه یافته‌اند. این روش‌ها عموماً محدود به تأیید پروتکل‌هایی هستند که به صورت مجزا اجرا می‌شوند. برای پروتکلی که بر روی یک شبکه نامعتبر استفاده می‌شود، مدل‌های صوری معمولاً فرض می‌کنند که تنها یک پروتکل در شبکه اجرا می‌شود. با این حال، این فرض که یک پروتکل تنها پروتکلی است که بر روی شبکه غیرقابل اعتماد اجرا می‌شود، واقع‌بینانه نیست. متأسفانه، هنگامی که چندین پروتکل در یک شبکه نامعتبر استفاده می‌شوند، مشکل تأیید ادعاهای امنیتی به طور قابل ملاحظه‌ای سخت‌تر می‌گردد. دلیل این واقعیت این است که ویژگی‌ها و ادعاهای امنیتی ترکیبی نیستند. ممکن است که دو پروتکل در هنگام اجرا به صورت مجزا امن باشند، ولی در برابر حملات جدید در صورت

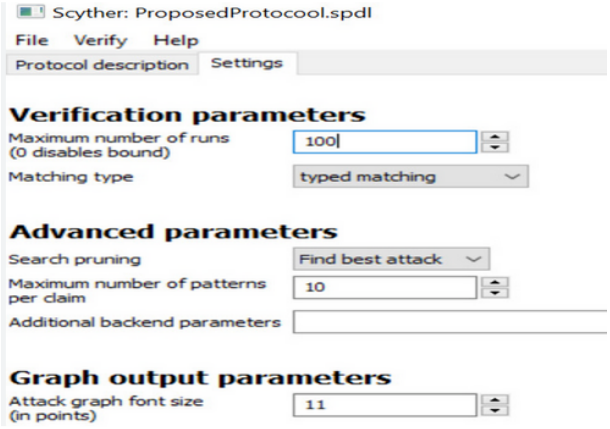


شکل ۷. محیط ابزار تحلیل امنیت سایتر [۱۰]

استفاده ترکیبی در همان شبکه آسیب‌پذیر باشند. حمله‌ای که لزوماً شامل بیش از یک پروتکل است، حمله چند پروتکلی نامیده می‌شود. وجود چنین حملاتی ابتدا توسط کلسیی، اسشنیر و واگنر معرفی شد [۷، ۱۰]. آن‌ها رویه‌ای را طراحی کردند که از یک پروتکل امن شروع می‌شود. سپس نشان دادند که امکان ساخت یک پروتکل اختصاصی وجود دارد، به طوری که پروتکل دوم امن است و زمانی‌که این دو پروتکل در یک شبکه اجرا می‌شوند، مهاجم می‌تواند پیام‌های پروتکل دوم را برای حمله به پروتکل اول استفاده کند [۱]. اگر همه پروتکل‌هایی که از شبکه و زیرساخت کلیدی یکسان استفاده می‌کنند، الزامات خاصی را برآورده کنند، ترکیب خواص امنیتی تضمین شده است [۱]. در این صورت برای اثبات صحت سامانه، اثبات صحت پروتکل‌ها به صورت مجزا کافی است. به طور شهودی، یک تک پروتکل به معنای مجموعه‌ای از نقش‌هاست که از طریق رابطه تعاملی به هم متصل می‌شوند. اگر ما دو پروتکل را به هم متصل کنیم، لزوماً همه نقش‌ها به هم متصل نمی‌شوند [۷، ۱۰].

- نقش‌های به‌هم‌متصل: دو نقش را به هم متصل می‌نامیم اگر تحت رابطه هم ارزی معادل باشند.
- پروتکل‌های تک/چندگانه: P به عنوان تعداد کلاس‌ها نشان داده می‌شود. اگر تعداد برابر با یک باشد گفته می‌شود P حاوی یک پروتکل واحد است. اگر بیش از یک باشد، گفته می‌شود این حاوی چندین پروتکل است. اگر مشخصات نقش‌های دو یا چند پروتکل را با مجموعه‌ای از نقش‌های غیرمتصل ترکیب کنیم، نتیجه شامل پروتکل‌های چندگانه می‌شود [۷، ۱۰].

از ابزار سایتر می‌توان برای بررسی حملات به پروتکل‌های چندگانه استفاده نمود. این حملات به تعامل بین زیرپروتکل‌های مختلف بستگی



شکل ۸. تنظیمات مدل دشمن در نسخه استاندارد ابزار سایتر [۷]

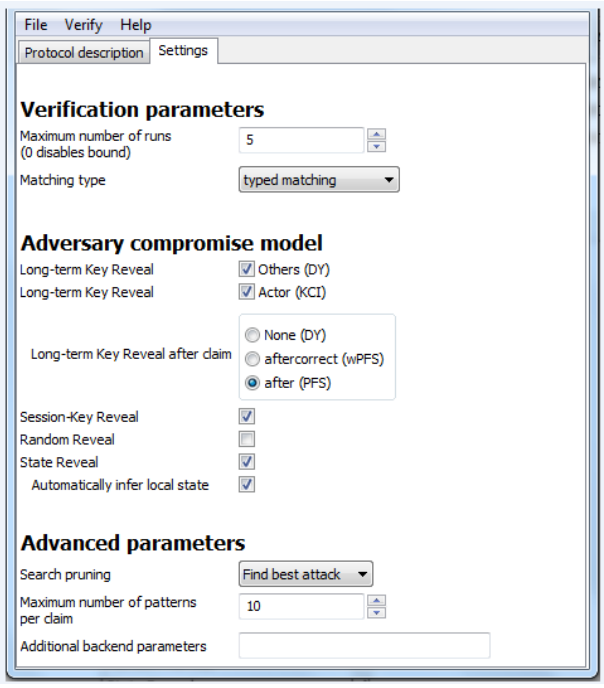
دارد. گاهی اوقات دشمن می‌تواند از پیام‌ها یا اجزای پیام یک پروتکل برای حمله به پروتکل دیگر استفاده کند. ساده‌ترین راه برای بررسی حملات چندپروتکلی در ابزار سایتر ترکیب توصیف دو پروتکل در یک فایل واحد است. فایل به دست‌آمده محیطی را مدلسازی می‌کند که هر دو پروتکل در آن اجرا شوند. درنهایت، از ابزار سایتر جهت ارزیابی ادعاهای موجود در فایل ترکیب‌شده می‌توان استفاده نمود [۷، ۱۰].

۶.۴ نسخه‌های ابزار سایتر

این ابزار از دو نسخه پشتیبانی می‌کند: نسخه استاندارد و نسخه پشتیبان مدل کلید تسخیرشده. نسخه دوم سایتر تمام پروتکل‌های موجود برای نسخه استاندارد را نیز پشتیبانی می‌کند. علاوه بر این، بسیاری از پروتکل‌هایی که در برابر دشمنان تسخیرشده انعطاف‌پذیری بیشتری دارند در این نسخه گنجانده شده است. بیسین و همکاران قدرت این ابزار را به منظور بررسی سناریوهای دشمن بررسی کردند [۸-۱۸]. این نسخه از سایتر، چارچوبی را برای مدل‌سازی دشمنان از جمله دشمن دالو-یائو تا دشمنان قدرتمندتر ارائه می‌کند. تنظیمات مدل دشمن را مطابق شکل ۸ به DY و KCI در بخش کلید بلندمدت تغییر می‌دهیم تا به مدل دشمن CK دست یابیم. گزینه‌های تنظیمات مدل دشمن در نسخه‌های استاندارد و پشتیبان مدل کلید تسخیرشده می‌تواند در شکل ۸ و ۹ مشاهده شود [۱].

۷.۴ تحلیل امنیتی صوری پروتکل ECCPWS با استفاده از ابزار سایتر

در این بخش، طرح ECCPWS با استفاده از زبان SPDL مدل شده‌است تا ادعاهای طراحان پروتکل را ارزیابی نماید. دیده شده است که طرح پیشنهادی مبتنی بر تعریف نقش شرکت‌کنندگان پروتکل از قبیل NM، AS و U مدل می‌شود و سپس این نقش‌ها با یکدیگر از طریق کانال‌های recv و send ارتباط برقرار می‌کنند. پیاده‌سازی SPDL پروتکل ECCPWS و نتایج ارزیابی آن در پیوست و شکل‌های ۱۰، ۱۱ و ۱۲ به ترتیب، نشان داده شده است. نتایج نشان می‌دهد که ابزار سایتر نتوانسته



شکل ۹. تنظیمات مدل دشمن در نسخه پشتیبان مدل کلید تسخیرشده ابزار سایتر [۷]

Claim				Status	Comments
proposed	U	Proposed, U2	Secret r	OK	No attacks within bounds.
		Proposed, U3	Secret idu	OK	No attacks within bounds.
		Proposed, U4	Secret B	OK	No attacks within bounds.
		Proposed, U5	Secret t	OK	No attacks within bounds.
		Proposed, U6	Secret y	OK	No attacks within bounds.
		Proposed, U7	Alive	OK	No attacks within bounds.
		Proposed, U8	Weakagree	OK	No attacks within bounds.
		Proposed, U9	Niagree	OK	No attacks within bounds.
		Proposed, U10	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM1	Secret t	OK	No attacks within bounds.
		Proposed, NM2	Secret idu	OK	No attacks within bounds.
		Proposed, NM3	Secret B	OK	No attacks within bounds.
		Proposed, NM4	Alive	OK	No attacks within bounds.
		Proposed, NM5	Weakagree	OK	No attacks within bounds.
		Proposed, NM6	Niagree	OK	No attacks within bounds.
		Proposed, NM7	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS1	Secret B	OK	No attacks within bounds.
		Proposed, AS2	Secret idu	OK	No attacks within bounds.
		Proposed, AS3	Secret y	OK	No attacks within bounds.
		Proposed, AS4	Secret t	OK	No attacks within bounds.

شکل ۱۰. نتایج ارزیابی ادعاهای امنیتی کاربر در طرح ECCPWS با استفاده از ابزار سایتر

است ضعف و یا حمله‌ای را برای پروتکل ECCPWS بیابد.

۵ ابزار تحلیل امنیت صوری خودکار پرووریف

پرووریف ابزاری جهت تأیید امنیت خودکار پروتکل‌های رمزنگاری است. شرط‌ها و قضایایی در این ابزار به اثبات خصوصیات محرمانه بودن و احراز اصالت متقابل افراد شرکت‌کننده در پروتکل کمک می‌کنند. این ابزار نیز از مدل مهاجم دالو-یائو به عنوان یک مدل مهاجم مبتنی بر فضای پیام نامحدود پشتیبانی می‌کند. ابزار پرووریف دو نوع فایل را به عنوان ورودی می‌پذیرد. یک فایل قضایای Horn و دیگری مجموعه‌ای از محاسبات Pi است. هم‌چنین پروتکل با تعداد نامحدودی جلسه تحلیل می‌شود. در این ابزار نیازی به مدل‌سازی صریح مهاجم نیست. پرووریف

Claim				Status	Comments
proposed	U	Proposed, U1	Secret r	OK	No attacks within bounds.
		Proposed, U2	Secret idu	OK	No attacks within bounds.
		Proposed, U3	Secret B	OK	No attacks within bounds.
		Proposed, U4	Secret t	OK	No attacks within bounds.
		Proposed, U5	Secret y	OK	No attacks within bounds.
		Proposed, U6	Alive	OK	No attacks within bounds.
		Proposed, U7	Weakagree	OK	No attacks within bounds.
		Proposed, U8	Niagree	OK	No attacks within bounds.
		Proposed, U9	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM2	Secret t	OK	No attacks within bounds.
		Proposed, NM3	Secret idu	OK	No attacks within bounds.
		Proposed, NM4	Secret B	OK	No attacks within bounds.
		Proposed, NM5	Alive	OK	No attacks within bounds.
		Proposed, NM6	Weakagree	OK	No attacks within bounds.
		Proposed, NM7	Niagree	OK	No attacks within bounds.
		Proposed, NM8	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS1	Secret B	OK	No attacks within bounds.
		Proposed, AS2	Secret idu	OK	No attacks within bounds.
		Proposed, AS3	Secret y	OK	No attacks within bounds.
		Proposed, AS3	Secret t	OK	No attacks within bounds.

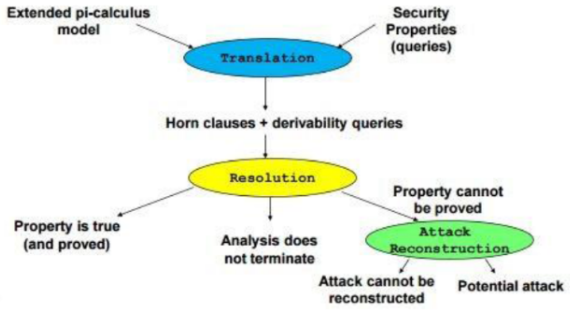
شکل ۱۱. نتایج ارزیابی ادعاهای امنیتی سرور NM در طرح ECCPWS با استفاده از ابزار سایتر

Claim				Status	Comments
proposed	U	Proposed, U1	Secret r	OK	No attacks within bounds.
		Proposed, U2	Secret idu	OK	No attacks within bounds.
		Proposed, U3	Secret B	OK	No attacks within bounds.
		Proposed, U4	Secret t	OK	No attacks within bounds.
		Proposed, U5	Secret y	OK	No attacks within bounds.
		Proposed, U6	Alive	OK	No attacks within bounds.
		Proposed, U7	Weakagree	OK	No attacks within bounds.
		Proposed, U8	Niagree	OK	No attacks within bounds.
		Proposed, U9	Nisynch	OK	No attacks within bounds.
	NM	Proposed, NM1	Secret t	OK	No attacks within bounds.
		Proposed, NM2	Secret idu	OK	No attacks within bounds.
		Proposed, NM3	Secret B	OK	No attacks within bounds.
		Proposed, NM4	Alive	OK	No attacks within bounds.
		Proposed, NM5	Weakagree	OK	No attacks within bounds.
		Proposed, NM6	Niagree	OK	No attacks within bounds.
		Proposed, NM7	Nisynch	OK	No attacks within bounds.
	AS	Proposed, AS2	Secret B	OK	No attacks within bounds.
		Proposed, AS3	Secret idu	OK	No attacks within bounds.
		Proposed, AS4	Secret y	OK	No attacks within bounds.
		Proposed, AS5	Secret t	OK	No attacks within bounds.

شکل ۱۲. نتایج ارزیابی ادعاهای امنیتی سرور AS در طرح ECCPWS با استفاده از ابزار سایتر

ابزاری از نوع خط فرمان است، که می‌تواند با استفاده از فرمان proverif./ <filename> [options] اجرا شود [۱۹–۲۲].

<filename> فایل ورودی و [options] مربوط به مقادیر خط فرمان هستند. پروورِیف می‌تواند فایل‌های ورودی رمزگذاری‌شده را با چندین زبان استفاده کند. فایل‌هایی از نوع محاسبات Pi با پسوند pv. مشخص می‌شوند. هدف اصلی این ابزار درستی‌یابی پروتکل‌های رمزنگاری است. مهاجم مورد استفاده در این ابزار کنترل کاملی بر روی کانال‌های ارتباطی دارد. مهاجم می‌تواند پیام‌های مبادله‌شده را بخواند، تغییر دهد، حذف کند، به شبکه تزریق نماید و داده‌ها را دستکاری کند. لازم به ذکراست، فقط شرکت‌کنندگان معتمد در این ابزار مدل می‌شوند. این ابزار به طور خودکار بررسی خصوصیات امنیتی ادعاشده را انجام می‌دهد. در این‌جا فرض‌می‌شود که رمزنگاری از نوع کامل است. یعنی اینکه، مهاجم تنها قادر به انجام عملیات رمزنگاری با داشتن کلیدهای موردنیاز است. این ابزار ویژگی‌های امنیتی محرمانگی و احراز اصالت را مبتنی بر فرضیاتی که اولیه‌های رمزنگاری ایده‌ال و امن هستند، تأیید می‌کند. در مدل دالو–یائو مورد استفاده در این ابزار، مهاجم بدون دانستن کلیدهای موردنیاز،

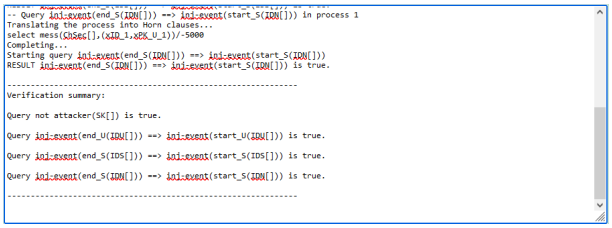


شکل ۱۳. ساختار ابزار پروورِیف [۱۱]

نمی‌تواند از پیام‌های رمزشده چیزی را به دست آورد و در صورتی که یک ویژگی قابل اثبات نباشد، این ابزار یک حمله را نشان می‌دهد. همچنین روند حمله نیز مشخص می‌گردد. در واقع، این ابزار با استفاده از فضای پیام نامحدود خود، برای تعداد نامحدودی اجرا انجام می‌شود. مقاومت پروتکل در برابر حملات و محافظت از نشت کلید جلسه با این ابزار بررسی شده است. در این ابزار مدل پروتکل در سه بخش نوشته می‌شود. بخش فراخوانی به عنوان بخش اول، برای توصیف اصول رمزنگاری مانند انواع تعریف‌شده توسط کاربر، نام‌های آزاد و توابع نمادین استفاده می‌شود. نام‌های آزاد به طور پیش‌فرض برای مهاجم شناخته شده‌اند. بنابراین، برای مخفی‌نمودن این مقادیر باید به صورت «private» اعلان شوند. همچنین، کانال امن، کانال عمومی، نقطه پایه P بر روی خم بیضوی، شناسه کاربر، رمزعبور کاربر در قسمت اعلان اعلام می‌شوند. همچنین توابع اعلان مانند عملیات جمع پیمانه‌ای، توابع چکیده‌ساز، عملیات ضرب نقطه‌ای بر روی خم بیضوی و غیره در این قسمت بیان می‌شوند. فرایند به عنوان بخش دوم، از ماکروهایی تشکیل شده است که برای تعریف فرایندهای فرعی استفاده می‌شوند. بخش اصلی به عنوان بخش نهایی است و برای اجرای پروتکل استفاده می‌گردد [۱]. پس از اجرای موفقیت‌آمیز بخش اصلی، خروجی دو حالت را نشان می‌دهد. اگر حمله‌ای امکان‌پذیر نباشد، RESULT [query] is true نشان داده می‌شود و اگر یک حمله شناسایی شود، RESULT [query] is false نشان داده می‌شود. همچنین، اگر مقدار X برای مهاجم قابل‌دسترس نباشد، پیام RESULT not attacker (X) is true نشان داده می‌شود. درنهایت، ساختار کلی ابزار پروورِیف در شکل ۱۳ نشان داده شده است [۱۷–۲۲].

۱.۵ تحلیل امنیتی صوری پروتکل ECCPWS با استفاده از ابزار پروورِیف

ابزار پروورِیف یک ابزار شناخته‌شده جهت تأیید امنیت طرح‌های احراز هویت است. در این بخش، از این ابزار برای تحلیل امنیت و بررسی خصوصیات احراز هویت و محرمانه بودن طرح ECCPWS استفاده شده است. در این ابزار اصول اولیه رمزنگاری از قبیل توابع چکیده‌ساز، توابع رمزنگاری و رمزگشایی تعریف می‌شوند. این ابزار می‌تواند ویژگی‌هایی از قبیل محرمانه بودن و احراز هویت متقابل را نیز اثبات کند. در این ابزار، ما ابتدا کانال‌ها را به صورت نام‌های آزاد مشخص می‌کنیم که



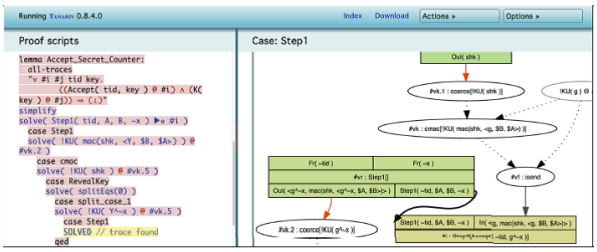
شکل ۱۴. نتایج ارزیابی طرح ECCPWS در ابزار پروویریف

در شکل ۱۴ کانال ChPub یک کانال ارتباطی عمومی بین U، AS و ChSec یک کانال ارتباطی خصوصی و امن بین U و NM است. در واقع، ما در پروتکل پیشنهادی از دو کانال استفاده نمودیم. یک کانال ثبت‌نام ChSec بین U و NM و یک کانال احراز هویت ChPub بین U و AS. هم‌چنین، تمام متغیرهایی که به صورت آزاد تعریف می‌شوند برای مهاجم شناخته شده‌اند. برای ارزیابی یک پروتکل با استفاده از این ابزار، مدل پروتکل باید در سه بخش نوشته شود. وظیفه اصلی ابزار پروویریف اثبات ویژگی دسترسی‌پذیری است. not attacker (X) برای بررسی ویژگی دسترسی‌پذیری متغیر X استفاده می‌شود. این ابزار تلاش می‌کند که اثبات کند آیا یک متغیر برای مهاجم شناخته شده است یا نه. در اینجا، سه فرآیند فرعی مختلف برای کاربر، AS و NM وجود دارد که در پیوست نشان داده شده است. این فرایندهای فرعی در حین اجرای فرآیند اصلی به عنوان ماکرو گسترش خواهند یافت. فرایند و بخش اصلی، شروع و پایان فرایند شرکت‌کنندگان پروتکل را تعریف می‌کند. اجرای فرایندهای شرکت‌کنندگان به صورت موازی است. بعد از اجرای موفقیت‌آمیز فرآیند اصلی، خروجی دو حالت را نشان می‌دهد. اگر حمله‌ای امکان‌پذیر نباشد، خروجی RESULT[query] is true را نمایش می‌دهد و اگر حمله‌ای یافت شود، خروجی RESULT[query] is false را نمایش می‌دهد. هم‌چنین، اگر مقدار متغیر X در دسترس مهاجم قرار نداشته باشد، ابزار پروویریف پیام RESULT not attacker (X) is true را نمایش می‌دهد. شکل ۱۴ نتیجه تحلیل امنیت پروتکل پیشنهادی ECCPWS را با استفاده از ابزار پروویریف نشان می‌دهد. صحت طرح پیشنهادی به طور موفقیت‌آمیز اثبات شده است و امنیت آن به دلیل ناموفق بودن دسترسی مهاجم به کلید جلسه SK تأیید شده است.

۶ ابزار تحلیل امنیت صوری خودکار تامارین

این ابزار نیز از تجزیه و تحلیل خودکار پروتکل‌های امنیتی پشتیبانی می‌کند و دارای زبانی رسا جهت مدل نمودن مهاجم است. ابزار تامارین به زبان برنامه‌نویسی هاسکل نوشته می‌شود و با استفاده از صفحات HTML و جاوا اسکریپت مورد استفاده قرار می‌گیرد. شکل ۱۵ معماری این ابزار را برای تجزیه و تحلیل خودکار نشان می‌دهد. این ابزار، کشف حالت‌های مختلف اثبات و شکل‌های مختلف حمله را با استفاده از روش جست‌وجوی خودکار فراهم می‌کند. پیاده‌سازی یک پروتکل امنیتی در این ابزار شامل موارد زیر است:

- پیاده‌سازی دقیق پیام‌های ارسال‌شده در یک پروتکل با مجموعه‌ای از



شکل ۱۵. معماری ابزار تحلیل امنیت تامارین [۱۲]

قوانین بازنویسی‌شده

- پیاده‌سازی توابع نمادین مانند توابع چکیده‌ساز و توابع رمزنگاری

در نتیجه ورودی این ابزار یک پروتکل امنیتی است که به عنوان یک spthy file در نظر گرفته می‌شود. اقدامات انجام‌شده توسط طرفین شرکت‌کننده در پروتکل در نقش‌های مختلف مشخص می‌شوند. به طور مثال، آغازگر پروتکل، پاسخ‌دهنده و سرور. به طور کلی، این ابزار سعی در بررسی نمودن هر لم در تئوری پروتکل دارد [۱، ۲۳].

این ابزار دو روش اثبات امنیت را ارائه می‌دهد. در واقع یک حالت کاملاً خودکار دارد که در حالت خودکار اثبات را نشان می‌دهد. یعنی این ابزار یا اثبات صحت را برمی‌گرداند و یا سناریوی یک حمله را نشان می‌دهد، به طوری که ویژگی‌های بیان‌شده را نقض می‌نماید [۱۲، ۲۴].

۷ منطق‌های موجهات

منطق‌های موجهات شامل مجموعه‌ای از گزاره‌ها و قوانین هستند. این گزاره‌ها و عبارت‌ها باور و دانش را در مورد پیام‌های مورد استفاده در پروتکل‌های امنیتی بیان می‌نمایند. در این رویکرد، با توجه به پروتکل امنیتی مورد نظر، در ابتدا فرض‌ها و اهداف اولیه تعریف می‌شوند. سپس شکل اصلی پروتکل برای درستی‌یابی و تحلیل آماده می‌شود. درنهایت، از قوانین و باورهای منطقی موجود جهت به دست آوردن اهداف اولیه تعریف‌شده استفاده می‌شود. مشهورترین منطق‌های موجهات، منطق BAN است، که توسط سه پژوهشگر به نام‌های باروس- آبادی- نیدهام معرفی شد. به دلیل موفقیت‌آمیز بودن این منطق، زمینه جهت تحقیقات قابل‌توجه درباره تأیید صوری پروتکل‌های امنیتی ایجاد شد. هم‌چنین این منطق به منطق‌های دیگر از قبیل GNY [۱۳]، VO [۱۴]، SVO [۱۴] و غیره گسترش یافته است [۱، ۲۵-۲۷].

۱.۷ منطق BAN

منطق BAN یکی از روش‌های صوری جهت اثبات امنیت پروتکل‌های امنیتی است. این منطق توسط سه پژوهشگر ارائه شد و به عنوان منطق BAN معرفی گردید. منطق BAN مبتنی بر مجموعه‌ای از باورهای طرفین شرکت‌کننده در پروتکل ساخته می‌شود. پروتکل‌های امنیتی جهت بررسی احراز اصالت متقابل با استفاده از این منطق مورد بررسی قرار می‌گیرند [۱]. در این منطق، مجموعه‌ای از فرض‌ها و قوانین برای اثبات احراز

اصالت متقابل طرفین شرکت‌کننده استفاده می‌شود. به‌ویژه، به کاربران کمک می‌کند، تعیین کنند که آیا پیام‌های مبادله‌شده قابل اعتماد هستند یا نه. منطق BAN شامل سه نوع آزمایش به شرح ذیل است [۱۴، ۲۸]:

- تأیید منبع پیام
- تأیید اعتبار منبع
- تأیید تازگی پیام

قوانین منطق BAN

- تمامی قوانین این منطق با استفاده از علائم نشان داده شده در جدول ۳، در جدول ۴ بیان شده است. هم‌چنین این قوانین با جزئیات کامل به صورت زیر تعریف می‌شوند.
- IR1: این قانون جهت تضمین تازگی کل رابطه است، اگر یک بخش از رابطه تازه باشد. بنابراین، این معنی را می‌دهد که، اگر شرکت‌کننده P باور داشته باشد که پیام X تازه است، پس استنباط می‌شود، شرکت‌کننده P باور دارد که پیام (X,Y) نیز تازه است.
- IR2: این بدین معنی است، اگر شرکت‌کننده P پیام X و Y را باور داشته باشد، سپس، P رابطه ترکیب‌شده (X,Y) را نیز باور دارد.
- IR2*: اگر شرکت‌کننده P رابطه ترکیب شده (X,Y) را باور داشته باشد، پس شرکت‌کننده P هر یک از اجزای پیام مثلاً پیام X را نیز باور دارد.
- IR3: این قانون جهت تأیید کنترل یک شرکت‌کننده بر روی پیام استفاده می‌شود. بنابراین، این معنی را می‌دهد که، اگر شرکت‌کننده P باور داشته باشد که شرکت‌کننده Q بر روی X کنترل دارد و شرکت‌کننده P باور داشته باشد که شرکت‌کننده Q پیام X را باور دارد، سپس شرکت‌کننده P پیام X را باور دارد.
- IR4: این قانون جهت بررسی تازه بودن یک پیام و متعاقباً اعتقاد فرستنده به تازگی آن استفاده می‌شود. بنابراین، این معنی را می‌دهد که، اگر شرکت‌کننده P باور داشته باشد که پیام X تازه است و شرکت‌کننده P باور داشته باشد که شرکت‌کننده Q پیام X را یک‌مرتبه گفته است، سپس شرکت‌کننده P باور دارد که شرکت‌کننده Q پیام X را باور دارد.
- IR5: این قانون جهت تفسیر پیام‌های رمزشده استفاده می‌شود. بنابراین، این معنی را می‌دهد که، اگر شرکت‌کننده P باور داشته باشد که کلید K بین خودش و شرکت‌کننده Q به اشتراک گذاشته‌شده و هم‌چنین P می‌بیند که پیام X با کلید K رمز شده است، درنتیجه استنباط می‌شود که، P باور دارد که شرکت‌کننده Q پیام X را نیز یک‌مرتبه گفته است.
- IR5*: این قانون جهت تفسیر پیام‌های رمزشده استفاده می‌شود. بنابراین، این معنی را می‌دهد که، اگر شرکت‌کننده P باور دارد که K^+ کلید عمومی شرکت‌کننده Q است و هم‌چنین P می‌بیند که پیام X با کلید K^- رمز شده است، سپس استنباط می‌شود که، شرکت‌کننده P باور دارد که، شرکت‌کننده Q پیام Y را یک‌مرتبه گفته است.
- IR6: این معنی را می‌دهد که، اگر شرکت‌کننده P باور داشته باشد که شرکت‌کننده Q مقدار (X,Y) را گفته است، درنتیجه استنباط می‌شود که، شرکت‌کننده P باور دارد، شرکت‌کننده Q مقدار X را گفته است.
- IR7: این بدین معنی است که، اگر شرکت‌کننده P پیام X را باور داشته باشد، درنتیجه، شرکت‌کننده P مقدار چکیده X یعنی $h(X)$ را نیز باور دارد.

معمولاً روش اثبات امنیت پروتکل‌های امنیتی با منطق BAN در چند

جدول ۳. برخی از نمادهای استفاده‌شده در منطق BAN

علائم	توضیح
$P \triangleleft X$	شرکت‌کننده P پیام X را می‌بیند
$P \mid\equiv X$	شرکت‌کننده P پیام X را باور دارد
$P \mid\Rightarrow X$	شرکت‌کننده P بر روی پیام X کنترل دارد، یعنی اگر پیام تغییر کند، متوجه می‌شود
$\#X$	پیام X یک پیام تازه است
$\{X\}_K$	پیام X با استفاده از کلید K رمز شده است
$(X)_K$	پیام X با استفاده از کلید K چکیده‌سازی شده است
$\langle X\rangle_Y$	رابطه X با رابطه Y ترکیب شده است
(X,Y)	X یا Y بخشی از رابطه (X,Y) هستند
SK	کلید جلسه
$P \overset{K}{\rightleftharpoons} Q$	K یک مقدار مخفی به اشتراک گذاشته‌شده بین P و Q است
$P \overset{K}{\leftrightarrow} Q$	شرکت‌کننده‌های P و Q با یکدیگر از طریق کلید به اشتراک گذاشته‌شده K ارتباط برقرار می‌کنند
$P \mid\sim X$	شرکت‌کننده P یک‌مرتبه پیام X را گفته است
$\overset{K^+}{\mapsto} P$	کلیدهای عمومی/خصوصی شرکت‌کننده P به ترتیب برابر K^+ و K^- هستند

جدول ۴. قوانین منطق BAN استفاده‌شده

علائم	توضیح
$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$	IR1: قانون تازگی
$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X,Y)}$	IR2: قانون باور و اعتقاد
$\frac{P \mid\equiv (X,Y)}{P \mid\equiv X}$	IR2*: قانون باور و اعتقاد
$\frac{P \mid\equiv Q \mid\Rightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$	IR3: قانون کنترل
$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$	IR4: قانون تأیید عدد یک‌بار مصرف
$\frac{P \mid\equiv P \overset{K}{\rightleftharpoons} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$	IR5: قانون معنی پیام: برای رمزنگاری کلید متقارن
$\frac{P \mid\equiv \overset{K^+}{\mapsto} Q, P \triangleleft \{X\}_{K^-}}{P \mid\equiv Q \mid\sim X}$	IR5*: قانون معنی پیام: برای رمزنگاری کلید عمومی
$\frac{P \mid\equiv Q \mid\sim (X,Y)}{P \mid\equiv Q \mid\sim X}$	IR6: قانون حذف پیام‌های چند قسمتی
$\frac{P \mid\equiv X}{P \mid\equiv (X)_h}$	IR7: قانون باور توابع چکیده‌ساز

مرحله به صورت زیر خلاصه می‌شود:

- گام یک (تعیین شکل کلی پیام‌ها): در این گام، پیام‌های مبادله‌شده در طول مرحله احراز اصالت مشخص می‌شوند.
- گام دوم (تعیین شکل مرتبط و ایده‌ال‌شده پروتکل با منطق BAN): در این گام، پیام‌های مبادله‌شده به شکل منطق BAN تبدیل می‌شوند. به‌عبارتی، پیام‌های مبادله‌شده ایده‌آل می‌شوند. این بدین معنی است، پیام‌های ارسال‌شده‌ی رمزنشده که موجب افزایش امنیت نمی‌شوند، حذف می‌گردد.
- گام سوم (تعیین فرض‌های اولیه): در این مرحله، فرض‌هایی جهت شروع فرایند برای بررسی ویژگی امنیتی احراز اصالت متقابل در نظر گرفته می‌شود. این فرض‌ها مبتنی بر وضعیت هر پیام هستند و قبل از اینکه پروتکل اجرا شود، با منطق BAN رسمیت می‌یابند.
- گام چهارم (تعریف اهداف امنیتی): به منظور بررسی امنیت، در این گام اهداف امنیتی تعریف می‌شوند.
- گام پنجم (حصول اهداف امنیتی): با استفاده از پیام‌ها، فرض‌های اولیه و طبق قوانین منطق BAN بررسی می‌شود که آیا اهداف امنیتی قابل حصول هستند یا خیر.

۱.۱.۷ تحلیل امنیتی صوری پروتکل ECCPWS با استفاده از منطق BAN

به دلیل اینکه کانال استفاده‌شده در مرحله ثبت‌نام امن است، واضح است که اهداف امنیتی مرحله ثبت‌نام از طریق منطق BAN قابل دسترس است. بنابراین، در این بخش، منطق BAN برای اثبات امنیت مرحله احراز هویت طرح ECCPWS استفاده می‌شود. اثبات امنیت مرحله احراز هویت طرح ECCPWS با استفاده از منطق BAN به صورت زیر انجام شده است.

شکل عمومی پیام‌ها پیام‌های مبادله‌شده در طول مرحله احراز هویت در طرح ECCPWS، می‌تواند به صورت زیر بیان شود:

- پیام یک:

$$U_i \mapsto AS\{E_U, R, T_1\} : \{\{c, U_U, Ind_U, T_1\}_{K_{UA}}, r \cdot B, T_1\}$$

- پیام دو:

$$AS \mapsto U_i\{Auth, Y, T_2\} : \{h(U_U, R, Q', Y, T_2), y \cdot B, T_2\}$$

تعیین پروتکل به صورت شکل ایده‌ال‌شده منطق BAN در این گام، پیام‌های مبادله‌شده در مرحله احراز هویت طرح ECCPWS به شکل ایده‌ال منطق BAN تبدیل می‌شوند. به عبارتی، پیام‌های مبادله‌شده ایده‌ال می‌شوند. به گونه‌ای‌که، پیام‌های ارسالی رمزنشده که هیچ افزایش امنیتی ایجاد نمی‌کنند، حذف می‌شوند. بنابراین، شکل ایده‌آل‌شده‌ی پیام‌ها در مرحله احراز هویت طرح ECCPWS به گونه‌ای هستند که K^{-AS} به عنوان کلید خصوصی AS، PK_{AS} به عنوان کلید عمومی آن و K_{UA} کلید مخفی به‌اشتراک گذاشته‌شده بین U و AS را نشان می‌دهد.

$$\text{IM1} : AS \triangleleft \{c, U_U, Ind_U, T_1\}_{K_{UA}}, r \cdot B$$
$$\text{IM2} : U_i \triangleleft \{U_U, R, Y, T_2\}_{K^{-AS}}, y \cdot B$$

تعیین فرض‌های اولیه فرض‌های منطق BAN طرح ECCPWS در جدول ۵ بیان شده است.

تعریف اهداف امنیتی به منظور بررسی امنیت طرح پیشنهادی ECCPWS، باید اهداف امنیتی آن را تعریف کنیم. مطابق با روش تحلیل منطق BAN، طرح پیشنهادی ECCPWS دو هدف بیان‌شده در جدول ۵ را دنبال می‌کند.

حصول اهداف امنیتی در این مرحله، با اعمال قوانین منطق BAN ارائه‌شده در فصل چهارم به فرض‌ها و پیام‌های ایده‌ال شده پروتکل پیشنهادی ECCPWS، اهداف امنیتی مطابق با جدول ۶ اثبات می‌شود.

بنابراین، اهداف امنیتی «Goal1» و «Goal2» به دست خواهد آمد، که نشان می‌دهد AS و کاربر U_i به کلید مخفی به اشتراک گذاشته شده محاسبه شده یعنی SK، اعتقاد و باور دارد. مبتنی بر اثبات‌های بالا، استنباط می‌شود که ECCPWS هر دو هدف را تأمین می‌کند. منظور از «Goal1» این است که، AS به کلید مخفی به اشتراک گذاشته شده یعنی SK، اعتقاد و باور دارد و منظور از «Goal2» این است که، U_i به کلید مخفی به اشتراک گذاشته شده یعنی SK، اعتقاد و باور دارد. پس درنتیجه، تحلیل بالا مشخص می‌کند که طرح ECCPWS امنیت قابل قبولی دارد و می‌تواند احراز هویت متقابل را فراهم کند.

۲.۷ مدل پیشگوی تصادفی

مدل پیشگوی تصادفی، امنیت کلید جلسه را که بین طرفین شرکت‌کننده در پروتکل ایجاد شده است در برابر حملات فعال و غیرفعال بررسی می‌کند. به عبارتی، هدف اصلی اثبات با استفاده از این مدل، اثبات امنیت کلید جلسه است. پیشگوهای تصادفی جهت فراهم نمودن پاسخ‌های منحصربه‌فرد به چالش‌هایی که توسط مهاجم ایجاد می‌شود، استفاده می‌شوند. قابل‌توجه است که، این پاسخ همیشه برای داده‌های تکراری مشابه است. مدل پیشگوی تصادفی نیز از مدل مهاجم دالو-یائو پیروی می‌کند. در واقع، رمزنگاری پیشگوی تصادفی شامل یک پیشگوی تصادفی است که به هر سوالی که از آن می‌شود، یک پاسخ کاملاً تصادفی می‌دهد. یعنی، پیشگوی تصادفی به ازای هر ورودی ثابت هر بار یک جواب می‌دهد. از پیشگوی تصادفی در اثبات‌های رمزنگاری استفاده می‌شود. در طرحی که از پیشگوی تصادفی استفاده می‌شود، همه طرفین شرکت‌کننده از جمله مهاجم با یکدیگر تعامل دارند. علاوه‌براین، آن‌ها می‌توانند پرسش‌های تصادفی را ایجاد کنند. فرض بر این است که، همه پرسش‌های تصادفی صرف‌نظر از هویت طرفین شرکت‌کننده با یک تابع واحد مانند $\mathcal{O}$ پاسخ داده شوند. یک سامانه، امن در نظر گرفته می‌شود، اگر مهاجم با توانایی‌های داده‌شده مانند دسترسی به پیشگوی تصادفی، احتمال موفقیت ناچیزی داشته باشد [۱۶، ۱۷].

جدول ۵. فرض‌ها و اهداف امنیتی در اثبات امنیت طرح ECCPWS با استفاده از منطق BAN

علائم	توضیح
$U_i \equiv r$	A1: U_i مقدار r را باور دارد.
$U_i \equiv R$	A2: U_i مقدار R را باور دارد.
$AS \equiv y$	A3: AS مقدار y را باور دارد.
$AS \equiv Y$	A4: AS مقدار Y را باور دارد.
$AS \equiv U_i \Rightarrow U_U$	A5: AS باور دارد که U_i بر روی U_U کنترل دارد. یعنی، اگر دچار تغییر شود، متوجه می‌شود.
$AS \equiv U_i \Rightarrow \mathrm{Ind}_U$	A6: AS باور دارد که U_i بر روی Ind_U کنترل دارد. یعنی، اگر دچار تغییر شود، متوجه می‌شود.
$U_i \equiv U_i \overset{K_U A}{\rightleftharpoons} AS$	A7: U_i باور دارد که U_i و AS با یکدیگر از طریق کلید به اشتراک گذاشته شده $K_U A$ ارتباط برقرار می‌کنند.
$AS \equiv AS \overset{K_U A}{\rightleftharpoons} U_i$	A8: AS باور دارد که AS و U_i با یکدیگر از طریق کلید به اشتراک گذاشته شده $K_U A$ ارتباط برقرار می‌کنند.
$U_i \equiv \overset{PK_{AS}}{\mapsto} AS$	A9: U_i باور دارد که PK_{AS} کلید عمومی AS است.
$U_i \equiv \#r$	A10: U_i باور دارد که r تازه است.
$U_i \equiv AS \Rightarrow Y$	A11: U_i باور دارد که AS بر روی Y کنترل دارد. یعنی، اگر دچار تغییرشود، متوجه می‌شود.
$U_i \equiv U_U$	A12: U_i مقدار U_U را باور دارد.
$AS \equiv U_i \Rightarrow R$	A13: AS باور دارد که U_i بر روی R کنترل دارد. یعنی، اگر دچار تغییرشود، متوجه می‌شود.
$AS \equiv S_{AS}$	A14: AS مقدار S_{AS} را باور دارد.
$AS \equiv \#U_U$	A15: AS باور دارد که U_U تازه است.
$AS \equiv \#r$	A16: AS باور دارد که r تازه است.
$AS \equiv U_i \equiv R$	A17: AS باوردارد که U_i مقدار R را باور دارد.
$U_i \equiv \#T_1$	A18: U_i باور دارد که مقدار T_1 تازه است.
$U_i \equiv \#T_2$	A19: U_i باور دارد که مقدار T_2 تازه است.
$AS \equiv \#T_3$	A20: AS باور دارد که مقدار T_3 تازه است.
$AS \equiv \#T_4$	A21: AS باور دارد که مقدار T_4 تازه است.
$AS \equiv SK$	Goal1: کلید مخفی به اشتراک گذاشته شده، یعنی AS SK را باور دارد.
$U_i \equiv SK$	Goal2: کلید مخفی به اشتراک گذاشته شده، یعنی U_i SK را باور دارد.

مهاجم: در این مدل، مهاجم A با استفاده از مدل دالو–یائو مدل می‌شود، این یعنی همه‌ی ارتباطات می‌تواند با A کنترل شوند و A کنترل کاملی سرتاسر کانال دارد. این توانایی شامل شنود، تغییر، حذف پیام‌های مبادله‌شده، ساخت پیام‌های جدید و تزریق پیام‌ها بین دو شرکت‌کننده در طول ارتباطات است. بنابراین، مهاجم A به درخواست‌های زیر دسترسی دارد:

- $Execute(\mathcal{O}^t, \mathcal{O}^u, \mathcal{O}^v)$: مهاجم $\mathcal{A}$ درخواست شرکت‌کنندگان قانونی پروتکل را در زمان‌های u، v و t برای شنود پیام‌های ارسال‌شده در میان آن‌ها اجرا می‌نماید.
- $Reveal(\mathcal{O}^t)$: در این درخواست، کلید جلسه SK تولید شده با $\mathcal{O}^t$ و طرف مقابلش، برای A در جلسه جاری افشا می‌شود.
- $Test(\mathcal{O}^t)$: مبتنی بر این درخواست، امنیت کلید جلسه بین شرکت‌کننده‌ها مدل می‌شود. یک سکه‌ی سالم در شروع آزمایش انتخاب شده، به گونه‌ای که نتیجه آن تنها برای مهاجم شناخته شده است. مبتنی بر خروجی، تصمیم‌گیری انجام می‌شود.
- $Send(\mathcal{O}^t, msg)$: این درخواست جهت ارسال پیام msg به $\mathcal{O}^t$ و یا دریافت پاسخ پیام از $\mathcal{O}^t$ استفاده می‌شود.

شرکت‌کننده‌ها: شرکت‌کنندگان پروتکل به عنوان پیشگوهای تصادفی مشخص می‌شوند.

حالت پذیرفته‌شده: پیشگوهای تصادفی $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ بعد از دریافت آخرین پیام پروتکل و احراز اصالت به یکدیگر به حالت پذیرفته شده می‌روند و شناسه جلسه sid را برای جلسه جاری در نظر می‌گیرند.

مدل پیشگوی تصادفی موارد زیر را در نظر می‌گیرد [۱]:

شراکت: دو پیشگوی تصادفی $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ با یکدیگر شریک و همکار هستند، اگر شرایط زیر برقرار باشد:

- پیشگوهای $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ در حالت‌های پذیرفته‌شده باشند.
- پیشگوهای $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ برای یکدیگر احراز اصالت متقابل شوند و sid یکسان به‌اشتراک گذارند.

امنیت معنایی کلید جلسه: در این مدل، امنیت معنایی کلید جلسه در بین شرکت‌کننده‌ها، براساس ویژگی عدم تمایز بین کلید جلسه واقعی و مقدار تصادفی، توسط مهاجم $\mathcal{A}$ تعریف می‌شود. مهاجم $\mathcal{A}$ می‌تواند چندین درخواست Test تولید کند که خروجی این درخواست‌ها باید مرتبط با بیت تصادفی a باشد.

۳.۷ تحلیل امنیتی صوری پروتکل ECCPWS با استفاده از مدل پیشگوی تصادفی

در این بخش، از مدل پیشگوی تصادفی برای اثبات امنیت طرح پیشنهادی ECCPWS استفاده شده است. در این تحلیل، هدف اصلی اثبات امنیت کلید جلسه یعنی SK با استفاده از مدل پیشگوی تصادفی است. از این رو، اثبات در قضیه یک بیان شده است [۱]. همچنین، این مدل موارد زیر را در نظر می‌گیرد:

مهاجم: در مدل مهاجم دالو–یائو، همه‌ی ارتباطات می‌تواند با $\mathcal{A}$ کنترل شوند و $\mathcal{A}$ کنترل کاملی سرتاسر کانال دارد. مهاجم توانایی شنود، تغییر،

جدول ۶. اثبات امنیت طرح پیشنهادی ECCPWS با استفاده از منطق BAN

پیام	مفروضات اولیه	قوانین	فرضیه‌ها	هدف امنیتی
IM1	A8	IR5	$AS\|\equiv U_i\|\approx \{c, U_U, \mathrm{Ind}_U, T_1\}$:P1	-
P1	-	IR6	$AS\|\equiv U_i\|\approx c$:P2	-
P1	-	IR6	$AS\|\equiv U_i\|\approx U_U$:P3	-
P1	-	IR6	$AS\|\equiv U_i\|\approx \mathrm{Ind}_U$:P4	-
P1	-	IR6	$AS\|\equiv U_i\|\approx T_1$:P5	-
P2	-	IR6	$AS\|\equiv U_i\|\approx r$:P6	-
P6	A13,A17	IR3,IR4	$AS\|\equiv R$:P7	-
P7	A14	-	$AS\|\equiv RS_{AS} = Q'$:P8	-
P3	A15	IR4	$AS\|\equiv U_i\|\equiv U_U$:P9	-
P9	A5	IR3	$AS\|\equiv U_U$:P10	-
P7,P8,P10	A4,A17,A21	IR2	$AS\|\equiv (U_U, R, Q', Y, T_2)$:P11	-
P11	-	IR7	$AS\|\equiv (U_U, R, Q', Y, T_2)_h = \mathrm{Auth}$:P12	-
P7	A3	-	$AS\|\equiv yR = Z$:P13	-
P10,P12,P13	-	IR2	$AS\|\equiv (\mathrm{Auth}, U_U, Z)$:P14	-
P14	-	IR7	$AS\|\equiv (\mathrm{Auth}, U_U, Z)_h = SK$:P15	Goal1
IM2	A9	IR5*	$U_i\|\equiv AS\|\approx (U_U, R, Y, T_2)$:P16	-
-	A10	IR1	$U_i\|\#(U_U, R, Y, T_2)$:P17	-
P16,P17	-	IR4	$U_i\|\equiv AS\|\equiv (U_U, R, Y, T_2)$:P18	-
P18	-	IR6	$U_i\|\equiv AS\|\equiv U_U$:P19	-
P18	-	IR6	$U_i\|\equiv AS\|\equiv R$:P20	-
P18	-	IR6	$U_i\|\equiv AS\|\equiv Y$:P21	-
P18	-	IR6	$U_i\|\equiv AS\|\equiv T_2$:P22	-
P21	A11	IR3	$U_i\|\equiv Y$:P23	-
-	A1,A9	-	$U_i\|\equiv rPK_{AS} = Q$:P24	-
P23,P24	A12,A2,A21	IR2	$U_i\|\equiv (U_U, R, Q, Y, T_2)$:P25	-
P25	-	IR7	$U_i\|\equiv (U_U, R, Q, Y, T_2)_h = \mathrm{Auth}'$:P26	-
P23	A1	-	$U_i\|\equiv rY$:P27	-
P26,P27	A12	IR2	$U_i\|\equiv (\mathrm{Auth}', U_U, rY)$:P28	-
P28	-	IR7	$U_i\|\equiv (\mathrm{Auth}', U_U, rY)_h = SK$:P29	Goal2

حذف پیام‌های مبادله شده، ساخت پیام‌های جدید و تزریق پیام‌ها را بین دو شرکت‌کننده در طول ارتباطات دارد. بنابراین، $\mathcal{A}$ به درخواست‌های زیر دسترسی خواهد داشت:

- $Execute(\mathcal{O}^t, \mathcal{O}^u, \mathcal{O}^v)$: مهاجم درخواست به پیشگوهای NM، AS و U را در زمان‌های u، v و t برای شنود پیام‌های ارسال‌شده در میان شرکت‌کنندگان قانونی U، NM و AS اجرا می‌نماید.
- $Reveal(\mathcal{O}^t)$: در این درخواست، کلید جلسه SK تولیدشده با پیشگوی $\mathcal{O}^t$ و طرف مقابلش، در جلسه جاری برای $\mathcal{A}$ افشا می‌شود.
- $Test(\mathcal{O}^t)$: در این درخواست، امنیت کلید جلسه SK بین شرکت‌کنندگان U و AS بررسی می‌شود. بنابراین، سکه‌ی سالم a در شروع بازی انتخاب شده، به گونه‌ای که نتیجه پرتاب آن تنها برای مهاجم شناخته شده است. فرض می‌شود که $\mathcal{A}$ این درخواست را اجرا کند و سپس $\mathcal{O}^t$ مقدار کلید جلسه SK را برمی‌گرداند، اگر $a = ۱$ و اگر $a = ۰$ باشد، یک مقدار تصادفی را برمی‌گرداند، در غیر این صورت، یک مقدار تهی به نام $\perp$ برگردانده می‌شود.
- $Send(\mathcal{O}^t, msg)$: این درخواست جهت ارسال پیام msg به

پیشگوی $\mathcal{O}^t$ و یا دریافت پاسخ پیام از $\mathcal{O}^t$ استفاده می‌شود.

شرکت‌کننده‌ها: پیشگوهای $\mathcal{O}^t_U$، $\mathcal{O}^u_{NM}$ و $\mathcal{O}^v_{AS}$ استفاده‌شده برای شرکت‌کنندگان U، NM و AS در زمان‌های t، u و v به ترتیب، مشخص می‌شوند.

حالت پذیرفته‌شده: پیشگوهای $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ بعد از دریافت آخرین پیام پروتکل و احراز هویت به یکدیگر به حالت پذیرفته‌شده می‌روند. همچنین، پیشگوهای تصادفی شناسه جلسه sid را برای جلسه جاری در نظر می‌گیرند.

شراکت: دو پیشگوی تصادفی $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ برای یکدیگر شریک و همکار هستند، اگر شرایط زیر برقرار گردد:

- پیشگوهای $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ در حالت‌های پذیرفته شده قرار بگیرند.
- پیشگوهای $\mathcal{O}^{t_1}$ و $\mathcal{O}^{t_2}$ برای یکدیگر احراز هویت متقابل شوند و sid یکسان به اشتراک گذارند.

امنیت معنایی کلید جلسه: در این مدل، امنیت معنایی کلید جلسه بین شرکت‌کنندگان U و AS بر اساس ویژگی عدم تمایز بین کلید جلسه واقعی و مقدار تصادفی توسط $\mathcal{A}$ بررسی می‌شود. مهاجم می‌تواند چندین درخواست $Test$ برای پیشگوهای $\mathcal{O}^t_U$، $\mathcal{O}^u_{NM}$ و $\mathcal{O}^v_{AS}$ تولید کند. خروجی این درخواست باید مرتبط با بیت تصادفی a باشد. در پایان بازی، $\mathcal{A}$ بیت a' را حدس‌می‌زند. اگر شرط $a = a'$ برقرار بود، مهاجم بازی را می‌برد. Succ داده‌شده، رخدادی است که، $\mathcal{A}$ بازی را می‌برد، برتری $\mathcal{A}$ در شکستن امنیت طرح پیشنهادی ECCPWS به صورت رابطه زیر تعریف می‌شود:

$$Adv_{\mathcal{A}}^{ECCPWS} = \left|2 \cdot \Pr\left[Succ_{\mathcal{A}}^{Game_i}\right] - 1\right| \quad (1)$$

ECCPWS امن است، اگر رابطه $Adv_{\mathcal{A}}^{ECCPWS} \leq \epsilon$ برای مقادیر ناچیز $\epsilon > 0$ برقرار گردد.

پیشگوی تصادفی: ECCPWS از یک تابع چکیده‌ساز یک‌طرفه مقاوم در برابر برخورد به نام $h(\cdot)$ استفاده می‌کند. این تابع به عنوان یک پیشگوی تصادفی به شکل $h(\cdot)$ مدل می‌شود. فرض می‌شود که $\mathcal{A}$ و همه شرکت‌کنندگان به تابع $h(\cdot)$ دسترسی دارند.

اثبات امنیت: امنیت کلید جلسه در طرح پیشنهادی ECCPWS مبتنی بر قضیه ۱ و براساس مدل پیشگوی تصادفی اثبات می‌شود.

قضیه ۱. فرض‌کنید q_{hash}، $|Hash|$ و $Adv_{\mathcal{A}}^{ECDDHP}(t)$ به ترتیب تعداد درخواست‌های چکیده‌سازی، محدوده فضای تابع چکیده‌ساز $H(\cdot)$ و برتری $\mathcal{A}$ در شکستن $ECDDHP$ را نشان می‌دهد. آنگاه، برتری $\mathcal{A}$ در شکستن امنیت $ECCPWS$ جهت به دست آوردن SK بین U و AS در طول مرحله احراز هویت می‌تواند به صورت زیر تخمین زده شود:

$$Adv_{\mathcal{A}}^{ECCPWS}(t) \leq \frac{q_{hash}^2}{|Hash|} + 2 \cdot Adv_{\mathcal{A}}^{ECDDHP}(t) \quad (2)$$

اثبات یک. این اثبات شامل سه بازی به شکل $Game_i$ با $(i = 0, 1, 2)$ است. $Succ_{\mathcal{A}}^{Game_i}$ رخدادی است که $\mathcal{A}$ می‌تواند بیت تصادفی a را از یک سکه‌ی سالم پرتاب شده در $Game_i$ با استفاده از درخواست $Test$ به طور موفق تخمین بزند. برتری $\mathcal{A}$ در برنده‌شدن $Game_i$ به صورت $Adv_{\mathcal{A},Game_i}^{ECCPWS} = \Pr\left[Succ_{\mathcal{A}}^{Game_i}\right]$ بیان می‌شود. جزئیات بازی‌ها به ترتیب به صورت زیر نشان داده شده است.

- $Game_0$: در این بازی به عنوان بازی اولیه، $\mathcal{A}$ «حمله واقعی» را به ECCPWS مبتنی بر مدل پیشگوی تصادفی پیاده‌سازی می‌کند. در $Game_0$، بیت a به طور تصادفی و به صورت زیر انتخاب می‌گردد:

$$Adv_{\mathcal{A}}^{ECCPWS}(t) = \left|2 \cdot Adv_{\mathcal{A},Game_0}^{ECCPWS} - 1\right| \quad (3)$$

- $Game_1$: این بازی به عنوان «حمله شنود» با $\mathcal{A}$ انجام می‌شود، که $\mathcal{A}$ می‌تواند درخواست $Execute(\mathcal{O}^t, \mathcal{O}^u, \mathcal{O}^v)$ را اجرا کند. $\mathcal{A}$ می‌تواند همه پیام‌های مبادله‌شده $\{E_U, R, T_1\}$ و $\{Auth, Y, T_3\}$ را در طول مرحله احراز هویت با استفاده از درخواست $Execute$ شنود کند. $\mathcal{A}$ نیاز به اجرای درخواست‌های $Reveal$ و $Test$ در پایان بازی دارد. از این رو، $\mathcal{A}$ تشخیص می‌دهد که آیا خروجی درخواست $Test$ کلید جلسه واقعی است یا یک مقدار تصادفی. در نظر داشته باشید مقدار کلید جلسه به صورت $SK = h(Auth, U_U, Z)$ محاسبه می‌شود. از این رو، امنیت SK، وابسته به مقادیر مخفی بلندمدت S_{AS}، S_U و مقادیر تصادفی r و y است. بنابراین، شنود پیام‌های مبادله‌شده $\{E_U, R, T_1\}$ و $\{Auth, Y, T_3\}$ احتمال بردن $\mathcal{A}$ را در به دست آوردن کلید جلسه در $Game_1$ بدون این مقادیر مخفی، افزایش نمی‌دهد. بدیهی است که، $Game_0$ و $Game_1$ غیرقابل تمایز هستند. یعنی:

$$\Pr\left[Succ_{\mathcal{A}}^{Game_0}\right] = \Pr\left[Succ_{\mathcal{A}}^{Game_1}\right] \quad (4)$$

هم‌چنین، $\mathcal{A}$ نیاز به اعداد تصادفی (r, y) دارد که این مقادیر برای $\mathcal{A}$ شناخته‌شده نیست. درنتیجه، داریم:

$$Adv_{\mathcal{A},Game_0}^{ECCPWS} = Adv_{\mathcal{A},Game_1}^{ECCPWS} \quad (5)$$

- $Game_2$: در این بازی، درخواست $Hash$ یا همان تابع چکیده‌ساز مدل شده است. $\mathcal{A}$ تلاش می‌کند که چندین درخواست $Hash$ را جهت یافتن یک برخورد در خروجی پیام‌های تابع چکیده‌ساز تولید کند. زمانی‌که درخواست $Hash$ با $\mathcal{A}$ اجرا می‌شود، هیچ برخوردی رخ نمی‌دهد. در پیام $\{Auth, Y, T_3\}$، مقدار $Auth$ با تابع چکیده‌ساز یک‌طرفه مقاوم در برابر برخورد $h(\cdot)$ محافظت می‌شود. چون همه‌ی مقادیر موقتی، اعداد تصادفی، مهرهای زمانی موجود در پیام‌ها و تابع چکیده‌ساز ایده‌ال هستند، در نتیجه هیچ برخوردی رخ نمی‌دهد، اگر درخواست $Hash$ با $\mathcal{A}$ اجرا شود. هم‌چنین، به دلیل وجود روابط $R = r \cdot B$ و $Y = y \cdot B$، یک مسئله ناممکن محاسباتی ECDDHP برای $\mathcal{A}$ وجود دارد که بتواند $Z = yr \cdot B$ را به دست آورد. ازاین‌رو، جهت به دست آوردن SK بین U و AS، مهاجم نیاز به مقدار Z جهت حل ECDDHP در زمان چندجمله‌ای t دارد و هیچ برخوردی در خروجی تابع چکیده‌ساز در $Auth$ وجود ندارد. شایان ذکر است،

هر دو $Game_۱$ و $Game_۲$ به جز در شبیه‌سازی درخواست $Hash$ در $Game_۲$ «غیرقابل تمایز» هستند. بنابراین، ما نتایج زیر را با استفاده از ویژگی ناممکن بودن حل مسئله ECDDHP و پارادوکس روز تولد به‌دست می‌آوریم:

$$\left|Adv^{ECCPWS}_{\mathcal{A},Game_۱} - Adv^{ECCPWS}_{\mathcal{A},Game_۲}\right| \leq \frac{q^۲_{hash}}{۲|Hash|} + Adv^{ECDDHP}_{\mathcal{A}}(t) \quad (۶)$$

$\mathcal{A}$ بازی را تنها با حدس یک‌مرتبه بیت a با اجرای درخواست‌های $Test$ و $Reveal$ می‌برد. درنتیجه داریم:

$$Adv^{ECCPWS}_{\mathcal{A},Game_۲} = \frac{۱}{۲} \quad (۷)$$

بنابراین، مبتنی بر معادله (۴) و (۷) داریم:

$$\frac{۱}{۲}Adv^{ECCPWS}_{\mathcal{A}}(t) = \left|Adv^{ECCPWS}_{\mathcal{A},Game_۰} - \frac{۱}{۲}\right| = \left|Adv^{ECCPWS}_{\mathcal{A},Game_۱} - \frac{۱}{۲}\right| \quad (۸)$$

مبتنی بر معادلات (۵)، (۶)، (۷) و (۸)، نیز داریم:

$$\frac{۱}{۲}Adv^{ECCPWS}_{\mathcal{A}}(t) = \left|Adv^{ECCPWS}_{\mathcal{A},Game_۱} - Adv^{ECCPWS}_{A,Game_۲}\right| \leq \frac{q^۲_{hash}}{۲|Hash|} + Adv^{ECDDHP}_{\mathcal{A}}(t) \quad (۹)$$

درنتیجه، هر دو طرف معادله (۹) را با عامل دو ضرب می‌کنیم. بنابراین، نتیجه نهایی که همان قضیه ۱ است، به صورت زیر اثبات می‌شود:

$$Adv^{ECCPWS}_{\mathcal{A}}(t) \leq \frac{q^۲_{hash}}{|Hash|} + ۲ \cdot Adv^{ECDDHP}_{\mathcal{A}}(t) \quad (۱۰)$$

این رابطه برتری $\mathcal{A}$ را در شکستن امنیت طرح ECCPWS جهت به دست آوردن SK بین U و AS در مرحله احراز هویت نشان می‌دهد و درنهایت امنیت کلید جلسه در پروتکل پیشنهادی مبتنی بر مدل پیشگوی تصادفی اثبات می‌شود.

□

۴.۷ منطق GNY

در سال ۱۹۹۰، سه پژوهشگر به نام‌های گنگ، نیدهام، یاهالم منطق BAN را گسترش دادند و نام آن را منطق GNY نامیدند. به طور خاص، این منطق آن‌چه را که شخص در اختیار دارد از آن‌چه شخص به آن اعتقاد و باور دارد، متمایز می‌کند [۱۳]. با این حال، این منطق قواعد و قوانین منطق BAN را تعمیم می‌دهد. مبتنی بر تعمیم فوق، منطق GNY دارای دامنه کاربرد گسترده‌ای است. درحقیقت، این منطق هم شبیه منطق BAN هدفش تحلیل پروتکل‌ها به صورت گام‌به‌گام است. درواقع، در ابتدا هرگونه فرضی را که لازم است بیان می‌کند و سپس نتیجه‌گیری‌های لازم را انجام می‌دهد. این منطق نسبت به منطق BAN مزایای بیشتری دارد [۱]. از جمله اینکه، این منطق شامل قوانین کامل‌تری است. هم‌چنین، این منطق به ما امکان تجزیه و تحلیل طیف وسیع‌تری از پروتکل‌ها را می‌دهد. هر هستار دو مجموعه را نشان می‌دهد: اولاً مجموعه‌ای از باورها که شامل همه باورهای جاری آن هستار است و دوماً مجموعه‌ای از دارایی‌ها که شامل هر چیزی که آن هستار دریافت می‌کند و هر چیزی که آن هستار خودش تولید کرده (مانند اعداد تصادفی در جلسه جاری) هستند. هستارها، جلسه را با باورها و دارایی‌های اولیه شروع می‌کنند. یک هستار می‌تواند باورهای جدید به‌دست آورد و مجموعه باورهای خودش را گسترش دهد. قوانین استنتاجی موجب ایجاد باورهای جدید از پیام‌های جاری و دریافتی می‌شوند. به‌طور مشابه، یک هستار می‌تواند دارایی‌های خودش را افزایش دهد. در این بخش، مفاهیم اساسی و علائم اختصاری این منطق توضیح داده خواهد شد. رابطه نامی است که برای نشان دادن یک رشته بیت مانند نام یک متغیر استفاده می‌شود. فرض کنید X و Y محدوده‌ی رابطه‌ها هستند. همچنین کلیدهای مخفی به اشتراک گذاشته شده جهت برقراری ارتباط و کلیدهای رمزنگاری به ترتیب با S و K نشان داده می‌شوند. رابطه‌های موجود در این منطق به صورت زیر بیان می‌شوند [۱۳]:

- (X, Y): ترکیب دو رابطه
- $\{X\}_K$ و $\{X\}^{-۱}_K$: رمزنگاری و رمزگشایی مرسوم مانند AES. فرض می‌شود که سامانه‌های رمزنگاری در مقابل حملات متن اصلی معلوم و متن فقط رمزشده مقاوم هستند. علاوه بر این، هر بیت در متن رمزشده بستگی به همه بیت‌های متن اصلی و کلید دارد، به طوری که هرگونه تغییر در متن اصلی موجب یک تغییر تصادفی در متن رمزشده و برعکس می‌شود.
- $\{X\}_{+K}$ و $\{X\}_{-K}$: رمزنگاری و رمزگشایی کلید عمومی. علاوه بر الزامات بیان‌شده برای سامانه‌های رمز معمولی، رابطه $\{\{X\}_{+K}\}_{-K} = X$ برقرار است. هم‌چنین، بعضی از سامانه‌های رمز کلید عمومی دارای رابطه $\{\{X\}_{-K}\}_{+K} = X$ هستند.
- $H(X)$: یک تابع چکیده‌ساز یک‌طرفه از X. اگر X به ما داده شود محاسبه $H(X)$ آسان است ولی برعکس نه. محاسبه X و X' به طوری که $X \neq X'$ و $H(X) = H(X')$ باشد، ناممکن است.
- $F(X_۱, \ldots, X_n)$: F یک تابع چند به یک و از لحاظ محاسباتی شدنی است به طوری که برای هر مقدار X_i، $۱ \leq i \leq n$ و ثوابت $C_۱, \ldots, C_{n-۱}$، $F(C_۱, \ldots, C_{i-۱}, X_i, C_i, \ldots, C_{n-۱})$ یک تابع یک‌به‌یک و از لحاظ محاسباتی شدنی است. به عنوان مثال، XOR یک تابع است. بنابراین، $F(X)$ یک تابع یک‌به‌یک و از لحاظ محاسباتی شدنی است.

نمادها و علائم منطق GNY تمامی علائم استفاده‌شده در این منطق با ذکر جزئیات در جدول ۷ به صورت زیر تعریف می‌شوند. فرض کنید P و Q دو هستار ارتباطی هستند [۱].

قوانین منطق GNY تمامی قوانین این منطق با استفاده از علائم نشان داده شده در جدول ۷، در جدول ۸ بیان می‌شود. هم‌چنین این قوانین با

جدول ۷. نمادها و علائم استفاده‌شده در منطق GNY

علائم	توضیح
$P \triangleleft X$	شرکت‌کننده P مقدار X را احتمالاً بعد از انجام برخی محاسبات مانند رمزگشایی دریافت می‌کند.
$P \triangleleft *X$	شرکت‌کننده P مقدار X را دریافت می‌کند که آن را قبلاً در جلسه جاری منتقل نکرده است.
$\{X\}_K$	پیام X با استفاده از کلید K رمز شده است.
$P \mid\approx X$	شرکت‌کننده P پیام X را ارسال کرده است.
$P \ni X$	شرکت‌کننده P دارای و یا قادر به تملک پیام X است.
$P \overset{K}{\leftrightarrow} Q$	شرکت‌کنندگان P و Q با یکدیگر از طریق کلید رمزنگاری K ارتباط برقرار می‌کنند.
$P \mid\equiv \emptyset(X)$	شرکت‌کننده P باور دارد و یا حق دارد باور کند که پیام X قابل تشخیص است. به این معنا که، P پیام X را تشخیص خواهد داد اگر P انتظار خاصی در مورد محتوای X قبل از اینکه پیام X را واقعاً دریافت کند داشته باشد. P ممکن است یک مقدار خاص (به‌طور مثال هویت خودش) و یک ساختار خاص (به‌طور مثال نوع مهرزمانی) را تشخیص دهد.
$P \mid\equiv P \overset{S}{\leftrightarrow} Q$	شرکت‌کننده P باور دارد و یا حق دارد باورکند که مقدار S یک مقدار مخفی مناسب برای P و Q است. آنها ممکن است از آن برای اثبات هویت متقابل و یا همچنین کلیدی برای برقراری ارتباط استفاده کنند. این علائم اختصاری به صورت متقارن $P \overset{S}{\leftrightarrow} Q$ و $Q \overset{S}{\leftrightarrow} P$ می‌تواند به جای هم استفاده شوند.
$P \mid\equiv \overset{+K}{\mapsto} Q$	شرکت‌کننده P باور دارد و یا حق دارد باور کند که مقدار $+K$ کلید عمومی Q است. کلید خصوصی K هرگز با هیچ شرکت‌کننده‌ای بجز Q یا یک نهاد مورداعتماد با Q کشف نخواهد شد.

جزئیات کامل به صورت زیر تعریف می‌شوند [۱، ۱۳].

- IR1: قانون T_1 بیان می‌کند که گفتن یک رابطه به معنای گفتن هر یک از اجزای آن است. همچنین قانون T_2 نشان می‌دهد اگر یک طرف ارتباطی مقدار X رمزشده و کلید را بگوید در نتیجه او محتوای رمزگشایی‌شده آن پیام را نیز گفته است.
- IR2: قانون P_1 نشان می‌دهد یک طرف ارتباطی قادر است هر چیزی را که او گفته است در اختیار داشته باشد. همچنین قانون P_2 بیان می‌کند اگر یک طرف ارتباطی دارای دو مقدار باشد در نتیجه او قادر به داشتن الحاق دو مقدار و یک تابع F از آن‌ها نیز است.
- IR3: قوانین تازگی بیان می‌کند که باور یک طرف ارتباطی به تازگی یک رابطه، نشان‌دهنده این است که رابطه هرگز ارزش یکسانی در هیچ یک از اجراهای قبلی پروتکل ندارد. قانون F_1 نشان می‌دهد اگر P باور دارد که X تازه است، سپس او حق دارد باور کند که هر رابطه‌ای که X یک جزء از آن است نیز تازه است و تابع F از X یک‌به‌یک و از لحاظ محاسباتی شدنی و تازه است. قانون F_2 نشان می‌دهد اگر P با کلید K باور دارد که X تازه است، سپس P حق دارد باور کند که هر ترکیب از رمزنگاری و حتی رمزگشایی X با آن کلید نیز تازه است.
- IR4: قوانین قابلیت تشخیص، اعتقادات و باور یک طرف ارتباطی را درمورد تشخیص رابطه‌های دیگر بیان می‌کند. قانون R_1 نشان می‌دهد اگر P باور دارد که X قابلیت تشخیص دارد، سپس او حق دارد که باور کند که هر ترکیبی از X نیز قابل تشخیص است و یا محاسبه تابع شدنی است. همچنین، قانون R_2 بیان می‌کند اگر P باور دارد که X قابلیت تشخیص دارد و دارای کلید K است، سپس P حق دارد باور کند که رمزنگاری و رمزگشایی X با کلید K نیز قابلیت تشخیص دارد.
- IR5: در اینجا قوانینی داریم که طرفین ارتباطی را قادر می‌سازد تا با بررسی پیام‌هایی که آن‌ها دریافت می‌کنند، باورهای خود را در مورد سایر طرفین پیش ببرند. قانون I_1 نشان می‌دهد اگر برای P، همه این شرایط برقرار باشد: (۱) P یک مقدار X رمزشده با K دریافت کند. (۲) P دارای کلید K باشد. (۳) P باور دارد که K یک مقدار مخفی مناسب برای خودش و Q است. (۴) P باور دارد که رابطه X قابلیت تشخیص دارد. (۵) P باور دارد که K یا X تازه هستند. درنتیجه، P حق دارد که چنین مواردی را باور کند: (۱) Q یک مرتبه X را منتقل کرده است. (۲) Q یک مرتبه X رمزشده با کلید K را منتقل کرده است و (۳) Q مالک کلید K است. همچنین، قانون I_2 بیان می‌کند، اگر P باور دارد که Q یک مرتبه X را منتقل کرده است و P باور دارد که X تازه است در نتیجه، P حق دارد باور کند که Q مالک کلید K است.
- IR6: فرضیه‌ها را با قانونی تکمیل می‌کنیم که به آن قانون عقلانیت گوییم. این قانون بیان می‌کند اگر Q باور دارد که P قادر به دراختیار داشتن دو مقدار است، سپس Q باور دارد که P قادر به در اختیار داشتن ترکیب این دو مقدار و هر تابع Fای از آنها نیز است. قانون عقلانیت نشان می‌دهد که طرفین ارتباطی قادر به اخذ نتایج منطقی در مورد وضعیت طرف‌های دیگر هستند.

معمولاً روش اثبات امنیت پروتکل‌های امنیتی با منطق GNY در چند مرحله به صورت زیر خلاصه می‌شود:

- گام یک (بیان پیام‌های پروتکل به صورت عبارت‌های ریاضی): در این گام، پیام‌های مبادله‌شده در پروتکل به صورت عبارت‌های ریاضی مشخص می‌شوند.
- گام دوم (بیان پیام‌های پروتکل مبتنی بر علائم اختصاری منطق GNY): در این گام، پیام‌های مبادله‌شده با استفاده از علائم منطق GNY بیان می‌شوند. باید توجه داشت که به دلیل سهولت تجزیه‌وتحلیل، پیام‌های ارسال‌شده در پروتکل به عنوان تابعی از ورودی‌های آن‌ها نوشته می‌شوند.
- گام سوم (تعیین شکل ایده‌ال‌شده پروتکل با منطق GNY): در این

جدول ۸. برخی از قوانین منطق GNY [۱۳]

توضیح	علائم
IR1: قوانین گفته شدن	$T_1 = \frac{P \triangleright (X,Y)}{P \triangleright X}$
	$T_2 = \frac{P \triangleright \{X\}_K, P \ni K}{P \triangleright X}$
IR2: قوانین مالکیت	$P_1 = \frac{P \triangleright X}{P \ni K}$
	$P_2 = \frac{P \ni X, P \ni Y}{P \ni (X,Y), P \ni F(X,Y)}$
IR3: قوانین تازگی	$F_1 = \frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y), P \mid\equiv \#(F(X))}$
	$F_2 = \frac{P \mid\equiv \#(X), P \ni K}{P \mid\equiv \#(\{X\}_K), P \mid\equiv \#(\{X\}_K^{-1})}$
IR4: قوانین قابلیت تشخیص	$R_1 = \frac{P \mid\equiv \emptyset(X)}{P \mid\equiv \emptyset(X,Y), P \mid\equiv \emptyset(F(X))}$
	$R_2 = \frac{P \mid\equiv \emptyset(X), P \ni K}{P \mid\equiv \emptyset(\{X\}_K), P \mid\equiv \emptyset(\{X\}_K^{-1})}$
IR5: قوانین تفسیر پیام	$I_1 = \frac{P \triangleright * \{X\}_K, P \ni K, P \mid\equiv P \overset{K}{\rightleftharpoons} Q, P \mid\equiv \emptyset(X), P \mid\equiv \#(X,K)}{P \mid\equiv Q \mid\approx X, P \mid\equiv Q \mid\approx \{X\}_K, P \mid\equiv Q \ni X}$
	$I_2 = \frac{P \mid\equiv Q \mid\approx X, P \mid\equiv \#(X)}{P \mid\equiv Q \ni X}$
IR6: قانون عقلانیت	$M_1 = \frac{Q \mid\equiv P \ni X, Q \mid\equiv P \ni Y}{Q \mid\equiv P \ni F(X,Y)}$

گام، پیام‌های مبادله‌شده به شکل منطق GNY تبدیل می‌شوند. به عبارتی، پیام‌های مبادله‌شده ایده‌آل می‌شوند. این بدین معنی است که، پیام‌های ارسال‌شده‌ی رمزنشده که موجب افزایش امنیت نمی‌شوند حذف می‌گردد.

- گام چهارم (تعریف فرض‌ها و اهداف امنیتی پروتکل): در این مرحله فرض‌هایی جهت شروع فرایند و اهداف امنیتی به منظور بررسی امنیت در نظر گرفته می‌شود.
- گام پنجم (حصول اهداف امنیتی): در این گام، با استفاده از فرض‌های پروتکل و طبق قوانین منطق GNY، اهداف امنیتی به دست خواهند آمد.

۸ مقایسه ابزارها و روش‌های تحلیل و اثبات امنیت پروتکل‌های امنیتی

به طور کلی، ابزارهای مختلفی با کاربردهای متعدد، جهت تحلیل و اثبات امنیت پروتکل‌های احراز اصالت وجود دارد. مشاهده شده است، در بعضی مواقع، تحلیل یک پروتکل امنیتی با استفاده از یک ابزار و مدل امن و با استفاده از ابزار و مدل دیگر ناامن تشخیص داده می‌شود. بنابراین، ضروری است که طراحان پروتکل‌های امنیتی با توجه به هدف موردنظر خود، از ابزارهای تحلیل امنیت مختلفی استفاده نمایند. از این‌رو، نتایج مقایسه این ابزارها و روش‌های تحلیل امنیت در جدول ۹ بیان شده است.

۹ جمع‌بندی

استفاده از ابزارها، روش‌های اثبات و تحلیل امنیت خودکار به دلیل اهمیت تحلیل امنیتی پروتکل‌های احراز اصالت ارائه‌شده در حوزه‌های مختلف اینترنت اشیاء از قبیل پزشکی، حمل‌و نقل و غیره امری بسیار ضروری است. در این مقاله به معرفی انواع روش‌های اثبات و تحلیل صوری پروتکل‌های امنیتی پرداخته شد. هم‌چنین، مشاهده شده درصورتی‌که ابزار موردنظر، مدل امنیتی در نظر گرفته شده برای پروتکل مطرح‌شده را امن اعلام نکند، باید تدابیر امنیتی مناسبی برای پروتکل در نظر گرفته شود. در هر کدام از این ابزارها، پروتکل با استفاده از فرمان‌های مختلف اجرا شده و خروجی مورد تحلیل و بررسی قرار خواهد گرفت. به‌طورکلی، ابزارهای مختلفی جهت تحلیل و اثبات امنیت پروتکل‌های احراز اصالت وجود دارد. به دلیل متفاوت بودن فضاهای حالت و سناریوهای حمله، برای تحلیل امنیت پروتکل‌های احراز اصالت از ابزارهای تحلیل امنیت مختلفی استفاده می‌کنند. به بیانی دیگر، در بعضی مواقع، تحلیل یک پروتکل امنیتی با استفاده از یک ابزار و مدل امن و با استفاده از ابزار و مدل دیگر ناامن تشخیص داده می‌شود، بنابراین ضروری است که طراحان پروتکل‌های امنیتی جهت اثبات و تحلیل امنیت پروتکل‌های احراز اصالت از ابزارها و مدل‌های مختلفی استفاده نمایند. هم‌چنین، علاوه بر بیان اثبات و تحلیل امنیت پروتکل ECCPWS با استفاده از برخی از روش‌های تحلیل و اثبات امنیت صوری پروتکل‌های امنیتی از قبیل سایتر، اویسپا، مدل پیشگوی تصادفی و منطق BAN به مقایسه این ابزارهای تحلیل امنیت از جهات نقاط قوت، نقاط ضعف، قابلیت‌ها و کاربردها پرداخته شد.

واژه‌نامه

ابزاری جهت تحلیل امنیت پروتکل‌های امنیتی ..... ProVerif
مدلی جهت اثبات امنیت پروتکل‌های امنیتی ..... Real Or Random
ابزاری جهت تحلیل امنیت پروتکل‌های امنیتی ..... AVISPA
منطقی جهت اثبات امنیت پروتکل‌های امنیتی ..... BAN Logic
ابزاری جهت تحلیل امنیت پروتکل‌های امنیتی ..... Scyther
منطقی جهت اثبات امنیت پروتکل‌های امنیتی ..... GNY Logic
ابزاری جهت تحلیل امنیت پروتکل‌های امنیتی ..... TAMARIN
مدل مهاجم خارجی ..... EXTernal model
مدل مهاجم داخلی ..... ENTernal model
مدل مهاجم دالو یائو ..... Dolev Yao model
مدل مهاجم بازیگر تسخیرشده ..... Compromised Actor (CA) model
مدل مهاجم ..... AFter model
مدل مهاجم ..... AFter Correct model
رازمانی پیش سوی کامل ..... Perfect Forward Secrecy
رازمانی پس سوی کامل ..... Perfect Backward Secrecy
مدل مهاجم ..... Bellare Rogaway model
مدل مهاجم ..... Canetti Krawczyk model
مدل مهاجم ..... Ephemeral Canetti Krawczyk model
ادعای امنیتی محرمانگی ..... Secret

جدول ۹. مقایسه انواع روش‌های اثبات و تحلیل پروتکل‌های امنیتی

ابزار	ویژگی‌ها
اویسپا	• یک نوع ابزار خودکار جهت تحلیل پروتکل‌های امنیتی • پروتکل با استفاده از زبان HLPSL مدل می‌شود. • عدم بررسی حملات چند پروتکلی با استفاده از این ابزار • نمودار و گراف حمله در این ابزار نشان داده نمی‌شود. • سناریو حمله نشان داده نمی‌شود. • تأیید پروتکل تنها با استفاده از جلسات محدود صورت می‌پذیرد. • مدل مهاجم آن: DY و KCI • دانش مهاجم به این ابزار داده می‌شود. • بررسی دو حمله تکرار و فرد در میانه
سایتر	• یک نوع ابزار خودکار جهت تحلیل پروتکل‌های امنیتی • پروتکل با استفاده از زبان SPDL مدل می‌شود. • بررسی حملات چند پروتکلی در این ابزار • در صورت وجود حمله، نمودار و گراف حمله نشان داده می‌شود. • درستی‌یابی و اجرای پروتکل با تعداد نامحدود و محدود جلسات انجام می‌شود. • این ابزار با رابط کاربر گرافیکی ارائه می‌شود. • این ابزار خروجی‌های احتمالی تولید می‌کند. • در صورت وجود حمله، گراف حملاتی که نشان می‌دهد نشان‌دهنده حمله و ردیابی حمله است. • طرفین شرکت‌کننده به عنوان نقش‌ها مدل می‌شوند. • این ابزار محرمانگی همه‌ی متغیرهای ممکن را با صلاحدید خود بررسی می‌نماید و هیچ ادعای صریحی لازم نیست. • شرکت‌کنندگان قانونی و مهاجم با agent مشخص می‌شوند. • در این ابزار مفهومی به نام کانال وجود ندارد.
پروویریف	• یک نوع ابزار خودکار جهت تحلیل پروتکل‌های امنیتی • پروتکل با استفاده از horn clauses یا pi calculus مدل می‌شود. • ابزار بایستی از طریق خط فرمان اجرا شود. • ردیابی به صورت گام‌به‌گام انجام می‌شود که اجرای پروتکل و سناریوی حمله را توضیح می‌دهد. • ردیابی تنها برای ویژگی که در دست بررسی است تولید می‌شود. • شرکت‌کنندگان ارتباطی به عنوان processها مدل می‌شوند. • تساوی با استفاده از "if… then" و یا "let…in" بررسی می‌گردد. • این ابزار تنها حملاتی را بررسی می‌نماید که "query" آن در کد مشخص شده است. • جهت بررسی حمله تکرار و عدم تازگی کلید جلسه، هیچ کد خاصی نیاز نیست. • در این ابزار پروتکل ممکن است تنها برای تعداد نامحدودی جلسه اجرا شود. • جهت ارتباط باید کانال‌ها تعیین گردند. • در این ابزار کاربر می‌تواند سناریو حمله را بنویسد.
تامارین	• یک نوع ابزار خودکار جهت تحلیل پروتکل‌های امنیتی • زبان برنامه‌نویسی هاسکل • پیاده‌سازی دقیق پیام‌های ارسال‌شده در یک پروتکل با مجموعه‌ای از قوانین بازنویسی‌شده • طرفین شرکت‌کننده به عنوان نقش‌ها مدل می‌شوند. • این ابزار دو روش اثبات امنیت را ارائه می‌دهد. یا اثبات صحت را برمی‌گرداند و یا سناریوی یک حمله را نشان می‌دهد.
منطق BAN	• روش دستی جهت تحلیل پروتکلهای امنیتی • محدود به قوانین • مدل دشمن دالو یائو • دارای فضای حالت محدود • استفاده از مفاهیم ریاضی در این منطق • فرض می‌شود توابع رمزنگاری مورداستفاده کامل هستند. به دلیل اینکه جزئیات توابع نوشته نمی‌شود، فرض می‌شود توابع مورد استفاده قوی‌ترین هستند.
منطق GNY	• روش دستی جهت تحلیل پروتکلهای امنیتی • این روش محدود به قوانین • بسیار شبیه به منطق BAN بوده با این تفاوت که قوانین بیش‌تر و کامل‌تری را شامل می‌شود. • مدل دشمن دالو یائو • دارای فضای حالت محدود • استفاده از مفاهیم ریاضی در این منطق • فرض می‌شود توابع رمزنگاری مورداستفاده کامل هستند. به دلیل اینکه جزئیات توابع نوشته نمی‌شود، فرض می‌شود توابع مورد استفاده قوی‌ترین هستند.
مدل اوراکل تصادفی	• روش دستی جهت تحلیل امنیت پروتکلهای امنیتی • مدل مهاجم: دالو یائو • استفاده از تعداد فرضیات بیش‌تری در تحلیل پروتکل • همه طرفین شرکت‌کننده از جمله مهاجم با یکدیگر تعامل دارند.

ادعای امنیتی برخط بودن ..... Alive
ادعای امنیتی توافق ضعیف ..... Weak Agreement
ادعای امنیتی غیر یک به یک ..... Non Injective Agreement
ادعای امنیتی یک به یک ..... Injective Agreement
ادعای امنیتی تازگی و جدید بودن ..... Recentness
ادعای امنیتی همزمان‌سازی ..... Synchronization
ادعای امنیتی دردسترس‌پذیری ..... Reachability
حملات چند پروتکلی در ابزار سایتر ..... Multi protocol attacks
کلسیی، اسشنیر واگنر ..... Kelsey Schneier Wagner
زبان برنامه‌نویسی ابزار تحلیل امنیت تامارین ..... Haskell
قانون تازگی ..... Freshness Rule
قانون باور و اعتقاد ..... Belief Rule
قانون کنترل ..... Jurisdiction Rule
قانون تایید عدد یک‌بار مصرف ..... Nonce Verification Rule
قانون معنی پیام ..... Message Meaning Rule
قانون باور توابع چکیده‌ساز ..... Hash Functions Belief Rule
قوانین مالکیت ..... Being Told Rules
قوانین قابلیت تشخیص ..... Recognizability Rules
قوانین تفسیر پیام ..... Message Interpretation Rules
قانون عقلانیت ..... Rationality Rule
قوانین تفسیر پیام ..... Message Interpretation Rules
هستار ..... Entity
سبک وزن ..... Lightweight
قانون حذف پیام‌های چند قسمتی ..... Elimination of Multipart Message Rule

فهرست علائم و اختصارات

AVISPA Automated Validation of Internet Security Protocols and Application
BAN ..... Burrows Abadi Needham
CL-AtSe ..... Constraint Logic based Attack Searcher.BAN Logic
ECCPWS ..... ECC based Protocol for WBAN Systems
GNY Logic ..... Gong Needham Yahalom
HLPSL ..... High Level Protocols Specification Language
IF ..... Intermediate Format
IR ..... Inference Rules
OF ..... Output Format
OFMC ..... On the Fly Model Checker
ProVerif ..... Protocol Verifivation
RoR ..... Real Or Random
SPAN ..... Security Protocol ANimator for AVISPA
SATMC ..... SAT Based Model Checker
SPDL ..... Security Protocol Description Language
TA4SP ..... Tree Automata based Protocol Analyzer
WBAN ..... Wireless Body Area Networks
WHMS ..... Wearable Health Monitoring Systems

CA ..... Compromised Actor
AF ..... AFter model
AFC ..... AFter Correct model
CK ..... Canetti Krawczyk model
BR ..... Bellare Rogaway model
eCK ..... Ephemeral Canetti Krawczyk model
SeCK ..... Strengthened Ephemeral Canetti Krawczyk model
SPAN ..... Security Protocol ANimator
VO ..... Van Oorschot Logic
SVO ..... Syverson Van Oorschot Logic
IoT ..... Internet of Things
PFS ..... Perfect Forward Secrecy

پیوست یک

پیاده‌سازی پروتکل پیشنهادی ECCPWS با استفاده از ابزار سایتر

```
hashfunction h;
hashfunction ECC;
const ADD: Function;
secret B;
secret idu;
secret t;
secret y;
secret r;
usertype Timestamp;
macro IndU=ECC(t,B);
macro UU= {t}pk(U);
macro x=h(UU,IndU);
macro v=ADD({x}sk(NM),t);
macro R=ECC(r,B);
macro Qprim={R}sk(AS);
macro c=ADD(v,r, sk(U));
macro EU={C,UU,IndU,T1}pk(AS);
macro Y=ECC(y,B);
macro Auth=h(UU,R, Qprim,Y,T3);
macro M=ECC(idu,B);
protocol @oracle (X){
role X {
var Y:Agent;
const P;
recv_!X1(X, X, ECC(X,ECC(Y,P)));
send_!X2(X, X, ECC(Y,ECC(X,P)));
}
}
protocol @mad (X){
role X {
var Y:Agent;
const P;
recv_!X1(X,X,ECC(sk(Y),pk(X)));
send_!X2(X,X,ECC(sk(X),pk(Y)));
}
}
protocol @oracleM (X){
role X {
var Y:Agent;
const P;
var r;
fresh y;
recv_!X1(X,X,ECC(r,ECC(y,B)));
send_!X2(X,X,ECC(y,ECC(r,B)));
}
}
protocol proposed (U,NM, AS){
role U {
secret B;
secret idu;
fresh r;
fresh T1: Timestamp;
fresh T4: Timestamp;
var y;
var t;
var T3: Timestamp;
var T2: Timestamp;
send_1 (U, NM, {M,PK(U)}k(U,NM));
recv_2 (NM, U, {IndU,v}k(U,NM));
send_3 (U, AS, EU,R,T1);
recv_4 (AS, U, Auth,Y,T3);
claim(U, Alive);
claim(U, Weakagree);
claim(U, Niagree);
claim(U, Secret, sk(U));
claim(U, Secret, r);
};
role NM{
secret B;
secret idu;
fresh t;
recv_1(U, NM, {M,PK(U)}k(U,NM));
send_2(NM, U, {IndU,v}k(U,NM));
claim(NM, Alive);
claim(NM, Weakagree);
claim(NM, Niagree);
claim(NM, Secret, sk(NM));
};
role AS{
var r;
fresh y;
fresh T2: Timestamp;
fresh T3: Timestamp;
var t;
var T1: Timestamp;
var T4: Timestamp;
```

```
secret B;
secret idu;
recv_3 (U, AS, EU,R,T1);
recv_4 (AS, U, Auth,Y,T3);
claim(AS, Alive);
claim(AS, Weakagree);
claim(AS, Niagree);
claim(AS, Secret, sk(AS));
};
};
```

پیوست دو

پیاده‌سازی پروتکل پیشنهادی ECCPWS با استفاده از ابزار اویسپا

```
role alice(U, S, A:agent, SKus:symmetric_key, MUL,H,Add
    ↪ ,ECC:hash_func,SND,RCV:channel(dy))
played_by U
def=local State:nat,
SU, B, PKU, IDU, UU, X, PKas, T1, T3, R,Q, Kua, C, AuthU,
    ↪ SK:text, ID, IndU,V, RR, EU, Auth, YY:text,
const auth_1, auth_2, auth_3, subs1, subs2, subs3,
    ↪ subs4, subs5, subs6:
protocl_id
init State:=0
transition
1. State=0 ^ RCV(start) =|>
State ':= 1 ^ IDU':=new() ^ PKU':=ECC(SU'.B) ^ ID':=ECC
    ↪ (IDU'.B)
^ secret ({SU,IDU}, subs1, {U})
^ SND ({ID'}_SKus. {PKU'}_SKus)
2. State=1 ^RCV({IndU'}_SKus .{V'}_SKus)=|>
State ':= 2 ^ UU':= MUL(SU. IndU) ^ X':=H(UU'. IndU) ^
    ↪ Kua':=MUL(SU. PKas) ^ R':=new() ^ T1':=new() ^
    ↪ RR':=ECC(R'. B) ^ Q':=MUL(R'. PKas)
^ C':=Add(V. R'. SU) ^ EU':={C'. UU'. IndU. T1'}_ Kua'
^ SND (EU'. RR'. T1)
^ secret ({R', C', UU'}, subs2, {U,A})
^ witness(U, A, auth_1, C)
^ request(S, U, auth_3, UU)
3. State=2 ^ RCV(Auth', YY', T3') =|>
State ':= 3
^ AuthU':= H(UU. RR. Q. YY'. T3')
^ SK':=H(AuthU'. UU. R,YY')
^ secret ({SK'}, subs3, {U,A})
^ request(A, U, auth_2, Auth')
end role
role server (S, U, A:agent, SKus:symmetric_key,H,MUL,
    ↪ ECC,Add:hash_func,
SND,RCV:channel(dy))
played_by S
def=
local State:nat,
SN, UU, K, X, B, PKU:text, ID, IndU, V:text,
const auth_1, auth_2, auth_3, subs1,subs2,subs3,subs4,
    ↪ subs5,subs6:protocol_id
init State:=0
transition
1. State=0 ^ RCV({ID}_SKus. {PKU}_SKus) =|>
State ':= 1 ^ SN': new() ^ K':=new() ^ IndU':=ECC(K'. B)
^ UU':=MUL(K'. PKU) ^ X':=H(UU'. IndU)
^ V':=Add(MUL(X'. SN). K')
^ SND ({IndU}_SKus, {V'}_SKus)
^ secret ({SN'}, subs6, {S})
^ witness(S, U, auth_3, UU)
end role
role bob (A, S, U: agent, SKus:symmetric_key, H,ECC,MUL
    ↪ ,Add:hash_func, SND,RCV:channel(dy))
played_by A
def=
local State:nat,
SA, SU, PKas, B, Q, X, PKU, KKua, T1,T3, Z, Y, UU, C:text
    ↪ , YY, Auth, EU,
RR, IndU: text,
const auth_1, auth_2, auth_3,
subs1, subs2, subs3, subs4, subs5, subs6: protocol_id
init State:=0
transition
1. State=0 ^ RCV(EU', RR, T1') =|>
State ':= 1 ^ SA':= new()
^ PKas':=ECC(SA'. B)
^ secret ({SA'}, subs5, {U, A, S})
^ QQ':=MUL(SA. RR') ^ KKua':=MUL(SA. PKU') ^ X':=H(UU'.
    ↪ IndU) ^ Y':=new()
^ T3':=new() ^ YY':=ECC(Y'. B) ^ Z':=MUL(Y' . RR')
^ Auth':=H(UU. RR'. QQ. YY'. T3') ^ SK':=MUL(Auth'. UU.
    ↪ Z')
^ SND (Auth'. YY'. T3')
^ secret ({SK}, subs4, {A,U})
^ request(U, A, auth_1, C)
^ witness(S, U, auth_2, Auth')
end role
role session (U, S, A: agent, SKus:symmetric_key, MUL,
    ↪ ECC,Add:hash_func)
def=local
SND1, RCV1, SND2, RCV2, SND3, RCV3:
channel(dy)
^ composition
alice(U, S, A, SKus, H, ECC,Add,MULSND1, RCV1)
^server(U, S, A, SKus, H, ECC,Add, MUL,SND2, RCV2)
^bob(U, S, A, SKus, H,ECC, Add,MUL, SND3, RCV3)
end role
role environment()
```

```
def=const u,s,a:agent,
skus: symmetric_key,
ecc,mul,h,add: hash_func,
v,eu,rr,auth,yy,indu,id:text,
auth_1, auth_2, auth_3, subs1, subs2,
subs3, subs4, subs5, subs6: protocl_id
intrusder_knowledge={u, s, a,mul, h, f, ecc,add, id,
indu, v, eu, rr, auth, yy}
composition
session(u, s, a, skus, h, mul,ecc,add)
^session(i, s, a, skus, h, mul,eccadd)
^session(u, i, a, skus, h, mul,ecc,add)
^session(u, s, i, skus, h, mul,ecc,add)
end role
goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
secrecy_of subs6
authentication_on auth_1
authentication_on auth_2
authentication_on auth_3
end goal
environment()
```

پیوست سه

پیاده‌سازی پروتکل پیشنهادی ECCPWS با استفاده از ابزار پروورِیف

```
(***********Declarations********)
(***********Channels************)
free ChSec: channel [private].
free ChPub: channel.
(***********Constants and Variables****)
free IDU: bitstring.
free S_NM: bitstring [private].
free PK_NM: bitstring.
free S_U: bitstring [private].
free S_AS: bitstring [private].
free PK_AS: bitstring.
free PK_U: bitstring.
free B: bitstring.
free SK: bitstring [private].
free IDS: bitstring.
free xIndU: bitstring.
free xUU: bitstring.
free xc: bitstring.
(****Functions and Constructors*********)
fun h1(bitstring): bitstring.
fun CONCAT(bitstring, bitstring): bitstring.
fun ADD(bitstring, bitstring): bitstring.
fun ECMul(bitstring , bitstring): bitstring.
fun MULT(bitstring, bitstring): bitstring.
fun EKUA(bitstring): bitstring.
 (*************User Process**********)
let pU= event start_U (IDU);
let ID= ECMul (IDU, B) in
out(ChSec, (ID, PK_U));
in (ChSec, (IndU: bitstring, v: bitstring));
let UU=MULT (S_U, IndU) in
let x=h1(CONCAT (UU, IndU)) in
let vB =ECMul(v, B) in
let 'vB =ADD)MULT(x, PK_NM), IndU) in
if (vB='vB) then
let K_UA=MULT(S_U, PK_AS) in
new r: bitstring;
new T1: bitstring;
let R= ECMul (r, B) in
let Q=MULT(r, PK_AS) in
let c= ADD (ADD (v, r), S_U) in
let EU= EKUA(CONCAT (UU, (IndU, T1, c))) in
out (ChPub, (EU, R, T1));
in (ChPub,(Auth: bitstring, Y: bitstring, T3: bitstring
    ↪ ));
let xAuth=h1( CONCAT (UU , (R, Q, Y, T3))) in
if (Auth=xAuth) then
let xSK=h1(CONCAT(Auth, (UU, MULT(r, Y))) in
event end_U (IDU)
else 0.
let pNM=
event start_NM (IDN);
in (ChSec, (xID: bitstring, xPK_U: bitstring));
new k:bitstring;
let IndU=ECMul(k, B) in
let UU=MULT(k, PK_U) in
let x=h1(CONCAT(UU, IndU)) in
let v=ADD (MULT (x, S_NM), k) in
out (ChSec, (IndU, v)),
event end_NM (IDN)
let pS=
event start_S (IDS)
in (ChPub, (xEU:bitstring, R: bitstring, T1: bitstring)
    ↪ );
let 'Q= MULT (S_AS, R) in
let 'KUA=MULT(S_AS, PK_U) in
let x=h1(CONCAT (XUU, xIndU)) in
let xcB=ECMul (xc, B)
let cB=ADD ((ADD((ADD(MULT(x, PK_NM), R)), PK_U)),
    ↪ XIndU) in
if (xcB=cB) then
```

```
new y:bitstring;
new T3: bitstring;
let Y = ECMul(y, B ) in
let Z = MULT(y, R) in
let Auth = h1(CONCAT(xUU,(R, 'Q, Y, T3))) in
let xSK=h1(CONCAT(Auth, (xUU, Z))) in
out(ChPub, (Auth, Y, T3));
event end_S (IDS)
else 0.
(**********Events***************)
event start_U (bitstring).
event end_U (bitstring).
event start_S (bitstring).
event end_S (bitstring).
event start_NM (bitstring).
event end_NM (bitstring).
(*****main and Process Replication*******)
process ((!pS) | (!pU) | (!pNM))
(******************Queries*************)
query id: bitstring; inj-event(end_U (IDU)) ==> inj-
    ↪ event(start_U(IDU)).
query id: bitstring; inj-event(end_S(IDS))==> inj-
    ↪ event (start_S(IDS)).
query id: bitstring; inj-event(end_S(IDN))==> inj-
    ↪ event (start_S(IDN)).
query attacker (SK).
```

مراجع

[1] Pirmoradian, Fatemeh. *Design and Security Analysis of Protocols Used in Telecare Medicine Information Systems (TMIS)*. Ph.D. thesis, Isfahan University of Technology, 2023.

[2] Nikkhah, F. Improving the privacy security of telecare medical information systems. Master's thesis, Shahid Rajaee Training University, 2020.

[3] Dolev, Danny and Yao, Andrew. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[4] Basin, David and Cremers, Cas. Know your enemy: Compromising adversaries in protocol analysis. *ACM Transactions on Information and System Security (TISSEC)*, 17(2):1–31, 2014.

[5] Do, Quang, Martini, Ben, and Choo, Kim-Kwang Raymond. The role of the adversary model in applied security research. *Computers & Security*, 81:156–181, 2019.

[6] Cremers, Cas and Feltz, Michele. Beyond eck: Perfect forward secrecy under actor compromise and ephemeral-key reveal. in *Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings 17*, pp. 734–751. Springer, 2012.

[7] Sarr, Augustin P, Elbaz-Vincent, Philippe, and Bajard, Jean-Claude. A new security model for authenticated key agreement. in *Security and Cryptography for Networks: 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings 7*, pp. 219–234. Springer, 2010.

[8] Cremers, Cas JF. Session-statereveal is stronger than ecks ephemeralkeyreveal: using automatic analysis to attack the naxos protocol. *International Journal of Applied Cryptography*, 2(2):83–99, 2010.

[9] Takkinen, Laura. Analysing security protocols with avispa. in *TKK T-110.7290 research seminar on network security*, vol. 12, 2006.

[10] Cremers, Cas JF. The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper. in *International conference on computer aided verification*, pp. 414–418. Springer, 2008.

[11] Küsters, Ralf and Truderung, Tomasz. Using proverif to analyze protocols with diffie-hellman exponentiation. in *2009 22nd IEEE Computer Security Foundations Symposium*, pp. 157–171. IEEE, 2009.

[12] Cremers, Cas. Symbolic security analysis using the tamarin prover. in *2017 Formal Methods in Computer Aided Design (FMCAD)*, pp. 5–5. IEEE, 2017.

[13] Gong, Li, Needham, Roger M, and Yahalom, Raphael. Reasoning about belief in cryptographic protocols. in *IEEE Symposium on Security and Privacy*, vol. 1990, pp. 234–248. Citeseer, 1990.

[14] Narwal, Bhawna and Mohapatra, Amar Kumar. A survey on security and authentication in wireless body area networks. *Journal of Systems Architecture*, 113:101883, 2021.

[15] Burrows, Michael, Abadi, Martin, and Needham, Roger. A logic of authentication. *ACM Transactions on Computer Systems (TOCS)*, 8(1):18–36, 1990.

[16] Das, Ashok Kumar, Wazid, Mohammad, Yannam, Animi Reddy, Rodrigues, Joel JPC, and Park, Youngho. Provably secure ecc-based device access control and key agreement protocol for iot environment. *IEEE Access*, 7:55382–55397, 2019.

[17] Das, Ashok Kumar, Wazid, Mohammad, Kumar, Neeraj, Vasilakos, Athanasios V, and Rodrigues, Joel JPC. Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment. *IEEE Internet of Things Journal*, 5(6):4900–4913, 2018.

[18] Amintoosi, Haleh, Nikooghadam, Mahdi, Shojafar, Mohammad, Kumari, Saru, and Alazab, Mamoun. Slight: A lightweight authentication scheme for smart healthcare services. *Computers and Electrical Engineering*, 99:107803, 2022.

[19] Meshram, Chandrashekhar, Obaidat, Mohammad S, Ibrahim, Rabha W, Meshram, Sarita Gajbhiye, and Raikwar, Arpit Vijay. An efficient privacy-preserved authentication technique based on conformable fractional chaotic map for tmis under smart homes environments. *The Journal of Supercomputing*, 80(2):2514–2537, 2024.

[20] Hosseinzadeh, Mehdi, Malik, Mazhar Hussain, Safkhani, Masoumeh, Bagheri, Nasour, Le, Quynh Hoang, Tightiz, Lilia, and Mosavi, Amir H. Toward designing a secure authentication protocol for iot environments. *Sustainability*, 15(7):5934, 2023.

[21] Roy, Prasanta Kumar and Bhattacharya, Ansuman. An anonymity-preserving mobile user authentication protocol for global roaming services. *Computer Networks*, 221:109532, 2023.

[22] Qiao, Hui, Dong, Xuewen, Jiang, Qi, Ma, Siqi, Liu, Chao, Xi, Ning, and Shen, Yulong. Anonymous lightweight authenticated key agreement protocol for fog-assisted healthcare iot system. *IEEE Internet of Things Journal*, 2023.

[23] Wu, Tsu-Yang, Wang, Liyang, and Chen, Chien-Ming. Enhancing the security: A lightweight authentication and key agreement protocol for smart medical services in the ioht. *Mathematics*, 11(17):3701, 2023.

[24] Kumar, Ajay, Abhishek, Kumar, Liu, Xuan, and Haldorai, Anandakumar. An efficient privacy-preserving id centric authentication in iot based cloud servers for sustainable smart cities. *Wireless Personal Communications*, 117(4):3229–3253, 2021.

[25] Son, Seunghwan, Park, Yohan, and Park, Youngho. A secure, lightweight, and anonymous user authentication protocol for iot environments. *Sustainability*, 13(16):9241, 2021.

[26] Hajian, Rahman, Haghighat, A, and Erfani, Seyed Hossein. A secure anonymous d2d mutual authentication and key agreement protocol for iot. *Internet of Things*, 18:100493, 2022.

[27] Alzahrani, Bander A, Chaudhry, Shehzad Ashraf, Barnawi, Ahmed, Al-Barakati, Abdullah, and Shon, Taeshik. An anonymous device to device authentication protocol using ecc and self certified public keys usable in internet of things based autonomous devices. *Electronics*, 9(3):520, 2020.

[28] Pirmoradian, Fatemeh, Safkhani, Masoumeh, and Dakhilalian, Seyed Mohammad. Eccpws: An ecc-based protocol for wban systems. *Computer Networks*, 224:109598, 2023.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2024, Volume 21, Number 1 (pp. 112–136)
https://monadi.isc.org.ir

An overview of methods for analyzing and proving the security of security protocols

Mohammad Dakhilalian [1,*], Masoumeh Safkhani [2] and Fatemeh Pirmoradian [1]

[1] *Department of Electrical and Computer Engineering, Electrical and Computer Engineering, Isfahan University of Technology, Esfahan, Iran*
[2] *Department of Computer Engineering, Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: December 18, 2023
Accepted: March 11, 2024
Published Online: March 26, 2024

Keywords:
Authentication Protocols
ProVerif Tool
TAMARIN Tool
AVISPA Tool
Scyther Tool
BAN Logic
GNY Logic

Type: Review paper

ABSTRACT

The provision of all remote services requires the mutual authentication of the participating parties. The framework by which this authentication is done is called authentication protocols. In other words, cryptography or encryption protocol is a distributed cryptographic algorithm that establishes interactions between at least two or more entities with a specific purpose. These protocols have provided secure and insecure channels for communication between the parties participating in the protocol. Usually, secure channels are used for registration and insecure channels are used for mutual authentication. After registering on the server and verifying its authenticity, the user can benefit from the services provided by the server. Many authentication protocols have been proposed in fields such as e-medical care, the Internet of Things, cloud computing, etc. The privacy and anonymity of users in these schemes is the biggest challenge in implementing a platform to benefit from remote services. Because authentication of users takes place on the insecure platform of the Internet, it can be vulnerable to all existing Internet attacks. Generally, there are two methods to analyze and prove the security of authentication protocols. Formal method and Informal method. The informal method, which is based on intuitive arguments, the analyst's creativity, and mathematical concepts, tries to find errors and prove security. The formal method, which is done both manually and automatically, has used a variety of mathematical logic and automatic security analysis tools. The manual method has been done using mathematical models such as real or random models and mathematical logic such as BAN logic, GNY logic, etc., and the automatic method has been done using AVISPA, Scyther, ProVerif, TAMARIN tools, etc. The methods of proving and analyzing the security of security protocols are divided into two general categories based on proof of theorem and model verification, in this article the details of each of these methods of proving and analyzing security, analyzing the security of ECCPWS protocol with some of these methods and finally comparing these methods are described together in terms of strengths, weaknesses, etc. In this article, the methods based on model checking and then the methods based on proving the theorem are described.

* Corresponding author
Email addresses: mdalian@iut.ac.ir (Mohammad Dakhilalian), safkhani@sru.ac.ir (Masoumeh Safkhani), f.pirmoradian@ec.iut.ac.ir (Fatemeh Pirmoradian)