مجله علمی- پژوهشی "پدافند الکترونیکی و سایبری"
سال پنجم، شماره ۴، زمستان ۱۳۹۶، ص ۱۲۹-۱۰۹

تحلیل و ارزیابی صوری پروتکل‌های امنیتی شبکه تترا با استفاده از ابزارهای تحلیل خودکار

مهدی ملازاده گل‌محله[1*]، محمد سبزی‌نژاد فراش[2]، روح‌اله رستاقی[3]

۱- مربی، عضو هئیت علمی دانشگاه امام حسین(ع) ۲- دکتری ریاضی رمز دانشگاه خوارزمی ۳- کارشناس ارشد مخابرات رمز دانشگاه امام حسین (ع)
(دریافت: ۹۵/۰۵/۰۱، پذیرش: ۹۶/۰۵/۰۱)

چکیده

در این مقاله، ساختار نسخه‌های مختلف پروتکل امنیتی تترا در "مدل صوری" و با استفاده از ابزارهای تحلیل خودکار پرووریف و اسکایتر مورد ارزیابی قرار می‌گیرند. پروتکل امنیتی شبکه تترا از نوع پروتکل‌های تبادل کلید تصدیق شده است که در آن، طرفین ضمن احراز هویت یکدیگر، یک کلید نشست نیز می‌سازند. این پروتکل همچنین از کلیدهای محرمانه از پیش توزیع شده استفاده می‌کند که مبتنی بر ساز و کارهای رمزنگاری متقارن است. تحلیل امنیتی پروتکل مذکور در "مدل صوری" و با استفاده از ابزارهای تحلیل خودکار پرووریف و اسکایتر انجام شده است. در ابتدا، هشت ویژگی امنیتی: محرمانگی، احراز هویت، امنیت پیشرو، امنیت کلید ناشناخته، کلید نشست یکسان، امنیت کلید معلوم، گمنامی و تمامیت را در بستر این ابزارها مدل‌سازی نموده و سپس با استفاده از هر دو ابزار، امنیت پروتکل مذکور را نسبت به این ویژگی‌ها مورد بررسی قرار می‌دهیم. مقایسه نتایج حاصل از تحلیل صوری این ویژگی‌ها با نتایج حاصله از تحلیل‌های غیرصوری در منابع آشکار دلالت بر وجود ضعف‌هایی جدید در ویژگی‌های "امنیت پیشرو" و "تمامیت" در ساختار این پروتکل دارد. در نهایت، روش‌هایی برای غلبه بر این ضعف‌ها ارایه شده است.

واژه‌های کلیدی: تحلیل امنیتی، مدل‌های صوری، ابزار تحلیل خودکار، شبکه تترا

۱- مقدمه

پروتکل‌های رمزنگاری معمولا بمنظور ایجاد امنیت در سامانه‌های مخابراتی و شبکه‌های رایانه‌ای مورد استفاده قرار می‌گیرند. حال آنکه حصول اطمینان از امنیت این پروتکل‌ها به لحاظ طبیعت واکنشی آنها، کار چندان ساده‌ایی نیست. لذا حصول اطمینان از امنیت این پروتکل‌ها قبل از به‌کارگیری آنها بسیار ضروری است. برای تحلیل امنیت پروتکل‌ها و طرح‌های رمزنگاری چندین مدل وجود دارد:

۱- مدل منطقی یا مدل BAN
۲- مدل‌های صوری[1] یا مدل Dolev-Yao
۳- مدل محاسباتی[2] (شرط: مهاجم دارای محدودیت محاسباتی است)
۴- مدل تئوری اطلاعات (یا امنیت غیر مشروط)

مدل‌ها از ضعیف‌ترین به قوی‌ترین لیست شده‌اند. مدل منطقی بسیار ضعیف است و اثبات امنیت در این مدل نمی‌تواند تضمین کننده امنیت طرح یا پروتکل رمزنگاری در عمل باشد [۲]. از طرف دیگر، مدل تئوری-اطلاعات یک مدل بسیار قوی است و دستیابی به امنیت در این مدل بدون فرضیات فیزیکی مانند کانال نویزی، مکانیک کوانتومی و ... عملا غیرممکن است. این مدل با احتمالات و تئوری اطلاعات سر و کار دارد و برای اثبات امنیت در این مدل بایستی اثبات نمود که "احتمال" این‌که مهاجم بتواند طرح یا پروتکل رمزنگاری را بشکند ناچیز است.

امنیت پروتکل‌های رمزنگاری معمولا در مدل‌های صوری یا مدل‌های محاسباتی مورد بررسی و ارزیابی قرار می‌گیرند. در مدل‌های صوری، فرض بر این است که اولیه‌های رمزنگاری[3] مورد استفاده در پروتکل رمزنگاری امن یا به‌عبارت دیگر ایده‌آل هستند. در این مدل پروتکل رمزنگاری بعنوان یک سیستم و به‌عنوان مجموعه‌ایی از اعضا درنظر گرفته شده و سپس کارآیی سیستم مورد

* رایانامه نویسنده مسئول: mollazadeh@dspri.com

1- Formal Model
2 - Computational Model
3 - Cryptographic Primitives

تحلیل و بررسی قرار می‌گیرد. در مدل صوری، دو روش برای اثبات امنیت پروتکل‌ها وجود دارد [۸-۱]:

۱) روش وارسی مدل[1]: در این روش بررسی می‌شود که آیا سیستم مورد نظر، یک مدل برای مشخصه مورد نظر می‌باشد یا خیر؟ به‌عبارت دیگر کارائی سیستم مورد وارسی قرار می‌گیرد.

۲- روش اثبات قضیه[2]: در این روش به دنبال اثبات کارائی سیستم با استفاده از برهان خلف و به روش استقراء هستیم. معمولا اثبات امنیت در این روش بسیار پیچیده تر و دشوارتر از روش وارسی مدل است.

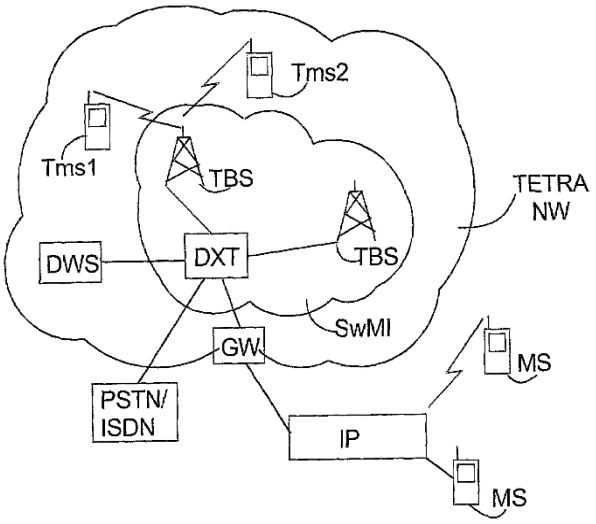
همان‌طور که گفته شد، در مدل صوری، فرض بر این است که همه اولیه‌های رمزنگاری مورد استفاده در پروتکل رمزنگاری امن هستند و پروتکل نیز بدرستی پیاده‌سازی شده است. در این مدل، مهاجم تنها به کانال ارتباطی دسترسی داشته و قادر است اطلاعات را از روی کانال ارتباطی شنود نموده و آنها را دست‌کاری و تغییر دهد. در این مدل هدف بررسی ساختار پروتکل امنیتی و درستی‌یابی عملکرد آن است.

در این مقاله، پروتکل امنیتی شبکه تترا[3] را در مدل صوری مورد تحلیل و بررسی قرار خواهیم داد. برای تحلیل، از دو ابزار تحلیل خودکار پرکاربرد اسکایتر [۱۱-۹] و پروویرف [۶-۴] استفاده خواهیم نمود. در ابتدا، سرویس‌های امنیتی محرمانگی، احراز هویت، امنیت پیشرو، امنیت کلید ناشناخته، کلید نشست یکسان، امنیت کلید معلوم، گمنامی و تمامیت در بستر دو ابزار مدل خواهند گردید و سپس پروتکل مربوطه با استفاده از این ویژگی‌های امنیتی مورد تحلیل قرار خواهند گرفت.

ادامه ساختار این مقاله به شرح زیر است. در بخش دوم، پروتکل امنیتی شبکه تترا معرفی خواهد گردید. در ادامه و در بخش سوم، تحلیل امنیتی با استفاده از ابزار پروویرف و در بخش چهارم با استفاده از اسکایتر ارائه خواهد گردید و در پایان، در بخش پنجم، جمع‌بندی و نتیجه‌گیری تحلیل‌های امنیتی ارائه شده آورده شده است.

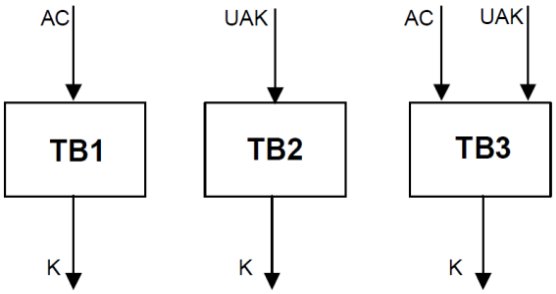
۲- احراز هویت در شبکه TETRA

شکل (۱) ساختار کلی شبکه تترا [۱۵-۱۲] را نشان می‌دهد. همان‌طور که مشاهده می‌شود، شبکه از دو بخش اصلی زیرساخت شبکه (SwMI) که نقش سرویس دهنده را دارد و ایستگاه موبایل (MS) که نقش سرویس گیرنده را دارد، تشکیل شده است. درسمت سرویس گیرنده، هر مشترک دارای یک دستگاه موبایل است که یک سیم کارت بر روی آن قرار دارد. بر روی این سیم کارت یک کلید محرمانه UAK ذخیره شده که در سمت سرویس دهنده نیز وجود دارد.



شکل (۱): ساختار کلی شبکه تترا [۱۶]

برای احراز هویت مشترک توسط شبکه و بالعکس، یک کلید احراز هویت K مورد نیاز است. برای ساخت این کلید سه راه کار وجود دارد که در شکل (۲) نشان داده شده است. همان‌طور که مشاهده می‌شود، یک روش ساخت K استفاده از کد فعال ساز AC است که توسط کاربر وارد می شود. روش دوم استفاده از کلید ذخیره شده در سیم کارت UAK است؛ و روش سوم ترکیبی از دو روش اول است. الگوریتم‌های مورد استفاده برای هر یک از روش‌ها می‌توانند توابع چکیده‌ساز باشند.



شکل (۲): ساخت کلید احراز هویت [۱۶]

احراز هویت در شبکه تترا بر دو نوع یک طرفه و دو طرفه می‌تواند باشد. در احراز هویت یک طرفه، کاربر شبکه را احراز هویت می‌کند یا بالعکس؛ در احراز هویت دو طرفه، هم کاربر و هم شبکه

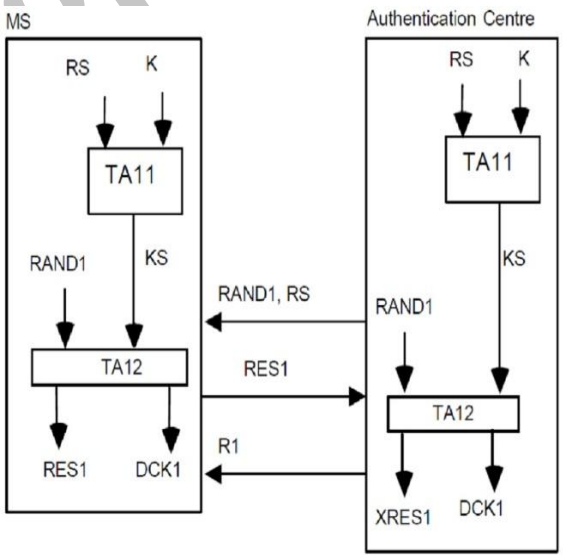
1- Model Checking
2- Theorem Proving
3- TETRA

یکدیگر را احراز هویت می‌کنند. شکل‌های (۳ و ۴) احراز هویت یک طرفه را نشان می‌دهند و شکل‌های (۵ و ۶) احراز هویت دو طرفه را نشان می‌دهند. همان‌طور که در این شکل‌ها قابل مشاهده است، علاوه بر احراز هویت، در پایان پروتکل یک کلید محرمانه مشترک DCK نیز تولید می‌شود که برای رمز کردن ارتباطات بعدی و اطمینان از صحت داده مورد استفاده قرار می گیرد.

۲-۱- پروتکل احراز هویت کاربر توسط زیرساخت شبکه (TETRA 1)

در این پروتکل، کاربر توسط شبکه احراز هویت می‌شود. با توجه به شکل (۳)، در این پروتکل از توابع TA11 و TA12 استفاده می شود که معمولا توابع چکیده ساز هستند. روند اجرای این پروتکل به شرح زیر است.

گام ۱: ابتدا سرور شبکه مقدار KS = TA11(K,RS) را محاسبه و یک مقدار تصادفی RAND1 را انتخاب می‌کند. سپس مقادیر RAND1 و RS را برای کاربر MS ارسال می‌کند.



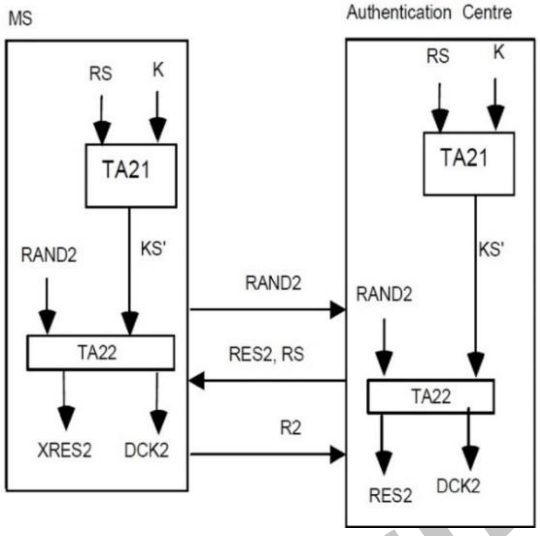
شکل (۳): پروتکل احراز هویت کاربر توسط زیرساخت شبکه (TETRA1) [۱۶]

گام ۲: کاربر MS پس از دریافت RAND1 و RS، ابتدا KS = TA11(K,RS) را محاسبه کرده و سپس مقادیر RES1 و DCK1 را به صورت (RES1,DCK1) = TA12(KS,RAND1) به‌دست می‌آورد. در پایان، کاربر مقدار RES1 را برای سرور شبکه ارسال می‌کند.

گام ۳: پس از دریافت RES1، سرور شبکه مقادیر (XRES1,DCK1) = TA12(KS,RAND1) را محاسبه و RES1 را با XRES1 مقایسه می‌کند. در صورت تساوی پیغامی مبنی بر پذیرش و در صورت عدم تساوی پیغام عدم پذیرش برای کاربر MS ارسال می‌شود.

۲-۲- پروتکل احراز هویت زیرساخت شبکه توسط کاربر (TETRA 2)

در این پروتکل، شبکه توسط کاربر احراز هویت می شود. در این پروتکل از توابع TA21 و TA22 استفاده می شود که معمولا توابع چکیده ساز هستند. روند اجرای این پروتکل با توجه به شکل (۴) به شرح زیر است:



شکل (۴): پروتکل احراز هویت زیرساخت شبکه توسط کاربر (TETRA2) [۱۶]

گام ۱: ابتدا کاربر MS مقدار تصادفی RAND2 را انتخاب و آنرا برای سرور شبکه ارسال می‌کند.

گام ۲: سرور شبکه پس از دریافت RAND2، ابتدا یک مقدار تصادفی RS را انتخاب و مقدار KS′ = TA21(K,RS) را محاسبه کرده و سپس مقادیر RES2 و DCK2 را به صورت (RES2,DCK2) = TA22(KS′,RAND2) بدست می آورد. در پایان، سرور مقدار RES2 و RS را برای کابر MN ارسال می‌کند.

گام ۳: پس از دریافت RES2 و RS، کاربر مقادیر KS′ = TA21(K,RS) و (XRES2,DCK2) = TA22(KS′,RAND2) را محاسبه و RES2 را با XRES2 مقایسه می‌کند. در صورت تساوی پیغامی مبنی بر پذیرش و در صورت عدم تساوی پیغام عدم پذیرش برای سرور شبکه ارسال می‌شود.

۲-۳- پروتکل احراز هویت دوطرفه شروع شده از طرف زیرساخت شبکه (TETRA 3)

در این پروتکل با توجه به شکل (۵) هم کاربر توسط شبکه احراز هویت می‌شود و هم زیرساخت شبکه توسط کاربر. در این پروتکل از توابع TA11، TA12، TA21 و TA22 استفاده می‌شود که معمولا توابع چکیده ساز هستند. روند اجرای این پروتکل به شرح زیر است.

گام ۱: ابتدا سرور شبکه یک مقدار تصادفی RS را انتخاب و مقادیر KS = TA11(K,RS) و KS′= TA21(K,RS) را محاسبه می‌نماید. سرور سپس یک مقدار تصادفی RAND1 را انتخاب و مقادیر RAND1 و RS را برای کاربر MS ارسال می‌کند.

گام ۲: کاربر پس از دریافت RAND1 و RS، ابتدا KS = TA11(K,RS) و KS′ = TA21(K,RS) را محاسبه و سپس مقادیر RES1 و DCK1 را به صورت (RES1,DCK1) = TA12(KS,RAND1) به‌دست می‌آورد. کاربر در ادامه یک مقدار تصادفی RAND2 را انتخاب کرده و مقادیر RAND2 و RES1 را برای سرور شبکه ارسال می‌کند.

گام ۳: سرور شبکه پس از دریافت RAND2 و RES1، مقادیر (XRES1,DCK1) = TA12(KS,RAND1) را محاسبه و RES1 را با XRES1 مقایسه می‌کند. در صورت تساوی، سرور شبکه مقادیر (RES2,DCK2) = TA22(KS′,RAND2) را محاسبه و RES2 را به همراه پیغام R1 مبنی بر پذیرش احراز هویت برای کاربر ارسال می‌کند.

گام ۴: پس از دریافت RES2 و R1، کاربر مقادیر (XRES2,DCK2) = TA22(KS′,RAND2) را محاسبه و RES2 را با XRES2 مقایسه می‌کند. در صورت تساوی پیغام R2 مبنی بر پذیرش و در صورت عدم تساوی پیغام R2 عدم پذیرش برای سرور شبکه ارسال می‌کند.

۲-۴- پروتکل احراز هویت دوطرفه شروع شده از طرف کاربر (TETRA 4)

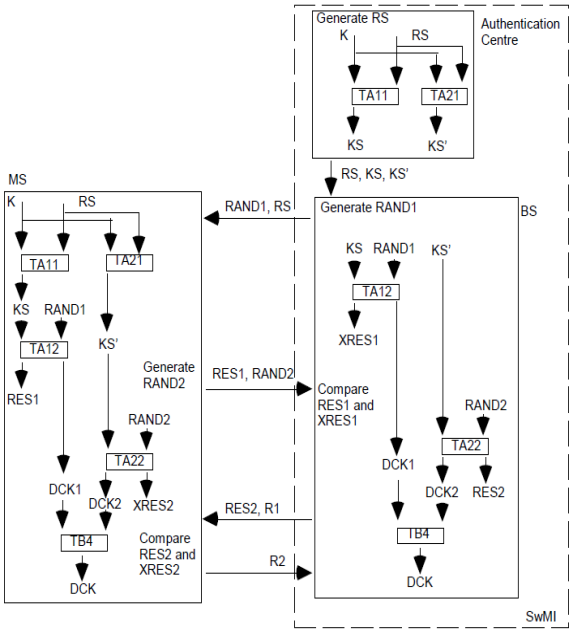
نمای کلی این پروتکل در شکل (۶) نشان داده شده است. در این پروتکل، هم کاربر توسط شبکه احراز هویت می‌شود و هم زیرساخت شبکه توسط کاربر. در این پروتکل از توابع TA11 ، TA12، TA21 و TA22 استفاده می‌شود که معمولا توابع چکیده ساز هستند. روند اجرای این پروتکل به شرح زیر است.

گام ۱: ابتدا کاربر یک مقدار تصادفی RAND2 را انتخاب و آن‌را برای سرور ارسال می‌کند.

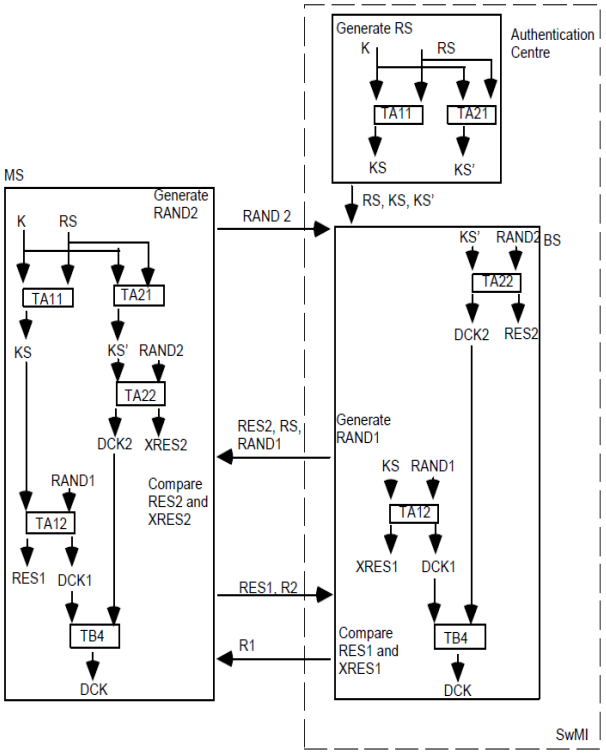
گام ۲: سرور در ابتدا مقدار تصادفی RS را انتخاب و مقادیر KS = TA11(K,RS) و KS′ = TA21(K,RS) را محاسبه نموده و سپس مقادیر RES2 و DCK2 را به صورت (RES2,DCK2) = TA12(KS′,RAND2) به‌دست می‌آورد. سرور در ادامه یک مقدار تصادفی RAND1 انتخاب کرده و (XRES1,DCK1) = TA12(KS,RAND1) را محاسبه می‌کند. سپس مقادیر RAND1، RES2 و RS را برای کاربر ارسال می‌کند.

گام ۳: پس از دریافت RAND1، RES2 و RS، کاربر ابتدا مقادیر KS = TA11(K,RS) و KS′ = TA21(K,RS) و سپس مقادیر (XRES2,DCK2) = TA22(KS′,RAND2) را محاسبه و RES2 را با XRES2 مقایسه می‌کند. در صورت تساوی، کاربر مقادیر (RES1,DCK1) = TA12(KS,RAND1) را محاسبه و RES1 را به همراه پیغام R2 مبنی بر پذیرش احراز هویت برای سرور ارسال می‌کند.

گام ۴: سرور پس از دریافت RES1 و R2، مقدار RES1 را با XRES1 که در گام ۲ محاسبه کرده بود مقایسه می‌کند. در صورت تساوی پیغام R1 مبنی بر پذیرش و در صورت عدم تساوی پیغام R1 عدم پذیرش برای کاربر ارسال می‌کند.



شکل (۵): پروتکل احراز هویت دو طرفه، شروع شده از طرف زیرساخت شبکه (TETRA3) [۱۶]



شکل (۶): احراز هویت دوطرفه، شروع شده توسط کاربر (TETRA4) [۱۶]

۳- ارزیابی پروتکل‌های امنیتی TETRA با استفاده از نرم‌افزار پروورِیف

در این بخش، نسخه‌های مختلف پروتکل امنیتی تترا شامل TETRA1, TETRA2, TETRA3 و TETRA4 را با استفاده از نرم‌افزار تحلیل خودکار پروورِیف مورد ارزیابی قرار می‌دهیم. همان‌طور که قبلا عنوان گردید، نرم‌افزار پروورِیف از نرم‌افزارهای تحلیل خودکار ارائه شده در مدل صوری یا همان Dolev-Yao است. در این مدل، فرض بر این است که اولیه‌های رمزنگاری مورد استفاده در پروتکل امنیتی ایده‌آل هستند و مهاجم تنها به کانال عمومی که برای ارسال و دریافت اطلاعات بین فرستنده و گیرنده مورد استفاده قرار می‌گیرد دسترسی دارد و قادر است که اطلاعات را از روی این کانال بخواند، آنها را دست‌کاری و تغییر داده و سپس جایگزین نماید.

از هشت ویژگی امنیتی مورد بررسی، هر چهار نسخه پروتکل تترا امنیت پیشرو (امنیت پیشرو به این معنا که اگر کلید محرمانه مشترک طولانی مدت فاش شود، کلیدهای نشست قبل به خطر نیافتد) را برآورده نمی‌سازند. دلیل آن این است که با فاش شدن کلید K، کلیه کلیدهای وابسته به آن نیز فاش می‌شوند. بنابراین، مهاجم با در اختیار داشتن کلید K و همچنین پیام‌های تبادل شده در نشست‌های قبل، به راحتی می‌تواند کلیدهای نشست DCK را به‌دست آورد. البته این یک ویژگی رایج در تمامی پروتکل‌های مبتنی بر رمزنگاری متقارن است. تنها راه برای برآورده کردن ویژگی امنیت پیشرو، بکارگیری ساز و کارهای مبتنی بر رمزنگاری نامتقارن است. همچنین، چون در پروتکل تترا از کدهای احراز هویت[1] (MAC) بر روی پیام‌های مبادله شده مابین کاربر MS و سرور swMI استفاده نمی‌شود، لذا ویژگی امنیتی تمامیت پیام‌های ارسالی برقرا نیست. ویژگی امنیت کلید معلوم نیز تنها در نرم‌افزار اسکایتر قابل بررسی است، که نتایج بررسی در بخش‌های مربوطه آمده است. نتایج حاصل از بررسی پنج ویژگی باقیمانده همراه با توضیحات مربوطه در ادامه آمده است.

۳-۱- نتایج ارزیابی محرمانگی TETRA1 در نرم‌افزار پروورِیف

نتایج حاصل از اجرای این کد و تفسیر ویژگی‌های امنیتی مختلف به تفضیل در ادامه آمده است.

۳-۱-۱- نتایج ارزیابی محرمانگی برای TETRA1

در این پروتکل، کلید مشترک K و کلید نشست DCK1 باید محرمانه باشند. کلید مشترک K در طی فرایند اجرای پروتکل، قابل کشف نیست زیرا در هیچ یک از پیام های تبادل شده به صورت متن آشکار تبادل نمی‌شود. کلید نشست DCK1 نیز قابل کشف نیست زیرا تابعی از کلید مشترک K است. در واقع، امنیت DCK1 وابسته به امنیت K است. از این رو، اگر کلید K محرمانه باشد، کلید DCK1 نیز محرمانه خواهد بود. خلاف این گزاره نیز صادق است. نتایج حاصل از ارزیابی با استفاده از پروورِیف محرمانگی این دو پارامتر را تایید می کند. همان‌طور که از پیوست (۱- الف) قابل مشاهده است، نتایج حاصل از ارزیابی این پارامترها هم از منظر MS و هم از منظر SwMI با عبارت true گزارش شده است که نشان دهنده برقرار بودن محرمانگی پارامترهای مورد ارزیابی است.

۳-۱-۲- نتایج ارزیابی احراز هویت برای TETRA1

همان‌طور که در بخش قبلی عنوان گردید، پروتکل TETRA1 در واقع دارای مکانیزم احراز هویت یک‌طرفه است و برای احراز هویت

1- Message authentication code

MS توسط SwMI طراحی شده است. بدیهی است، احراز هویت SwMI توسط MS در آن لحاظ نشده است.

نتایج ارزیابی امنیتی احراز هویت در پروتکل TETRA1 با استفاده از ابزار پرووریف در پیوست (۱-ب) آمده است. همان‌طور که در این بخش قابل مشاهده است، بررسی تقدم پیشامد beginMS بر پیشامد endSwMI نتیجه true داده است. این امر به این معنی است که نشست SwMI خاتمه پیدا نمی‌کند مگر این‌که نشست متناظری توسط MS وجود داشته باشد. بنابراین، ویژگی احراز هویت MS توسط SwMI برای این پروتکل برقرار است.

ویژگی احراز هویت SwMI توسط MS با بررسی تقدم پیشامد endMS بر beginSwMI انجام شده است. همان‌طور که انتظار می‌رفت، نتیجه ارزیابی false است به این معنا که احراز هویت SwMI توسط MS برای این پروتکل برقرار نیست. برای این وضعیت نیز یک ساختار حمله پیشنهاد شده که در آن یک مهاجم خود را به عنوان SwMI به MS معرفی می‌کند.

۳-۱-۳- نتایج ارزیابی UKS برای TETRA1

در حمله کلید ناشناخته، مهاجم تلاش دارد تا بین سه بازیگر A، B و C به نحوی اجرای پروتکل را دستکاری کند تا A تصور کند طرف مقابلش B بوده در حالی که B بر این باور باشد که طرف مقابلش C بوده است. پروتکل‌هایی که در آن‌ها طرفین از کلیدهای رمزنگاری متقارن استفاده می‌کنند، امکان اعمال این حمله وجود ندارد. زیرا اگر A با B یک نشست موفقیت‌آمیز داشته باشد، باید از کلید محرمانه مشترک منحصر به‌فردی استفاده کرده باشند که این کلید در اختیار هیچ شخص دیگری نیست. بنابراین، B به‌طور قطع مطمئن است با کسی نشست داشته که کلید مشترک را در اختیار دارد. نتایج حاصل از ارزیابی این ویژگی امنیتی برای پروتکل TETRA1 با استفاده از ابزار پرووریف در پیوست (۱-۱) آمده است که امنیت این پروتکل در برابر حملات کلید ناشناخته را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه ارزیابی true گزارش شده که به معنای مثبت بودن نتیجه ارزیابی است.

۳-۱-۴- نتایج مربوط به ارزیابی کلید نشست مشترک برای TETRA1

در پروتکل‌های احراز هویت که در آن‌ها کلید نشست نیز ساخته می‌شود، یک نکته امنیتی این است که در صورت احراز هویت موفقیت آمیز، کلیدهای نشست ساخته شده توسط طرفین باید یکسان باشند. ساختار پروتکل TETRA1 به نحوی است که در صورت احراز هویت موفقیت آمیز، طرفین کلید نشست یکسانی می‌سازند. این امر به دلیل استفاده از کلید محرمانه K توسط طرفین و همچنین بکارگیری مقادیر تصادفی می‌باشد.

نتایج ارزیابی مربوط به کلید نشست مشترک برای پروتکل TETRA1 در پیوست (۱-ج) آمده است. همان‌طور که قابل مشاهده است، نتیجه ارزیابی true است که این موضوع را تایید می‌کند.

۳-۱-۵- نتایج ارزیابی گمنامی برای TETRA1

نتایج ارزیابی گمنامی برای پروتکل TETRA1 در پیوست (۱-ه) آمده است. از آنجایی که در پروتکل TETRA1، شناسه ثابت طرفین بر روی کانال ارسال نمی‌شود، ویژگی گمنامی طرفین توسط این پروتکل برآورده می‌شود. نتایج ارزیابی پرووریف نیز این موضوع را تایید می‌کند. همان‌طور که مشاهده می‌شود، نتایج ارزیابی true گزارش شده است.

۳-۲- ارزیابی پروتکل امنیتی TETRA 2 با استفاده از پرووریف

نتایج حاصل از اجرای این کد و تفسیر ویژگی‌های امنیتی مختلف به تفضیل در ادامه آمده است.

۳-۲-۱- نتایج ارزیابی محرمانگی برای TETRA2

همانند پروتکل TETRA1، در پروتکل TERTRA2 کلید مشترک K و کلید نشست DCK2 باید محرمانه باشند. کلید مشترک K در طی فرایند اجرای پروتکل، قابل کشف نیست زیرا در هیچ یک از پیام‌های تبادل شده به صورت متن آشکار تبادل نمی‌شود. کلید نشست DCK2 نیز قابل کشف نیست زیرا تابعی از کلید مشترک K است. در واقع، امنیت DCK2 وابسته به امنیت K است. از این رو، اگر کلید K محرمانه باشد، کلید DCK2 نیز محرمانه خواهد بود. خلاف این گزاره نیز صادق است.

نتایج حاصل از ارزیابی با استفاده از پروویرف محرمانگی این دو پارامتر را تایید می کند. همانطور که در پیوست (۲-الف) قابل مشاهده است، نتایج حاصل از ارزیابی این پارامترها هم از منظر MS و هم از منظر SwMI با عبارت true گزارش شده است که نشان‌دهنده برقرار بودن محرمانگی پارامترهای مورد ارزیابی است.

۳-۲-۲- نتایج ارزیابی احراز هویت برای TETRA2

پروتکل TETRA2، همان‌طور که در بخش ۲-۳ توصیف شد، برای احراز هویت SwMI توسط MS طراحی شده است. بدیهی است، احراز

هویت MS توسط SwMI در آن لحاظ نشده است. نتایج ارزیابی با استفاده از پروووریف نیز این ویژگی احراز هویت یک‌طرفه را تایید می‌کند.

همان‌طور که در پیوست (۲-ب) قابل مشاهده است، بررسی تقدم پیشامد beginSwMI بر پیشامد endMS نتیجه true داده است. این امر به این معنی است که نشست MS خاتمه پیدا نمی‌کند مگر این‌که نشست متناظری توسط SwMI وجود داشته باشد. بنابراین، ویژگی احراز هویت SwMI توسط MS برای این پروتکل برقرار است.

همچنین، بررسی احراز هویت MS توسط SwMI با بررسی تقدم پیشامد endSwMI بر beginMS انجام می‌شود. همان‌طور که انتظار می‌رفت، نتیجه این ارزیابی false است، بدین معنا که احراز هویت MS توسط SwMI برای این پروتکل برقرار نیست. برای این وضعیت نیز یک ساختار حمله پیشنهاد شده که در آن یک مهاجم خود را به عنوان MS به SwMI معرفی می‌کند.

۳-۲-۳- نتایج ارزیابی UKS برای TETRA2

در حمله کلید ناشناخته، مهاجم تلاش دارد تا بین سه بازیگر A، B و C به نحوی اجرای پروتکل را دست‌کاری کند تا A تصور کند طرف مقابلش B بوده در حالی که B بر این باور باشد که طرف مقابلش C بوده است. پروتکل‌هایی که در آن‌ها طرفین از کلیدهای رمزنگاری متقارن استفاده می‌کنند، امکان اعمال این حمله وجود ندارد. زیرا اگر A با B یک نشست موفقیت آمیز داشته باشد، باید از کلید محرمانه مشترک منحصربه‌فردی استفاده کرده باشند که این کلید در اختیار هیچ شخص دیگری نیست. بنابراین، B به‌طور قطع مطمئن است با کسی نشست داشته که کلید مشترک را در اختیار دارد. نتایج حاصل از ارزیابی این ویژگی امنیتی در پیوست (۲-ج) آمده است که امنیت این پروتکل در برابر حملات کلید ناشناخته را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه ارزیابی true گزارش شده که به معنای مثبت بودن نتیجه ارزیابی است.

۳-۲-۴- ارزیابی کلید نشست مشترک برای TETRA2

در پروتکل‌های احراز هویت که در آن‌ها کلید نشست نیز ساخته می‌شود، یک نکته امنیتی این است که در صورت احراز هویت موفقیت آمیز، کلیدهای نشست ساخته شده توسط طرفین باید یکسان باشد. ساختار پروتکل TETRA2 به نحوی است که در صورت احراز هویت موفقیت آمیز، طرفین کلید نشست یکسانی می‌سازند. این امر به دلیل استفاده از کلید محرمانه K توسط طرفین و همچنین بکارگیری مقادیر تصادفی می‌باشد. نتایج حاصل از ارزیابی با استفاده از پروووریف که در پیوست (۱-۲) آمده است نیز این موضوع را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه این ارزیابی true گزارش شده است.

۳-۲-۵- نتایج ارزیابی گمنامی برای TETRA2

از آنجایی که در پروتکل TETRA2، شناسه ثابت طرفین بر روی کانال ارسال نمی‌شود، ویژگی گمنامی طرفین توسط این پروتکل برآورده می‌شود. نتایج حاصل از ارزیابی با استفاده از پروووریف که در پیوست (۲-ه) آمده است نیز این موضوع را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه این ارزیابی true گزارش شده است.

۳-۳- ارزیابی پروتکل امنیتی TETRA3 با استفاده از نرم افزار پروووریف

نتایج حاصل از اجرای این کد و تفسیر ویژگی‌های امنیتی مختلف در ادامه آمده است.

۳-۳-۱- نتایج ارزیابی محرمانگی برای TETRA3

در این پروتکل، کلید مشترک K و کلید نشست DCK باید محرمانه باشند. کلید مشترک K در طی فرایند اجرای پروتکل، قابل کشف نیست زیرا در هیچ یک از پیام های تبادل شده به صورت متن آشکار تبادل نمی‌شود. کلید نشست DCK نیز قابل کشف نیست زیرا تابعی از کلید مشترک K است. در واقع، امنیت DCK وابسته به امنیت K است. از این رو، اگر کلید K محرمانه باشد، کلید DCK نیز محرمانه خواهد بود. خلاف این گزاره نیز صادق است. نتایج حاصل از ارزیابی با استفاده از پروووریف محرمانگی این دو پارامتر را تایید می‌کند (پیوست ۳-الف).

۳-۳-۲- نتایج ارزیابی احراز هویت برای TETRA3

پروتکل TETRA3 برای احراز هویت دوطرفه SwMI و MS طراحی شده است. با این حال، نتایج ارزیابی با استفاده از پروووریف نشان می‌هد که ویژگی احراز هویت SwMI توسط MS برقرار است اما ویژگی احراز هویت MS توسط SwMI دارای ضعف است.

نتایج بررسی ویژگی احراز هویت پروتکل TETRA3 در پیوست (۳-ب) آمده است. همان‌طور که قابل مشاهده است، بررسی تقدم پیشامد beginSwMI بر پیشامد endMS نتیجه true داده است. این امر به این معنی است که نشست MS خاتمه پیدا نمی‌کند مگر اینکه نشست متناظری توسط SwMI وجود داشته باشد. بنابراین، ویژگی احراز هویت SwMI توسط MS برای این پروتکل برقرار است.

همچنین، بررسی ویژگی احراز هویت MS توسط SwMI با بررسی تقدم پیشامد endSwMI بر beginMS انجام می‌شود. همان‌طور که مشاهده می‌شود، نتیجه ارزیابی false است به این معنا که احراز هویت MS توسط SwMI برای این پروتکل برقرار نیست.

برای این وضعیت نیز یک ساختار حمله پیشنهاد شده که در آن مهاجم پارامتر RAND2 (ارسال شده توسط MS) را تغییر داده و برای SwMI ارسال می‌کند. از آنجایی که هیچ گونه صحت سنجی روی این پارامتر صورت نمی‌گیرد، SwMI نمی‌تواند این دستکاری را تشخیص دهد. در پایان نیز، مهاجم پیام R2 مبنی بر تایید را برای SwMI ارسال می‌کند. در این صورت، SwMI به درستی MS را تایید کرده و تصور می‌کند که MS نیز او را پذیرفته است، در حالی که MS، SwMI را نپذیرفته است.

۳-۳-۴- ارزیابی کلید نشست مشترک برای TETRA3

در پروتکل‌های احراز هویت که در آن‌ها کلید نشست نیز ساخته می‌شود، یک نکته امنیتی این است که در صورت احراز هویت موفقیت‌آمیز، کلیدهای نشست ساخته شده توسط طرفین باید یکسان باشند. همانند نسخه‌های دیگر پروتکل تترا، ساختار پروتکل TETRA3 به نحوی است که در صورت احراز هویت موفقیت‌آمیز، طرفین کلید نشست یکسانی می‌سازند. این امر به دلیل استفاده از کلید محرمانه K توسط طرفین و همچنین بکارگیری مقادیر تصادفی می‌باشد. نتایج حاصل از ارزیابی با استفاده از پروورِیف که در پیوست (۳-د) آمده است نیز این موضوع را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه این ارزیابی true گزارش شده است

۳-۳-۵- نتایج ارزیابی گمنامی برای TETRA3

از آنجایی که در پروتکل TETRA3، شناسه ثابت طرفین بر روی کانال ارسال نمی‌شود، ویژگی گمنامی طرفین توسط این پروتکل برآورده می‌شود. نتایج ارزیابی پروورِیف که در پیوست (۳-ه) آمده است نیز این موضوع را تایید می‌کند. همان‌طور که مشاهده می‌شود، نتایج ارزیابی true گزارش شده است.

۳-۴- ارزیابی پروتکل امنیتی TETRA4 با استفاده از نرم افزار پروورِیف

نتایج حاصل از اجرای این کد و تفسیر ویژگی‌های امنیتی مختلف در ادامه آمده است.

۳-۴-۱- نتایج ارزیابی محرمانگی برای TETRA4

در این پروتکل، کلید مشترک K و کلید نشست DCK باید محرمانه باشند. کلید مشترک K در طی فرایند اجرای پروتکل، قابل کشف نیست زیرا در هیچ یک از پیام‌های تبادل شده به صورت متن آشکار تبادل نمی‌شود. کلید نشست DCK نیز قابل کشف نیست زیرا تابعی از کلید مشترک K است. در واقع، امنیت DCK وابسته به امنیت K است. از این رو، اگر کلید K محرمانه باشد، کلید DCK نیز محرمانه خواهد بود. خلاف این گزاره نیز صادق است. نتایج حاصل از ارزیابی با استفاده از پروورِیف محرمانگی این دو پارامتر را تایید می‌کند. همان‌طور که در ذیل قابل مشاهده است؛ نتایج حاصل از ارزیابی با استفاده از پروورِیف که در پیوست (۴-الف) آمده است نیز این موضوع را تایید می‌کند. همان‌طور که قابل مشاهده است، نتیجه این ارزیابی true گزارش شده است

۳-۴-۲- نتایج ارزیابی احراز هویت برای TETRA4

پروتکل TETRA4، همان‌طور که در بخش ۲-۵ توصیف شد، برای احراز هویت دوطرفه SwMI و MS طراحی شده است. نتایج ارزیابی با استفاده از پروورِیف نیز نشان می‌هد که ویژگی احراز هویت دوطرفه در این پروتکل برقرار است.

نتایج بررسی ویژگی احراز هویت پروتکل TETRA4 در پیوست (۴-ب) آمده است. همان‌طور که مشاهده می‌شود، بررسی تقدم پیشامد beginSwMI بر پیشامد endMS نتیجه true داده است. این امر به این معنی است که نشست MS خاتمه پیدا نمی‌کند مگر اینکه نشست متناظری توسط SwMI وجود داشته باشد. بنابراین ویژگی احراز هویت SwMI توسط MS برای این پروتکل برقرار است.

همچنین، بررسی ویژگی احراز هویت MS توسط SwMI با بررسی تقدم پیشامد endSwMI بر beginMS انجام می‌شود. همانطور که مشاهده می‌شود، بررسی تقدم پیشامد beginMS بر پیشامد endSwMI نتیجه true داده است. این امر به این معنی است که نشست SwMI خاتمه پیدا نمی‌کند مگر این‌که نشست متناظری توسط MS وجود داشته باشد. بنابراین، ویژگی احراز هویت MS توسط SwMI برای این پروتکل برقرار است.

۳-۴-۳- نتایج ارزیابی UKS برای TETRA4

در بخش پیوست (۴-ج)، نتایج مربوط به ارزیابی امنیت UKS آورده شده است. همان‌طور که مشخص است، نتیجه این ارزیابی true گزارش شده که به معنای امنیت در برابر کلید ناشناخته است.

۳-۴-۴- ارزیابی کلید نشست مشترک برای TETRA4

در پروتکل‌های احراز هویت که در آن‌ها کلید نشست نیز ساخته می‌شود، یک نکته امنیتی این است که در صورت احراز هویت

موفقیت‌آمیز، کلیدهای نشست ساخته شده توسط طرفین باید یکسان باشد. ساختار پروتکل TETRA3 به نحوی است که در صورت احراز هویت موفقیت آمیز، طرفین کلید نشست یکسانی می‌سازند. این امر به دلیل استفاده از کلید محرمانه K توسط طرفین و همچنین به‌کارگیری مقادیر تصادفی می‌باشد. نتایج حاصل از ارزیابی با استفاده از پروووریف نیز این موضوع را تایید می‌کند. همان‌طور که در ذیل قابل مشاهده است، نتیجه ارزیابی true است.

۳-۴-۵- نتایج ارزیابی گمنامی برای TETRA4

از آنجایی که در پروتکل TETRA4، شناسه ثابت طرفین بر روی کانال ارسال نمی‌شود، ویژگی گمنامی طرفین توسط این پروتکل برآورده می‌شود.

در پیوست (۴- د) ، نتایج ارزیابی پروووریف برای این ویژگی آمده است. همان‌طور که مشاهده می‌شود، نتایج ارزیابی true گزارش شده است که این موضوع را تایید می‌کند.

۴- ارزیابی امنیت نسخه‌های مختلف پروتکل TETRA با استفاده از اسکایتر

در این بخش، نتایج ارزیابی امنیتی نسخه‌های مختلف پروتکل TETRA با استفاده از نرم‌افزار اسکایتر آورده شده است. برخلاف نرم‌افزار پروووریف که بایستی کلیه ویژگی‌های امنیتی با استفاده از تعریف پیشامدها در کد برنامه و با استفاده از دستورات در خط فرمان انجام شود، در نرم‌افزار اسکایتر تنها برای بررسی ویژگی‌های امنیتی محرمانگی و احراز هویت نیاز است که از تعاریف پیشامدها توسط خط فرمان استفاده نمود و بقیه ویژگی‌های امنیتی را می‌توان از پنجره تنظیمات و تنظیمات پیشرفته انتخاب نمود.

۴-۱- نتایج ارزیابی پروتکل TETRA1 در نرم‌افزار اسکایتر

همان‌طور که عنوان گردید، ویژگی‌های قابل ارزیابی از طریق تعریف پیشامدها در خط فرمان شامل محرمانگی پارامترهای حساس و احراز هویت از منظر طرفین پروتکل می‌باشد که در کد برنامه با استفاده از claim ها برای طرفین مشخص شده است. نتایج حاصل از ارزیابی، محرمانگی تمامی پارامترهای حساس را تایید می‌کند. در رابطه با ویژگی احراز هویت نیز مطابق انتظار، احراز هویت SwMI توسط MS تایید می‌شود اما احراز هویت MS توسط SwMI تایید نمی‌شود. این امر به دلیل ذات پروتکل TETRA1 است که برای احراز هویت یک‌طرفه طراحی شده است.

در تنظیمات انجام شده در نرم‌افزار اسکایتر، مهاجم در حالت پایه در نظر گرفته شده و تعداد اجراها نامحدود تنظیم شده است. نتایج حاصل از ارزیابی نشان می‌دهد، برای نقش SwMI، محرمانگی پارامترهای K و DCK1 و همچنین ویژگی احراز هویت (که با دستور Nisynch بررسی می‌شود) برقرار است. برای نقش MS نیز محرمانگی پارامترهای مورد ارزیابی برقرار است اما ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت سمت MS که تایید نشده، یک حمله گزارش شده است. طرح کلی این حمله در شکل پیوست (۵ - الف) نشان داده شده است.

۴-۲- نتایج ارزیابی پروتکل TETRA2 در نرم‌افزار اسکایتر

در نرم‌افزار اسکایتر، ویژگی‌های قابل ارزیابی از طریق خط فرمان همان محرمانگی پارامترهای حساس و احراز هویت از منظر طرفین پروتکل می‌باشد که با استفاده از claimها برای هر طرف مشخص شده است. نتایج حاصل از ارزیابی، محرمانگی تمامی پارامترهای حساس را تایید می‌کند. در رابطه با ویژگی احراز هویت نیز مطابق انتظار، احراز هویت SwMI توسط MS تایید می‌شود اما احراز هویت MS توسط SwMI تایید نمی‌شود. این امر به دلیل ذات پروتکل TETRA2 است که برای احراز هویت یک‌طرفه طراحی شده است.

تنظیمات انجام شده در نرم‌افزار اسکایتر به این صورت است که، مهاجم در حالت پایه در نظر گرفته شده و تعداد اجراها نامحدود تنظیم شده است. همان‌طور که ملاحظه می‌شود، برای نقش MS، محرمانگی پارامترهای K و DCK2 و همچنین ویژگی احراز هویت (که با دستور Nisynch بررسی می‌شود) برقرار است. برای نقش SwMI نیز محرمانگی پارامترهای مورد ارزیابی برقرار است اما ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت سمت MS که تایید نشده است یک حمله گزارش شده است. طرح کلی این حمله در پیوست (۵-ب) نشان داده شده است.

۴-۳- نتایج ارزیابی پروتکل TETRA3 در نرم‌افزار اسکایتر

نتایج حاصل از ارزیابی TETRA3 با استفاده از اسکایتر، محرمانگی تمامی پارامترهای حساس را تایید می‌کند. در رابطه با ویژگی احراز هویت برخلاف انتظار، احراز هویت SwMI توسط MS و احراز هویت MS توسط SwMI تایید نمی‌شود.

از نتایج حاصل از ارزیابی مشاهده می‌شود، برای نقش MS، محرمانگی پارامترهای K و DCK برقرار است اما ویژگی احراز هویت

(که با دستور Nisynch بررسی می‌شود) برقرار نیست. برای نقش SwMI نیز محرمانگی پارامترهای مورد ارزیابی برقرار است اما ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت که تایید نشده حملاتی گزارش شده است. همان‌طور که در شکل (۹) از پیوست ۵ مشاهده می‌شود، مهاجم پارامتر RAND2 ارسالی از طرف MS را از روی کانال دست‌کاری کرده و به‌جای آن پارامتر انتخابی خود یعنی IntruderNonce1 را برای SwMI ارسال می‌کند. در گام recv_4 نیز پارامتر R2 را مهاجم برای SwMI ارسال می‌کند. چنانچه مشاهده می‌شود، SwMI با موفقیت نشست را به پایان می‌رساند. در حالی که نشست متناظر آن در نقش MS به پایان نرسیده است. همان‌طور که در شکل پیوست (۵-ج) مشاهده می‌شود، در Run2، مهاجم پارامتر RAND1 را با IntruderNonce1 عوض می‌کند. در گام recv_3 نیز پارامتر R1 را به انتخاب خود برای Run2 ارسال می‌کند. همان‌طور که مشاهده می‌شود، MS در Run2 با موفقیت نشست را به پایان می‌رساند و بر این باور است که SwMI نیز به‌طور متناظر چنین نشستی را اجرا و به پایان رسانده است. در حالی که این چنین نیست و SwMI نشستی را که در Run1 برگزار کرده با دخالت مهاجم با دو نشست از MS در ارتباط بوده است.

۴-۴- نتایج ارزیابی پروتکل TETRA4 در نرم افزار اسکایتر

در ابزار اسکایتر، ویژگی‌های قابل ارزیابی از طریق خط فرمان شامل محرمانگی پارامترهای حساس و احراز هویت از منظر طرفین پروتکل می‌باشد که برای هر طرف با استفاده از claimها مشخص شده است. نتایج حاصل از ارزیابی، محرمانگی تمامی پارامترهای حساس را تایید می‌کند. در رابطه با ویژگی احراز هویت برخلاف انتظار، احراز هویت MS توسط SwMI تایید نمی‌شود. نتایج حاصل از ارزیابی نشان می‌دهد که برای نقش SwMI، محرمانگی پارامترهای K و DCK و همچنین ویژگی احراز هویت (که با دستور Nisynch بررسی می‌شود) برقرار است. برای نقش MS محرمانگی پارامترهای مورد ارزیابی برقرار است اما ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت که تایید نشده است یک حمله گزارش شده که در شکل (۱۰) از پیوست (۵-د) نشان داده شده است. همان‌طور که در شکل مشاهده می‌شود، مهاجم پارامتر RAND1 ارسالی از طرف SwMI را از روی کانال برداشته و بجای آن پارامتر انتخابی خود یعنی IntruderNonce1 را برای MS ارسال می‌کند. در گام recv_4 نیز پارامتر R1 را مهاجم برای MS ارسال می‌کند. همان‌طور که مشاهده می‌شود، MS با موفقیت نشست را به پایان می‌رساند. در حالی که نشست متناظر آن در نقش SwMI به پایان نرسیده است.

جدول (۱): خلاصه نتایج حاصل از ارزیابی پروتکل‌های TETRA در نرم افزارهای پروورید و اسکایتر

ردیف	ویژگی امنیتی	نتایج ارزیابی پروورید				نتایج ارزیابی اسکایتر			
		TETRA1	TETRA2	TETRA3	TETRA4	TETRA1	TETRA2	TETRA3	TETRA4
۱	محرمانگی	☑	☑	☑	☑	☑	☑	☑	☑
۲	احراز هویت	☑	☑	☒	☑	☑	☑	☒	☒
۳	امنیت پیشرو	☒	☒	☒	☒	☒	☒	☒	☒
۴	امنیت کلید ناشناخته	☑	☑	☑	☑	–	–	–	–
۵	کلید نشست یکسان	☑	☑	☑	☑	–	–	–	–
۶	امنیت کلید معلوم	–	–	–	–	☑	☑	☑	☑
۷	گمنامی	☑	☑	☑	☑	–	–	–	–
۸	تمامیت	☒	☒	☒	☒	☒	☒	☒	☒

۵- نتیجه‌گیری

در این مقاله، امنیت نسخه‌های مختلف پروتکل ارتباطی TETRA را با استفاده از دو نرم‌افزار تحلیل خودکار پروورید و اسکایتر مورد ارزیابی صوری قرار دادیم. نتایج حاصل از ارزیابی با استفاده از هر دو نرم‌افزار نشان می‌دهد که این پروتکل‌ها ویژگی‌های امنیتی تمامیت و امنیت پیشرو (امنیت پیشرو به این معنا که اگر کلید محرمانه مشترک طولانی مدت فاش شود، کلیدهای نشست قبل به خطر نیافتد) را برآورده نمی‌کنند. همچنین، ویژگی احراز هویت برای پروتکل‌های TETRA3 و TETRA4 دارای ضعف جزئی است.

تنها راه برای برآورده کردن ویژگی امنیت پیشرو، به‌کارگیری ساز و کارهای مبتنی بر رزنگاری نامتقارن است.

ویژگی تمامیت برای تمامی پروتکل‌های مورد ارزیابی دارای مشکل است. دلیل این امر، عدم استفاده از توابع چکیده‌ساز کلیددار برای پیام‌های R1 و R2 در این پروتکل‌ها است. R1 و R2 پیام‌های تک بیتی هستند که برای اعلام پذیرش یا عدم‌پذیرش طرفین ارسال می‌شوند. مطابق استانداردها، این پارامترها به صورت رمز نشده و بدون هیچ گونه صحت‌سنجی ارسال و دریافت می‌شوند. بدیهی

است، دستکاری این پارامترها توسط مهاجم امکان‌پذیر است بدون این‌که گیرنده از این دستکاری مطلع شود.

ضعف ویژگی احراز هویت در پروتکل‌های TETRA3 و TETRA4 نیز از R1 و R2 ناشی می‌شود. مهاجم امکان تغییر این مقادیر را در گام‌های اجرایی پروتکل دارد. این امر سبب می‌شود که مهاجم بتواند طرفین را به اشتباه بیاندازد. مثلا در صورت تایید، به آن‌ها اعلام کند که تایید نشده‌اند یا در صورت عدم تایید به آن‌ها اعلام کند که تایید شده‌اند. این امر در واقع نوعی از حملات منع سرویس است.

بدیهی است برای رفع مشکل تمامیت و احراز هویت در انواع پروتکل‌های TETRA، کافی است از ساز و کارهای صحت سنجی بدرستی بهره گرفته شود. اما برای ویژگی امنیت پیشرو، با ساختارهای مبتنی بر رمزنگاری متقارن نمی‌توان چاره‌ای اندیشید بلکه باید به‌طور مبنایی پروتکل را از متقارن به نامتقارن تغییر داد.

در نهایت نشان دادیم که با استفاده از تحلیل صوری پروتکل‌ها علاوه بر آن‌که تمامی تحلیل های غیرصوری که به این پروتکل در منابع آشکار ارائه‌شده را پوشش دهد توانسته ضعف‌های جدید نیز استخراج نماید.

۶- مراجع

[1] B. Torke Ladani, “Formal Analyzing of Security Protocols,” PHD Dissertation, Tarbiat Modars Universiry, 2004.

[2] M. Abadi, B. Blanchet, and C. Fournet, “The applied pi calculus: Mobile values, new names, and secure communication,” Report arXiv: 1609.03003v1, September 2016. Available at http://arxiv.org/abs/1609.03003v1.

[3] B. Blanchet, “Automatic Verification of Security Protocols in the Symbolic Model: the Verifier ProVerif. In FOSAD 7,” LNCS, vol. 8604, pp. 54-87, 2014.

[4] B. Blanchet, “Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif,” Foundations and Trends in Privacy and Security, pp. 1-135, October 2016.

[5] Proverif, http://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[6] D. Basin, C. Cremers, and C. Meadow, “Handbook of Model Checking: Model Checking Security Protocols,” springer, 2017.

[7] C. Cremers and S. Mauw, “Operational Semantics and Verification of Security Protocols,” Springer, 2012.

[8] C. Cremers, “The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols,” CAV 2008, LNCS, vol. 5123, pp. 414-418, 2008.

[9] S. Duan, “Security Analysis of TETRA,” Master thesis, Norwegian University of Science and Technology, 2013.

[10] Scyther, https://www.cs.ox.ac.uk/people/cas.cremers/scyther/index.html

[11] “ETSI Technical Standard ETSI EN 302 109 V1.1.1 Terrestrial Trunked Radio (TETRA),” Security Synchronization mechanism for end-to-end encryption, 2003.

[12] “TETRA Association SFPG Information document,” Overview of Standard TETRA Cryptographic Algorithms and their rules for management and distribution.

[13] ETSI Technical Report TR 052101 V1.1.1: SAGE Rules for the management of the TETRA standard authentication and key management algorithm set TAA1.

[14] ETSI Technical Report TR 1-0530101 V1.1.2: SAGE Rules for the management of the TETRA standard encryption algorithms Part1 TEA1.

[15] ETSI Technical Standard ETSI EN 300 392-7 V2.1.1: Terrestrial Trunked Radio (TETRA), Voice plus Data) V+D), Part 7: Security.

پیوست ۱ : نتایج حاصل از اجرای کد پروویریف TETRA1

در این بخش، نتایج حاصل از اجرای کدهای پیاده‌سازی نسخه‌های مختلف پروتکل تترا در نرم‌افزار پروویریف آورده شده است.

الف) نتایج مربوط به محرمانگی

--Query not attacker(secretSwMI_DCK([]

Completing...

Starting query not attacker(secretSwMI_DCK([]

RESULT not attacker(secretSwMI_DCK ([]is **true**.

--Query not attacker(secretMS_DCK([]

Completing...

Starting query not attacker(secretMS_DCK([]

RESULT not attacker(secretMS_DCK[]) is **true**.

--Query not attacker(secretSwMI_K([]

Completing...

Starting query not attacker(secretSwMI_K([]

RESULT not attacker(secretSwMI_K[]) is **true**.

--Query not attacker(secretMS_K([]

Completing...

Starting query not attacker(secretMS_K([]

RESULT not attacker(secretMS_K[]) is **true**.

ب) نتایج مربوط به ارزیابی احراز هویت

۱) احراز هویت MS توسط SwMI

--Query inj-event(endSwMI(x_4280,y_4281,r_4282,k_4283 ((<==inj-event(beginMS(x_4280,y_4281,r_4282((

Completing...

Starting query inj-event(endSwMI(x_4280,y_4281,r_4282,k_4283 ((<==inj-event(beginMS(x_4280,y_4281,r_4282((

goal reachable: egin(beginMS(hostMS[],hostSwMI[],RAND1_34[!1 = endsid_5171]), RS = RS_35[!1 = endsid_5171], RAND1 = RAND1_34[!1 = endsid_5171], @sid = @sid_5172, @occ8 = @occ_cst <- (end(endsid_5171,endSwMI(hostMS[],hostSwMI[],RAND1_34[!1 = endsid_5171],kdf((TA11(RS_35[!1 = endsid_5171],K[]),RAND1_34[!1 = endsid_5171((((

RESULT inj-event(endSwMI(x_4280,y_4281,r_4282,k_4283 ((<==inj-event(beginMS(x_4280,y_4281,r_4282)) is **true**.

۲) احراز هویت MS توسط SwMI

--Query inj-event(endMS(x_3411,y_3412,r_3413,k_3414 <== ((inj-event(beginSwMI(x_3411,y_3412,r_3413((

Completing...

goal reachable: attacker(r_4234 && (attacker(RS_4235 (<-end(endsid_4236,endMS(hostMS[],hostSwMI[],r_4234,kdf((TA11(RS_4235,K[]),r_4234((((

attacker((r_4250,RS_4245.((

.۴Using the function R1 the attacker may obtain R1. attacker(R1.(

.۵The message (r_4250,RS_4245) that the attacker may have by 3 may be received at input {4.{

The message R1 that the attacker may have by 4 may be received at input.{۱۰}

So event

endMS(hostMS[],hostSwMI[],r_4250,kdf((TA11(RS_4245,K[]),r_4250))) may be executed at {13} in session endsid_4247.

end(endsid_4247,endMS(hostMS[],hostSwMI[],r_4250,kdf((TA11(RS_4245,K[]),r_4250.((((

A more detailed output of the traces is available with set traceDisplay = long.

new K creating K_4254 at {1{

insert db(hostMS,hostSwMI,K_4254) at {2{

in(pubch, (a_4252,a_4253)) at {4} in copy a

event(beginMS(hostMS,hostSwMI,a_4252)) at {8} in copy a

out(pubch, TA12(TA11(a_4253,K_4254),a_4252)) at {9} in copy a

in(pubch, R1) at {10} in copy a

event(endMS(hostMS,hostSwMI,a_4252,kdf((TA11(a_4253,K_4254),a_4252 ((((at {13} in copy a

The event

endMS(hostMS,hostSwMI,a_4252,kdf((TA11(a_4253,K_4254),a_4252))) is executed in session a.

A trace has been found.

RESULT inj-event(endMS(x_3411,y_3412,r_3413,k_3414 <== ((inj-event(beginSwMI(x_3411,y_3412,r_3413)) is **false**.

RESULT (even event(endMS(x_4237,y_4238,r_4239,k_4240 <== ((event(beginSwMI(x_4237,y_4238,r_4239)) is false(.

ج) نتایج مربوط به ارزیابی UKS

--Query

event(endMSSwMI(x_1557,y_1558,r_1559,rs_1560,k_1561,x_1557,y_1558,r_1559,rs_1560,k'_1566 <== ((k_1561 = k'_1566

Completing...

Starting query

event(endMSSwMI(x_1557,y_1558,r_1559,rs_1560,k_1561,x_1557,y_1558,r_1559,rs_1560,k'_1566 <== ((k_1561 = k'_1566

goal reachable:

end(endMSSwMI(hostMS[],hostSwMI[],RAND1_34[!1 @ =sid_2471],RS_35[!1 = @sid_2471],kdf((TA11(RS_35[!1 @ =sid_2471],K[]),RAND1_34[!1 = @sid_2471])),hostMS[],hostSwMI[],RAND1_34[!1 = @sid_2471],RS_35[!1 = @sid_2471],kdf((TA11(RS_35[!1 = @sid_2471],K[]),RAND1_34[!1 = @sid_2471((([

RESULT event(endMSSwMI(x_1557,y_1558,r_1559,rs_1560,k_1561, x_1557,y_1558,r_1559,rs_1560,k'_1566 <== ((k_1561 = k'_1566 is true.

د) نتایج مربوط به ارزیابی کلید نشست مشترک

--Query

event(endMSSwMI(x_2483,y'_2489,r_2485,rs_2486,k_2487,x'_2488,y_2484,r_2485,rs_2486,k_2487) <== ((x_2483 = x'_2488 && y_2484 = y'_2489(

Completing...

Starting query

event(endMSSwMI(x_2483,y'_2489,r_2485,rs_2486,k_2487,x'_2488,y_2484,r_2485,rs_2486,k_2487) <== ((x_2483 = x'_2488 && y_2484 = y'_2489(

goal reachable: end(endMSSwMI(hostMS[],hostSwMI[],RAND1_34[!1 = @sid_3398],RS_35[!1@ = sid_3398],kdf((TA11(RS_35[!1 = @sid_3398],K[]),RAND1_34[!1 = @sid_3398])),hostMS[],hostSwMI[],RAND1_34[!1 = @sid_3398],RS_35[!1@ = sid_3398],kdf((TA11(RS_35[!1 = @sid_3398],K[]),RAND1_34[!1 = @sid_3398((([

RESULT event(endMSSwMI(x_2483,y'_2489,r_2485,rs_2486,k_2487, x'_2488,y_2484,r_2485,rs_2486,k_2487) <== ((x_2483 = x'_2488 && y_2484 = y'_2489) is **true**.

ه) نتایج ارزیابی گمنامی

--Query not attacker(hostSwMI([]

Completing...

Starting query not attacker(hostSwMI([]

RESULT not attacker(hostSwMI[]) is **true**.

--Query not attacker(hostMS([]

Completing...

Starting query not attacker(hostMS([]

RESULT not attacker(hostMS[]) is **true**.

پیوست۲ : نتایج حاصل از اجرای کد پروریف TETRA2

نتایج حاصل از اجرای کد پروریف TETRA2 به صورت زیر می باشد

الف) نتایج ارزیابی محرمانگی

--Query not attacker(secretSwMI_DCK([]

Completing...

Starting query not attacker(secretSwMI_DCK([]

RESULT not attacker(secretSwMI_DCK ([]is **true**.

--Query not attacker(secretMS_DCK([]

Completing...

Starting query not attacker(secretMS_DCK([]

RESULT not attacker(secretMS_DCK[]) is **true**.

--Query not attacker(secretSwMI_K([]

Completing...

Starting query not attacker(secretSwMI_K([]

RESULT not attacker(secretSwMI_K[]) is **true**.

--Query not attacker(secretMS_K([]

Completing...

Starting query not attacker(secretMS_K([]

RESULT not attacker(secretMS_K[]) is **true**.

ب) نتایج مربوط به ارزیابی احراز هویت

۱) احراز هویت MS توسط SwMI

--Query inj-event(endMS(x_2838,y_2839,r_2840,k_2841 <== ((inj-event(beginSwMI(x_2838,y_2839,r_2840((

Completing...

Starting query inj-event(endMS(x_2838,y_2839,r_2840,k_2841 <== ((inj-event(beginSwMI(x_2838,y_2839,r_2840((

RESULT inj-event(endMS(x_2838,y_2839,r_2840,k_2841 <== ((inj-event(beginSwMI(x_2838,y_2839,r_2840 ((is **true**.

۲) احراز هویت MS توسط SwMI

--Query inj-event(endSwMI(x_3702,y_3703,r_3704,k_3705 <== ((inj-event(beginMS(x_3702,y_3703,r_3704((

Completing...

Starting query inj-event(endSwMI(x_3702,y_3703,r_3704,k_3705 <== ((inj-event(beginMS(x_3702,y_3703,r_3704((

goal reachable: attacker(r_4538 <- (end(endsid_4539,endSwMI(hostMS[],hostSwMI[],r_4538,kdf((TA21(RS_36[RAND2_35 = r_4538,k_34 = K[],!1 = endsid_4539],K[]),r_4538((((

Abbreviations:

RS_4553 = RS_36[RAND2_35 = r_4551,k_34 = K[],!1 = endsid_4548[

.۱The entry db(hostMS[],hostSwMI[],K[]) may be inserted in a table at insert {2 .{ table(db(hostMS[],hostSwMI[],K.(([]

.۲We assume as hypothesis that attacker(r_4551.(

.۳Using the function R2 the attacker may obtain R2. attacker(R2.(

.۴The entry db(hostMS[],hostSwMI[],K[]) that may be in a table by 1 may be read at get {33.{

The message r_4551 that the attacker may have by 2 may be received at input {20.{

The message R2 that the attacker may have by 3 may be received at input {27.{

So event endSwMI(hostMS[],hostSwMI[],r_4551,kdf((TA21(RS_4553,K[]),r_4551 (((may be executed at {29} in session endsid_4548.

RESULT event(endMSSwMI(x_1433,y_1434,r,rs,k_1435,x_1433,y_1434,r,rs,k <== (('k_1435 = k' is **true**.

د) نتایج مربوط به ارزیابی کلید نشست مشترک

--Query

event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137(

Completing...

Starting query

event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137(

RESULT event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137) is **true**.

ه) نتایج ارزیابی گمنامی

--Query not attacker(hostSwMI([]

Completing...

Starting query not attacker(hostSwMI([]

RESULT not attacker(hostSwMI[]) is **true**.

--Query not attacker(hostMS([]

Completing...

Starting query not attacker(hostMS([]

RESULT not attacker(hostMS[]) is **true**.

end(endsid_4548,endSwMI(hostMS[],hostSwMI[],r_4551,kdf((TA21(RS_4553,K[]),r_4551.((((

A more detailed output of the traces is available with set traceDisplay = long.

new K creating K_4555 at {1{

insert db(hostMS,hostSwMI,K_4555) at {2{

get db(hostMS,hostSwMI,K_4555) at {33} in copy a

in(pubch ,a_4554) at {20} in copy a

new RS_36 creating RS_4556 at {21} in copy a

event(beginSwMI(hostMS,hostSwMI,a_4554 ((at {24} in copy a

out(pubch ,)TA22(TA21(RS_4556,K_4555),a_4554),RS_4556)) at {25} in copy a

in(pubch ,R2) at {27} in copy a

event(endSwMI(hostMS,hostSwMI,a_4554,kdf((TA21(RS_4556,K_4555),a_4554 ((((at {29} in copy a

The event endSwMI(hostMS,hostSwMI,a_4554,kdf((TA21(RS_4556,K_4555),a_4554 (((is executed in session a.

A trace has been found.

RESULT inj-event(endSwMI(x_3702,y_3703,r_3704,k_3705 <== ((inj-event(beginMS(x_3702,y_3703,r_3704)) is **false**.

RESULT (even event(endSwMI(x_4540,y_4541,r_4542,k_4543 <== ((event(beginMS(x_4540,y_4541,r_4542)) is **false**(.

ج) نتایج مربوط به ارزیابی UKS

Query -- event(endMSSwMI(x_1433,y_1434,r,rs,k_1435,x_1433,y_1434,r,rs,k 'k_1435 = k <== (('34,r,rs,k

...Completing

query Starting event(endMSSwMI(x_1433,y_1434,r,rs,k_1435,x_1433,y_1434,r,rs,k 'k_1435 = k <== (('34,r,rs,k

پیوست ۳ : نتایج حاصل از اجرای کد پروریف TETRA3

نتایج حاصل از اجرای کد پروریف TETRA3 به صورت زیر می باشد:

الف) نتایج ارزیابی محرمانگی

--Query not attacker(secretSwMI_DCK([])

Completing...

Starting query not attacker(secretSwMI_DCK([])

RESULT not attacker(secretSwMI_DCK ([])is **true**.

--Query not attacker(secretMS_DCK([])

Completing...

Starting query not attacker(secretMS_DCK([])

RESULT not attacker(secretMS_DCK[]) is **true**.

--Query not attacker(secretSwMI_K([])

Completing...

Starting query not attacker(secretSwMI_K([])

RESULT not attacker(secretSwMI_K[]) is **true**.

--Query not attacker(secretMS_K([])

Completing...

Starting query not attacker(secretMS_K([])

RESULT not attacker(secretMS_K[]) is **true**.

ب) نتایج مربوط به ارزیابی احراز هویت

۱) احراز هویت MS توسط SwMI

---Query

inj-event(endMS(x_5123,y_5124,r1_5125,r2_5126,k_5127 ((<==inj-event(beginSwMI(x_5123,y_5124,r1_5125((

Completing...

Starting query

RESULT

inj-event(endMS(x_5123,y_5124,r1_5125,r2_5126,k_5127 ((<==inj-event(beginSwMI(x_5123,y_5124,r1_5125)) is **true**.

۲) احراز هویت MS توسط SwMI

--Query

inj-event(endSwMI(x_6488,y_6489,r1_6490,r2_6491,k_6492 ((<==inj-event(beginMS(x_6488,y_6489,r1_6490,r2_6491((

Completing...

Starting query

inj-event(endSwMI(x_6488,y_6489,r1_6490,r2_6491,k_6492 ((<==inj-event(beginMS(x_6488,y_6489,r1_6490,r2_6491((

...

.۱۰Using the function R2 the attacker may obtain R2.

attacker(R2.(

.۱۱The entry db(hostMS[],hostSwMI[],K[]) that may be in a table by 1 may be read at get {45.{

.....

The event

endSwMI(hostMS,hostSwMI,RAND1_8008,a_8005,TB4 ((kdf((TA11(RS_8009,K_8007),RAND1_8008)),kdf((TA21(RS_8009,K_8007),a_8005 (((((is executed in session a.

A trace has been found.

RESULT

inj-event(endSwMI(x_6488,y_6489,r1_6490,r2_6491,k_6492 ((<==inj-event(beginMS(x_6488,y_6489,r1_6490,r2_6491)) is **false**.

RESULT

)even

event(endSwMI(x_7969,y_7970,r1_7971,r2_7972,k_7973 ((

RESULT not attacker(hostMS[]) is **true**.

پیوست ۴ : نتایج حاصل از اجرای کد پرووریف TETRA4

نتایج حاصل از اجرای کد پرووریف TETRA4 به صورت زیر می باشد:

الف) نتایج ارزیابی محرمانگی

--Query not attacker(secretSwMI_DCK([]

Completing...

RESULT not attacker(secretSwMI_DCK ([]is **true**.

Completing...

Starting query not attacker(secretMS_DCK([]

RESULT not attacker(secretMS_DCK[]) is **true**.

--Query not attacker(secretSwMI_K([]

Completing...

Starting query not attacker(secretSwMI_K([]

RESULT not attacker(secretSwMI_K[]) is **true**.

--Query not attacker(secretMS_K([]

Completing...

Starting query not attacker(secretMS_K([]

RESULT not attacker(secretMS_K[]) is **true**.

ب) نتایج مربوط به ارزیابی احراز هویت

۱) احراز هویت MS توسط SwMI

--Query

inj-event(endSwMI(x_6259,y_6260,r1_6261,r2_6262,k_6263 ((
<==inj-event(beginMS(x_6259,y_6260,r1_6261,r2_6262((

Completing...

Starting query

inj-event(endSwMI(x_6259,y_6260,r1_6261,r2_6262,k_6263 ((
<==inj-event(beginMS(x_6259,y_6260,r1_6261,r2_6262((

<==event(beginMS(x_7969,y_7970,r1_7971,r2_7972)) is **false**(.

ج) نتایج مربوط به ارزیابی UKS

--Query

event(endMSSwMI(x_3745,y'_3752,r1_3747,r2_3748,rs_3749,k_3750,x'_3751,y_3746,r1_3747,r2_3748,rs_3749,k_3750) <== ((x_3745 = x'_3751 && y_3746 = y'_3752(

Completing...

Starting query

RESULT event(endMSSwMI(x_3745,y'_3752,r1_3747,r2_3748,rs_3749,k_3750,x'_3751,y_3746,r1_3747,r2_3748,rs_3749,k_3750) <== ((x_3745 = x'_3751 && y_3746 = y'_3752) is **true**.

د) نتایج مربوط به ارزیابی کلید نشست مشترک

--Query

event(endMSSwMI(x_2369,y_2370,r1_2371,r2_2372,rs_2373,k_2374,x_2369,y_2370,r1_2371,r2_2372,rs_2373,k'_2380 <== ((k_2374 = k'_2380

Completing...

Starting query

event(endMSSwMI(x_2369,y_2370,r1_2371,r2_2372,rs_2373,k_2374,x_2369,y_2370,r1_2371,r2_2372,rs_2373,k'_2380 <== ((k_2374 = k'_2380

RESULT event(endMSSwMI(x_2369,y_2370,r1_2371,r2_2372,rs_2373,k_2374,x_2369,y_2370,r1_2371,r2_2372,rs_2373,k'_2380 <== ((k_2374 = k'_2380 is **true**.

ه) نتایج ارزیابی گمنامی

--Query not attacker(hostSwMI([]

Completing...

Starting query not attacker(hostSwMI([]

RESULT not attacker(hostSwMI[]) is **true**.

--Query not attacker(hostMS([]

Completing...

Starting query not attacker(hostMS([]

RESULT

inj-event(endSwMI(x_6259,y_6260,r1_6261,r2_6262,k_6263 ((<==inj-event(beginMS(x_6259,y_6260,r1_6261,r2_6262)) is **true**.

۲) احراز هویت MS توسط SwMI

--Query

inj-event(endMS(x_4972,y_4973,r1_4974,r2_4975,k_4976 ((<==inj-event(beginSwMI(x_4972,y_4973,r1_4974,r2_4975((

Completing...

Starting query

inj-event(endMS(x_4972,y_4973,r1_4974,r2_4975,k_4976 ((<==inj-event(beginSwMI(x_4972,y_4973,r1_4974,r2_4975((

RESULT

inj-event(endMS(x_4972,y_4973,r1_4974,r2_4975,k_4976 ((<==inj-event(beginSwMI(x_4972,y_4973,r1_4974,r2_4975)) is **true**.

ج) نتایج مربوط به ارزیابی UKS

--Query

event(endMSSwMI(x_3638,y'_3645,r1_3640,r2_3641,rs_3642,k_3643,x'_3644,y_3639,r1_3640,r2_3641,rs_3642,k_3643) <== ((x_3638 = x'_3644 && y_3639 = y'_3645(

Completing...

Starting query event(endMSSwMI(x_3638,y'_3645,r1_3640,r2_3641,rs_3642,k_3643,x'_3644,y_3639,r1_3640,r2_3641,rs_3642,k_3643) <== ((x_3638 = x'_3644 && y_3639 = y'_3645(

RESULT event(endMSSwMI(x_3638,y'_3645,r1_3640,r2_3641,rs_3642,k_3643,x'_3644,y_3639,r1_3640,r2_3641,rs_3642,k_3643) <== ((x_3638 = x'_3644 && y_3639 = y'_3645) is **true**.

د) نتایج مربوط به ارزیابی کلید نشست مشترک

--Query

event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137(

Completing...

Starting query

event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137(

RESULT event(endMSSwMI(x_2131,y'_2137,r_2133,rs_2134,k_2135,x'_2136,y_2132,r_2133,rs_2134,k_2135) <== ((x_2131 = x'_2136 && y_2132 = y'_2137) is **true**.

ه) نتایج ارزیابی گمنامی

--Query not attacker(hostSwMI([]

Completing...

Starting query not attacker(hostSwMI([]

RESULT not attacker(hostSwMI[]) is **true**.

--Query not attacker(hostMS([]

Completing...

Starting query not attacker(hostMS([]

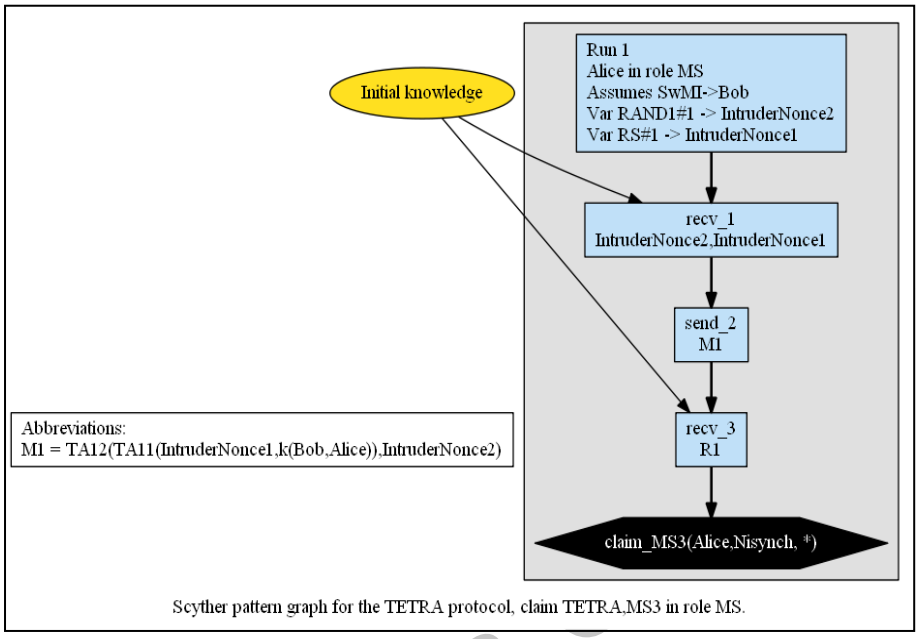
RESULT not attacker(hostMS[]) is **true**.

پیوست ۵ : نتایج حاصل از اجرای کد اسکایتر

در این بخش، نتایج حاصل از اجرای کدهای پیاده‌سازی نسخه‌های مختلف پروتکل تترا در نرم‌افزارهای اسکایتر آورده شده است.

الف – نتایج حاصل از اجرای کد اسکایتر TETRA1

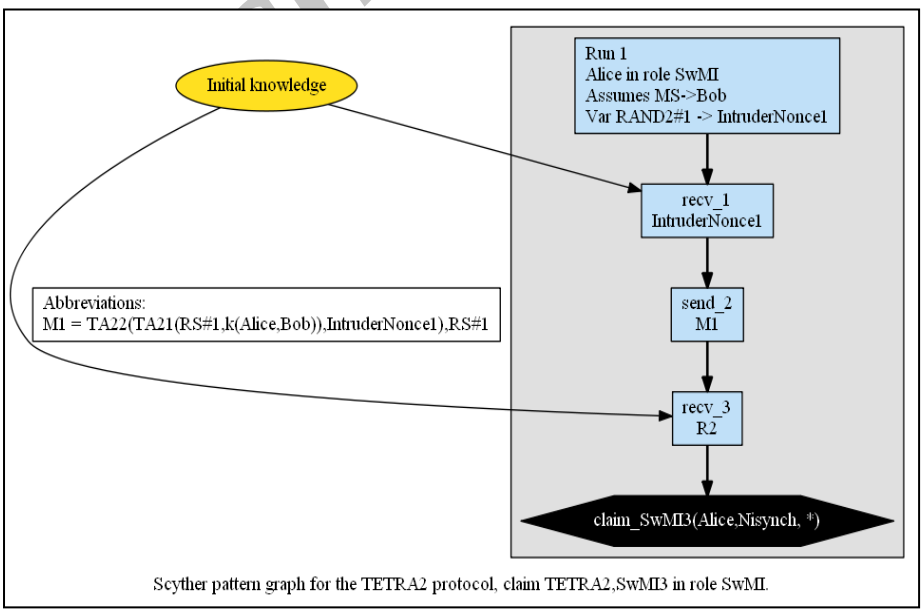
حمله گزارش شده توسط اسکایتر بر روی پروتکل TETRA1 در شکل (۷) آمده است. احراز هویت MS توسط SwMI تایید می‌شود اما احراز هویت SwMI توسط MS تایید نمی‌شود.



شکل (۷): حمله جعل هویت SwMI بر روی پروتکل TETRA1

ب- نتایج حاصل از اجرای کد اسکایتر TETRA2

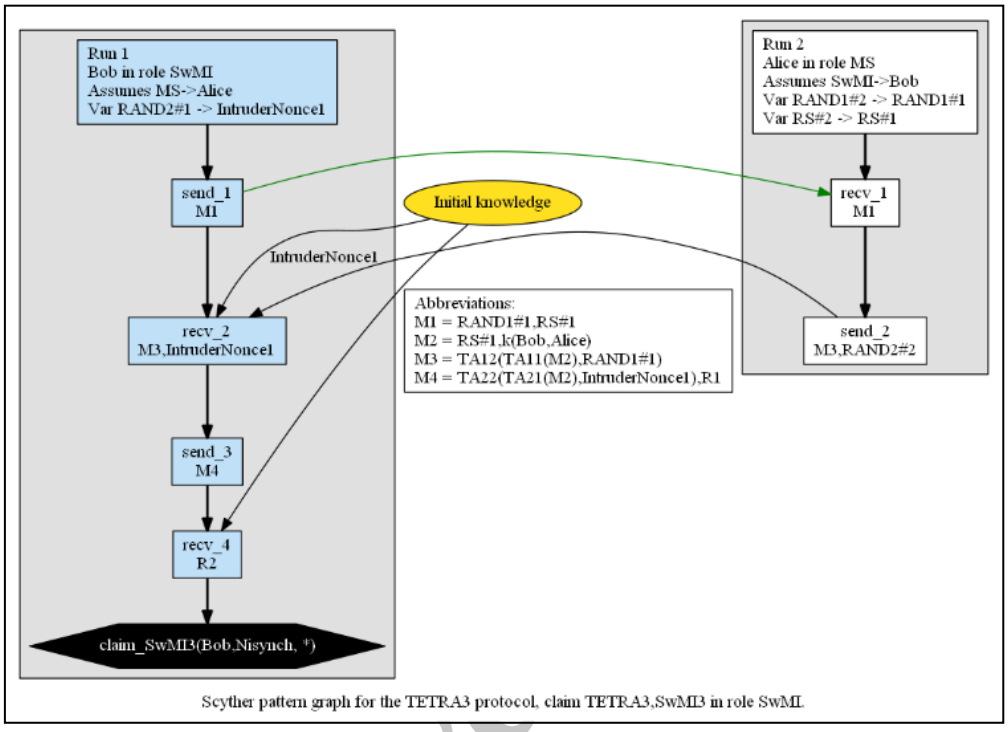
حمله گزارش شده توسط اسکایتر بر روی پروتکل TETRA2 در شکل (۸) آمده است. احراز هویت SwMI توسط MS تایید می‌شود اما احراز هویت MS توسط SwMI تایید نمی‌شود.



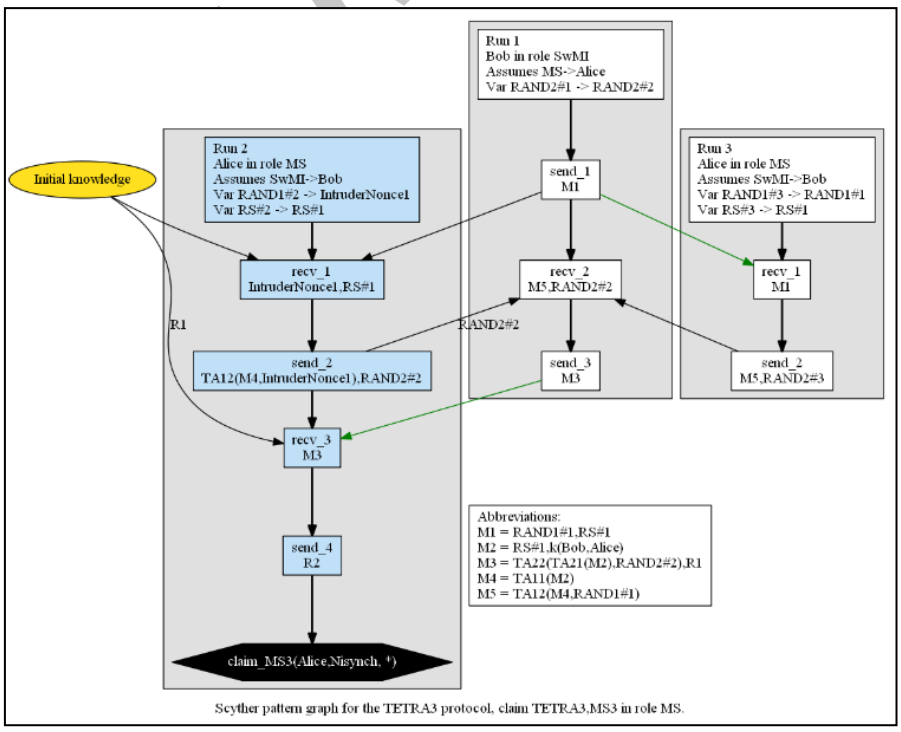
شکل (۸): حمله جعل هویت MS بر روی پروتکل TETRA2

ج- نتایج حاصل از اجرای کد اسکایتر TETRA3

از نتایج حاصل از ارزیابی TETRA3 مشاهده می‌شود که برای برای نقش MS ویژگی احراز هویت برقرار نیست. برای نقش SwMI نیز ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت که تایید نشده حملاتی گزارش شده است. این حملات در شکل‌های (۱۰-۹) مشاهده می‌شود.



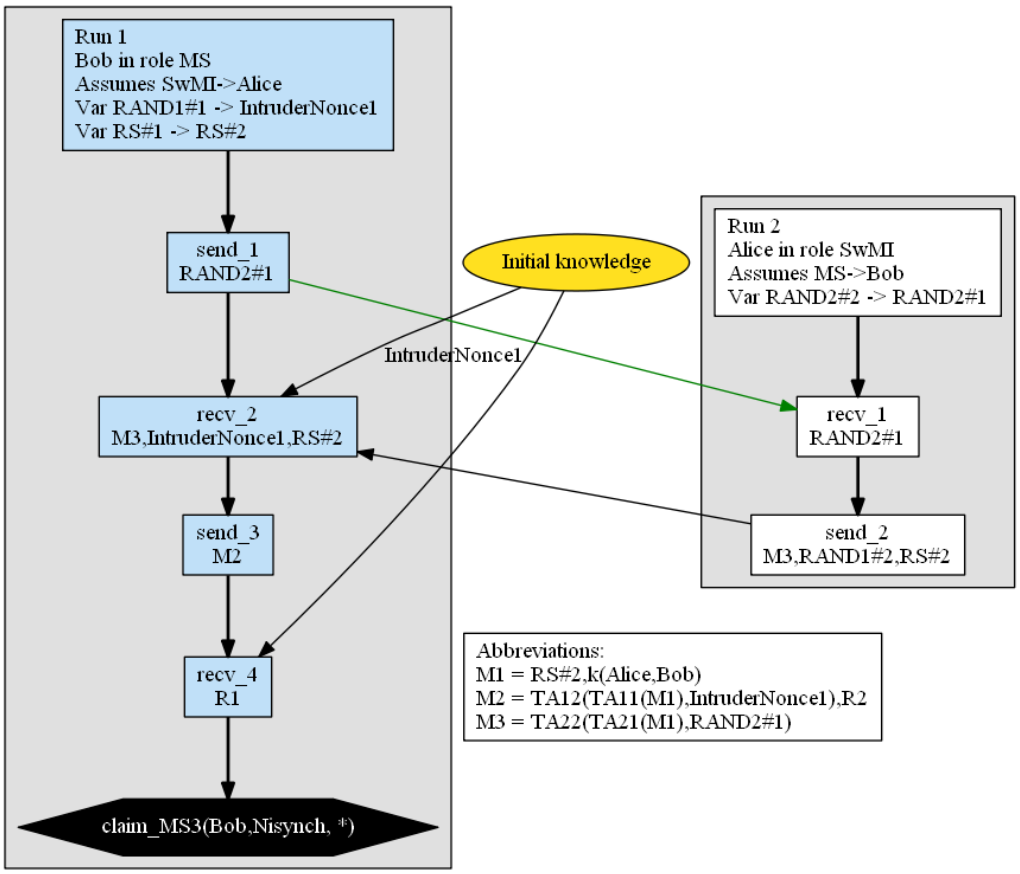
شکل (۹): دستکاری اطلاعات دریافتی توسط SwMI در پروتکل TETRA3



شکل (۱۰): دستکاری اطلاعات دریافتی توسط MS در پروتکل TETRA3

د- نتایج حاصل از اجرای کد اسکایتر TETRA4

نتایج حاصل از ارزیابی TETRA4 نشان می دهد که برای نقش SwMI ویژگی احراز هویت برقرار است اما برای نقش MS ویژگی احراز هویت برقرار نیست. برای ویژگی احراز هویت که تایید نشده است یک حمله گزارش شده که در شکل (۱۱) نشان داده شده است..



Scyther pattern graph for the TETRA4 protocol, claim TETRA4,MS3 in role MS.

شکل (۱۱): دستکاری اطلاعات دریافتی توسط SwMI در پروتکل TETRA4

Analysis and Evaluation of the TETRA Network Security Protocols Using Automated Analysis Tools

M. Mollazadeh*, M. Sabzineghad, R. Rastaghi

*Imam Hossein University

(Received: 22/07/2016, Accepted: 33/07/2017)

ABSTRACT

In this paper, the structure of the various versions of the TETRA security protocol is investigated in the "formal model" using Proverif and Scyther automatic analysis tools. The TETRA's network security protocol is a key-exchange one, in which two parties also establish a session key while authenticating each other. This protocol also uses pre-distributed secret keys which are based on the symmetric-encryption schemes. The security analysis of the protocol has been done in the "formal model", using the Proverif and Scyther automatic analysis tools. Firstly, eight security features including Confidentiality, Authentication, Forward Secrecy, Unknown Key-Share security, Identical Session Key, Unknown Key Security, Anonymity, and Integrity are modeled in these frameworks, and then using both of the two tools, the security of the protocol is investigated regarding the mentioned features. Comparing the results of the formal analysis of these features with the informal analysis resulted from the open sources indicates that there are new security flows in the structure of the protocol respect to "Forward Secrecy" and "integrity". Finally, several solutions are suggested to overcome these weaknesses.

Keywords: Security Analysis, Formal Models, Automatic Analysis Tools, TETRA Network

* Corresponding Author Email: mollazadeh@dspri.com