

# یک پروتکل احراز اصالت جدید در سامانه‌های RFID بر مبنای ماتریس

فاطمه باقرنژاد<sup>۱</sup>، سید فرهاد عقیلی<sup>۲</sup>، منصور باقری<sup>۳</sup>

۱ کارشناسی ارشد الکترونیک سیستم، دانشگاه تربیت دبیر شهید رجایی

۲ دانشجوی دکتری امنیت اطلاعات، دانشگاه اصفهان

۳ استادیار دانشکده برق و کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، nbagheri@srttu.edu

تاریخ دریافت: ۹۲/۸/۱۸ تاریخ پذیرش: ۹۴/۱۲/۵

## چکیده

شناسایی بسامد رادیویی یک نوع فناوری است که در کاربردهای زیادی مورد استفاده قرار می‌گیرد. امنیت و حفظ حریم خصوصی کاربران دو نگرانی بزرگ در فناوری شناسایی بسامد رادیویی است به همین دلیل، پروتکل‌های زیادی پیشنهاد شده‌اند که هر کدام معایب و مزایای خود را دارند. در این مقاله، بعد از معرفی سه پروتکل احراز اصالت سبک وزن مبتنی بر ماتریس، آسیب‌پذیری این پروتکل‌ها نسبت به چند حمله مانند غیرهمزمان سازی، جعل هویت قرائت‌گر و ردیابی برجسب را ذکر خواهیم کرد. در این مقاله سعی خواهیم کرد که آسیب‌پذیری‌های این سه پروتکل را با پیشنهاد یک پروتکل احراز اصالت بهبود یافته برطرف کنیم. در طراحی پروتکل پیشنهادی از دو مولد اعداد شبه تصادفی سبک وزن (AKARI-1 و AKARI-2) با تعداد بیت حداکثر ۱۲۸ bits و ۶۴ bits استفاده شده است بطوریکه پروتکل پیشنهادی همچنان مبتنی بر استاندارد EPC Class-1 Gen-2 است و نیازی به تغییرات آن را برآورده می‌کند.

## کلیدواژه

شناسایی بسامد رادیویی (RFID)، احراز اصالت، ماتریس، غیر همزمان سازی، جعل هویت.

## مقدمه

(PUF<sup>۲</sup>)، مولد اعداد شبه تصادفی (PRNG<sup>۳</sup>)، توابع چکیده ساز و همچنین توابع منطقی مانند XOR استفاده کرد [۵،۴،۳]. در سال ۲۰۰۴ میلادی، استاندارد (EPC Class-1 Gen-2) توسط EPC Global برای برجسب‌های ارزان قیمت ارائه شد که این استاندارد دارای نقاط ضعف بسیاری می‌باشد. در این راستا و در جهت رفع این نقاط ضعف، پروتکل‌های احراز اصالت مختلفی ارائه شده است [۶-۱۴].

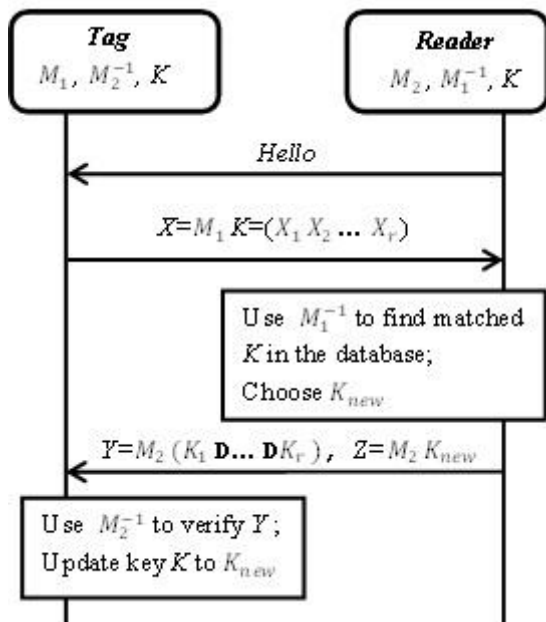
برخی از پروتکل‌های سبک وزن ارائه شده در زمینه‌ی احراز اصالت، بر مبنای ماتریس می‌باشند. در ادامه مقاله، در بخش کارهای انجام شده، قابلیت اجرای برخی حملات به پروتکل‌های ارائه شده توسط کارفیکیان و نسترنکو<sup>۴</sup> (پروتکل KN) [۱۵]، راماجاندرا<sup>۵</sup> و همکاران (پروتکل RRS) [۱۶] و شاوهای و ساجان<sup>۶</sup> (پروتکل SS) [۱۷] نشان داده می‌شود. سپس در بخش پروتکل

سامانه‌های شناسایی بسامد رادیویی<sup>۱</sup> یکی از پرکاربردترین سامانه‌های ارتباطی می‌باشند که در آنها برقراری ارتباط از طریق سیگنال‌های رادیویی انجام می‌شود. این فناوری شامل قرائت‌گر و برجسب می‌باشد، در مواردی که نیاز به ظرفیت گسترده‌ای از محاسبات و اطلاعات باشد از پایگاه داده نیز استفاده می‌شود که معمولاً کانال ارتباطی بین قرائت‌گر و پایگاه داده ایمن در نظر گرفته می‌شود [۱]. این سامانه‌ها به دلیل اینکه اطلاعات را در یک فضای آزاد منتشر می‌کنند، همیشه در معرض خطر حملات امنیتی می‌باشند [۲]. انرژی پردازشی کم و حافظه محدود برجسب‌های ارزان قیمت، باعث ایجاد محدودیت در تعداد و نوع الگوریتم‌های رمزنگاری مورد استفاده در این برجسب‌ها شده است. در عوض می‌توان از رمزگذاری سبک وزن، توابع شبیه ساز فیزیکی

2 Physically Unclonable Function  
3 Pseudo Random Number Generator  
4 Karthikeyan and Nesterenko  
5 Ramachandra  
6 Shaohui and Sujuan

1 Radio Frequency Identification (RFID)

مرحله ۴: برچسب با استفاده از  $M_2^{-1}$ ، درستی  $Y$  را بررسی کرده و قرائت‌گر را احراز اصالت می‌کند. سپس کلید خود را با استفاده از  $M_2^{-1}Z$  به روز می‌کند. در غیر اینصورت، قرائت‌گر را رد می‌کند.



شکل ۱. الگوی عملکرد پروتکل KN [۱۵]

### تحلیل امنیت پروتکل KN

پروتکل KN در برابر حملات غیرهمزمان سازی<sup>۷</sup>، تکرار<sup>۸</sup> [۱۸، ۱۹]، تبانی<sup>۹</sup>، جعل هویت<sup>۱۰</sup> [۱۷] و ردیابی برچسب آسیب‌پذیر می‌باشد.

### حمله‌ی غیر همزمان سازی

اگر مهاجم در مرحله ۳، پیام  $Z$  را به  $Z^*$  تغییر دهد، برچسب، کلید خود را با کلید نامعتبر  $K_{new}^*$  به روز خواهد کرد. با انجام این کار، مهاجم بین قرائت‌گر و برچسب با یک بار اجرای پروتکل و احتمال موفقیت "۱" غیر همزمانی ایجاد می‌کند.

### حمله‌ی تکرار

اگر مهاجم در مرحله ۳،  $Z$  را با  $Z'$  عوض کند (فرض کنید که  $Y'$  و  $Z'$  در اجرای قبلی پروتکل ارسال شده‌اند)، مهاجم می‌تواند  $Y'$  را در دور بعدی پروتکل تکرار کند و با برچسب ارتباط برقرار کند. این حمله با سه بار اجرای پروتکل و با احتمال موفقیت "۱" امکان‌پذیر است.

پیشنهادی، پس از معرفی یک مولد اعداد شبه تصادفی سازگار با استاندارد EPC Class-1 Gen-2 به توصیف پروتکل پیشنهادی می‌پردازیم و تحلیل امنیت پروتکل پیشنهادی را بیان خواهیم کرد و در نهایت نتیجه‌گیری حاصل از مقاله بیان خواهد شد.

### کارهای انجام شده

ما در این بخش، پس از توصیف پروتکل‌های KN [۱۵] و RRS [۱۶] و بیان نقاط ضعف آنها، به توصیف پروتکل SS [۱۷] که به جهت رفع نقاط ضعف پروتکل‌های KN و RRS طراحی شده است، خواهیم پرداخت و در نهایت، عدم امنیت این پروتکل را در برابر برخی از حملات نشان خواهیم داد. نمادهای مورد استفاده در این مقاله در جدول ۱ آورده شده است.

جدول ۱. نمادهای به کار رفته در مقاله

ماتریس $P \times P$	$M_1, M_2$
وارون ماتریس‌های $M_1, M_2$	$M_1^{-1}, M_2^{-1}$
پارامترهای مخفی مشترک بین قرائت‌گر و برچسب	$X, K$
بردارهای تصادفی	$N, N', N''$
نیمه‌ی سمت چپ مقدار تصادفی $N$	$N_L$
نیمه‌ی سمت راست مقدار تصادفی $N$	$N_R$

### توصیف پروتکل KN

الگوی پروتکل KN [۱۵] در شکل ۱ نشان داده شده است. این پروتکل شامل چهار ماتریس  $p \times p$  است که هر برچسب، ماتریس‌های  $M_1$  و  $M_2^{-1}$  و قرائت‌گر، ماتریس‌های  $M_2$  و  $M_1^{-1}$  را ذخیره می‌کند. ماتریس‌های  $M_1^{-1}$  و  $M_2^{-1}$  به ترتیب وارون ماتریس‌های  $M_1$  و  $M_2$  هستند. هر برچسب، بردار کلید  $K = (K_1 K_2 \dots K_r)$  با اندازه‌ی  $q = r.p$  را نیز با قرائت‌گر به اشتراک می‌گذارد، که در آن هر  $K_i$  با طول یکسان  $p$  بوده و همچنین  $r$  اعداد صحیح بزرگتر از یک می‌باشد.  $M_1$  و  $K$  به صورت تصادفی برای هر برچسب انتخاب شده‌اند و ماتریس مخفی در پایگاه داده است.

مراحل پروتکل KN به صورت زیر است:

مرحله ۱: قرائت‌گر  $Hello$  را برای برچسب ارسال می‌کند.  
 مرحله ۲: برچسب با ارسال پیام  $X = M_1 K = (X_1 X_2 \dots X_r)$  به قرائت‌گر پاسخ می‌دهد، که در آن  $X_i = M_1 K_i$  ( $1 \leq i \leq r$ ) می‌باشد.  
 مرحله ۳: قرائت‌گر همه‌ی  $M_1^{-1}$ ‌های موجود در پایگاه داده را برای محاسبه  $K' = M_1^{-1} X$  استفاده می‌کند، اگر  $K'$  با کلید ذخیره شده در پایگاه داده مطابقت داشته باشد، احراز اصالت، موفق است. سپس قرائت‌گر کلید جدید تصادفی  $K_{new}$  را انتخاب کرده و پیام‌های  $Y = M_2 (K_1 \oplus \dots \oplus K_r)$  و  $Z = M_2 K_{new}$  را برای برچسب ارسال می‌کند. در غیر اینصورت، برچسب را به عنوان برچسب نامعتبر رد می‌کند.

7 De-synchronization Attack  
 8 Replay Attack  
 9 Compromising Attack  
 10 Impersonation Attack

در نتیجه برچسب با احتمال موفقیت "۱" و پیچیدگی دو بار اجرای پروتکل ردیابی خواهد شد.

### توصیف پروتکل RRS

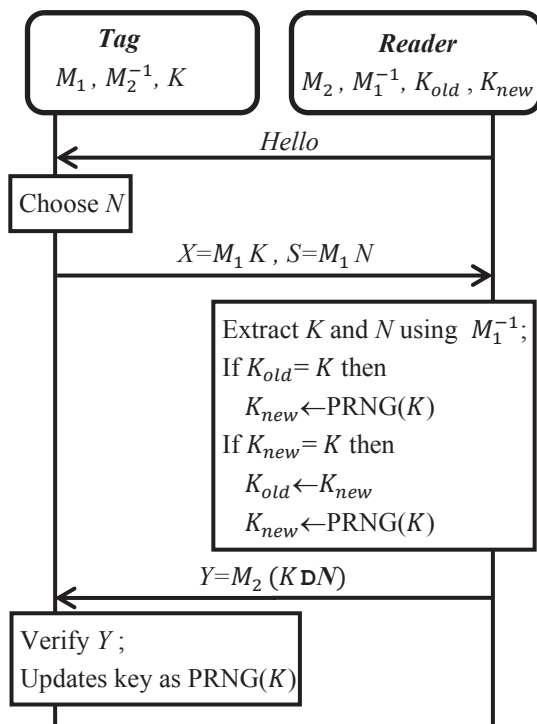
الگوی پروتکل RRS [۱۶] در شکل ۲ نشان داده شده است. در این پروتکل برای جلوگیری از حمله‌ی غیر همزمان‌سازی، قرائت‌گر دو کلید  $K$  و  $K_{new}$  را ذخیره خواهد کرد، که در ابتدا  $K_{new}$  خالی است و  $K$  کلید به اشتراک گذاشته شده بین قرائت‌گر و برچسب می‌باشد. مراحل پروتکل به صورت زیر می‌باشد، در اینجا  $r=1$  است.

مرحله ۱: قرائت‌گر پیام *Hello* را به منظور شروع پروتکل برای برچسب ارسال می‌کند.

مرحله ۲: برچسب بردار  $N$  را به صورت تصادفی انتخاب می‌کند، سپس  $S=M_1N$  و  $X=M_1K$  را برای قرائت‌گر ارسال می‌کند.

مرحله ۳: قرائت‌گر از ماتریس  $M_1^{-1}$  موجود در پایگاه داده برای استخراج مقادیر  $K'=M_1^{-1}X$  و  $N'=M_1^{-1}S$  استفاده می‌کند. اگر بین  $K'$  و هر یک از  $K$  یا  $K_{new}$  تطابقی وجود داشته باشد، به صورت زیر عمل خواهد کرد:

اگر  $K_{new}=K'$  باشد، قرائت‌گر، مقادیر  $K_{new}=K$  و  $K_{new}=PRNG(K)$  را به روز می‌کند و اگر  $K=K'$  باشد، مقدار  $K_{new}=PRNG(K)$  را به روز می‌کند. سپس قرائت‌گر پیام  $Y=M_2(K \otimes N)$  را برای برچسب ارسال می‌کند. در غیر اینصورت، برچسب را به عنوان برچسب نامعتبر رد می‌کند.



شکل ۲. الگوی عملکرد پروتکل RRS [۱۶]

### حمله‌ی تباری

ماتریس‌های  $M_2^{-1}$  و  $M_2$  برای همه برچسب‌ها یکسان هستند. بنابراین اگر مهاجم توانایی تباری با فقط یک برچسب را داشته باشد می‌تواند ماتریس  $M_2^{-1}$  را بدست آورد. چنین حمله‌ای برای برچسب‌های ارزان قیمت با احتمال موفقیت "۱" امکان پذیر است. در این پروتکل، چون کلید مخفی به روز شده از طریق پیام  $Z=M_2K_{new}$  برای برچسب ارسال می‌شود، مهاجم می‌تواند کلید  $K_{new}$  را با استفاده از  $M_2^{-1}$  استخراج کند.

### حمله‌ی جعل هویت

در این حمله مهاجم تعداد  $n$  دور ارتباط را استراق سمع می‌کند و پیام‌های (۱) و (۲) را بدست می‌آورد.

$$\{Z^{(i)}=M_2K^{(i)}\}_{i=1,\dots,n}=\{Z_j^{(i)}\}_{i=1,\dots,n;j=1,\dots,r} \quad (1)$$

$$\{X^{(i)}=M_1K^{(i)}\}_{i=1,\dots,n}=\{X_j^{(i)}\}_{i=1,\dots,n;j=1,\dots,r} \quad (2)$$

سپس بردارهای  $p$  را در مجموعه  $\{Z_j^{(i)}\}$  به صورت مستقل پیدا کرده و برای جعل هویت برچسب، ابتدا پیام ارسالی  $Z=M_2K_{new}=\{Z_j\}_{j=1,\dots,r}$  از قرائت‌گر را استراق سمع کرده و  $\{\alpha_{ij}\}_{i=1,\dots,p;j=1,\dots,r}$  را محاسبه می‌کند. با توجه به اینکه  $\{X_j^{(i)}\}_{i=1,\dots,p}$  ها بردارهای مستقل هستند و در دور بعدی اجرای پروتکل، پیام ارسالی از طرف برچسب،  $X=M_1K_{new}$  می‌باشد، مهاجم می‌تواند پیام را به صورت رابطه (۳) محاسبه کند و برچسب را بدون دانستن مقادیر مخفی جعل کند.

$$M_1K_{new}=M_1(\oplus_{i=1,\dots,p}\alpha_{i,1}K_1^{(i)}||\dots||\oplus_{i=1,\dots,p}\alpha_{i,r}K_r^{(i)}) \\ = \oplus_{i=1,\dots,p}\alpha_{i,1}X_1^{(i)}||\dots||\oplus_{i=1,\dots,p}\alpha_{i,r}X_r^{(i)} \quad (3)$$

مهاجم می‌تواند قرائت‌گر قانونی را در روش یکسانی با احتمال موفقیت "۱" جعل کند.

### حمله‌ی ردیابی برچسب

مراحل انجام این حمله به ترتیب زیر است:

مرحله ۱: مهاجم ارتباط بین قرائت‌گر و برچسب (T1) مشروع را استراق سمع می‌کند.

مرحله ۲: مهاجم، مانع پیام‌های  $Z$  و  $Y$  ارسال شده از طرف قرائت‌گر می‌شود و رشته بیت‌های تصادفی را برای (T1) ارسال می‌کند.

مرحله ۳: در نتیجه (T1)، کلید خود ( $K$ ) را به روز نخواهد کرد.

مرحله ۴: مهاجم، پیام *Hello* را برای برچسب ناشناخته ( $T_x$ ) ارسال می‌کند.

مرحله ۵: ( $T_x$ )، پیام  $X'=M_1K'$  را برای قرائت‌گر که با مهاجم جایگزین شده است ارسال می‌کند.

در صورتیکه ( $T_x$ ) همان (T1) باشد و با توجه به اینکه  $K$  همان مقدار اجرای قبلی پروتکل می‌باشد و  $M_1$  نیز برای هر برچسب ثابت

و منحصر بفرد است، در نتیجه  $X=X'$  خواهد بود.

مرحله ۴: برچسب مقدار  $T=M_2^{-1}Y$  را محاسبه می‌کند و اگر  $T=(K\otimes N)$  باشد، کلید خود را به صورت  $K=PRNG(K)$  به روز می‌کند و در غیر اینصورت، قرائت‌گر را به عنوان قرائت‌گر نامعتبر رد می‌کند.

$$\begin{aligned} U &= \alpha_1 U^{(1)} \oplus \dots \oplus \alpha_p U^{(p)} \\ &= \bigoplus_{i=1, \dots, p} \alpha_i M_1(K^{(i)} \oplus N^{(i)}) \\ &= M_1(\bigoplus_{i=1, \dots, p} \alpha_i (K^{(i)} \oplus N^{(i)})) \end{aligned} \quad (7)$$

$$\begin{aligned} M_2(K\otimes N) &= M_2(\bigoplus_{i=1, \dots, p} \alpha_i (K^{(i)} \oplus N^{(i)})) \\ &= \bigoplus_{i=1, \dots, p} \alpha_i Y^{(i)} \end{aligned} \quad (8)$$

بنابراین مهاجم می‌تواند قرائت‌گر قانونی را با احتمال موفقیت "۱" جعل کند.

### حمله‌ی ردیابی برچسب

مراحل انجام این حمله به ترتیب زیر است:

- مرحله ۱: مهاجم ارتباط بین قرائت‌گر و برچسب (T1) مشروع را استراق سمع می‌کند.
  - مرحله ۲: مهاجم، مانع پیام  $Y$  ارسال شده از طرف قرائت‌گر می‌شود و رشته بیت تصادفی را برای (T1) ارسال می‌کند.
  - مرحله ۳: در نتیجه (T1)، کلید خود ( $K$ ) را به روز نخواهد کرد.
  - مرحله ۴: مهاجم، پیام  $Hello$  را برای برچسب ناشناخته ( $T_x$ ) ارسال می‌کند.
  - مرحله ۵: ( $T_x$ )، پیام  $X'=M_1K'$  و  $S'$  را برای قرائت‌گر که با مهاجم جایگزین شده است ارسال می‌کند.
- در صورتیکه ( $T_x$ ) همان (T1) باشد و با توجه به اینکه  $K$  همان مقدار اجرای قبلی پروتکل می‌باشد و  $M_1$  نیز برای هر برچسب ثابت و منحصر بفرد است، در نتیجه  $X=X'$  خواهد بود. در نتیجه برچسب با احتمال موفقیت "۱" و پیچیدگی دو بار اجرای پروتکل ردیابی خواهد شد.

### توصیف پروتکل SS

اخیراً، شاولهای و ساجان [۱۷] با اصلاح پروتکل‌های احراز اصالت سبک وزن مبتنی بر ماتریس پیشین [۱۵، ۱۶]، پروتکل خود را ارائه کرده‌اند و ادعا کرده‌اند که حتی اگر مهاجم با چند برچسب تبانی کند، پروتکل پیشنهادی آنها می‌تواند ایمن باقی بماند. در ادامه به شرح پروتکل پیشنهادی آنها می‌پردازیم. مراحل انجام پروتکل در شکل ۳ نشان داده شده است.

مرحله ۱: قرائت‌گر بردار تصادفی  $N$  را انتخاب کرده و  $S_1=M_2N$  را برای برچسب ارسال می‌کند.

مرحله ۲: برچسب،  $N$  را استخراج کرده  $S_2=M_2^{-1}(X\otimes N)$  و  $S_3=M_1(K\otimes N)$  را محاسبه می‌کند و برای قرائت‌گر ارسال می‌کند.

این پروتکل در برابر حملات تبانی، جعل هویت [۱۷] و ردیابی برچسب آسیب پذیر می‌باشد.

### تحلیل امنیت پروتکل RRS

در این پروتکل، مهاجم پس از بدست آوردن ماتریس  $M_2^{-1}$  طی مراحل زیر می‌تواند به کلید مخفی دست پیدا کند.

### حمله‌ی تبانی

مرحله ۱: مهاجم به عنوان قرائت‌گر برای برچسب یک درخواست ارسال می‌کند و پاسخ ( $S^{(i)}=M_1N^{(i)}$ )،  $X^{(i)}=M_1K^{(i)}$  را دریافت می‌کند، که اندیس ( $i$ ) به معنای  $i$  امین درخواست است.

مرحله ۲: مهاجم به عنوان برچسب جعلی برای جواب دادن به درخواست ارسال شده از طرف قرائت‌گر مشروع عمل می‌کند و پاسخ‌های  $X^{(i)}$  و  $S^{(i)*}=X^{(i)}\oplus S^{(i)}=M_1(K^{(i)}\oplus N^{(i)})$  را از برچسب دریافت می‌کند. مهاجم می‌تواند از احراز اصالت عبور کند و پیام قرائت‌گر که به صورت  $(Y^{(i)}=M_2(K^{(i)}\oplus K^{(i)}\oplus N^{(i)}))=M_2N^{(i)}$  است را به دست آورد و با استفاده از  $M_2^{-1}$  بردار تصادفی  $N^{(i)}$  را استخراج کند.

مرحله ۳: بعد از اینکه مهاجم به اندازه‌ی کافی بردارهای تصادفی  $N^{(i)}$  و  $S^{(i)}=M_1N^{(i)}$  مربوطه را بدست آورد، می‌تواند بردارهای  $p$  را به صورت مستقل پیدا کند، بنابراین می‌تواند  $M_1$  را بدست آورد و در نهایت، مهاجم می‌تواند با استفاده از معادله  $X^{(i)}=M_1K^{(i)}$  و با احتمال موفقیت "۱" به کلید مخفی دست پیدا کند.

برای جعل هویت قرائت‌گر قانونی، در ابتدا  $n$  دور از پروتکل را استراق سمع کرده و پیام‌های (۴)، (۵) و (۶) را جمع‌آوری می‌کند.

### حمله‌ی جعل هویت

برای جعل هویت قرائت‌گر قانونی، در ابتدا  $n$  دور از پروتکل را استراق سمع کرده و پیام‌های (۴)، (۵) و (۶) را جمع‌آوری می‌کند.

$$\{X^{(i)}=M_1K^{(i)}\}_{i=1, \dots, n} \quad (4)$$

$$\{S^{(i)}=M_1N^{(i)}\}_{i=1, \dots, n} \quad (5)$$

$$Y^{(i)}=M_2\{K^{(i)}\oplus N^{(i)}\}_{i=1, \dots, n} \quad (6)$$

سپس مقادیر  $\{U^{(i)}=X^{(i)}\oplus S^{(i)}=M_1(K^{(i)}\oplus N^{(i)})\}_{i=1, \dots, n}$  را با بردارهای مستقل  $p$  محاسبه می‌کند. مهاجم، بعد از دریافت پیام-های  $U=X\otimes S$  و  $S=M_1N$  و  $X=M_1K$  توسط برچسب،

### تحلیل امنیت پروتکل SS

این پروتکل در برابر حملات ردیابی برچسب<sup>۱۱</sup>، جعل هویت قرائت-گر و غیرهمزمان سازی آسیب پذیر می‌باشد.

#### حمله ردیابی برچسب

مراحل انجام این حمله به ترتیب زیر است:

مرحله ۱: مهاجم ارتباط بین قرائت‌گر و برچسب (T1) مشروع را استراق سمع می‌کند.

مرحله ۲: مهاجم، مانع پیام‌های  $S_4$  و  $S_5$  ارسال شده از طرف قرائت-گر می‌شود و رشته بیت‌های تصادفی را برای (T1) ارسال می‌کند. در نتیجه (T1)،  $K$  و  $X$  خود را به روز خواهد کرد.

مرحله ۳: مهاجم به عنوان قرائت‌گر عمل کرده و پیام  $S_1$  استراق-سمع شده از اجرای قبلی پروتکل را برای برچسب ناشناخته ( $T_x$ ) ارسال می‌کند.

مرحله ۴: (T<sub>x</sub>)، پیام  $S_2$  و  $S'_3 = M_1(K \otimes N)$  را برای قرائت‌گر که با مهاجم جایگزین شده است ارسال می‌کند.

در صورتیکه (T<sub>x</sub>) همان (T1) باشد و با توجه به اینکه  $N$  و  $K$  همان مقادیر اجرای قبلی پروتکل می‌باشند و  $M_1$  نیز برای هر برچسب ثابت و منحصر بفرد است، در نتیجه  $S'_3 = S_3$  خواهد بود.

در نتیجه برچسب با احتمال موفقیت "۱" و پیچیدگی دو بار اجرای پروتکل ردیابی خواهد شد.

#### حمله جعل هویت قرائت‌گر

با توجه به حمله ردیابی برچسب ارائه شده در زیر بخش قبل، مهاجم، پیام  $S_1$  استراق سمع شده از اجرای قبلی پروتکل را برای برچسب (T1) ارسال می‌کند. سپس، برچسب (T1) که  $K$  و  $X$  خود را به روز نکرده است، از پیام دریافتی،  $N$  را استخراج کرده، سپس  $S_2 = M_2^{-1}(X \otimes N)$  و  $S_3 = M_1(K \otimes N)$  را محاسبه می‌کند و برای قرائت‌گر که در واقع مهاجم می‌باشد ارسال می‌کند. در نتیجه مهاجم توانسته است با احتمال موفقیت "۱" و پیچیدگی دو بار اجرای پروتکل، قرائت‌گر مشروع را جعل کرده و توسط برچسب احراز اصالت شود.

#### حمله غیر همزمان سازی

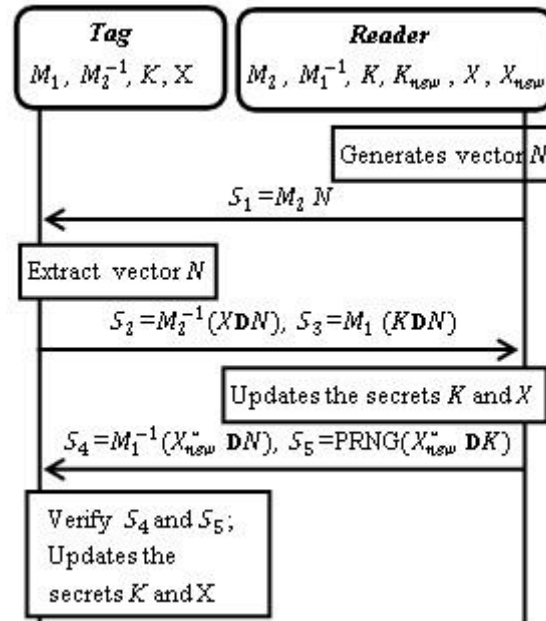
انجام این حمله، به سه بار اجرای پروتکل نیاز دارد:

اجرای اول پروتکل:

مرحله ۱: قرائت‌گر،  $S_1$  را برای برچسب ارسال می‌کند.

مرحله ۲: برچسب،  $S_2$  و  $S_3$  را به عنوان پاسخ برای قرائت‌گر ارسال می‌کند.

مرحله ۳: قرائت‌گر، برچسب را احراز اصالت کرده و  $S_4$  و  $S_5$  را برای برچسب ارسال می‌کند.



شکل ۳. الگوی عملکرد پروتکل SS [۱۷]

مرحله ۳: زمانیکه قرائت‌گر پیام‌های  $S_2$  و  $S_3$  را دریافت کرد،  $X' = M_2 S_2 \otimes N$  را بازیابی کرده و برای یافتن برچسب مورد نظر، پایگاه داده را جستجو می‌کند. در صورتیکه نتواند تطابق بین  $X'$  یا  $X_{new}$  پیدا کند، برچسب را رد کرده و پروتکل را بی-نتیجه باقی می‌گذارد و در غیر اینصورت با استفاده از  $M_1^{-1}$  استخراج شده، بررسی می‌کند که آیا  $K' = M_1^{-1} S_3 \otimes N$  معادل  $K$  یا  $K_{new}$  مربوطه می‌باشد یا خیر. در صورت عدم تساوی، برچسب را رد می‌کند و در غیر اینصورت مقادیر مخفی به اشتراک گذاشته را به روش زیر به روز خواهد کرد.

قرائت‌گر کلید جدید  $X_{new}^*$  را به صورت تصادفی انتخاب می‌کند. اگر  $X_{new} = X'$  باشد مقادیر،  $X = X_{new}$  و  $X_{new} = X_{new}^*$  را به روز می‌کند و اگر  $X = X'$  باشد،  $X_{new} = X_{new}^*$  را به روز می‌کند.

اگر  $K_{new} = K'$  باشد مقادیر،  $K = K_{new}$  و  $K_{new} = PRNG(K_{new})$  را به روز می‌کند. اگر  $K = K'$  باشد،  $K_{new} = PRNG(K)$  را به روز می‌کند. سپس  $S_4 = M_1^{-1}(X_{new}^* \otimes N)$  و  $S_5 = PRNG(X_{new}^* \otimes K)$  را برای برچسب ارسال می‌کند.

مرحله ۴: در صورت تساوی  $S_3$  و  $PRNG((M_1 S_4) \otimes N \otimes K)$  برچسب، کلیدهای مخفی را به صورت  $K = PRNG(K)$  و  $X = (M_1 S_4) \otimes N$  به روز می‌کند و در غیر اینصورت قرائت‌گر را رد می‌کند.

## معرفی یک PRNG سازگار با استاندارد EPC Class-1 Gen-2

مارتین<sup>۱۲</sup> و همکاران در [۲۰] دو مولد اعداد شبه تصادفی سبک وزن (AKARI-1 و AKARI-2) را پیشنهاد کرده‌اند که می‌تواند نیازمندی‌های سامانه‌های مبتنی بر برچسب‌های سبک وزن را برطرف کند و قابلیت اعتماد و امنیت آنها را افزایش دهد. مارتین و همکاران در AKARI-1 از یک تابع فیلتر استفاده می‌کنند که به نسبت تقریباً زیادی تکرار می‌شود ( $r=64$ ) و همچنین در AKARI-2 از دو تابع فیلتر ترکیب شده که باعث کاهش تعداد تکرارها در حلقه می‌شود ( $r=24$ ) استفاده می‌کنند. استفاده از تابع‌های فیلتر، غیر خطی بودن تابع را تضمین خواهد کرد. در هر دو مورد ذکر شده، از نیمه سمت راست توالی حاصل شده ( $m/2$ ) بیت کم ارزش) به عنوان خروجی نهایی استفاده می‌شود. شکل ۴، شبه کد PRNG های سبک وزن پیشنهادی را نشان می‌دهد که نمادهای «» و «» به ترتیب به معنای شیفت به راست و شیفت به چپ هستند.

AKARI-1
Initialize x0 and x1 of m-bits
$x0=x0+(x0*x0)V5$
$x1=x1+(x1*x1)V13$
$z=x0$
for r from 0 to 63
$z=(z>1)+(z<1)+z+x1$
%Output m/2 bits
Lower half of z

AKARI-2
Initialize x0 and x1 of m-bits
$x0=x0+(x0*x0)V5$
$x1=x1+(x1*x1)V13$
$z=x0^x1$
for r from 0 to 24
$z=(z<1)+(z+(0x56AB0A))>1$
$y=x1^z$
for r from 0 to 24
$y=(y>1)+(y<1)+y+(0x72A4FB)$
$z=z^y$
%Output m/2 bits
Lower half of z

شکل ۴. شبه کد AKARI-1 و AKARI-2 [۲۰]

نویسندگان در [۲۰]، پیاده‌سازی‌های مختلفی را برای دو PRNG (AKARI-1 و AKARI-2) ارائه کرده‌اند و هدف معماری‌های پیشنهادی آنها برآورده کردن نیازمندی‌های استاندارد EPC است درحالی‌که امنیت حداکثر را برآورده کند و قادر به استفاده از تعداد بیت بیشتر برای PRNG ها باشد.

اولین معماری در پیاده‌سازی AKARI-1 (AKARI-1A) با هدف کاهش تعداد سیکل‌های ساعت است و معماری پیشنهادی دوم (AKARI-1B) با هدف کاهش مساحت تراشه می‌باشد. اولین معماری در پیاده‌سازی AKARI-2 (AKARI-1A) نیز با هدف کاهش تعداد سیکل‌های ساعت است، معماری دوم (AKARI-2B)

مرحله ۴: مهاجم مانع  $S_4$  و  $S_5$  شده و رشته بیت تصادفی را برای برچسب ارسال می‌کند.

برچسب مقادیر کلید مخفی خود را به روز نخواهد کرد. در نتیجه مقادیر موجود در پایگاه داده مربوط به قرائت‌گر، چندتایی  $(M_2, M_1^{-1}, K, K', X, X')$  و مقادیر ذخیره شده در برچسب، چندتایی  $(M_1, M_2^{-1}, K, X)$  می‌باشند. اجرای دوم پروتکل:

مرحله ۱: قرائت‌گر  $S_1$  را برای برچسب ارسال می‌کند.

مرحله ۲: برچسب،  $S_2$  و  $S_3$  را برای قرائت‌گر می‌فرستد.

مرحله ۳: قرائت‌گر،  $S_4$  و  $S_5$  را برای برچسب می‌فرستد.

مرحله ۴: مهاجم مانع این دو پیام شده و رشته بیت تصادفی را برای برچسب ارسال می‌کند.

برچسب مقادیر کلید مخفی خود را به روز نخواهد کرد. در نتیجه مقادیر موجود در پایگاه داده مربوط به قرائت‌گر، چندتایی  $(M_2, M_1^{-1}, K, K'', X, X'')$  و مقادیر ذخیره شده در برچسب، چندتایی  $(M_1, M_2^{-1}, K, X)$  می‌باشند.

اجرای سوم پروتکل:

در این مرحله، مهاجم به عنوان قرائت‌گر عمل می‌کند.

مرحله ۱: مهاجم پیام  $S_1$  استراق سمع شده از اولین اجرای پروتکل را برای برچسب ارسال می‌کند.

مرحله ۲: برچسب،  $S_2$  و  $S_3$  را برای قرائت‌گر که با مهاجم جایگزین شده است ارسال می‌کند.

مرحله ۳: مهاجم  $S_4$  و  $S_5$  استراق سمع شده از اولین اجرای پروتکل را برای برچسب ارسال می‌کند.

مرحله ۴: برچسب، مهاجم را به عنوان یک قرائت‌گر معتبر احراز اصالت کرده و کلیدهای مخفی خود را به روز می‌کند.

در نتیجه مقادیر موجود در پایگاه داده قرائت‌گر، چندتایی  $(M_2, M_1^{-1}, K, K'', X, X'')$  و مقادیر ذخیره شده در برچسب چندتایی  $(M_1, M_2^{-1}, K', X')$  می‌باشند و از این پس، قرائت‌گر و برچسب قادر به احراز اصالت یکدیگر نیستند. در این حمله احتمال موفقیت برابر با "۱" و پیچیدگی آن، سه بار اجرای پروتکل می‌باشد.

## پروتکل پیشنهادی

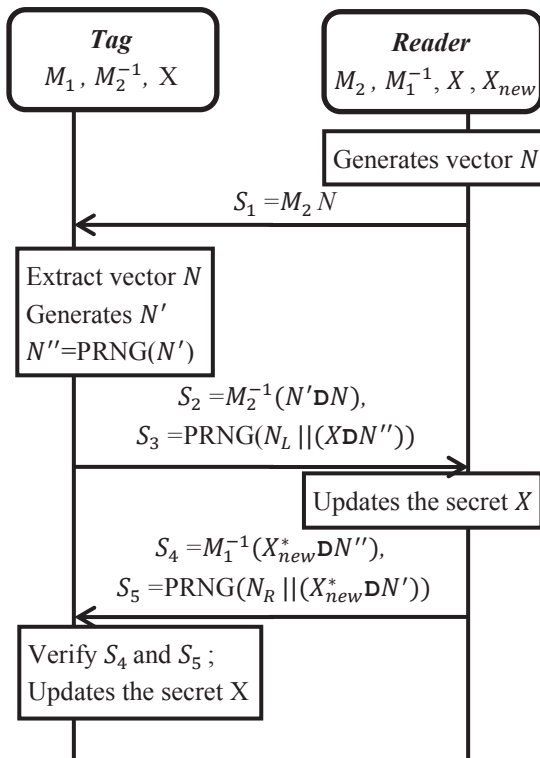
در این بخش در ابتدا به معرفی یک مولد اعداد شبه تصادفی سازگار با استاندارد EPC Class-1 Gen-2 می‌پردازیم و در ادامه پروتکل احراز اصالت جدیدی را که نقاط ضعف و آسیب پذیری‌های پروتکل‌های احراز اصالت قبلی را برطرف کرده است پیشنهاد خواهیم کرد.

با هدف بهبود مساحت اختصاص داده شده و سومین معماری (AKARI-2C) با هدف کاهش بیشتر مساحت با تعداد سیکل‌های ساعت بیشتر می‌باشد.

همانطور که می‌دانیم، تعداد گیت‌های معادل  $1^3$  (GE) اختصاص داده شده به منظور برقراری امنیت در برچسب‌های ارزان قیمت، 4000 GE، فرکانس کاری در حدود 100 KHz و تعداد سیکل‌های ساعت برای تولید یک عدد تصادفی در حدود 500 یا 600 سیکل می‌باشد و مصرف توان نباید از محدوده میکرو وات تجاوز کند.

مرحله 3: قرائت‌گر  $N'$  و  $N''$  را با استفاده از  $S_2$  بازیابی می‌کند. مرحله 4: قرائت‌گر، رابطه  $(PRNG(N_L || (X \oplus N'')))$  را برای همه برچسب‌های موجود در پایگاه داده محاسبه می‌کند، سپس حاصل را با  $S_3$  دریافت شده مقایسه می‌کند و در صورت تساوی، جزئیات برچسب مورد نظر را بازیابی می‌کند. در غیر اینصورت برچسب را به عنوان برچسب نامعتبر رد خواهد کرد.

در [20] نتایج بدست آمده توسط نویسندگان مقاله نشان می‌دهد که با تعداد بیت حداکثر 128 bits برای AKARI-1 و یا تعداد بیت حداکثر 64 bits برای AKARI-2 به حدود 3000-4000 GE نیاز است که می‌تواند نیازمندی‌های EPC را برآورده کند (جدول 2 و 3).



شکل 5. الگوی عملکرد پروتکل پیشنهادی

مرحله 5: قرائت‌گر کلید جدید تصادفی  $X_{new}^*$  را انتخاب می‌کند. اگر دریافتی برابر با  $X_{new}$  باشد مقادیر  $X = X_{new}$  و  $X_{new} = X_{new}^*$  را به روز می‌کند و اگر دریافتی برابر با  $X$  باشد،  $X_{new} = X_{new}^*$  را به روز می‌کند.

جدول 2. AKARI-1 PRNG [20]

سیکل‌های ساعت	توان (nW)	گیت‌های معادل (GE)	128 بیت
66	343	3898	AKARI-1A
450	350	3402	AKARI-1B

جدول 3. AKARI-2 PRNG [20]

سیکل‌های ساعت	توان (nW)	گیت‌های معادل (GE)	64 بیت
51	216	3743	AKARI-2A
290	255	3193	AKARI-2B
530	231	3040	AKARI-2C

### توصیف پروتکل پیشنهادی

با توجه به اینکه در PRNG 16 بیتی، مهاجم می‌تواند با تشکیل یک جدول جستجو با تعداد حالت کم ( $2^{16}$  حالت)، با استفاده از مقادیر خروجی به ورودی دست پیدا کند و با توجه به مطالبی که در زیربخش قبل در مورد دو مولد اعداد شبه تصادفی AKARI-1 و AKARI-2 مطرح شد، ما در طراحی پروتکل پیشنهادی می‌توانیم از این دو مولد اعداد شبه تصادفی با طول بیت حداکثر استفاده کنیم بطوریکه پروتکل پیشنهادی همچنان مبتنی بر EPC باقی بماند و نیازمندی‌های آن را برآورده کند.

مرحله 6: قرائت‌گر  $S_4 = M_1^{-1} (X_{new}^* \oplus N')$  و  $S_5 = PRNG(N_R || (X_{new}^* \oplus N'))$  را محاسبه و برای برچسب ارسال می‌کند.

مراحل انجام پروتکل پیشنهادی (شکل 5) به شرح زیر می‌باشد: مرحله 1: قرائت‌گر بردار تصادفی  $N = N_L || N_R$  را تولید کرده و  $S_1 = M_2 N$  را برای برچسب ارسال می‌کند.

مرحله 7: برچسب، مقدار  $X_{new}^* = S_4 M_1 \oplus N'$  را بازیابی می‌کند و سپس تساوی رابطه  $(PRNG(N_R || (S_4 M_1 \oplus N' \oplus N')))$  و  $S_5$  دریافت شده را بررسی می‌کند. در صورت تساوی،  $X = X_{new}^*$  را به روز می‌کند و در غیر اینصورت قرائت‌گر را رد خواهد کرد.

مرحله 2: برچسب، بردار تصادفی  $N$  را بازیابی کرده و بردار تصادفی  $N'$  را تولید می‌کند و با استفاده از  $N'$ ،  $N'' = PRNG(N')$  را محاسبه می‌کند. سپس برچسب، پیغام‌های  $(N' \oplus N)$  و  $S_2 = M_2^{-1} (N' \oplus N)$

## تحلیل امنیت پروتکل پیشنهادی

در این بخش نشان خواهیم داد که پروتکل پیشنهادی در برابر تمام حملات معرفی شده در این مقاله، مقاوم بوده و نسبت به پروتکل‌های پیشین از سطح امنیتی مناسبی برخوردار است.

### مقاومت در برابر حمله ردیابی برچسب

قرائت‌گر و برچسب در محاسبات و داده‌های ارسالی خود از بردارهای تصادفی استفاده می‌کنند و با توجه به اینکه این بردارها در هر اجرای پروتکل تغییر می‌یابند، مهاجم قادر به ردیابی برچسب نخواهد بود.

شرایطی را در نظر بگیرید که مهاجم در نظر دارد پیام  $S_1$  استراق-سمع شده از پاسخ برچسب (T1) در اجرای قبلی پروتکل را برای برچسب ناشناخته (Tx) که کلید هایش را به روز نکرده است ارسال کند. در جواب، برچسب (Tx)، پیغام‌های  $S_2 = M_2^{-1}(N' \oplus N)$  و  $S_3 = PRNG(N_L || (X \oplus N'))$  را محاسبه کرده و برای مهاجم ارسال می‌کند.

در صورتیکه (Tx) همان (T1) باشد، با توجه به اینکه  $N'$  و  $N''$  در هر دور اجرای پروتکل مقادیری متفاوت می‌باشند، در نتیجه  $S_2 \neq S_2'$  و  $S_3 \neq S_3'$  خواهند بود و مهاجم نمی‌تواند برچسب (T1) را شناسایی و در نتیجه ردیابی کند. لذا این پروتکل نسبت به حمله ردیابی برچسب ایمن می‌باشد.

### مقاومت در برابر حمله جعل هویت قرائت‌گر

در صورتیکه مهاجم بخواهد حمله جعل هویت قرائت‌گر را انجام دهد باید بتواند بردارهای تصادفی  $N'$  و  $N''$  تولید شده توسط برچسب را بازیابی کند و در  $S_4 = M_1^{-1}(X_{new}^* \oplus N')$  و  $S_5 = PRNG(N_R || (X_{new}^* \oplus N'))$  از آنها استفاده نماید و با توجه به اینکه این مقادیر در هر اجرای پروتکل تغییر پیدا می‌کنند و همچنین مقدار  $M_1^{-1}$  که فقط بین قرائت‌گر و برچسب مشترک است، مهاجم قادر به تولید  $S_4$  و  $S_5$  قابل قبول برای برچسب نیست و نمی‌تواند منجر به اجرای موفق این حمله شود.

### مقاومت در برابر حمله جعل هویت برچسب

با توجه به اینکه در پیغام‌های  $S_2 = M_2^{-1}(N' \oplus N)$  و  $S_3 = PRNG(N_L || (X \oplus N'))$  ارسالی از طرف برچسب، از بردارهای تصادفی  $N$  و  $N_L$  تولید شده توسط قرائت‌گر استفاده شده است و با توجه به اینکه در هر بار اجرای پروتکل این بردارها تغییر پیدا می‌کنند، مهاجم قادر به تولید  $S_2$  و  $S_3$  قابل قبول برای قرائت‌گر نیست و در نتیجه در حمله جعل هویت برچسب موفق نخواهد بود.

## مقاومت در برابر حمله غیرهمزمان سازی

به منظور جلوگیری از حمله غیر همزمان‌سازی، هر دو مقدار  $X$  و  $X_{new}$  در پایگاه داده قرائت‌گر ذخیره شده‌اند و در صورتیکه مهاجم در یک اجرای پروتکل، مانع از دریافت  $S_4$  و  $S_5$  و در نتیجه به روز نشدن کلید  $X$  توسط برچسب شود، قرائت‌گر در اجرای بعدی پروتکل باز هم قادر به احراز اصالت و شناسایی برچسب خواهد بود، همچنین به دلیل استفاده از مقادیر تصادفی  $N'$  و  $N''$  تولید شده توسط برچسب در پیغام‌های  $S_4$  و  $S_5$  مهاجم قادر به اجرای مرحله‌ی ۳ از حمله‌ی مشابه ارایه شده به پروتکل SS بر روی این پروتکل نمی‌باشد.

### حمله‌ی تبانی

در پروتکل پیشنهادی، اگر فرض کنیم مهاجم با تبانی با برچسب، ماتریس  $M_2^{-1}$  را بدست آورد فقط می‌تواند مقادیر  $N$  و  $N'$  و  $N''$  را بازیابی کند که با استفاده از این مقادیر قادر به انجام هیچ حمله‌ای نخواهد بود و همچنین به دلیل اینکه مقادیر بدست آمده برای هر برچسب منحصر بفرد بوده و در هر دور اجرای پروتکل به روز می‌شوند، برای برچسب‌های دیگر و همچنین در دوره‌های بعدی پروتکل قابل استفاده نمی‌باشند.

جدول ۴ نشان می‌دهد که پروتکل پیشنهادی نسبت به پروتکل‌های پیشین، از سطح امنیتی بالاتری برخوردار است.

در جدول ۵، پروتکل‌های KN، RRS، SS و پروتکل پیشنهادی را از نظر سخت افزاری مورد بررسی قرار می‌دهیم. در این جدول، L تعداد بیت دیتا می‌باشد که با توجه به نوع مولد اعداد شبه تصادفی می‌تواند ۱۶ بیت تا ۱۲۸ بیت باشد. با توجه به مقادیر این جدول و نوع مولدهای اعداد شبه تصادفی معرفی شده در این مقاله (AKARI-1 و AKARI-2)، پروتکل پیشنهادی سازگار با استاندارد EPC Class-1 Gen-2 می‌باشد.

## نتیجه گیری

سامانه‌های شناسایی بسامد رادیویی دارای مزیت‌های بسیاری می‌باشند که این امر موجب شده است که امروزه به طور گسترده مورد استفاده قرار گیرند. بنابراین تامین امنیت در این سامانه‌ها بسیار مهم می‌باشد. در این مقاله نشان داده شد که پروتکل‌های پیشنهاد شده توسط طراحان نسبت به حملات غیر همزمان‌سازی، جعل هویت قرائت‌گر و ردیابی برچسب آسیب پذیر می‌باشند و نمی‌توانند به عنوان یک الگوی مناسب مد نظر قرار گیرند.

در این مقاله پروتکل احراز اصالت جدیدی مبتنی بر ماتریس و با استفاده از PRNG های سبک وزن AKARI-1 و AKARI-2 با طول بیت حداکثر ارائه گردید بطوریکه پروتکل پیشنهادی همچنان مبتنی بر EPC باقی بماند و نیازمندی‌های آن را برآورده



## مرجع‌ها

کند. این پروتکل در مقایسه با دیگر پروتکل‌های موجود، دارای بیشترین سطح امنیتی می‌باشد.

[1] B. Alomair, A. Clark, J. Cuellart, and R. Poovendran, "Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification," accepted in IEEE/IFIP International Conference on Dependable Systems & Networks (DSN'10), Chicago, 2010.

[2] G. Avoine, "Cryptography in Radio Frequency Identification and Fair Exchange Protocols," PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, December 2005.

[3] J. Bringer, and H. Chabanne, "Trusted-HB: a low-cost version of HB+ secure against man-in-the-middle attacks," CoRR, abs/0802.0603, 2008.

[4] C.C. Tan, B. Sheng, and Q. Li, "Severless Search and Authentication Protocols for RFID," Proc. Fifth IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07), Mar. 2007.

[5] G. Tsudik, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol," Proc. Fourth IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '06), Mar. 2006.

[6] M. Burmester, B. de Medeiros, J. Munilla, and A. Peinado, "Secure EPC gen2 compliant radio frequency identification," In ADHOC-NOW 2009, LNCS 5793, Springer, pp. 227–240, 2009.

[7] E. Y. Choi, D. H. Lee, and J. I. Lim, "Anti-cloning protocol suitable to EPC global class-1 generation-2 RFID systems," Computer Standards & Interfaces, Vol.31, No.6, pp. 1124–1130, 2009.

[8] K. H. Yeh, and N. W. Lo, "Improvement of an EPC Gen2 compliant RFID authentication protocol," International Conference on Information Assurance and Security, Xi'an, China, IEEE Computer Society, pp. 532–535, 2009.

[9] P. Peris-Lopez, T. Li, and J. C. Hernandez-Castro, "Lightweight props on the weak security of EPC class-1 generation-2 standard," IEICE Transactions on Information and Systems, Vol.93-D, No.3, pp. 518–527, 2010.

[۱۰] معصومه صفحانی، تصور باقری و مجید نادری، "تحلیل امنیتی پروتکل SEAS: یک پروتکل احراز اصالت در سیستم‌های RFID"، پذیرفته شده در فصلنامه علمی و پژوهشی صنایع الکترونیک، دوره دوم، شماره ۲، ۱۳۹۰ صفحه ۷۷-۹۲، پاییز ۱۳۹۰.

[۱۱] محمد حسن حبیبی، محمود گردشی، "حمله به یک پروتکل احراز اصالت در سامانه های RFID"، پذیرفته شده در هشتمین کنفرانس بین المللی انجمن رمز ایران، دانشگاه فردوسی مشهد، ص ۹۹-۱۰۳، شهریور ۱۳۹۰.

[۱۲] محمد حسن حبیبی، "تحلیل امنیتی GW: یک پروتکل تصدیق اصالت برای سامانه های RFID"، پذیرفته شده در نشریه علمی خبری انجمن رمز ایران، شماره ۳۸، ص ۲-۸، زمستان ۸۹.

[13] M. H. Habibi, M. R. Alagheband, and M. R. Aref, "Attacks on a Lightweight Mutual Authentication Protocol under EPC C-1 G-2

جدول ۴. مقایسه پروتکل پیشنهادی با پروتکل‌های پیشین

پروتکل پیشنهادی	SS	RRS	KN	پروتکل
				مقاومت در برابر حملات
موفق	ناموفق	موفق	ناموفق	غیرهمزمان سازی
موفق	موفق	ناموفق	ناموفق	تکرار
موفق	موفق	موفق	ناموفق	تبانی
موفق	ناموفق	ناموفق	ناموفق	جعل هویت قرائتگر
موفق	موفق	موفق	ناموفق	جعل هویت برچسب
موفق	ناموفق	ناموفق	ناموفق	ردیابی برچسب

جدول ۵. مقایسه پروتکل پیشنهادی با پروتکل‌های پیشین

پروتکل پیشنهادی	SS	RRS	KN	
≈۳L	≈۴L	≈۳L	≈۳L	حافظه مصرفی در برچسب
≈۴L	≈۶L	≈۴L	≈۳L	حافظه مصرفی در قرائتگر
≈۲L	≈۲L	≈۲L	≈L	تعداد بیت‌های ارسالی توسط برچسب
≈۳L	≈۳L	≈L	≈۲L	تعداد بیت‌های ارسالی توسط قرائتگر
۴	۴	۳	.	PRNG
۴	۴	۱	۱	⊕

Standard,” WISTP, volume 6633 of Lecture Notes in Computer Science, pp.254-263, 2011.

[14] B. Abdolmaleki, H. Bakhsi, K. Baghery, and M. R. Aref, “Analysis of an RFID authentication protocol in accordance with EPC standards,” *International Journal of Information & Communication Technology Research*, vol. 6, pp. 7-12, 2014.

[15] S. Karthikeyan, and N. Nesterenko, “RFID security without extensive cryptography,” In *Workshop on Security of Ad Hoc and Sensor Networks — SASN’05*, Alexandria, VA, USA, pp. 63–67, 2005.

[16] V. Ramachandra, M. Rahman, and S. Sampalli, “Lightweight matrix-based authentication protocol for RFID,” *19th International Conference on Software, Telecommunications and Computer Networks*, Dubrovnik, Croatia, pp. 35–40, 2011.

[17] W. Shaohui, and L. Sujuan, “attacks and improvements on the RFID authentication protocols based on matrix,” *JOURNAL OF ELECTRONICS (CHINA)*, Vol.30, No.1, February 2013.

[18] H. Y. Chien, and C. H. Chen, “Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards,” *Computers Standards & Interfaces*, Vol.29, No.2, pp. 254–259, 2007.

[19] D. Seo, J. Baek, and D. Cho, “Secure RFID authentication scheme for EPC class Gen2,” *Proceedings of the 3rd International Conference on Ubiquitous information Management and Communication (ICUIMC’09)*, New York, USA, pp. 221–227, 2009.

[20] Honorio Martin, Enrique San Millán, Luis Entrena, Julio César Hernández Castro, and Pedro Peris-Lopez, “AKARI-X: A pseudorandom number generator for secure lightweight systems,” *IOLTS 2011*, pp. 228-233, 2011