

یک پروتکل احراز اصالت جدید مبتنی بر خم بیضوی برای سامانه‌های RFID

حامد سیاوشی^{۱*}، عبدالرسول میرقدری^۲، مهدی عزیزی^۳

۱- دانشجوی کارشناسی ارشد، دانشکده فاوا، دانشگاه جامع امام حسین (ع)

۲- دانشیار، گروه رمز و امنیت، دانشکده فاوا، دانشگاه جامع امام حسین (ع)

۳- استادیار، گروه رمز و امنیت، دانشکده فاوا، دانشگاه جامع امام حسین (ع)

چکیده

شناسایی با استفاده از امواج رادیویی (RFID) یک فناوری نوین است که به علت مزایای زیاد در زمینه‌های مختلف تجاری، صنعتی و نظامی، جهت شناسایی و احراز هویت استفاده می‌شود. با توجه به اهمیت و گسترش فناوری RFID از پروتکل‌های احراز اصالت برای ایجاد امنیت ارتباطات این سامانه استفاده می‌گردد. حمله منع سرویس یکی از مهم‌ترین آسیب‌پذیری‌های پروتکل‌های احراز اصالت RFID محسوب می‌گردد. در این مقاله ابتدا، یک پروتکل مبتنی بر ابرخم بیضوی طراحی می‌شود که در آن از یک الگوریتم امضاء دیجیتال و یک الگوریتم رمزنگاری متقارن به گونه‌های استفاده شده است که، حجم محاسباتی اضافی برای سیستم RFID ایجاد نمی‌کند. پروتکل پیشنهادی به گونه‌های طراحی شده که در برابر حمله‌های مطرح، مخصوصاً حمله منع سرویس مقاوم است. در نهایت پروتکل پیشنهادی را توسط نرم افزار تحلیل خودکار اسکایتر تحلیل کرده که نتایج حاصل از این تحلیل نشان از مقاوم بودن پروتکل طراحی شده، در برابر حمله‌های مطرح را دارد.

کلمات کلیدی: سامانه‌های RFID، پروتکل‌های احراز اصالت، خم بیضوی، حمله منع سرویس، الگوریتم.

۱. مقدمه

پیشرفت روز افزون فناوری مدارهای مجتمع در سال‌های اخیر و کاهش هزینه‌های استفاده از فناوری RFID شاهد رشد چشمگیر و به کارگیری سامانه‌های RFID در حوزه‌های مختلفی چون مدیریت زنجیره تامین، ترابری و پشتیبانی قوای نظامی و دفاعی، بارکدهای الکترونیکی، بلیط‌های حمل و نقل عمومی و گذرنامه‌های الکترونیکی می‌باشیم. در آینده‌های نزدیک یک بخش اساسی در اینترنت اشیاء [۱] را سامانه‌های RFID تشکیل خواهد داد. سامانه‌های RFID شامل سه بخش: برچسب، برچسب‌خوان و سرویس دهنده مرکزی (کارگزار) مطابق شکل (۱) می‌باشند. ارتباط میان برچسب و برچسب‌خوان از طریق کانال بی‌سیم و به واسطه امواج رادیویی است، در حالی که ارتباط میان برچسب‌خوان و کارگزار از طریق کانال باسیم و امن است.

¹ Email: hamedsiavashi@chmail.ir

² Radio Frequency IDentification



شکل ۱. نحوه عملکرد سیستم RFID [۲].

نحوه عملکرد این سامانه‌ها بدین گونه است که به هر برچسب، یک شناسه منحصر به فرد اختصاص داده می‌شود و این شناسه در حافظه برچسب ذخیره می‌گردد. هر برچسب دارای یک تراشه بسیار کوچک نیز هست که ابزارهای محاسباتی و پردازشی در این تراشه واقع می‌شوند. این برچسب‌ها طبق اهداف مورد نظر می‌توانند برای شناسایی بر روی کالا، انسان و یا حیوان، کار گذاشته شوند. دستگاه‌های برچسب‌خوان نیز در مکان‌های متناسبی نصب می‌گردند تا بتوانند با برچسب‌ها تبادل اطلاعات نمایند. برچسب‌خوان‌ها از طریق یک کانال امن به سرویس دهنده مرکزی متصل می‌شوند. در سرویس دهنده مرکزی نیز اطلاعات مربوط به همه برچسب‌ها در یک حافظه امن و بزرگ ذخیره می‌شود و با دریافت شناسه برچسب از طریق برچسب‌خوان در حافظه خود بررسی می‌کند، اگر برچسب جزو برچسب‌های مجاز باشد سرویس دهنده مرکزی برچسب را تایید هویت می‌نماید. در این مقاله به علت اینکه کانال ارتباطی بین برچسب‌خوان و سرویس دهنده مرکزی امن است، این دو را یکی و به عنوان سرور در نظر می‌گیریم.

ساختار بقیه مقاله به این شرح است که در بخش دوم، کارهای مرتبط در زمینه پروتکل‌های احراز اصالت مبتنی بر خم بیضوی را بیان و سپس در بخش سوم پروتکل طراحی شده را تشریح و در بخش چهارم تحلیل امنیتی این پروتکل را انجام می‌گردد و نشان می‌دهیم که پروتکل طراحی شده در برابر حمله‌های مطرح شده، به خصوص حمله منع سرویس مقاوم است. در بخش پنجم پروتکل طراحی شده نسبت به چهار پروتکل بررسی شده از لحاظ امنیتی مقایسه شده است. در بخش ششم یک تحلیل محاسباتی بر روی پروتکل طراحی شده انجام می‌گردد. در بخش هفتم پروتکل طراحی شده را توسط نرم افزار تحلیل خودکار پروتکل اسکایتر، مورد تحلیل نرم افزاری قرار داده گرفته است. در بخش هشتم نیز نتیجه گیری ارائه شده است.

۲. کارهای مرتبط

تاکنون پروتکل‌های زیادی برای احراز اصالت سامانه‌های RFID براساس الگوریتم‌های رمزنگاری شبیه [۵] IDEA و [۶] AES و [۷] ECC پیشنهاد شده است. در این مقاله تمرکز بر روی پروتکل‌هایی است که از سیستم رمزنگاری کلید عمومی خم بیضوی برای انجام احراز اصالت استفاده نموده‌اند. سیستم رمزنگاری خم بیضوی^۱ (ECC) برای سیستم RFID بسیار مناسب است زیرا می‌تواند یک سطح امنیتی با طول کلید کوتاه تر [۸] با حجم محاسباتی کمتری را فراهم نماید. در تحقیقات انجام شده [۱۱] نشان داده شده است که الگوریتم ECC با طول کلید ۱۶۰ بیت همان سطح امنیتی الگوریتم RSA با طول کلید ۱۰۲۴ بیت را فراهم می‌نماید. الگوریتم ECC بر روی بسیاری از برچسب‌های RFID قابل پیاده‌سازی است [۱۱].

^۱ Elliptic Curve Cryptography

طرح‌های احراز اصالت سامانه RFID مبتنی بر خم بیضوی طبق [۱۲] به سه دسته سنگین وزن، میان وزن، سبک وزن تقسیم می‌گردند. طرح‌های سنگین وزن مانند [۱۳] اغلب شامل عملیات‌های رمزنگاری کلید عمومی و امضاء دیجیتال هستند. در طرح‌های میان وزن مانند [۱۴] از عملیات خم بیضوی و تابع چکیده ساز استفاده می‌گردد. در طرح‌های سبک وزن مانند [۱۵] فقط از عملیات‌های خم بیضوی استفاده می‌شود. برای اولین بار در سال ۲۰۰۵ ولکراستروفر^۱ [۱۱] مفهوم طرح‌های احراز اصالت RFID مبتنی بر خم بیضوی را معرفی کرد. در سال ۲۰۰۶ توپلز^۲ و همکارانش [۱۶] یک طرح احراز اصالت RFID مبتنی بر ECC را ارائه کردند و نشان دادند که طرحشان در برابر حمله جعل برچسب مقاوم است. در سال ۲۰۰۸ لی^۳ و همکارانش [۱۷] نشان دادند که این طرح در برابر حمله ردیابی مقاوم نبوده و همچنین این طرح دارای ویژگی امنیت پیشرو نیست. در سال ۲۰۱۱ ژانگ^۴ و همکارانش [۱۹] یک طرح احراز اصالت بر اساس خم بیضوی و کلید تصادفی برای بهبود دو طرح لی [۱۷] و توپلز [۱۶] ارائه کردند و نشان داده‌اند که ضعف‌های این دو طرح را برطرف کرده است، اما باباهیدریان^۵ و همکارانش [۲۰] نشان دادند که طرح ژانگ [۱۹] دارای آسیب‌پذیری حمله جعل برچسب و سرور است. تمام این طرح‌های بیان شده دارای ویژگی احراز اصالت یکطرفه هستند و فقط سرور برچسب را احراز اصالت می‌کند که این موضوع باعث بروز حملات تکرار، فرد درمیان و منع سرویس به برچسب می‌گردد. در سال ۲۰۱۱ گودور و امیر^۶ [۲۱] یک پروتکل احراز اصالت دو طرفه سامانه‌های RFID مبتنی بر خم بیضوی را ارائه کردند که در آن از سیستم رمزنگاری الجمال^۷ و الگوریتم امضاء دیجیتال خم بیضوی^۸ (ECCDSA) استفاده شده بود. این پروتکل در برابر حمله منع سرویس در سمت برچسب مقاوم نیست. در سال ۲۰۱۳ هسیائو و لیائو^۹ [۲۲] یک پروتکل احراز اصالت مبتنی بر خم بیضوی ارائه کردند که ژائو^{۱۰} و همکارانش [۲۳] نشان دادند که این طرح دارای مشکل توافق کلید است به این معنی که مهاجم می‌تواند به اطلاعات مخفی ذخیره شده در سمت برچسب دسترسی داشته باشد. در سال ۲۰۱۴ چو^{۱۱} [۲۴] یک طرح احراز اصالت مبتنی بر خم بیضوی و یک تابع چکیده ساز را ارائه کرد و ژانگ [۱۹] و چی^{۱۲} [۲۵] نشان دادند که این طرح نیز همانند طرح هسیائو و لیائو دارای مشکل توافق کلید است. همچنین فرش^{۱۳} [۲۶] نشان داده است که طرح چو [۲۴] دارای ضعف امنیتی جعل برچسب است و راه کار بهبود این پروتکل را نیز بیان کرده است. در همین سال موسوی^{۱۴} و همکارانش [۲۷] یک پروتکل احراز اصالت دو طرفه بر اساس رمزنگاری خم بیضوی را ارائه کرده‌اند که در آن از یک الگوریتم امضاء دیجیتال استفاده شده است. در سال ۲۰۱۴ چن^{۱۵} و همکارانش [۲۹] یک پروتکل احراز اصالت دو طرفه مبتنی بر خم بیضوی را ارائه کردند که این پروتکل دارای ضعف‌های امنیتی مانند حمله جعل برچسب و تکرار است. در سال ۲۰۱۵ آقای جین^{۱۶} و همکارانش [۳۰] یک پروتکل احراز اصالت دو طرفه مبتنی بر خم بیضوی ارائه کرده‌اند که دارای ویژگی دسترس‌پذیری و امنیت پیشرو بوده و در برابر حملاتی مانند حمله تکرار، ردیابی، حمله استراق سمع، حمله جعل

¹ Wolkerstorfer

² Tuyls

³ Lee

⁴ Zhang

⁵ Babaheidarian

⁶ Godor and Imre

⁷ ElGamal

⁸ Elliptic Curve Cryptography Digital Signature Algorithm

⁹ Hsiao and Liao

¹⁰ Zhao

¹¹ Chou

¹² Qi

¹³ Farash

¹⁴ Moosavi

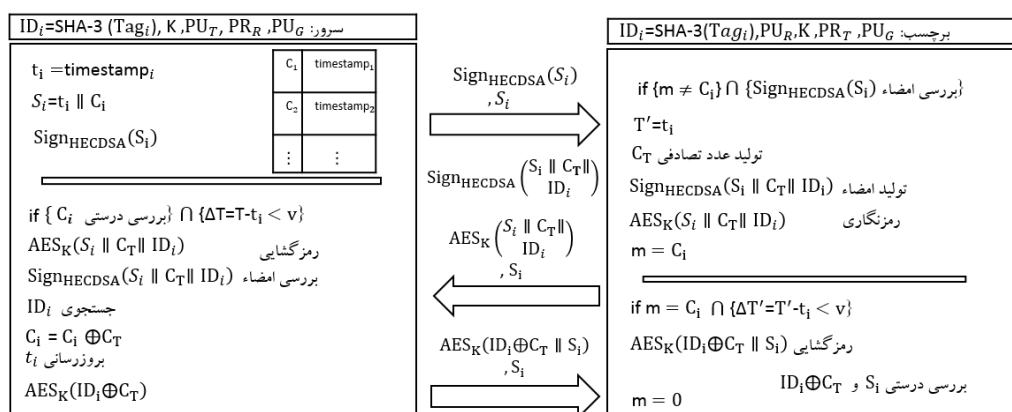
¹⁵ Chen

¹⁶ Jin

کارت‌خوان و جعل برچسب مقاوم است. در سال ۲۰۱۶ شان^۱ و همکارانش [۳۱] با بهبود پروتکل چن [۲۹] آن را در مقابل حمله جعل برچسب و حمله تکرار مقاوم ساختند، ولی این پروتکل همچنان در برابر حمله منع سرویس آسیب‌پذیر است.

۳. معرفی پروتکل پیشنهادی

بعد از تحلیل و بررسی چهار پروتکل منتخب به این نتیجه رسیده‌ایم که این پروتکل‌ها در برابر حمله منع سرویس آسیب‌پذیر هستند. با توجه به ساختار پروتکل‌های مختلف و ایده‌ای جدید، یک پروتکل احراز اصالت سامانه‌های RFID مبتنی بر ابر^۲ خم بیضوی را طراحی کرده‌ایم که در برابر حملات مطرح شده مخصوصاً حمله منع سرویس مقاوم است. در این پروتکل از یک امضاء دیجیتال مبتنی بر خم بیضوی ابر استفاده کرده‌ایم، که ترکیب الگوریتم امضاء دیجیتال DSA با سیستم رمزنگاری HECC است. در سال ۲۰۱۴ بارسگید^۳ و همکارانش [۲۸] با مقایسه خم‌های بیضوی ساده و ابر نشان دادند که سیستم رمزنگاری HECC نسبت به سیستم رمزنگاری خم بیضوی ECC دارای امنیت بالاتر و حجم محاسبات کمتری است به طوری که در این طرح نشان داده شده است که حل مسئله لگاریتم گسسته HECC با طول نقطه خم بیضوی ۸۰ بیت، سخت‌تر از حل مسئله لگاریتم گسسته ECC با طول نقطه خم بیضوی ۱۶۰ بیت است. بنابراین استفاده از سیستم رمزنگاری HECC در پروتکل‌های احراز اصالت سامانه‌های RFID بسیار مناسب‌تر از سیستم رمزنگاری ECC است. طبق [۳۲] در این پروتکل از امضاء دیجیتال ابر خم بیضوی با طول کلید ۱۶۰ بیت استفاده می‌کنیم که امنیت معادل یک امضاء دیجیتال ECC با طول کلید ۳۲۰ بیت را فراهم می‌نماید که در واقع همان سختی حل مسئله لگاریتم گسسته ابر خم بیضوی^۴ (HCDLP) است. در این پروتکل طراحی شده ما از یک الگوریتم متقارن AES برای افزایش امنیت و ایجاد محرمانگی در این پروتکل استفاده کرده‌ایم. این پروتکل با استفاده از تکنیک مهر زمانی و عملیات بروزرسانی در برابر حملات تکرار، منع سرویس مقاوم شده است. ابتدا توضیحاتی درباره نحوه عملکرد پروتکل را بیان و در ادامه پروتکل طراحی شده را مورد تحلیل و ارزیابی امنیتی و محاسباتی و نرم افزاری قرار می‌دهیم. در شکل (۳) جزئیات مربوط به این پروتکل آورده شده است.



شکل ۳. پروتکل طراحی شده

¹ Shen

² Hyper

³ Barsgade

⁴ Hyper elliptic Curve Discrete Logarithm Problem

• توضیحات پارامترهای استفاده شده در پروتکل:

- ۱- $Tag_i = SHA-3(Tag_i || ID_i)$: شناسه و اطلاعات اختصاصی مربوط به برچسب i ام است.
- ۲- PR_T : کلید خصوصی برچسب که یک عدد تصادفی $q < PR_T$ است. (p یک عدد تصادفی بزرگ و $q = p-1$)
- ۳- PU_R : کلید عمومی سرور $PU_R = PR_R * D$
- ۴- PU_G : پارامتر عمومی امضاء دیجیتال شامل: D, p, q : تقسیم کننده کاهش یافته یکتا بر روی ابر خم بیضوی C که توسط الگوریتم [۳۳] Harley's محاسبه می‌شود.
- ۵- K : کلید متقارن الگوریتم AES
- ۶- m : متغیر با مقدار اولیه صفر
- ۷- T' : یک شمارنده زمانی
- ۸- PR_R : کلید خصوصی سرور که یک عدد تصادفی $q < PR_R$ است. (p یک عدد تصادفی بزرگ و $q = p-1$)
- ۹- PU_T : کلید عمومی برچسب $PU_T = PR_T * D$
- ۱۰- یک جدول که مقادیر عدد تصادفی C_i و مهر زمانی ($timestamp_i$) متناظر با آن در آن ذخیره شده‌اند.
- ۱۱- یک جدول که تمام مقادیر ID_i و مقادیر T_i متناظر با آن برای شناسایی برچسب در آن ذخیره شده است.

• تشریح پروتکل

در این پروتکل سرویس دهنده مرکزی و برچسب‌خوان را به عنوان سرور در نظر گرفته‌ایم. عملکرد پروتکل طراحی شده به این صورت است که:

- ۱- شروع کار این پروتکل با انتخاب یک عدد تصادفی C_i از جدول اعداد تصادفی به صورت تصادفی در سمت سرور شروع می‌شود. این عدد تصادفی یک مهر زمانی t_i متناظر با خود را در جدول به همراه دارد. سرور یک عدد تصادفی C_i را به همراه t_i برای برچسب مورد نظر که در فضای کارت‌خوان این سرور قرار می‌گیرد، ارسال می‌نماید که این مقدار را S_i در نظر می‌گیریم.
- ۲- برچسب با دریافت پیام S_i ابتدا مقدار C_i را از آن جدا می‌کند و سپس C_i با مقدار m که مقدار اولیه آن m صفر است، مقایسه می‌کند که آیا C_i با m برابر است یا نه، اگر برابر نبود و همچنین بررسی امضاء $Sign_{HECDSA}(S_i)$ درست باشد یعنی پیام S_i دستکاری نشده باشد، برچسب ادامه مراحل را انجام می‌دهد و اگر این دو شرط همزمان برقرار نباشد برچسب مراحل احراز اصالت را ادامه نمی‌دهد. با این کار می‌توان از حمله منع سرویس و تکرار جلوگیری کرد.
- ۳- اگر این دو شرط برقرار باشد، برچسب عدد تصادفی C_T را تولید و سپس مقدار امضاء $Sign_{HECDSA}(S_i || C_T || ID_i)$ را محاسبه می‌نماید و در ادامه پیام امضاء شده را برای بررسی امضاء در سمت سرور با الگوریتم $AES_K(S_i || C_T || ID_i)$ رمز می‌کند. در آخر مقدار $m = C_i$ قرار می‌دهد و پیام‌های $AES_K(S_i || C_T || ID_i), S_i, Sign_{HECDSA}(S_i || C_T || ID_i)$ را برای سرور ارسال می‌نماید.

۴- سرور با دریافت این پیام‌ها ابتدا مقدار S_i را بررسی می‌کند که آیا مقدار آن معتبر است یا خیر. این کار با بررسی جدول اعداد تصادفی صورت می‌گیرد و اگر این مقدار تصادفی C_i در جدول وجود داشته باشد، سپس مقدار مهر زمانی t_i را بررسی می‌کند. این بررسی به این صورت است که تفاوت زمان مرجع زمانی T را از t_i بدست می‌آوریم $\Delta T = T - t_i$ ، اگر این مقدار ΔT از حد مجاز تعیین شده V (به طور مثال ۱۰ ثانیه) بیشتر نباشد سرور ادامه مراحل احراز اصالت را انجام می‌دهد در غیر این صورت اگر مقدار C_i معتبر نباشد و یا $\Delta T > V$ نباشد سرور برچسب را غیرمجاز تشخیص می‌دهد.

۵- در صورت برقراری این شروط، سرور ابتدا پیام رمز شده ارسالی از طرف برچسب را رمزگشایی و سپس به وسیله این پیام، امضاء دیجیتال را مورد بررسی قرار می‌دهد. سرور ابتدا به وسیله امضاء دیجیتال درستی مقدار S_i دریافتی را با مقدار امضاء شده آن بررسی می‌کند تا از تمامیت این پیام اطمینان حاصل نماید. در صورتی که این پیام دستکاری شده باشد سرور از حمله صورت گرفته مطلع می‌گردد و روند اجرای پروتکل را متوقف می‌نماید.

۶- در ادامه در صورت نبود حمله، سرور بوسیله ID_i دریافتی در جدول شناسه‌های برچسب جستجو می‌کند، در صورت وجود این مقدار در این جدول، سرور برچسب i ام را احراز اصالت می‌نماید. سپس سرور مقدار t_i و C_i را بصورت $C_i \oplus C_T = C_i$ بروزرسانی می‌کند. در نهایت سرور مقدار S_i ، $AES_K(ID_i \oplus C_T \parallel S_i)$ را برای برچسب ارسال می‌نماید.

۷- برچسب با دریافت پیام S_i ، $AES_K(ID_i \oplus C_T \parallel S_i)$ ابتدا بررسی می‌کند که آیا این رابطه قرار است یا نه $\Delta T' = T' - t_i$ (که T' یک شمارنده است و با دریافت t_i شروع به شمارش می‌کند، V یک مقدار زمانی مشخص برای جلوگیری از حمله تکرار و منع سرویس می‌باشد) در این رابطه اگر زمان دریافت پیام ارسال شده از طرف سرور از حد V بیشتر شود برچسب عملیات احراز اصالت را متوقف می‌نماید. شرط دیگر این است که آیا مقدار $C_i = m$ با هم برابر است یا نه. اگر این دو شرط به طور همزمان برقرار باشد برچسب پیام رمز شده دریافتی را رمزگشایی و سپس برچسب یکپارچگی و تمامیت S_i و درستی مقدار $ID_i \oplus C_T$ را با مقدار محاسبه شده در سمت خود را بررسی می‌نماید. در صورت درستی، برچسب سرور را احراز اصالت و در نهایت مقدار $m = 0$ قرار می‌دهد.

۴. تحلیل امنیتی پروتکل طراحی شده

در طراحی هر پروتکل باید یک سری از ویژگی‌ها و تحلیل‌های امنیتی را در نظر گرفت. در این قسمت پروتکل طراحی شده را در مقابل حملات مطرح شده در [۱۲] مورد تحلیل و بررسی قرار داده و نشان داده شده است که پروتکل در مقابل تمام حملات ذکر شده مخصوصاً حمله منع سرویس مقاوم است.

• احراز اصالت دو طرفه^۱

در این پروتکل سرور ابتدا پیامی را برای شروع فرایند احراز اصالت به برچسب ارسال می‌کند و در صورت برقراری شروط موجود برچسب، پاسخ این پیام را برای سرور ارسال می‌نماید، سرور هم با بررسی پیام دریافتی، برچسب را احراز اصالت می‌نماید و سپس پیامی را به برچسب ارسال و برچسب هم با بررسی این پیام دریافتی سرور را احراز اصالت می‌نماید و احراز اصالت به صورت دو طرفه صورت می‌گیرد.

• محرمانگی^۲

در این پروتکل اطلاعات مخفی که در کانال ناامن مبادله می‌شود فقط مقدار ID_i است که این مقدار توسط الگوریتم AES رمز شده و بعد بر روی کانال ارسال می‌گردد. مقدار تصادفی C_T هم که توسط برچسب تولید و در مراحل احراز اصالت مورد استفاده قرار می‌گیرد نیز، توسط الگوریتم AES رمز شده و مهاجم به آن دسترسی ندارد و همچنین C_T بعد از هر نشست بروزرسانی می‌گردد. بنابراین محرمانگی اطلاعات در این پروتکل حفظ می‌گردد.

• گمنامی^۳

در این پروتکل تنها پیامی که اطلاعات برچسب در آن قرار دارد مقدار ID_i است که این مقدار توسط الگوریتم رمزنگاری AES رمز می‌شود و مهاجم نمی‌تواند به آن دست پیدا کند. بر فرض دستیابی مهاجم به مقدار ID_i این مقدار چکیده شده شناسه برچسب است و توسط الگوریتم SHA-3 شناسه برچسب Tag_i چکیده شده و مهاجم با داشتن مقدار ID_i هم نمی‌تواند از هویت برچسب اطلاعی پیدا نماید، بنابراین این پروتکل دارای ویژگی گمنامی است.

• حمله جعل برچسب^۴

در این پروتکل پیام‌های رد و بدل شده بین برچسب و سرور توسط الگوریتم AES رمز شده و همچنین توسط الگوریتم امضاء دیجیتال HECDsa امضاء شده است بنابراین مهاجم بدون داشتن کلید مخفی AES و کلید خصوصی امضاء دیجیتال نمی‌تواند هیچ پیام مجازی را از طرف برچسب به سرور ارسال و خود را به عنوان یک برچسب مجاز جعل نماید.

• حمله جعل سرور^۵

در این پروتکل برچسب به وسیله آخرین پیام دریافتی از سرور و بررسی آن، سرور را احراز اصالت می‌نماید. این پیام شامل این مقدار $AES_K(ID_i \oplus C_T \parallel S_i \parallel K_{new} \parallel PU_R)$ است و مهاجم با ندانستن مقدار $ID_i \oplus C_T$ و هم چنین کلید رمزنگاری K نمی‌تواند این پیام را جعل نماید و به برچسب ارسال و برچسب را فریب دهد. به دلیل وجود مهر زمانی در

¹ Mutual authentication

² Confidentiality

³ Anonymity

⁴ Tag impersonation attack

⁵ Server spoofing attack

مقدار S_i این اطمینان بدست می‌آید که مهاجم زمان کافی برای کشف کلید مخفی K را نداشته باشد. بنابراین این پروتکل در برابر حمله جعل سرور مقاوم است.

• حمله تکرار^۱

مهاجم می‌تواند پیام‌های ارسالی از سمت برچسب به سرور $Sign_{HECDSA}(S_i \parallel C_T \parallel ID_i)$ را ذخیره و در زمان دیگری برای سرور ارسال نماید، اما سرور ابتدا مقدار S_i را بررسی و توسط مقدار مهرزمانی t_i ، تکراری بودن پیام را تشخیص می‌دهد. این کار به این صورت است که سرور ابتدا مقدار زمان ارسال و دریافت S_i را بوسیله معادله $\Delta T = T - t_i$ بدست می‌آورد و اگر این زمان از مقدار V (مثلاً ۳۰ ثانیه) بیشتر باشد و هم چنین اگر مقدار C_i دریافتی که بعد از هر نشست بروز می‌شود با مقدار موجود در جدول سرور برابر نباشد، سرور از تکراری بودن پیام دریافتی اطلاع پیدا می‌نماید و در نهایت از قبول این پیام خودداری و متوجه حمله تکرار می‌گردد. از طرف دیگر مهاجم می‌تواند پیام ارسالی از سمت سرور به سمت برچسب S_i ، $AES_K(ID_i \oplus C_T \parallel S_i \parallel K_{new} \parallel PU_R)$ را ذخیره و در زمان دیگری برای برچسب ارسال نماید، اما برچسب ابتدا با بررسی مقدار S_i ، تکراری بودن پیام را تشخیص می‌دهد. برچسب با بررسی معادله $\Delta T' = T' - t_i$ ، زمان دریافت پیام اول و دوم از سرور را مشخص و اگر مقدار $\Delta T'$ از مقدار V (مثلاً ۱ دقیقه) بیشتر شد، برچسب پیام دریافتی را نامعتبر تشخیص می‌دهد. همچنین برچسب مقدار m را که بعد از هر نشست موفقیت آمیز مقدارش صفر می‌گردد را بررسی، اگر معادله $m = C_i$ برقرار نباشد، برچسب پیام دریافتی را نامعتبر تشخیص می‌دهد. بنابراین این پروتکل در برابر حمله تکرار مقاوم است.

• حمله فرد در میان^۲

در این پروتکل به دلیل استفاده از عملیات رمزنگاری و احراز اصالت توسط الگوریتم امضاء دیجیتال و همچنین استفاده از مهرزمانی در هر دو سمت برچسب و سرور امکان انجام این حمله وجود ندارد زیرا طبق مکانیزمی که در قسمت‌های قبلی بیان شد به راحتی تاخیر و یا جعلی بودن در دریافت پیام در هر دو سمت برچسب و سرور قابل تشخیص است و مهاجم نمی‌تواند این حمله را بر روی این پروتکل انجام دهد.

• حمله ناهمزمانی^۳

در این پروتکل عملیات بروزرسانی در سمت سرور فقط بر روی مقادیر اعداد تصادفی C_i و مهرزمانی t_i انجام می‌گردد و اگر مهاجم از ارسال پیام دوم از سمت سرور به سمت برچسب خودداری نماید، برچسب به وسیله شمارنده T' و سرور توسط شمارنده T از دیر رسیدن پیام‌ها مطلع می‌شوند و عملیات احراز اصالت متوقف می‌گردد. در نتیجه مهاجم نمی‌تواند حمله ناهمزمانی را به دو سمت برچسب و سرور اعمال نماید.

¹ replay attack

² man-in-the-middle attack

³ desynchronization attack

• حمله ردیابی^۱

در این پروتکل همان طور که در قسمت‌های قبل بیان شد مهاجم نمی‌تواند اطلاعات محرمانه برچسب را بدست آورد زیرا تمام پیام‌های مبادله شده رمزنگاری می‌گردند، و پیام‌های $\text{Sign}_{\text{HECDSA}}(S_i \parallel C_T \parallel \text{ID}_i)$ در هر نشست دارای مقادیر تصادفی تازه هستند و همین مقادیر تصادفی رمز و خروجی متفاوتی را در هر نشست ایجاد می‌کنند. پس مهاجم با داشتن این پیام‌ها از چندین نشست و از برچسب‌های متفاوت، نمی‌تواند تشخیص دهد که پیام دریافتی مورد نظر مربوط به کدام برچسب است. بنابراین این پروتکل در برابر حمله ردیابی مقاوم است.

• حمله منع سرویس^۲

در این پروتکل به دلیل استفاده از مکانیزم مهرزمانی t_i و یک عدد تصادفی شاخص C_i در هر دو سمت برچسب و سرور از این حمله جلوگیری شده است. در این پروتکل با قرار یک شرط (if) در سمت برچسب و سرور در هنگام دریافت پیام‌ها، ابتدا مقدار مهرزمانی بررسی می‌گردد که زمان ارسال و دریافت پیام چه مدت بوده است و اگر از حد مجاز تعیین شده بیشتر باشد پیام را نامعتبر تشخیص می‌دهد. همچنین به وسیله عدد تصادفی C_i از تکرار همزمان پیام به هر دو سمت برچسب و سرور جلوگیری می‌گردد. در سمت سرور این کار به وسیله بروزسانی مقدار C_i بعد از هر بار دریافت پیام از سمت برچسب انجام می‌گردد و اگر مهاجم برای دومین بار پیام برچسب را به سرور ارسال نماید مقدار C_i با مقدار موجود در پایگاه داده سرور برابر نخواهد بود و سرور پیام دریافتی را قبول نمی‌نماید. در سمت برچسب یک متغیر m را در نظر می‌گیریم که مقدار اولیه آن صفر است و با دریافت اولین پیام از سمت سرور مقدار C_i دریافتی را با m مقایسه می‌نماید اگر با هم برابر نباشند برچسب محاسبات مشخص خود را انجام می‌دهد و سپس مقدار m را برابر مقدار C_i قرار داده و پیام مشخص شده در روند پروتکل را، برای سرور ارسال می‌نماید و مهاجم نمی‌تواند پیام اول ارسال شده از سمت سرور را چندین بار برای برچسب ارسال نماید زیرا مقدار C_i با مقدار m است و در شرط (if) ابتدای دریافت پیام مقدار C_i با مقدار m برابر است و شرط نابرابر آن معتبر نخواهد بود و برچسب پیام دریافتی را قبول نمی‌کند. در زمان دریافت پیام دوم از طرف سرور، برچسب ابتدا بررسی می‌نماید که آیا مقدار C_i دریافتی برابر مقدار m است یا نه، اگر برابر بود پیام دریافتی را مورد بررسی قرار می‌دهد و در انتها مقدار m را برابر صفر قرار می‌دهد. حال مهاجم نمی‌تواند برای دومین بار پیام معتبر سرور را به برچسب ارسال نماید زیرا مقدار m برابر صفر شده است و دیگر با مقدار C_i برابر نخواهد بود تا شرط دریافت پیام دوم از سمت سرور محقق گردد. بنابراین این پروتکل در برابر حمله منع سرویس در هر دو سمت برچسب و سرور مقاوم است.

¹ tracking attack

² Denial of Service attack

۵. مقایسه امنیتی پروتکل طراحی شده با چهار پروتکل بررسی شده

از بین پروتکل‌های بررسی شده، چهار پروتکل گودور و امیر [۲۱]، موسوی [۲۷]، جین [۳۰]، شان [۳۱] را مورد تحلیل و بررسی قرار داده‌ایم. دلیل انتخاب این چهار پروتکل این است که از یک الگوریتم رمزنگاری استفاده کرده‌اند و در بین پروتکل‌های چند سال گذشته دارای امنیت بالاتری نسبت به دیگر پروتکل‌ها بوده‌اند. به صورت مختصر تحلیل این چهار پروتکل به این صورت است که در پروتکل گودور و امیر [۲۱] از یک سیستم رمزنگاری الجمال و الگوریتم امضاء دیجیتال ECDSA استفاده شده است و به دلیل اینکه برچسب نمی‌تواند تکراری بودن پیام‌های دریافتی را تشخیص دهد، این پروتکل در برابر حمله منع سرویس در سمت برچسب مقاوم نیست. پروتکل موسوی [۲۷] از دو مرحله احراز اصالت و بررسی بر اساس مسئله لگاریتم گسسته خم در سمت سرور و برچسب تشکیل شده است و این پروتکل در برابر حملاتی همچون تکرار، حمله ردیابی برچسب، حمله استراق سمع، حمله جعل برچسب‌خوان و جعل برچسب امن است و در برابر حمله منع سرویس در دو سمت سرور و برچسب ناامن است. پروتکل جین [۳۰] هم دارای ویژگی‌های احراز اصالت دوطرفه، دسترس پذیری و امنیت پیشرو است و در برابر حملاتی همچون حمله تکرار، ردیابی، حمله جعل برچسب‌خوان و جعل برچسب مقاوم است و در برابر حمله منع سرویس در هر دو سمت برچسب و سرور ناامن است. در مورد پروتکل شان [۳۱] هم با وجود بهبودهای صورت گرفته در برابر حمله تکرار و جعل برچسب مقاوم، اما هم‌چنان این پروتکل در برابر حمله منع سرویس در سمت برچسب و سرور مقاوم نیست. ایده حمله منع سرویس به این صورت است که حمله کننده پیامی را که شامل گواهی تصدیق است، را ذخیره، و این پیام را به صورت انبوه به سمت سرور و برچسب ارسال می‌کند. به دلیل اینکه پروتکل‌های احراز اصالت RFID دارای ضعف ساختاری است و از تکنیک رمزنگاری و مهرزمانی^۱ و مقدار اولیه^۲ استفاده نمی‌کنند، برچسب و سرور توانایی تشخیص تکراری بودن پیام دریافتی را ندارند، بنابراین سرور و برچسب به اجبار زمان و منابع محاسباتی خود را از دست می‌دهد و کاربر معتبر از هرگونه سرویسی بی بهره می‌ماند. نتیجه تحلیل‌های امنیتی صورت گرفته بر روی پروتکل پیشنهادی و مقایسه آن با چهار پروتکل تحلیل شده در جدول (۱) آمده است.

جدول ۱. مقایسه امنیتی پروتکل‌های احراز اصالت.

پروتکل‌ها	گودور و امیر [۲۱]	شان [۳۰]	جین [۲۹]	موسوی [۲۷]	پروتکل طراحی شده
حملات بررسی شده					
حمله جعل برچسب	✓	✓	✓	✓	✓
جعل سرور	×	×	✓	✓	✓
تکرار	×	✓	✓	✓	✓
مرد در میان	✓	✓	✓	✓	✓
حمله ناهمزمانی	✓	✓	✓	✓	✓
ردیابی	✓	✓	✓	✓	✓
حمله منع سرویس در سمت سرور	✓	×	×	×	✓
حمله منع سرویس در سمت برچسب	×	×	×	×	✓

^۱ timestamps

^۲ nonce

۶. پیچیدگی محاسباتی پروتکل پیشنهادی

با بررسی محاسباتی این چهار پروتکل، نشان داده‌ایم که استفاده ترکیبی از یک امضاء دیجیتال و یک الگوریتم رمزنگاری متقارن در پروتکل طراحی شده حجم محاسباتی زیادی را برای سیستم فراهم نمی‌نماید و پروتکل طراحی شده بر روی سیستم‌های RFID قابل پیاده‌سازی است. محاسبات سمت برچسب و سرور در این پنج پروتکل در جدول (۲) به نمایش درآمده است و مقادیر T_{mul} , T_{inv} , T_{AES} , T_{ecm} , T_{eca} , $T_{mul,ecd}$, T_h به ترتیب به معنای پیچیدگی زمانی مورد نیاز برای اجرای ضرب پیمانه‌ای در میدان متناهی، پیچیدگی زمانی اجرای عمل جمع روی خم بیضوی، پیچیدگی زمانی اجرای عمل ضرب روی خم بیضوی، پیچیدگی زمانی اجرای عمل معکوس پیمانه‌ای، پیچیدگی زمانی محاسبه الگوریتم رمزنگاری AES، پیچیدگی زمانی اجرای ضرب پیمانه‌ای در میدان متناهی ابر خم بیضوی و زمان محاسبه تابع چکیده ساز است که مقادیر آن به ترتیب ۱۲۰۰ و ۵ و ۳ و ۱۰۰ و ۶ و ۰/۳۶ برابر T_{mul} هستند [۱۲].

جدول ۲. مقایسه پیچیدگی محاسباتی پروتکل پیشنهادی.

محاسبات سرور	محاسبات برچسب	پروتکل
$2T_{mul}+5T_{ecm} \approx 6002T_{mul}$	$2T_{mul}+5T_{ecm} \approx 6002T_{mul}$	گودور و امیر [۲۱]
$2T_{mul}+1T_{inv}+6T_{ecm}+2T_{eca}+1T_h \approx 7215T_{mul}$	$2T_{mul}+1T_{inv}+6T_{ecm}+3T_{eca}+1T_h \approx 7220T_{mul}$	موسوی [۲۷]
$2T_{mul}+5T_{ecm}+2T_{eca}+2T_h \approx 6012T_{mul}$	$2T_{mul}+5T_{ecm}+2T_{eca}+2T_h \approx 6012T_{mul}$	جین [۳۰]
$2T_{mul}+2T_{inv}+5T_{ecm}+2T_{eca}+5T_h \approx 6020T_{mul}$	$2T_{mul}+4T_{ecm}+2T_{eca}+4T_h \approx 4823T_{mul}$	شان [۳۱]
$4T_{mul,HECC}+2T_{inv}+6T_{ecm}+4T_{eca}+2T_{AES}+4T_h \approx 7428T_{mul}$	$4T_{mul,HECC}+2T_{inv}+6T_{ecm}+4T_{eca}+2T_{AES}+4T_h \approx 7428T_{mul}$	پروتکل طراحی شده

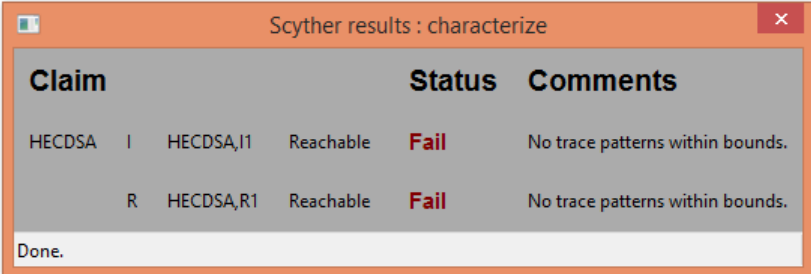
۷. تحلیل خودکار نرم افزاری پروتکل پیشنهادی

تحلیل و آنالیز پروتکل‌های امنیتی توسط انسان کاری دشوار است و در بسیاری از پروتکل‌های امنیتی پس از انتشار، نقض‌هایی مشاهده شده است، به همین جهت تحلیل پروتکل‌های امنیتی توسط ابزارهای تحلیل خودکار نظیر اسکایتر^۱ و پروویو^۲ می‌تواند راه حل مناسبی در جهت شناسایی مشکلات و حفره‌های امنیتی این پروتکل‌ها باشد. مزیت نرم افزار اسکایتر نسبت به نرم افزار پروویو، عدم نیاز به تعریف یک سناریو برای ورودی نرم افزار اسکایتر است که خود نرم افزار تمام حالت‌های مختلف موجود برای حمله را بر روی پروتکل به اجرا در می‌آورد، زیرا ذهن انسان توانایی در نظر گرفتن تمام سناریوهای حمله را ندارد. بنابراین طبق مطالب بیان شده ما از نرم افزار تحلیل خودکار پروتکل Scyther v1.1

^۱ Scyther

^۲ ProVerif

Compromise-0.9.2 [۳۵] برای تحلیل پروتکل طراحی شده استفاده کرده‌ایم. در این قسمت خروجی تحلیل نرم افزار اسکایتر بر روی پروتکل طراحی شده را مورد تحلیل و بررسی قرار داده می‌دهیم. نتیجه این نرم افزار به صورت شکل (۴) و (۵) نشان داده شده است که به صورت کلی نشان می‌دهد که این پروتکل دارای هیچ نقطه ضعف امنیتی نیست. توضیحات کامل مربوط به این ویژگی‌های امنیتی این نرم افزار در [۳۴] تشریح شده است. در خروجی نرم افزار اسکایتر اگر حمله‌های به پروتکل انجام شده باشد، یک صفحه گرافیکی را به صورت گراف نحوه انجام این حمله را نشان می‌دهد. همان طور که در شکل (۴) مشاهده می‌شود در تحلیلی که بر روی مشخصه‌های پروتکل انجام شده است، نشان می‌دهد که هیچ گونه الگویی برای ردیابی مشخصه‌های برچسب توسط پارامترهای موجود در نقش‌های I , R پیدا نشده است، و مهاجم نتوانسته است از روی مراحل مختلف اجرای پروتکل برچسب را ردیابی نماید. در نتیجه پروتکل طراحی شده در برابر حمله ردیابی در تمام مراحل اجرای پروتکل، مقاوم است.



Claim	Status	Comments
HECDSA, I	Fail	No trace patterns within bounds.
HECDSA, R	Fail	No trace patterns within bounds.

Done.

شکل ۴. نتیجه بررسی نرم افزار اسکایتر بر روی پارامترهای پروتکل پیشنهادی

در شکل (۵) نتیجه بررسی نرم افزار اسکایتر بر روی ویژگی‌های امنیتی پروتکل طراحی شده مانند محرمانگی، احراز اصالت و وجود یا عدم وجود حملاتی همچون جعل برچسب، تکرار، ناهمزمانی و مرد درمیان را نشان داده می‌شود. توضیح تحلیل امنیتی شکل (۵) بصورت زیر است:

۱. **Secret x** : اولین ادعای امنیتی است که در نرم افزار اسکایتر مورد بررسی قرار می‌گیرد ویژگی $Secret\ x$ است که مقدار x ، شامل پارامترهای مهم موجود در پروتکل است. دارا بودن این ویژگی به این معنا است که پارامترهای این پروتکل (C_0, C_1, ID_1) توسط مهاجم شنود و یا دستکاری نشده است و در واقع این پارامترها هم دارای محرمانگی و هم دارای جامعیت است. در شکل (۵) نشان داده شده است که این پروتکل در هر دو نقش I , R دارای این ویژگی امنیتی محرمانگی و جامعیت پیام‌های تبادل شده است.

۲. **Alive** : یک نوع احراز اصالت است که هدفش ایجاد یک شریک ارتباطی است و در واقع ادعا می‌کند که طرف مقابل احراز اصالت به نوعی زنده^۲ است. در واقع این ویژگی وجود احراز اصالت دوطرفه را اثبات می‌نماید و بیان می‌کند دو طرف در هر نقش I, R یکدیگر را شناخته‌اند و طبق شکل (۵)، این ویژگی در پروتکل طراحی شده موجود است. همچنین وجود این ویژگی نشان می‌دهد که مهاجم نتوانسته است حمله مرد درمیان را بر روی پروتکل اجرا نماید.

^۱ roles
^۲ Alive

Scyther results : verify

Claim	Status	Comments
HECCDSA I HECCDSA,I1 Secret C0	Ok	No attacks within bounds.
HECCDSA,I2 Secret C1	Ok	No attacks within bounds.
HECCDSA,I4 Alive	Ok	No attacks within bounds.
HECCDSA,I5 Weakagree	Ok	No attacks within bounds.
HECCDSA,I6 Commit R,C1,C0	Ok	No attacks within bounds.
HECCDSA,I7 Niagree	Ok	No attacks within bounds.
HECCDSA,I8 Nisynch	Ok	No attacks within bounds.
R HECCDSA,R1 Secret C0	Ok	No attacks within bounds.
HECCDSA,R2 Secret C1	Ok	No attacks within bounds.
HECCDSA,R4 Alive	Ok	No attacks within bounds.
HECCDSA,R5 Weakagree	Ok	No attacks within bounds.
HECCDSA,R6 Commit R,C1,C0	Ok	No attacks within bounds.
HECCDSA,R7 Niagree	Ok	No attacks within bounds.
HECCDSA,R8 Nisynch	Ok	No attacks within bounds.

Done.

شکل ۵. نتیجه بررسی نرم افزار اسکایتر بر روی ویژگی‌های امنیتی پروتکل

۳. **WeakAgree**: این ویژگی به معنای وجود یک احراز اصالت قوی است و به این معنا است که گیرنده یقین کند که تمام پیام‌های دریافتی از طرف فرستنده معتبر و مورد نظر خود ارسال شده است. در واقع این ویژگی نشان می‌دهد که پروتکل طراحی شده دارای یک احراز اصالت کامل است و مهاجم نتوانسته است حمله ناهمزمانی را بر روی این پروتکل اجرا نماید.

۴. **Niagree**^۱: این ویژگی یک نوع احراز اصالت است که ادعا می‌کند که فرستنده و گیرنده پیام بر روی مقدارها و متغیر-های تبادل شده، توافق می‌کنند و هیچ اطلاعاتی به دست مهاجم نیافتاده است تا بتواند پیام‌ها را در زمان دیگری برای برچسب یا سرور ارسال نماید و حمله تکرار را به اجرا در بیاورد. در واقع این ویژگی نشان می‌دهد که مهاجم نتوانسته است حمله تکرار را بر روی پروتکل طراحی شده اجرا نماید.

۵. **Nisynch**^۲: این ویژگی بیان می‌کند که طرفین مطمئن شده‌اند که پیام ارسال شده فقط از طرف فرد معتبر ارسال شده است و تکراری و یا جعلی نبوده و مهاجم نتوانسته است اطلاعات ردوبدل شده را دست کاری و یا اینکه به صورت تکراری برای برچسب یا سرور ارسال نماید. در واقع این ویژگی بیان می‌کند که حمله جعل سرور و جعل برچسب و منع

^۱ Non-injective agreement

^۲ Non-injective synchronization

سرویس توسط مهاجم به پروتکل اعمال نشده است، در نتیجه پروتکل طراحی شده در برابر حمله منع سرویس در هر دو سمت برچسب و سرور مقاوم است.

۸. نتیجه‌گیری

با بررسی نقاط ضعف پروتکل‌های احراز اصالت سامانه RFID به این نتیجه رسیدیم که اکثر پروتکل‌های احراز اصالت RFID دارای نقطه ضعف امنیتی حمله منع سرویس در سمت برچسب و سرور هستند. در این مقاله یک پروتکل مبتنی بر ابرخیم بیضوی را طراحی کرده‌ایم که در آن از یک الگوریتم امضاء دیجیتال HECDSA برای انجام عمل احراز اصالت و الگوریتم رمزنگاری AES برای ایجاد محرمانگی، استفاده شده است. در طراحی این پروتکل از یک مهر زمانی و عملیات روزرسانی به گونه‌ای استفاده شده است که علاوه بر حملات جعل برچسب، جعل سرور، مرد درمیان، تکرار، ردیابی و ناهمزمانی، در برابر حمله منع سرویس در هر دو سمت برچسب و سرور نیز مقاوم است. در ادامه پروتکل طراحی شده را نسبت به چهار پروتکل دیگر مورد تحلیل امنیتی و محاسباتی قرار داده و نشان داده‌ایم که استفاده از الگوریتم رمزنگاری نامتقارن خم بیضوی و الگوریتم رمزنگاری متقارن AES در پروتکل طراحی شده، حجم محاسباتی بالای را به سیستم تحمیل نمی‌کند و میتوان این پروتکل را بر روی سیستم‌های RFID پیاده‌سازی نمود. برای تحلیل نرم افزاری پروتکل پیشنهادی از نرم‌افزار تحلیل خودکار پروتکل اسکایتر استفاده کرده و در ۱۰۰ دور اجرای پروتکل به این نتیجه رسیدیم که پروتکل طراحی شده در برابر حملات مطرح شده امن است و هیچ الگوی حمله ردیابی و هیچ سناریویی برای حمله به این پروتکل گزارش نشده است و در واقع طبق نرم افزار اسکایتر هیچ نقطه ضعف امنیتی در پروتکل طراحی شده وجود ندارد.

مراجع

- [1] R. Weinstein, "RFID: a technical overview and its application to the," IT Professionals, vol. 7, pp. 27-33, 2005.
- [2] Q. S. S. Z. D. Ranasinghe, "Unique Radio Innovation for the 21st Century, Building Scalable and Global RFID Networks," New York, NY, USA, no. Springer, 2010.
- [3] T. Chothia, "A traceability attack against Epassports," in R. Sion (Ed.): FC 2010, LNCS 6052, pp. 20-34, 2010.
- [4] D. Henrici, "RFID Security and privacy: concepts, protocols and," Lecture Notes Electrical Engineering, vol. 17, no. SpringerVerlag Berlin Heidelberg, 2008.
- [5] D. Liu, Y. Yang, J. Wang, "A mutual authentication protocol for RFID using IDEA," Auto-ID Labs White Paper WPHARDWARE-048, 2009.
- [6] T. A. Pham, M. S. Hasan, "An RFID mutual authentication protocol based on AES algorithm," 2012 UKACC International Conference on Control, no. IEEE, p. 997-1002, 2012.

- [7] J. Chou, "An efficient mutual authentication RFID scheme based on elliptic curve cryptography," J. Supercomput, vol. 70, p. 75–94, 2014.
- [8] M. Hutter, M. Feldhofer, and T. Plos, "An ECDSA processor for RFID authentication," in Proc. Radio Freq. Identif. Secur.Privacy Issues, p. 189–202, 2010.
- [9] S. Wang, S.Liu, D.Chen, "Analysis and Construction of Efficient RFID Authentication Protocol with Backward Privacy," China, no. Advances in, pp. 58-466, 2012.
- [10] N. Koblitz, "Elliptic curve cryptosystems," Math. Comput, vol. 48, p. 203–209, 1987.
- [11] J. Wolkerstorfer, "Is elliptic-curve cryptography suitable to secure RFID tags?," in Proc. Workshop RFID Light-Weight Cryptogr, p. 11–20, 2005.
- [12] D.He, Sh.Zeadally, "An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography," IEEE INTERNET OF THINGS JOURNAL, vol. 2, 2015.
- [13] Y. Chen, J. Chou, and C. Lin, "A novel RFID authentication protocol based on elliptic curve cryptosystem," Cryptology ePrint, 2011.
- [14] S. Wang, S. Liu, and D. Chen, "Analysis and construction of efficient RFID authentication protocol with backward privacy," in Advances in Wireless Sensor Networks, no. SpringerVerlag, p. 458–466, 2014.
- [15] L. Batina, S. Seys, and D. Singelee, "Hierarchical ECC-based RFID authentication protocol," in Proc. RFID Secur. Privacy, p. 183–201, 2012.
- [16] P.Tuyls, L.Batina, "RFID-tags for anti-counterfeiting," Pointcheval, D. (ed.) CT-RSA, vol. 3860, no. Springer, Heidelberg, p. 115–131, 2006.
- [17] Y.K.Lee, L.Batina, D.Singelee, B.Preneel, "An-counterfeiting, untraceability and other security challenges for RFID systems: public-key-based protocols and hardware," Towards Hardware-Intrinsic Security, no. Springer, p. 237–257, 2010.
- [18] L. Batina, J.Guajardo, T.Kerins, N.Mentens, P.Tuyls, "Public-key cryptography for RFID-tags," Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops Communications Workshops, no. PerCom Workshops IEEE, p. 217–222, 2007.
- [19] X. Zhang, J. Li, Y. Wu, Q.Zhang, "An ECDLP-based randomized key RFID authentication protocol," International Conference on Network Computing and Information Security (NCIS), vol. 2, p. 146–149, 2011.
- [20] P. Babaheidarian, M. Delavar, and J. Mohajeri, "On the security of an ECC based RFID authentication protocol," Proc. 9th Int.ISC Conf. Inf. Secur. Cryptol. (ISCISC), p. 111–114, 2012.
- [21] G. Godor, S. Imre, "Elliptic Curve Cryptography Based Authentication Protocol for Low-Cost RFID Tags," IEEE International Conference on RFID-Technologies and Applications, 2011.
- [22] Y. Liao, C. Hsiao, "A secure ECC-based RFID authentication scheme using hybrid protocols," in Advances in Intelligent Systems and Applications, no. Berlin, Germany: Springer-Verlag, p. 1–13, 2013.

- [23] Z. Zhao, "A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem," J. Med. Syst, vol. 38, 2014.
- [24] J. Chou, "An efficient mutual authentication RFID scheme based on elliptic curve cryptography," J. Supercomput, no. Springer, p. 75–94, 2014.
- [25] Z. Zhang, Q. Qi, "An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography," J. Med. Syst, vol. 38, 2014.
- [26] M. Farash, "Cryptanalysis and improvement of an efficient mutual authentication RFID scheme based on elliptic curve cryptography," J. Supercomput, 2014.
- [27] S. Rahimi Moosavi*, E. Nigussie, S. Virtanen, J. Isoaho, "An Elliptic Curve-based Mutual Authentication Scheme for RFID Implant Systems," Procedia Computer Science 32, p. 198 – 206, 2014.
- [28] W. Barsgade, M.T. Meshram, "Comparative study of elliptic and hyper-elliptic curve cryptography in discrete logarithmic problem," IOSR J. Math, p. 61–63, 2014.
- [29] Y. Chen, J. Sa Chou, "ECC-based untraceable authentication for large-scale active-tag RFID systems," Springer Science+Business Media New York, 2014.
- [30] C. Jin, C. Xu, X. Zhang, J. Zhao, "A Secure RFID Mutual Authentication Protocol for Healthcare Environments Using Elliptic Curve Cryptography," Springer Science+Business Media New York, 2015.
- [31] H. Shen, J. Shen, M. Khurram Khan, J. Lee, "Efficient RFID Authentication Using Elliptic Curve Cryptography for the Internet of Things," Springer Science+Business Media New York, 2016.
- [32] A. Liza John, M. Thampi, "Mutual Authentication Based on HECC for RFID Implant Systems," Springer Nature Singapore Pte Ltd. P. Mueller et al, p. 18–29, 2016.
- [33] K. Garrett, S. Raghu Talluri, S. Roy, "On vulnerability analysis of several password authentication protocols," Innovations Syst Softw Eng, Springer-Verlag London, p. 167–176, 2015.
- [34] C. Cremers, S. Mauw, "Operational Semantics and Verification of Security Protocols," Information Security and Cryptography, no. Springer-Verlag Berlin Heidelberg, 2012.
- [35] C. Cremers, "The Scyther Tool," Dept. of Computer Science, no. university of OXFORD, 2014.
- [36] R. Patel, B. Borisaniya, A. Patel, D. Patel, "Comparative Analysis of Formal Model Checking Tools for Security Protocol Verification," Network Security and Applications, Springer Berlin Heidelberg, vol. 89, pp. 152–63, 2010.
- [37] B. Ray, M. Howdhury, J. Abawajy, M. Jesmin, "Secure Object Tracking Protocol for Networked RFID Systems," Software Engineering, Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), no. IEEE/ACIS International Conference, 2015.