# مروری بر پروتکل‌های اثبات گروهی در سامانه‌های شناسایی به‌وسیلهٔ امواج رادیویی

نصور باقری[1] و سارا مجیدی[2]

[1]استادیار دانشکده مهندسی برق،دانشگاه تربیت دبیر شهید رجایی، تهران، ایران

nbagheri@srttu.edu

[2] دانشجوی کارشناسی ارشد، دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران

s.majidi.ee@gmail.com

## چکیده

**به مجموعه‌ای از فناوری‌ها که در آن‌ها برای شناسایی افراد و اشیا، از امواج رادیویی استفاده می‌شود، سامانه‌های شناسایی با امواج رادیویی یا RFID[1] می‌گوییم. عملکرد RFID وابسته به دو دستگاه برچسب[2] و قرائت‌گر[3] است که جهت برقراری ارتباط با یکدیگر از امواج رادیویی استفاده می‌کنند. در بسیاری از کاربردهای سامانه‌های امواج رادیویی، اثبات حضور هم‌زمان تعدادی شیء یا شخص در کنار هم و در یک زمان معین اهمیت دارد. هدف از طراحی پروتکل‌های اثبات گروهی، پاسخ به این نیاز است. می‌توان گفت اثبات گروهی مدرکی است که نشان می‌دهد دو و یا تعداد بیش‌تری از برچسب‌ها به‌طور هم‌زمان توسط یک قرائت‌گر بررسی شده‌اند. این اثبات باید با بررسی‌کنندهٔ[4] متناظر قابل اثبات باشد. در این مقاله، این دسته از پروتکل‌ها ارائه و بررسی شده‌اند. در ابتدا ایدهٔ تولید اثبات گروهی بیان شده و به‌دنبال آن انواع مختلفی از این دسته از پروتکل‌ها به همراه تحلیل امنیتی آن‌ها آمده‌اند. درنهایت هم توصیه‌هایی برای طراحی یک پروتکل امن آورده شده است.**

**واژگان کلیدی**: پروتکل‌های اثبات گروهی، برچسب، قرائت‌گر، سامانه‌های شناسایی امواج رادیویی، کنترل‌کننده

## ۱- مقدمه

امروزه ضرورت شناسایی خودکار عناصر، بدون نیاز به دخالت انسان جهت ورود اطلاعات در بسیاری از عرصه‌های صنعتی احساس می‌شود. در پاسخ به این نیاز تاکنون فناوری‌های متعددی طراحی شده است. به مجموعه‌ای از فناوری‌ها که در آنان برای شناسایی اشیا، انسان و حیوانات از ماشین استفاده می‌شود، شناسایی خودکار گفته می‌شود. هدف بیشتر سامانه‌های شناسایی خودکار، افزایش کارایی است. سامانه‌های شناسایی امواج رادیویی به‌عنوان جدیدترین فناوری مورد توجه قرار گرفته است. عملکرد RFID وابسته به دو دستگاه برچسب و قرائت‌گر[۱] و پروتکل‌های بسیاری در این زمینه معرفی شده است. یکی از انواع این پروتکل‌ها، پروتکل اثبات گروهی است که ایده آن نخستین‌بار توسط جولس[5] در سال ۲۰۰۴ ارائه شد[۲]. در سال‌های اخیر مطالعات گسترده‌ای در زمینهٔ پروتکل‌های اثبات گروهی، صورت گرفته است که حاصل آن‌ها معرفی پروتکل‌های جدیدی به‌منظور بهبود طرح‌های قبلی است. در این مقاله به بررسی انواع پروتکل‌های اثبات‌گروهی و نقاط قوت و ضعف هر یک از آن‌ها در سامانه‌های RFID می‌پردازیم و حملات وارد بر آن‌ها را بررسی می‌کنیم.

## ۲- پروتکل ارائه‌شده توسط جولس

ایدهٔ تولید مدرک مبنی بر این‌که یک جفت برچسب به‌طور هم‌زمان با قرائت‌گر پویش شده‌اند، توسط جولس مطرح شده است. او این اثبات را یوکینگ[6] نامید و دو پروتکل برای

[1] Radio frequency identification

[2] Tag

[3] Reader

[4] Verifier

[5] Jules

[6] Yoking

تولید آن پیشنهاد کرد[۲]. ابتدا به معرفی این پروتکل‌ها می‌پردازیم. در این پروتکل $r_i$ و $C_i$ به‌ترتیب مقدار تصادفی تولیدشده توسط برچسب و شمارندهٔ برچسب است. هم‌چنین $x_i$ کلید مخفی برچسب و بررسی‌کننده است. تابع f چکیده‌سازی است که به‌صورت $f:\{0,1\}^d * \{0,1\}^* \rightarrow \{0,1\}^d$ تعریف می‌شود.

شکل۱ نشان‌دهندهٔ این پروتکل برای تولید اثباتی مانند $P_{AB}$ است. جولس هم‌چنین به معرفی یک پروتکل دیگر برای برچسب‌هایی که قادر به انجام توابع رمزنگاری استاندارد نیستند، پرداخته است. در این پروتکل از کد احراز هویت پیغام[1] استفاده شده است. نمایی از این پروتکل در شکل ۲ آمده است.

## ۳- حملات ارائه‌شده برای پروتکل‌های جولس

در سال ۲۰۰۵ سایتو[2] و ساکورای[3] نشان دادند که پروتکل دوم جولس در برابر حملهٔ تکرار[4] در امان نیست[۴]. علت این حمله استفاده از مقادیر تصادفی در تابع کد احراز هویت پیغام است. قرائت‌گر جعلی می‌تواند مقدار تصادفی را تولید و آن را برای برچسب نخست بفرستد تا مقدار $m_A$ را دریافت کرده و اثبات جعلی را بسازد. نمایی از این حمله در شکل ۳ آمده است. در سال ۲۰۰۶ پیراموتا[5] حملهٔ مطرح‌شده توسط سایتو و ساکورای را گسترش داد[۷]. در این مقاله حمله‌ای بسیار شبیه به حملهٔ تکرار مطرح‌شده است؛ با این تفاوت که حمله بر روی برچسب نخست اجرا می‌شود. به‌عنوان مثال چون $r_A$،که توسط برچسب نخست تولید شده است، بعد از انتقال به قرائت‌گر هرگز توسط خود برچسب نخست استفاده نخواهد شد؛ هر مقدار تصادفی دیگری را می‌توان به جای آن استفاده و نتیجه تفاوتی نخواهد کرد. این نوع از حمله در شکل ۴ آورده‌شده است.

## ۴-پروتکل ارائه‌شده توسط سایتو و ساکورای

در ادامه حملهٔ وارده بر پروتکل جولس در سال ۲۰۰۵، سایتو و ساکورای این پروتکل را بهبود داده و به معرفی پروتکلی دیگر پرداختند[۴]. آن‌ها در این پروتکل برای جلوگیری از حملهٔ تکرار، ایدهٔ استفاده از برچسب‌های زمانی[6] را مطرح کردند. استفاده از کد احراز هویت بدین صورت است که از برچسب‌های زمانی به‌عنوان ورودی تابع کد احراز هویت استفاده می‌شود؛ بنابراین بررسی‌کننده می‌تواند زمان تولید اثبات را بررسی کند. در مرحلهٔ آخر از پروتکل برچسب مقادیر $m_i$ دریافتی از قرائت‌گر را به همراه برچسب زمانی رمز کرده، و برای بررسی‌کننده می‌فرستد. نمایی از این پروتکل در شکل ۵ آورده شده است.

## ۵-حملهٔ مطرح‌شده برای پروتکل سایتو و ساکورایتوسط پیراموتا

در سال ۲۰۰۶، پیراموتا حمله‌ای برای پروتکل سایتو مطرح کرد[۷] به این صورت که قرائت‌گر جعلی برچسب‌های زمانی مختلف را برای برچسب نخست می‌فرستد و زوج‌های (TS,mA) را دریافت می‌کند؛ سپس وقتی یکی از برچسب‌های زمانی، واقعی شد، حملهٔ تکرار بدون وجود برچسب نخست قابل اجراست؛ این حمله در شکل ۶ آمده است.

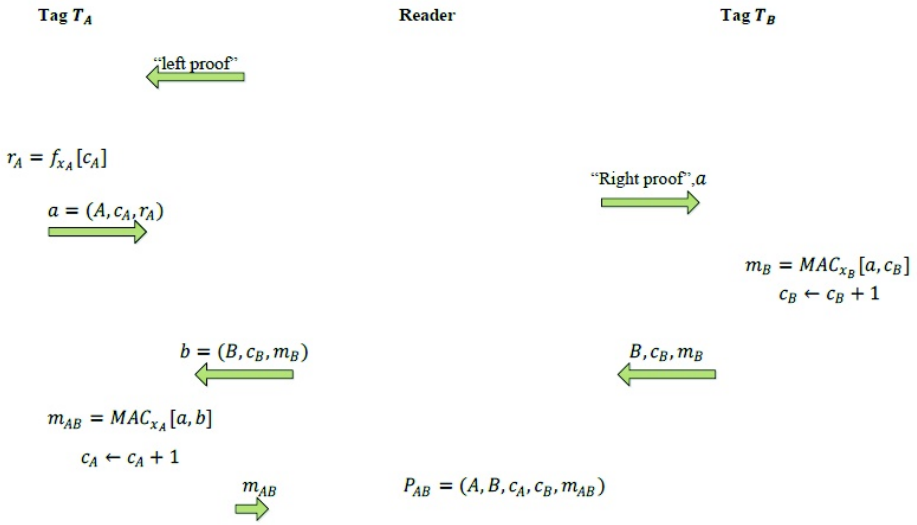
## ۶- پروتکل مطرح‌شده توسط پیراموتا

پروتکل مطرح‌شده توسط پیراموتا به‌منظور بهبود پروتکل‌های قبل بوده است[۷]. ایده طراحی این پروتکل به این صورت است که ورودی‌های یک برچسب براساس پارامترهایی هستند که این پارامترها از ملزومات برچسب دیگر است و محاسبات هر برچسب براساس محاسبات برچسب دیگر انجام می‌شود؛ بنابراین نمی‌توان آن‌ها را جدا از یکدیگر تحلیل کرد. در این‌جا فرض می‌کنیم که قرائت‌گر برای بررسی‌کننده، قبل از شروع اثبات گروهی، تصدیق هویت شده است. در شکل ۷ نمایی از این پروتکل آمده است.

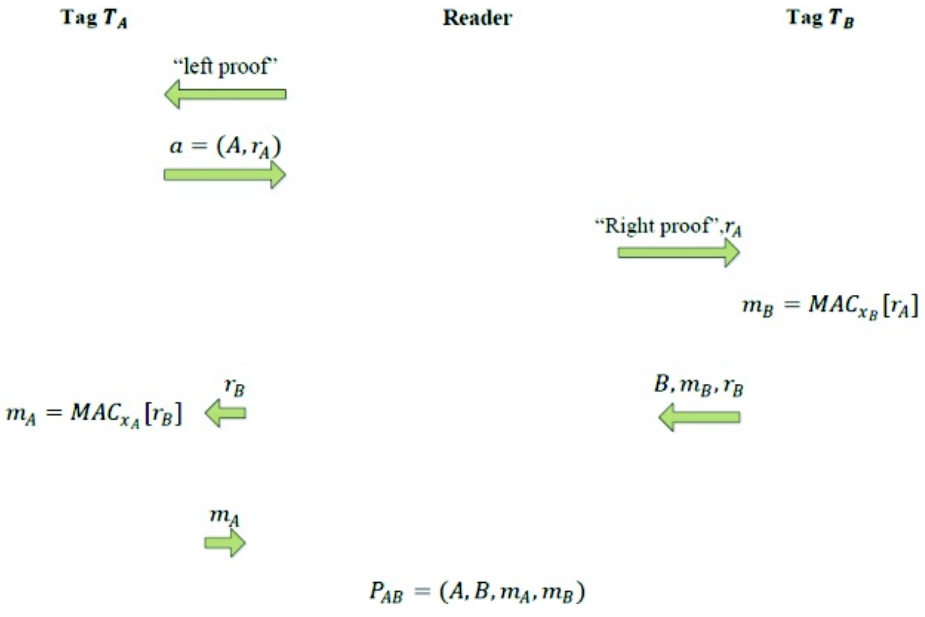
تفاوت اساسی این پروتکل با پروتکل مطرح‌شده توسط جولس این است که ۱) مقدار متغیر تصادفی r از طرف بررسی‌کننده برای هردوی برچسب‌ها فرستاده می‌شود. این امر کمک می‌کند تا بتوان زمان بین نخستین انتقال از قرائت‌گر به برچسب نخست تا فرستادن اثبات نهایی $P_{AB}$ از قرائت‌گر به بررسی‌کننده را کنترل کرد.۲) مقدار $m_B$ تولیدشده توسط برچسب دوم به مقادیر r و $r_A$ بستگی دارد.

[1]Message authentication code
[2]Saito
[3]Sakurai
[4]Reply attack
[5]Piramuthu
[6]Time stamps

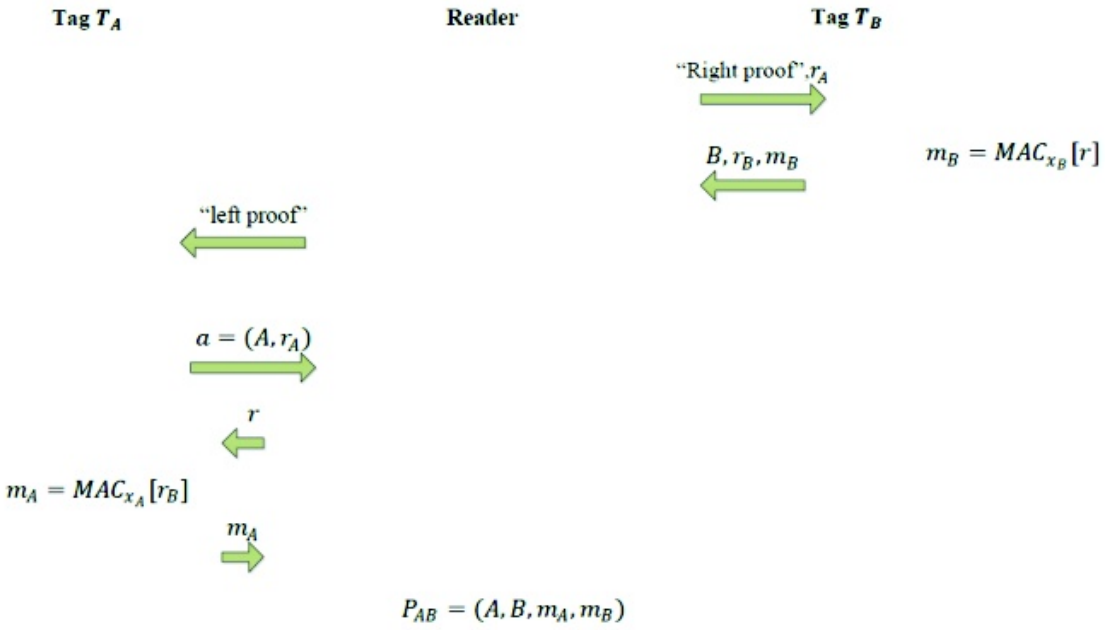
استفاده از $r_A$ امری ضروری است؛ زیرا برچسب نخست نیز از این مقدار برای تولید $m_A$ استفاده کرده است. بنابراین دشمن نمی‌تواند حملهٔ تکرار را بر روی هیچ کدام از برچسب‌ها انجام دهد. ۳) استفاده از $m_B$ در تولید $m_A$ امری ضروری است؛ زیرا برچسب نخست تا رسیدن پاسخ برچسب دوم باید منتظر بماند. بنابراین سهم برچسب نخست در تولید اثبات، هرگز قبل از برچسب دوم نخواهد بود.



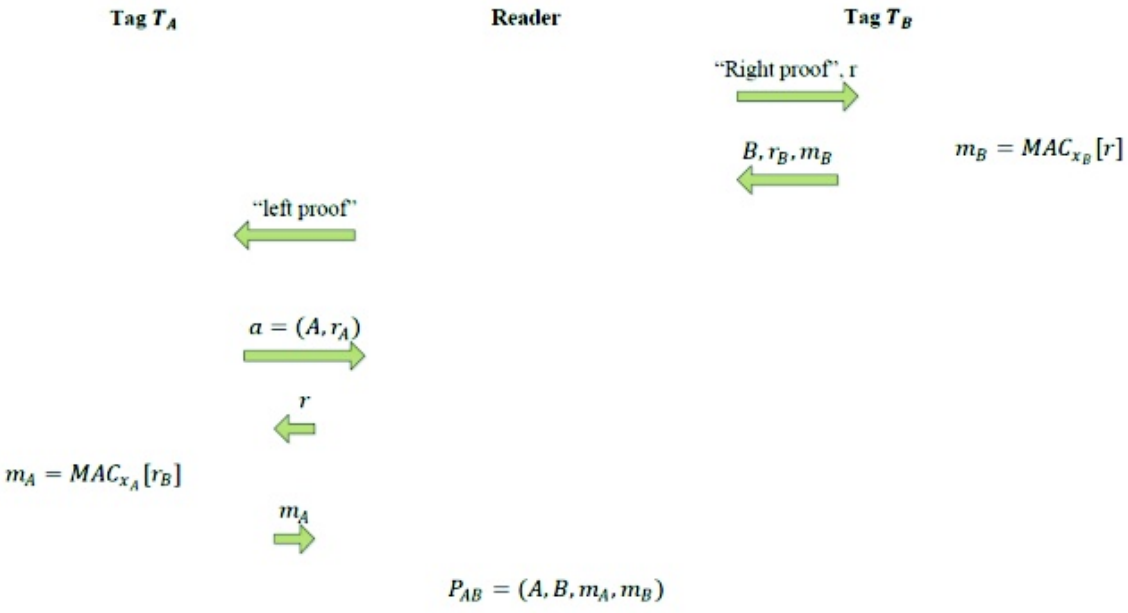

**شکل ۱: پروتکل یوکینگ با استفاده از توابع رمزنگاری اولیه [۳]**




**شکل ۲: پروتکل دوم جولس با استفاده از کد احراز هویت پیام[۳]**

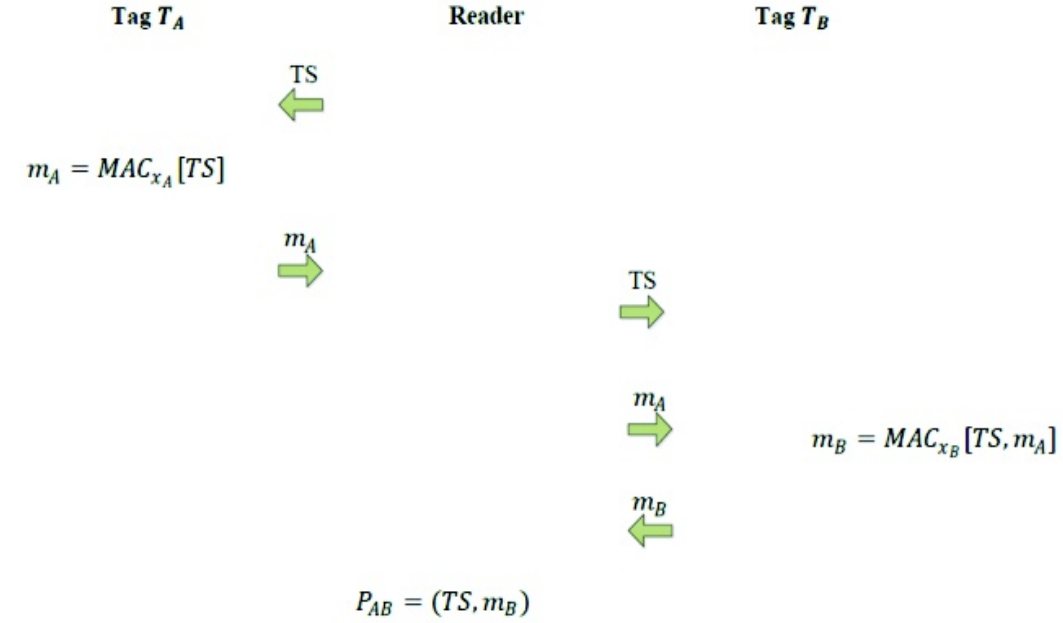



شکل ۳: حمله تکرار اعمال شده بر پروتکل دوم جولس [۵]

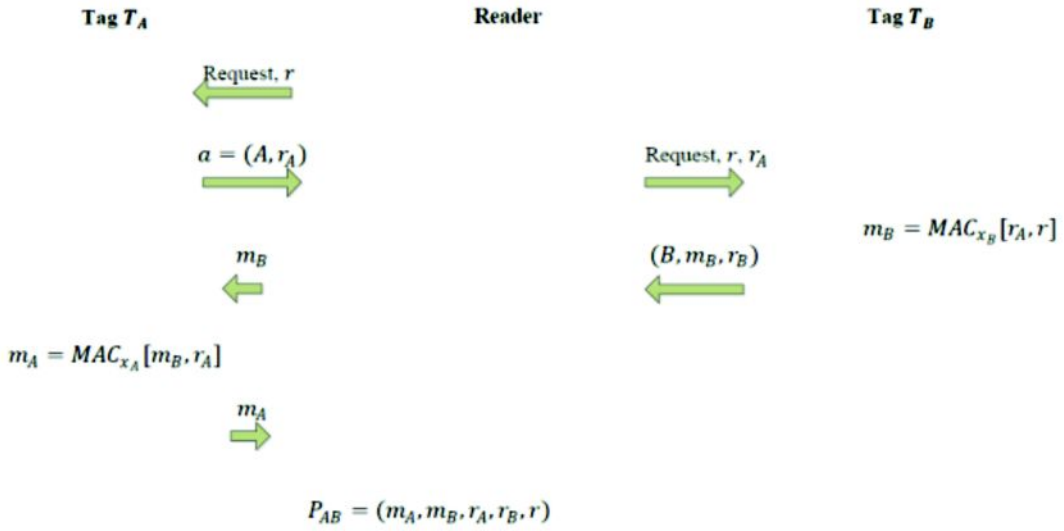



شکل ۴: حمله مطرح‌شده توسط پیراموتا برای پروتکل جولس[۶]


شکل ۵: پروتکل ارائه‌شده توسط سایتو و ساکورای با استفاده از برچسب‌های زمانی[۵]




شکل۶: حمله وارده بر پروتکل سایتو و ساکواری توسط پیراموتا[۶]




**شکل ۷: پروتکل معرفی شده توسط پیراموتا[۶]**

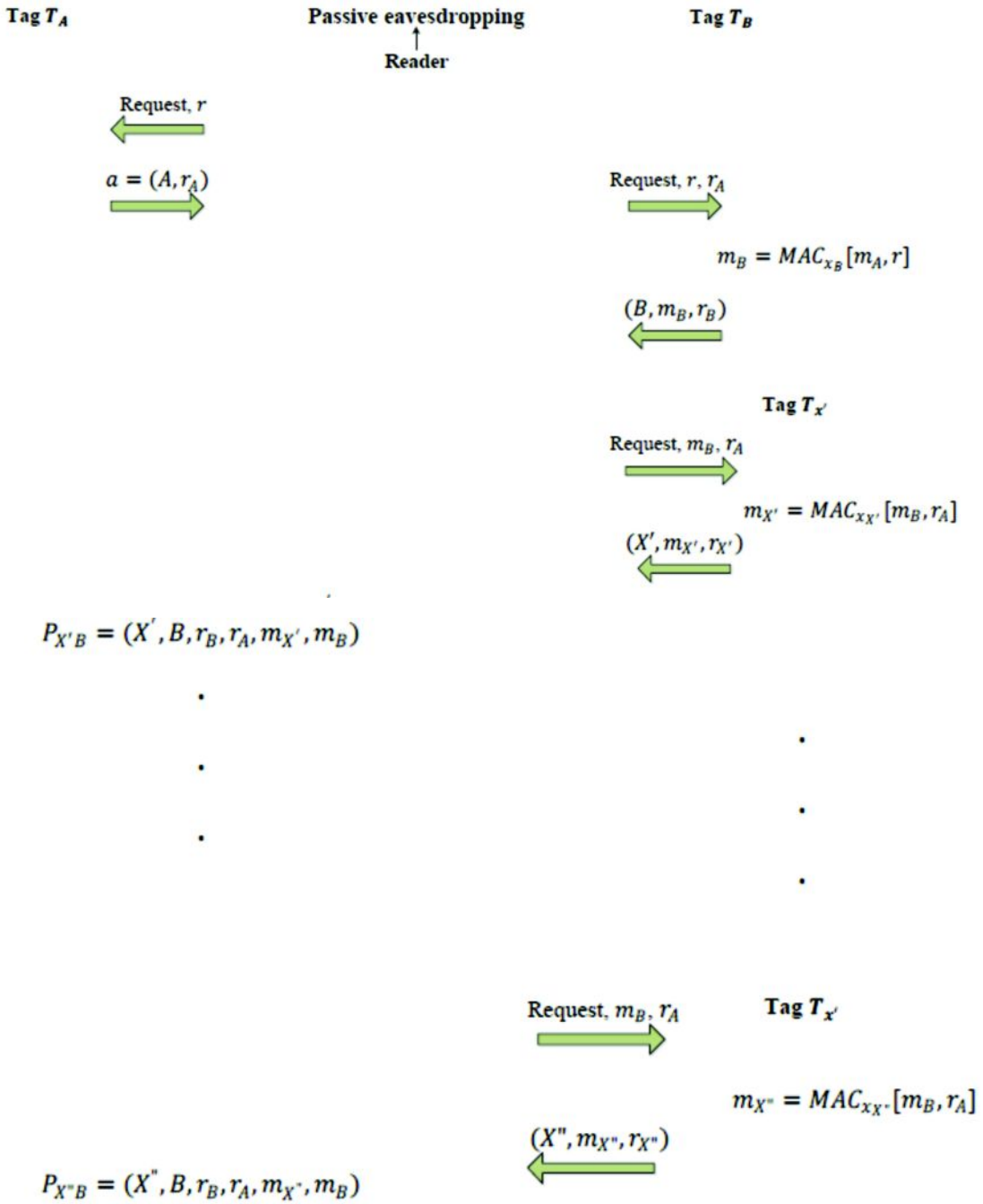
## ۷- حملهٔ وارده بر پروتکل پیراموتا توسط لوپز

در سال ۲۰۰۷ لوپز[1] و همکارانش حمله‌ای برای پروتکل پیراموتا ارائه دادند [۸]. در پروتکل پیراموتا فرض شده بود که $r$ به‌عنوان ورودی برای تولید $r_A$ استفاده می‌شود؛ اما هرگز در مورد این‌که چگونه $r_A$ می‌تواند احراز اصالت را انجام دهد، بحثی نشد؛ بنابراین $r_A$ هرگز نمی‌تواند تولید توسط برچسب نخست را تضمین کند؛ زیرا وجود هیچ نوع کلید مخفی قید نشده است. بنابراین پروتکل پیراموتا در معرض حملهٔ نشست چنداثباته[2] است. همان‌طور که در شکل ۸ نشان داده شده، روند انجام حمله به این صورت است که قرائت‌گر مقدار $r$ را از بررسی‌کننده گرفته و چالش خود را با برچسب نخست آغاز می‌کند. برچسب نخست مقدار $r_A$ را محاسبه کرده و مجموعه$a$ را برای قرائت‌گر می‌فرستد. قرائت‌گر $(request, r_A, r)$ را برای برچسب دوم ارسال می‌کند. این برچسب مقادیر $m_B = MAC_{x_B}[r_A, r]$ و $r_B$ را محاسبه می‌کند؛ سپس سه‌تایی $(B, m_B, r_B)$ را برای قرائت‌گر می‌فرستد. قرائت‌گر سه‌تایی $(request, m_B, r_A)$ را برای برچسب $X$ می‌فرستد. برچسب $X$ مقادیر $m_x = MAC_{x_X}[m_B, r_A]$ و $r_x$ را محاسبه و $(X, m_x, r_x)$ را برای قرائت‌گر ارسال می‌کند. قرائت‌گر مقدار $r_x$ را حذف کرده و مدرک اثبات جمعی $P_{XB} = (r_A, r_B, r, m_x, m_B)$ را برای بررسی‌کننده می‌فرستد. کنترل‌کننده درنهایت صحت این مدرک را بررسی می‌کند.

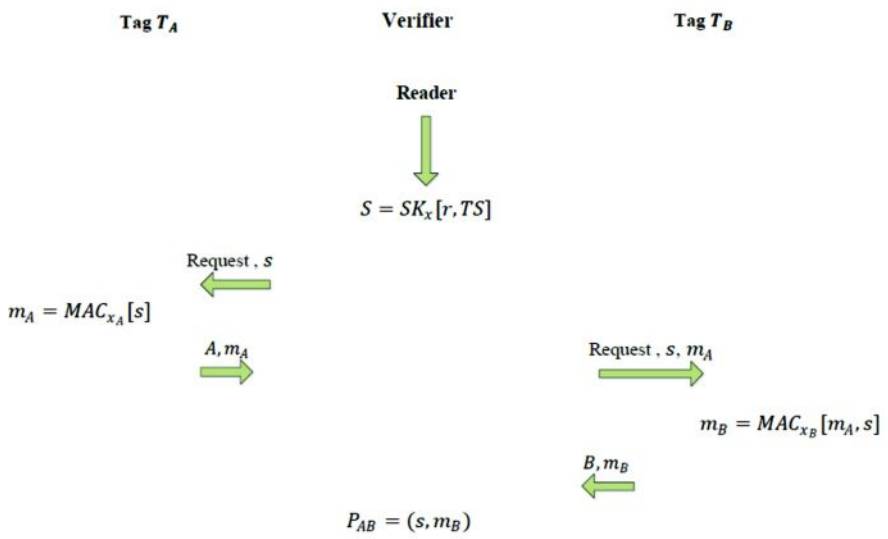
## ۸- پروتکل مطرح‌شده توسط لین و همکاران

در سال ۲۰۰۷ لین[3] و همکارانش به معرفی دو مشکل جدید در پروتکل پیراموتا پرداختند[۱۰]. نخستین مشکل زمانی پیش می‌آید که چندین قرائت‌گر در برابر یک برچسب وجود داشته باشد. سناریوی مطرح‌شده به این صورت است که قرائت‌گر شمارهٔ ۱ مقدار $r_1$ را برای برچسب نخست می‌فرستد. به‌طور هم‌زمان قرائت‌گر شمارهٔ ۲ مقدار $r_2$ را برای برچسب نخست می‌فرستد. این برچسب مقادیر $r_{1A}$ و $r_{2A}$ را تولید کرده و برای قرائت‌گرهای مربوطه ارسال می‌کند. این قرائت‌گرها با برچسب‌های مربوط به خودشان مبادلهٔ اطلاعات کرده و مقادیر $m_{1B}$ و $m_{2B}$ را برای برچسب نخست می‌فرستند. مشکل در اینجاست که برچسب نخست نمی‌داند از کدام مقدار تصادفی ($r_{1A}$ یا $r_{2A}$) همراه با $m_{1B}$ و $m_{2B}$ استفاده کرده تا بتواند $m_{1A}$ و $m_{2A}$ را تولید کند.

[1] Lopez
[2] Multi proof session attack
[3] Lin




شکل ۸: حمله وارده بر پروتکل پیراموتا توسط لوپز[۸]




**شکل شماره ۹: پروتکل برخط معرفی شده توسط لین[۹]**

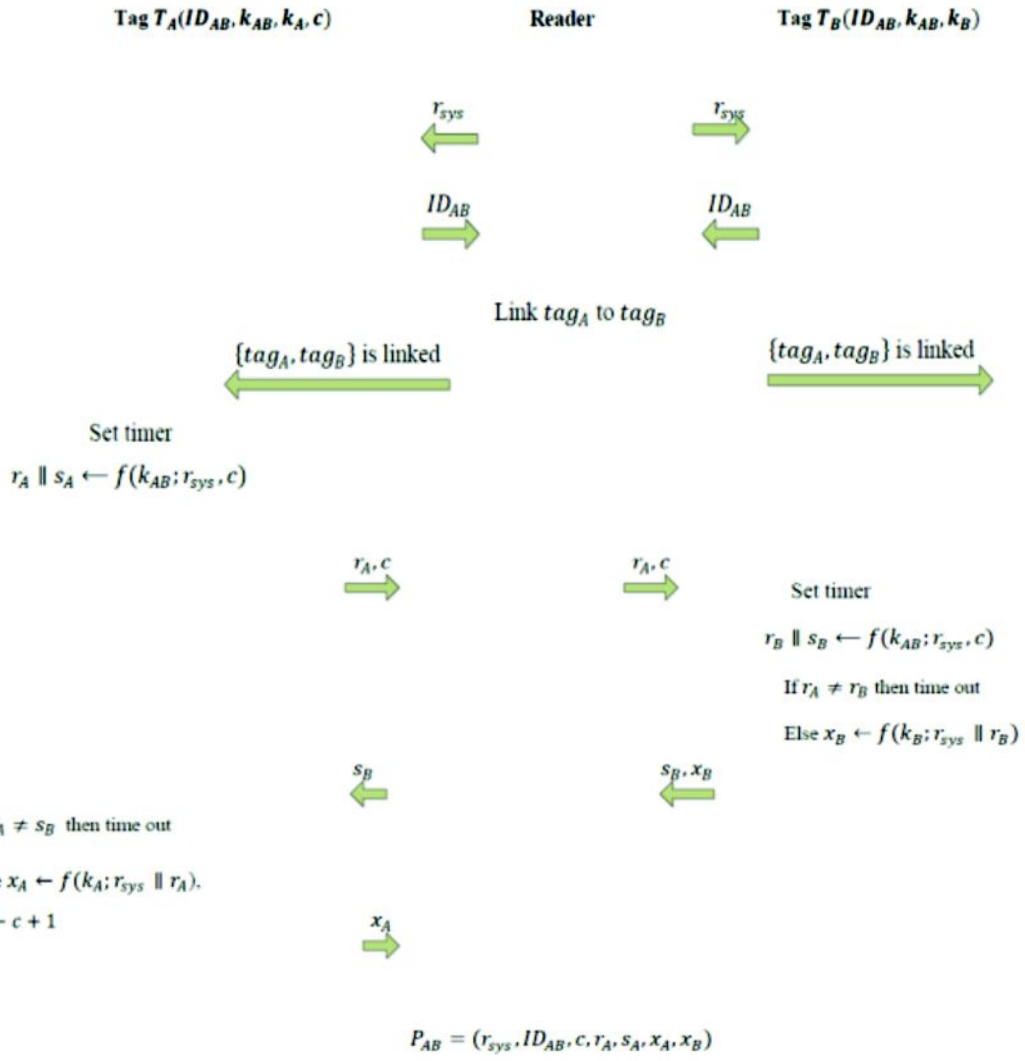
حالت دوم زمانی پیش می‌آید که چندین برچسب در مقابل برچسب نخست وجود داشته باشد. مشکل در اینجاست که برچسب نخست نمی‌داند که در مقابل خود چند برچسب وجود دارد تا به تعداد آن‌ها $r_A$ تولید کند. حتی اگر قرائت‌گر تعداد برچسب‌ها را در اختیار برچسب نخست قرار دهد، باز هم این برچسب نمی‌داند با استفاده از کدام $r_{Ai}$ وکدام $m_{Bi}$ باید $m_{Ai}$ را تولید کند. در ادامه لین و همکارانش به معرفی پروتکلی دیگر پرداختند. در این پروتکل بررسی‌کننده به‌صورت برخط[1] در اثبات حضور دارد. یعنی می‌تواند پیغامی از برچسب بگیرد و یا برای آن بفرستد (توسط قرائت‌گر). برای جلوگیری از این‌که دشمن بتواند برچسب‌های زمانی جعلی تولید کند، بررسی‌کننده مقدار $S = SK_x[r, TS]$ را به‌عنوان برچسب زمانی برای قرائت‌گر می‌فرستد، که در این رابطه x، کلیدی است که قرائت‌گر با بررسی‌کننده به اشتراک گذاشته است. در این پروتکل فرض می‌کنیم که اگر اثبات در زمان $TS + \Delta$ تولید شود، از سمت کنترل‌کننده رد خواهد شد. نمایی از این پروتکل در شکل ۹ نشان داده شده است.

## ۹- پروتکل‌های ارائه‌شده توسط بارمستر

بارمستر[2] در سال ۲۰۰۸ سه پروتکل در زمینه RFID معرفی کرد[۱۱]. پروتکل نخست گمنامی[3] را تضمین نمی‌کند، پروتکل دوم گمنامی را تضمین کرده و پروتکل سوم امنیت پیش‌رو[4] را به مجموعه خواصی که پروتکل دوم پشتیبانی می‌کند، اضافه کرده است. در پروتکل نخست اثبات از برچسب به قرائت‌گر و از قرائت‌گر به بررسی‌کننده فرستاده می‌شود؛ در پروتکل دوم هیچ شناساگری از برچسب برای قرائت‌گر فرستاده نمی‌شود و اثبات براساس شناساگرهای گروهی، کلیدهای گروهی و مقادیر فرستاده‌شده توسط قرائت‌گر می‌شود. در این حالت فقط کنترل‌کننده می‌تواند یک اثبات را برای گروهی از برچسب‌ها تطبیق دهد. این موضوع، گمنامی را تضمین می‌کند. در پروتکل سوم کلیدهای مخفی و کلیدهای گروهی برچسب‌ها در هر بار اجرای پروتکل، به‌هنگام می‌شود. در پروتکل نخست برچسب‌ها به دو دسته تقسیم که هر کدام از آن‌ها توسط یک $ID_{group}$ شناسایی می‌شوند. هر برچسب دو کلید را در خود ذخیره می‌کند: $k_{group}$ که عضویت برچسب را در گروه ثابت می‌کند و $k_{tag}$ که تصدیق هویت برچسب را انجام می‌دهد. ابتدا قرائت‌گر چالش تصادفی را منتشر می‌کند ($r_{sys}$)؛ سپس برچسب‌ها شناسهٔ گروه خود را می‌فرستند. برچسب نخست مقدار $r_A \parallel s_A = f(k_{group}; r_{sys}, c)$ را محاسبه کرده و زوج $\{r_A, c\}$ را برای قرائت‌گر ارسال کرده و شمارنده را یک واحد افزایش می‌دهد. قرائت‌گر مقدار به‌دست آمده را ذخیره کرده و برای برچسب دوم می‌فرستد. برچسب دوم مقدار

[1] Online
[2] Burmester
[3] Anonymity
[4] Forward security

$r_B \parallel s_B = f(k_{group}; r_{sys}, c)$را محاسبه کرده تا تعلق‌داشتن برچسب نخست به گروه را ثابت کند و درنهایت $x_B = f(k_B; r_{sys} \parallel r_B)$ را محاسبه کرده تا $\{s_B, x_B\}$ را برای قرائت‌گر ارسال کند. برچسب نخست تعلق‌داشتن برچسب دوم به گروه را بررسی می‌کند. در این صورت برچسب نخست پیغام تصدیق هویت خود، یعنی $x_A$ را محاسبه کرده و این مقدار را برای قرائت‌گر می‌فرستد. نمایی از این پروتکل در شکل ۱۰ آمده است.




**شکل ۱۰: پروتکل نخست معرفی شده توسط بارمستر[۱۲]**

در پروتکل دوم، شناساگر گروه با اسم مستعار گروهی جایگزین شده است. در این پروتکل یکی از برچسب‌ها باید نسخهٔ فعلی و قدیمی اسم مستعار خود را ذخیره کند. به‌منظور رسیدن به این هدف برچسب‌های گروه مقدار تصادفی $r_{tag}$ را ذخیره می‌کنند. برچسب آغازکننده تنها مقدار فعلی خود را ذخیره می‌کند، درصورتی‌که بقیه برچسب‌ها مقدار فعلی و قبلی، یعنی $(r_{tag}^{old}, r_{tag}^{current})$ را ذخیره می‌کنند.مراحل اجرای پروتکل به این صورت است که ابتدا مقدار $f(k_{group}; r_{sys} \parallel r_{tag})$ محاسبه می‌شود که در این رابطه $r_{sys}$ مقداری است که از سوی بررسی‌کننده فرستاده می‌شود. سپس این مقدار به دو قسمت $ps_{group}$ و $cnf_{tag}$ با طول‌های مساوی تجزیه می‌شود. مقدار $cnf_{tag}$ تاییدی است که برای احراز هویت اسم مستعار استفاده می‌شود. برچسب شروع‌کننده مقدار $ps_{group}$ را محاسبه می‌کند و برچسب‌های دیگر مقدارهای $ps_{group}^{old}$ و $ps_{group}^{current}$ را محاسبه می‌کنند. برچسب‌های یک گروه مقدار اسم مستعار خود را در هر بار اجرای پروتکل عوض می‌کنند. نمایی از این پروتکل در شکل ۱۱ آورده شده است.

در پروتکل سوم معرفی‌شده توسط بارمستر کلید مخفی و کلید گروه برچسب‌ها بعد از اجرای هر بار پروتکل، به‌هنگام می‌شود. همه برچسب‌ها، از جمله برچسب آغازکننده، دو گروه از کلید را ذخیره می‌کنند: $k_{group}^{i}$ و $k_{tag}^{i}$ . علاوه‌بر این مقادیر $r_{tag}^{i}$ نیز ذخیره می‌شود. در پایگاه داده نیز مقادیر $(r_{sys}, \{(k_{tag}^{t}, k_{group}^{t}, ps_{group}^{t})$ ذخیره می‌شوند. نمایی از این پروتکل در شکل ۱۲ آورده شده است.

## ۱۰- حملهٔ جعل هویت بر روی پروتکل نخست بارمستر

در سال ۲۰۱۱، لوپز و همکارانش نشان دادند پروتکل نخست بارمستر در معرض حمله جعل هویت قرار دارد[۱]. در این حمله دشمن مقدار $\{ID_{group}, c, r_A, s_A, x_A, x_B\}$ را که در کانال منتقل شده است، می‌داند. با داشتن این مقادیر حمله‌کننده می‌تواند یک اثبات هم‌زمانی برای خواندن برچسب دوم و هر برچسب دیگری مانند X را تولید کند. روندی که توسط دشمن طی می‌شود، درشکل ۱۲ آمده است.

## ۱۱- پروتکل معرفی‌شده توسط چین بر مبنای مدل درخت

در سال ۲۰۰۹، چین[1] و همکارانش پروتکلی بر مبنای مدل درخت ارائه دادند[۱۳]. در پروتکل آن‌ها برچسب‌ها به برگ‌های درخت نسبت داده شده‌اند و مسیر از ریشه تا برگ نشان‌دهنده هویت برچسب است. برچسب‌های متعلق به یک گروه به یک زیردرخت نسبت داده شده و پتانسیل وصل‌شدن به یک‌دیگر را دارند. در این‌جا فرض‌شده که بررسی‌کننده برون‌خط[2] است و کانال بین قرائت‌گر و بررسی‌کننده امن است. مشخصه یک برچسب که توسط یک مسیر نشان داده شده به دو بخش تقسیم می‌شود. بخش نخست نشان‌دهنده مشخصه یک گروه است و دومی برای مشخص‌کردن برچسب گروه به کار گرفته می‌شود. شکل ۱۴ نشان دهنده ساختار درخت است، که مثلث‌های مشخص‌شده با خط‌چین یک گروه را نشان می‌دهد. هر برچسب دارای دو مشخصه است:

۱) $Path_{Ti}^{1}$ نشان‌دهندهٔ مشخصه گروه است ۲) $Path_{Ti}^{2}$ یک مشخصهٔ جدا برای خود برچسب است. هر برچسب دارای سه کلید است:$gk_{GY}$. $gk_{GY}, IK_{Ti}, rk$ نشان‌دهندهٔ کلید مشترک در یک گروه است، $IK_{Ti}$ کلید مخفی یک برچسب و rkکه کلید ریشه و بین برچسب‌ها مشترک است. بررسی‌کننده تمام مسیرها و کلید مخفی هر برچسب را نگه‌داری می‌کند؛ درصورتی‌که قرائت‌گر فقط اطلاعات موجود در سطح گروه $rk, gk_{GY}, path_{Ti}^{1}$ را در اختیار دارد.

در مرحلهٔ نخست (مراحل ۱ تا ۳) براساس کلید گروه و پاسخی که از هر برچسب می‌گیرد، برچسب‌ها را به گروه‌های مربوطه متصل می‌کند. در مرحله دوم (مراحل ۴ تا ۸) قرائت‌گر و برچسب‌ها برای تولید یک مدرک مبنی بر حضور هم‌زمان دو برچسب در یک پنجرهٔ زمانی خاص، با یکدیگر همکاری می‌کنند. مراحل به‌طور دقیق به این صورت است که در مرحلهٔ نخست قرائت‌گر چالش تصادفی را از بررسی‌کننده دریافت و برای برچسب نخستو دوم می‌فرستد. در مرحلهٔ دوم بعد از دریافت چالش تصادفی برچسب نخست مقادیر زیر را محاسبه می‌کند:

$$h_A = h(gk_{GY}, r_{sys}, r_A) \quad (۱)$$

$$P'_{TA} = h(rk) \oplus path_{TA}^{1} \quad (۲)$$

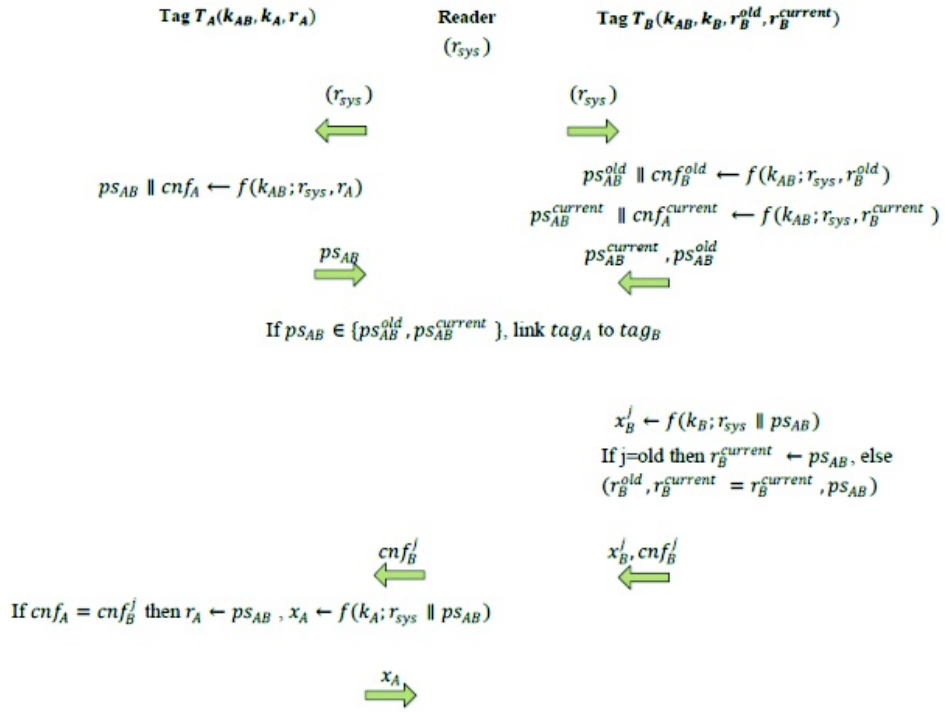
[1] Chien
[2] Offline

برچسب دوم نیز مقادیر زیر را محاسبه می‌کند:

$$h_B = h(gk_{GY}, r_{sys}, r_B) \quad (۳)$$

$$P'_{TB} = h(rk) \oplus path^1_{TB} \quad (۴)$$

در این معادلات $r_A$ و $r_B$ اعداد تصادفی هستند که توسط برچسب‌ها تولید می‌شوند. با دریافت $P'_{TA}$ و $P'_{TB}$ قرائت‌گر می‌تواند مقدار $path^1_{TA}$ و $path^1_{TB}$ را استنتاج کند؛ زیرا مقدار $h(rk)$ را محاسبه می‌کند. بنابراین ثابت می‌کند هر دو برچسب به یک گروه تعلق دارند. هم‌چنین قرائت‌گر $h_A$ و $h_B$ را محاسبه می‌کند.




**شکل شماره ۱۱: پروتکل دوم معرفی شده توسط بارمستر[۱۲]**

اگر اثبات موفقیت‌آمیز باشد، قرائت‌گر متقاعد می‌شود که هر دو برچسب به یک گروه تعلق دارند؛ در غیر این صورت پروتکل متوقف می‌شود. در مرحلهٔ سوم، قرائت‌گر مقادیر $h_A$،$h_B$ و $h(gk_{GY}, h_A, h_B)$ را برای برچسب‌ها می‌فرستد تا آن‌ها دریابند که به یک گروه تعلق دارند و در مراحل ۴ تا ۸ مدرکی مبنی بر حضور هم‌زمان خود تولید کنند. در مرحلهٔ چهارم برچسب نخست مقادیر زیر را محاسبه کرده تا بتواند به‌صورت امن مقدار $path^2_{TA}$ را حمل کند.

$$P''_{TA} = h(gk_{GY}, r_{sys}) \oplus path^2_{TA} \quad (۵)$$

$$a_1 = h(Ik_{TA}, h_A, h_B, r_{sys}) \quad (۶)$$

در مرحلهٔ پنجم قرائت‌گر مقدار $a_1$ را برای برچسب دوم ارسال می‌کند. در مرحله ششم، برچسب دوم مقدار $P''_{TB}$ را محاسبه کرده تا بتواند به‌صورت امن مقدار $path^2_{TB}$ را حمل کند. این برچسب هم‌چنین مقدار b را محاسبه می‌کند.

$$P''_{TB} = h(gk_{GY}, r_{sys}) \oplus path^2_{TB} \quad (۷)$$

$$b = h(Ik_{TB}, h_A, h_B, a_1, r_{sys}) \quad (۸)$$

مرحلهٔ هفتم: قرائت‌گر مقدار b را برای برچسب نخست می‌فرستد. مرحلهٔ هشتم: برچسب نخست مقدار $a_2$ را محاسبه کرده و برای قرائت‌گر می‌فرستد.

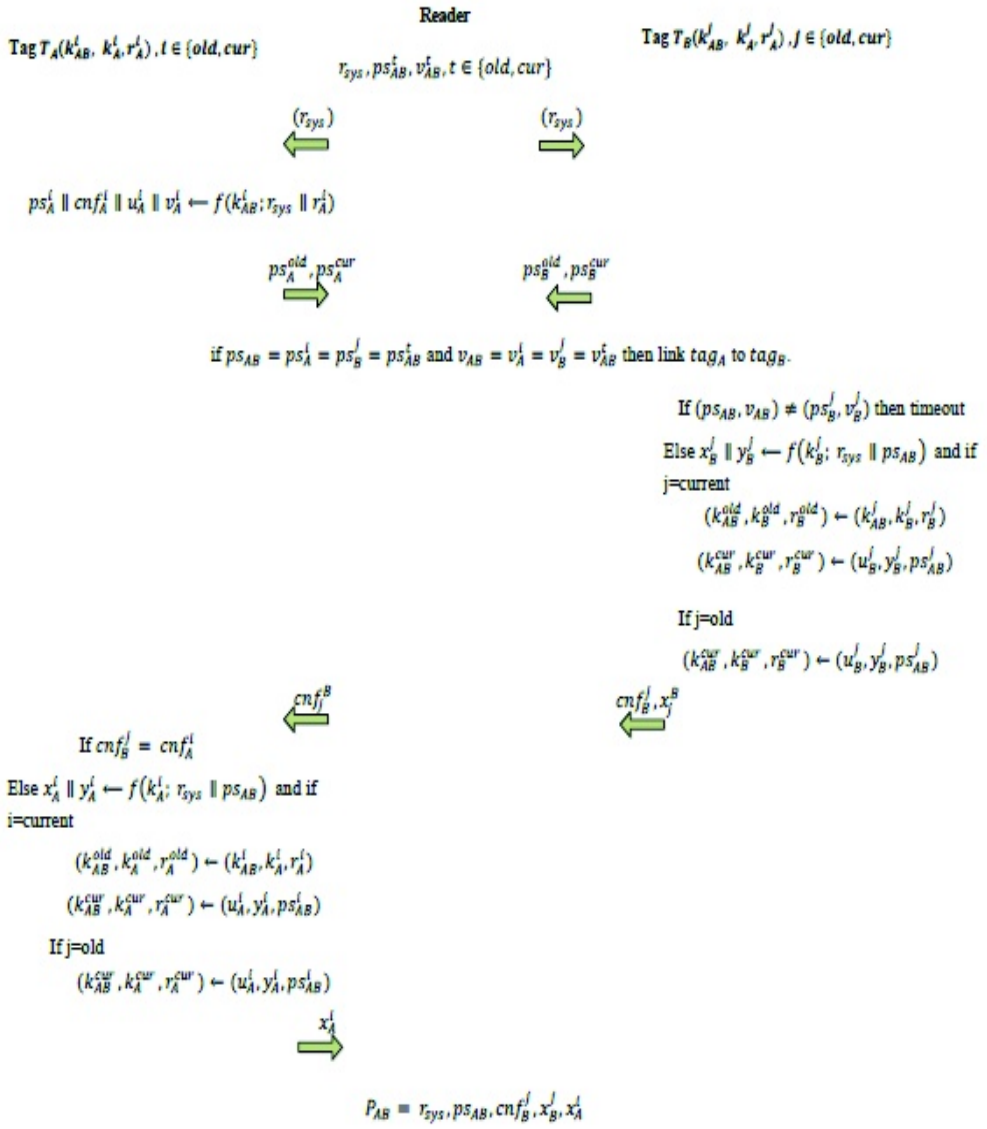
$$a_2 = h(Ik_{TA}, h_A, h_B, b, r_{sys}) \quad (۹)$$

اثبات نهایی شامل موارد زیر است:

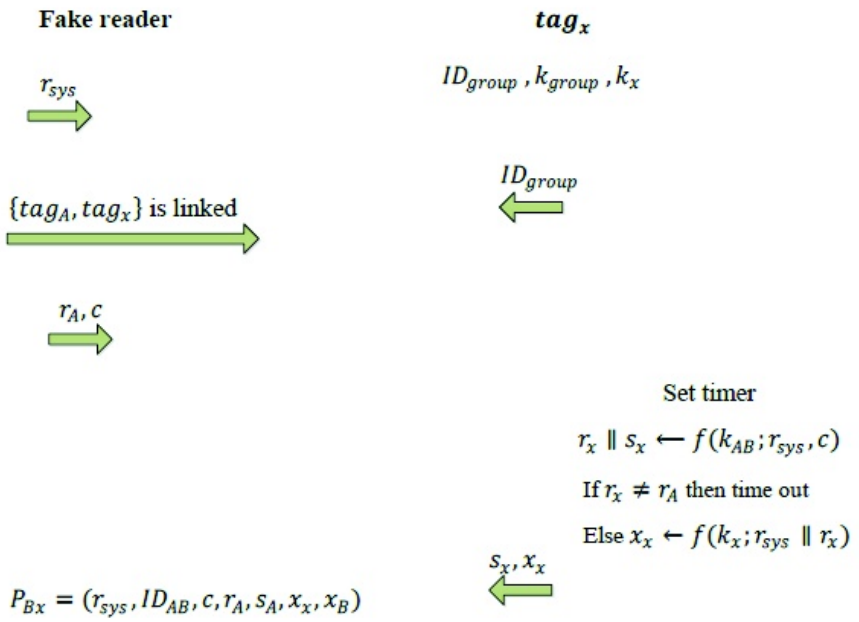
$$P_{AB} = (r_{sys}, P''_{TA}, P''_{TB}, h_A, h_B, b, r_{sys}, a_1, a_2) \quad (۱۰)$$

**۱۲- حملهٔ ردیابی در پروتکل چین**

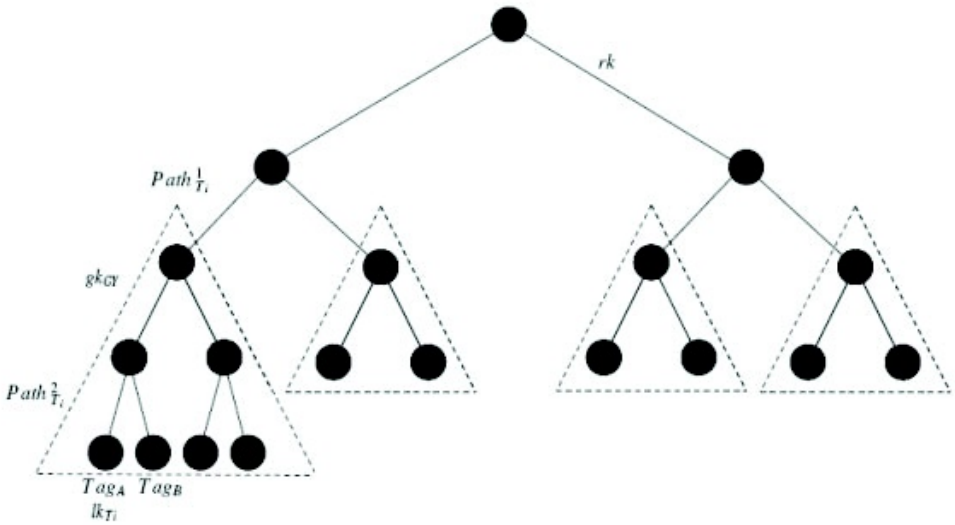
چین و همکارانش ادعا کردند که پروتکل آن‌ها در معرض حملهٔ ردیابی قرار ندارد. طبق نظر آن‌ها مشخصهٔ رمزشده $P''_{TA}$ تصادفی است و برای بخش‌های مختلف مستقل است؛ چیزی که ردیابی را غیر ممکن می‌کند. در سال ۲۰۱۱ لوپز و همکارانش نشان دادند حملات زیر در این پروتکل وجود دارد [۱]. نخست: شنودکننده یا دشمن، قادر است بفهمد دو برچسب که به‌طور هم‌زمان پویش شده‌اند به یک گروه تعلق دارند یاخیر. برای فهم این موضوع کافی است فقط مقدار $P'_{TA}$ و $P'_{TB}$ را داشته باشد. اگر مقدار $P'_{TA} \oplus P'_{TB} = 0$ باشد می‌توان نتیجه گرفت این دو برچسب متعلق به یک گروه هستند. دوم: اگر شنودکننده مقدار $P'_{Ti}$ را در دو بخش جدا به‌دست آورد می‌تواند بفهمد که هر دو برچسب از یک گروه فرستاده شده‌اند یا خیر. در چنین حالتی XOR پیغام برابر صفر می‌شود، $(P'^{old}_{Ti} \oplus P'^{new}_{Ti} = 0)$.




**شکل ۱۲: پروتکل سوم معرفی شده توسط بارمستر [۱۲]**




**شکل ۱۳: حمله جعل هویت بر روی پروتکل نخست بارمستر[۱]**




**شکل ۱۴: ساختار درخت در گروهی از برچسب‌ها[۱۳]**

سوم: شنودکننده می‌تواند برچسب‌ها را در یک جفت بررسی کند. به‌عبارت دیگر می‌توان فهمید که آیا تلاش‌های فعلی برای تولید یک اثبات جمعی بین دو برچسب به‌عنوان $P_{AB}$ متناظر با تلاش‌های گذشته است. شنودکننده می‌تواند یک جفت را در یک گروه ردیابی و آن‌ها را بعد تکرار دوبارهٔ مقدار شناسایی کند. به‌طورکلی از مراحل ۴ تا ۶ دشمن می‌تواند مقدار $P''_{TA}$و $P''_{TB}$ و سپس مقدار $P''_{TA} \oplus P''_{TB}$ را به‌دست بیاورد. این مقدار ذخیره می‌شود و اگر در آینده دوباره این مقدار تکرار شد، می‌تواند دریابد یک اثبات جمعی از همان جفت در گذشته تولید شده است. باید

متذکر شد، لازم نیست تمام ترکیبات ممکن بین دو برچسب در یک گروه بررسی شود؛ این امر به‌خاطر خواص عمل‌گر XOR است. به‌عنوان مثال اگر جفت‌های $(P''_{TA},P''_{TB})$ و$(P''_{TA},P''_{TC})$، را داشته باشیم می‌توان $P''_{TB}\oplus P''_{TC}$ را محاسبه کرد. به‌طورکلی برای یک گروه شامل N مقدار به جای محاسبهٔ $\binom{N}{2}$ حالت مختلف فقط کافی است N ترکیب مختلف شنود شود. در این حالت با وجود این‌که$Path^2_{Ti}$ فاش نشده، اما ردیابی انجام شده است.


**شکل ۱۵: پروتکل ارائه‌شده توسط هونگ و کو برای امنیت دارویی بیماران[۱۵]**

## ۱۳- پروتکل ارائه‌شده توسط هونگ و کو

هونگ[1] و کو[2] در سال ۲۰۰۹ پروتکلی ارائه دادند[۱۴] که نمایی از این پروتکل در شکل ۱۵ آمده است. از نظر آن‌ها خطاهای دارویی باعث مرگ بسیاری از انسان‌ها می‌شود. بنابراین شناسایی درست بیمار و دارو بسیار مهم است. پروتکل مدرکی مبنی بر این‌که $\{tag_1, \dots, tag_n, tag_{pallet}\}$ هم‌زمان قرائت شده‌اند، تولید می‌کند. $tag_{pallet}$ برچسب بیمار و $\text{tag}_\text{i}$ برچسب دارو است. در این نوع از پروتکل‌ها با توابع تولید اعداد تصادفی شانزده‌بیتی، عمل‌گرهای بیتی مانند XOR و توابع کد افزونگی چرخشی[3] سروکار داریم. این تابع برای اثبات انتقال اطلاعات در خطوط انتقال کاربرد دارد. علاوه‌براین برچسب‌ها دارای دو نوع رمز عبور هستند. ۱) رمز عبور دسترسی[4] که درحقیقت دسترسی به حافظه را کنترل می‌کند. ۲) رمز عبور کشتن[5] که برچسب را تا پذیرش مجدد آن غیرفعال می‌کند. در این پروتکل $\text{PIN}_\text{i}$[6] شمارهٔ شناسایی مختص برچسب و $\text{EPC}_\text{i}$[7] کد کالای الکترونیکی مربوط به برچسب است.

## ۱۴- ارائهٔ حملهٔ جعل توسط چین برای پروتکل هونگ و کو

در سال ۲۰۱۰ چین[8] و همکارانش به معرفی حملهٔ جعل در مقابل پروتکل ارائه‌شده در بخش قبل پرداختند[۱۷]. طبق خطی‌سازی تابع کد افزونگی چرخشی دارای خواص زیر است:

$$\text{CRC}(\text{A}\oplus\text{B}) = \text{CRC}(\text{A})\oplus\text{CRC}(\text{B}) \quad (۱۱)$$

$$\text{CRC}(\text{A} \parallel \text{B}) = \text{CRC}(\text{A} \ll \text{n})\oplus\text{CR}C(B) \quad (۱۲)$$

در رابطهٔ بالا n طول بیتی متغیر B است.

حمله‌کننده می‌تواند با استفاده از این خواص اطلاعات خصوصی یک برچسب را به‌دست آورد و در آینده این برچسب را جعل کند. حمله‌کننده مراحل زیر را طی می‌کند:

**مرحلهٔ نخست: به‌دست‌آوردن اطلاعات خصوصی**

دشمن یک مقدار شناخته‌شده را برای برچسب i می‌فرستد.(a)

این برچسب مقدار a و $\text{PIN}_\text{i}$ را به تابع مولد اعداد شبه‌تصادفی[9] خود می‌دهد و مقادیر زیر را محاسبه می‌کند.

$$\text{C}_\text{i} = \text{PRNG}(\text{PIN}_\text{i}) \quad (۱۳)$$

$$\text{r}_\text{i} = \text{PRNG}(\text{a}) \quad (۱۴)$$

همان‌طور که در قسمت قبل گفته شد، برچسب بر روی مقادیر $EPC_\text{i}$ و $C_i$ متمرکز شده و مقدار مربوط به تابع CRCرا تولید می‌کند. درنهایت مقدار $m_i$ محاسبه می‌شود و مقدار رمز عبور مخفی خود را عوض می‌کند.

دشمن می‌تواند مقدار $r_i$ را از روی تابع PRNGبه‌دست آورد. شناسهٔ برچسب یعنی $EPC_i$ نیز به‌وضوح در کانال منتقل شده و برای دشمن فاش می‌شود. با داشتن این اطلاعات و خواص تابع CRC دشمن می‌تواند اطلاعات خصوصی مربوط به این برچسب را به‌دست آورد.

$$\text{m}_\text{i} = \text{CRC}(\text{EPC}_\text{i}\|\text{C}_\text{i})\oplus\text{C}_\text{i} \oplus \text{r}_\text{i} = \text{CRC}(\text{EPC}_\text{i} \ll \text{n})\oplus\text{CRC}(\text{C}_\text{i})\oplus\text{C}_\text{i} \oplus \text{r}_\text{i} \quad (۱۵)$$

به‌طور خلاصه می‌توان گفت دشمن مقدار $\text{C}_\text{i}\oplus\text{CRC}(\text{C}_\text{i})$ را که مربوط به برچسب iام است، به‌دست می‌آورد.

$$\text{S}_\text{i} = \text{CRC}(\text{C}_\text{i})\oplus\text{C}_\text{i} = \text{m}_\text{i}\oplus\text{CRC}(\text{EPC}_\text{i} \ll 16)\oplus\text{r}_\text{i} \quad (۱۶)$$

**مرحلهٔ دوم: تولید یک اثبات جعلی**

قرائت‌گر قانونی به دشمنی که خود را به جای یکی از برچسب‌ها جا زده است، مقدار تصدیق هویت شدهٔ $\text{m}'_{\text{i}-1}$ رامی‌فرستد.

دشمن این مقدار را وارد تابع PRNG کرده و مقدار $\text{r}'_\text{i} = \text{PRNG}(\text{m}'_{\text{i}-1})$ را محاسبه می‌کند. در این مرحله با داشتن مقدار $\text{s}_\text{i} = \text{CRC}(\text{C}_\text{i})\oplus\text{C}_\text{i}$، کد کالای الکترونیکی و معادله می‌تواند پیغام تصدیق هویت‌شدهٔ جعلی را تولید کند. این پیغام به‌صورت زیر است:

$$\text{m}'_\text{i} = \text{s}_\text{i}\oplus\text{r}'_\text{i}\oplus\text{CRC}(\text{EPC}_\text{i} \ll 16) \quad (۱۷)$$

بنابراین دشمن می‌تواند قرائت‌گر را گمراه کند که برچسب i ام در گروه حضور دارد؛ درصورتی‌که این‌طور نیست و این برچسب غایب است.

[1] Huang
[2] Ku
[3] Cyclic redundancy check
[4] access password
[5] Kill password
[6] Personal identification number
[7] Electronic product code
[8] Chien
[9] Pseudo random number generator

## ۱۵- پروتکل‌های ارائه‌شده توسط چین‌برای افزایش امنیت دارویی بیماران

### ۱۵-۱ پروتکل برخط ارائه‌شده

در یک اقدام برای تصحیح خطاهای امنیتی الگوهای پیشین، چین در سال ۲۰۱۰ دو پروتکل اثبات گروهی را معرفی کرد[۱۶]. در پروتکل نخست کنترل‌کننده برخط و در دومی کنترل‌کننده برون‌خط است. برچسب و کنترل‌کننده یک مقدار مخفی $PIN_i$ را به اشتراک می‌گذارند. هم‌چنین برچسب کد کالای الکترونیکی را در حافظهٔ خود ذخیره می‌کند. به‌طور خاص الگوی زیر پیشنهاد شده است:

قرائت‌گر مقداری تصادفی مانند $N_R$ تولید کرده و با تمام برچسب‌ها چالش را آغاز می‌کند. روند زیر برای تمام برچسب‌ها تکرار می شود:

برچسب مقدار تصادفی $N_i$ را تولید کرده و پیغام احراز هویت را به‌صورت زیر تولید می کند:

$$MAC_i = PRNG\big(EPC_i \oplus PRNG(PIN_i) \oplus PRNG(N_R) \oplus PRNG(N_i)\big) \quad (۱۸)$$

برچسب سه‌تایی $\{EPC_i, N_i, MAC_i\}$ را برای قرائت‌گر می‌فرستد.

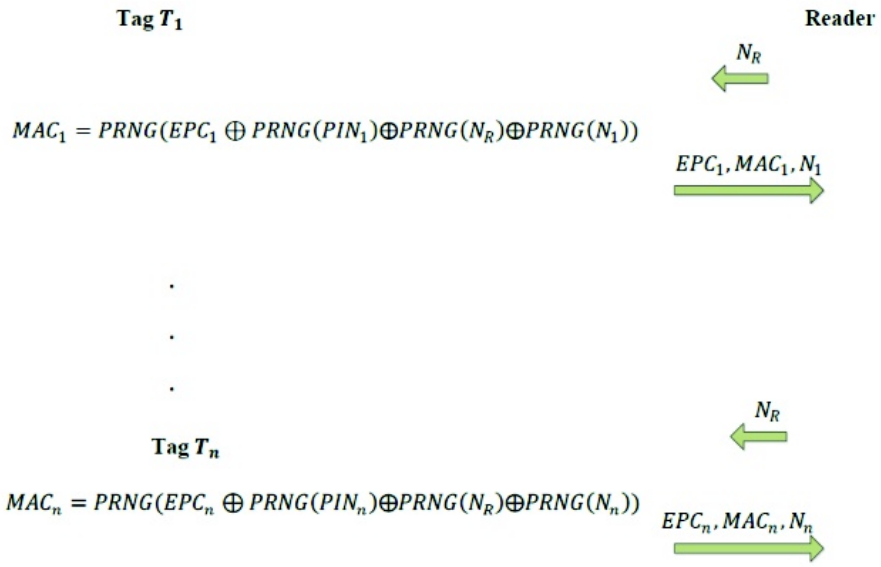
قرائت‌گر صحیح‌بودن مقدار $MAC_i$ را برای هر برچسب بررسی می‌کند. اگر بررسی موفقیت‌آمیز بود، این برچسب‌ها در یک گروه هستند. نمایی از این پروتکل در شکل ۱۶ آمده است.

### ۱۵-۲ پروتکل برون‌خط ارائه‌شده

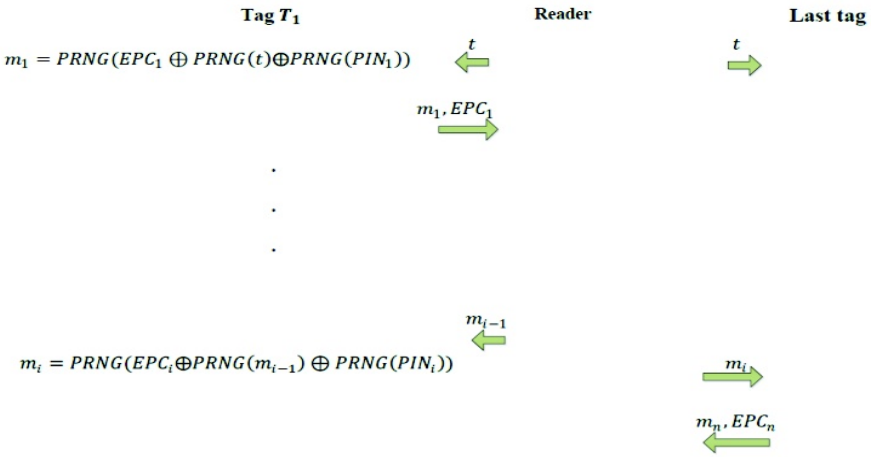
این پروتکل که در شکل ۱۷ نشان داده شده، در مورد این‌که مجموعه‌ای از داروهای خاص به بیمار داده شده است، بحث می‌کند. درحقیقت بررسی‌کنندهٔ برون‌خط می‌داند کدام دارو با کدام بیمار متناظر است. بنابراین هر دارو با یک برچسب همراه شده و هر بیمار نیز یک برچسب دارد (pallet). مراحل این پروتکل در زیر شرح داده شده است:

قرائت‌گر، مقدار رمزگذاری‌شدهٔ برچسب زمانی را که با استفاده از کلید مشترک قرائت‌گر و بررسی‌کننده محاسبه‌شده از بررسی‌کننده دریافت می‌کند($t$).

قرائت‌گر مقدار t را برای برچسب نخست می‌فرستد. برچسب i ام مقدار زیر را محاسبه کرده و همراه $EPC_i$ برای قرائت‌گر می فرستد.




شکل ۱۶: پروتکل برخط چین برای افزایش امنیت دارویی بیماران[ ۱۷]




**شکل ۱۷: پروتکل برون‌خط چین برای افزایش امنیت دارویی بیماران [ ۱۷]**

$$m_i = PRNG\left(EPC_i \oplus PRNG(m_{i-1}) \oplus PRNG(PIN_i)\right) \quad (۱۹)$$

در این معادله اگر $i = 1$ باشد، مقدار $m_0 = t$ را قرار می دهیم.

قرائت‌گر مقدار $m_i$ را برای برچسب مربوط به بیمار می‌فرستد. این برچسب مقدار زیر را محاسبه کرده و برای قرائت‌گر می‌فرستد. قرائت‌گر مجموعهٔ زیر را برای بررسی‌کننده ارسال می‌کند.

$$(t, EPC_1, m_1, \dots, EPC_i, m_i, EPC_n, m_n) \quad (۲۰)$$

نمایی از این پروتکل در شکل۱۷ آورده شده است. این پروتکل برای اثبات انتقال اطلاعات در خطوط انتقال کاربرد دارد. علاوه‌براین، در این پروتکل برچسب‌ها دارای دو نوع رمزعبور هستند. ۱) رمز عبور دسترسی[1] که درحقیقت دسترسی به حافظه را کنترل می کند. ۲) رمز عبور کشتن[2] که برچسب را تا پذیرش مجدد آن غیرفعال می‌کند. در این پروتکل $PIN_i$[3] شمارهٔ شناسایی مختص برچسب و $EPC_i$[4] کد کالای الکترونیکی مربوط به برچسب است.

## ۱۶- حملهٔ تکراردر پروتکل بـــرون‌خـــط چین

در سال ۲۰۱۱ لوپز و همکارانش به معرفی حملهٔ تکرار برای پروتکل برون‌خط معرفی‌شده توسط چین پرداختند[۱]. این پروتکل فرض می‌کند بررسی‌کننده نسخهٔ هر بیمار را می‌داند. فرض می‌کنیم نسخهٔ بیمار نخست زیرمجموعهٔ نسخهٔ بیمار دوم باشد. در این حالت، با شنودکردن پیغام مبادله‌شده هنگام تولید اثبات برای بیمار دوم، می‌توان یک اثبات جعلی مبنی بر این‌که بیمار نخست نسخهٔ خود را دریافت کرده، تولید کرد. به‌عنوان مثال فرض می‌کنیم نسخهٔ بیمار، نخست داروی ایبوپروفن و پاراستامول باشد و داروی بیمار دوم ایبوپروفن، پاراستامول و مورفین است. وقتی که یک‌بار پیغام متناظر با اثبات گروهی برای بیمار دوم $(t, m_1, EPC_1, \dots, m_B, EPC_B)$ شنود شود، قرائت‌گر جعلی می‌تواند مقدار $m_2$ را برای بیمار نخست تکرار کرده تا این بیمار زوج $(EPC_A, m_A)$ را برای قرائت‌گر جعلی بفرستد و $(t, EPC_1, m_1, EPC_2, m_2, EPC_A, m_A)$ تولید شود. در این‌جا باید متذکر شد علت این حمله این است که بررسی‌کننده، تناظری بین مقدار متغیر $t$ و مجموعه‌ای از برچسب‌ها برقرار نمی‌کند؛ بنابراین دشمن می‌تواند با پیغام‌های مربوط به یک گروه از برچسب‌ها (که شامل مقدار $t$ هستند) و استفاده مجدد از آن‌ها یک اثبات جعلی را برای گروه دیگری از برچسب‌ها به وجود بیاورد. جدول شماره ۱ مقایسه‌ای بین برخی از پروتکل‌های بررسی شده در این مقاله را نشان می‌دهد.

[1] Access password
[2] Kill password
[3] Personal identification number
[4] Electronic product code

## ۱۷- معرفی پروتکل کازاهایا، یک پروتکل اثبات گروهی برای برچسب‌های ارزان‌قیمت

در سال ۲۰۱۱، لوپز و همکارانش به معرفی پروتکلی برای برچسب‌های ارزان‌قیمت پرداختند که در شکل ۱۸ نشان داده شده است[۱]. عملکرد این برچسب‌ها به یک تابع PRNG و یک عمل‌گر XOR محدود می‌شود. در این پروتکل، برچسب‌ها به گروه‌هایی تقسیم می‌شوند که توسط شناسهٔ گروه شناسایی ($\mathrm{ID_{group}}$) می‌شوند. هر برچسب یک شناسهٔ یکتا ($\mathrm{ID_{Ti}}$) دارد و دو کلید را ذخیره می‌کند. یکی کلید گروه ($\mathrm{K_{group}}$) که عضویت برچسب را در یک گروه خاص ثابت می‌کند و کلید $\mathrm{K_{Ti}}$ است که احراز هویت برچسب را ممکن می‌کند. شناسهٔ ایستای برچسب‌ها $\{\mathrm{ID_{Ti}}, \mathrm{ID_{group}}\}$ هرگز به‌صورت آشکار و واضح در کانال منتقل نمی‌شوند تا امنیت حریم خصوصی حفظ شود. در مرحله نخست بررسی‌کننده مقدار رمزگذاری‌شده $t_n = E_{KV}(timestamp_n)$ را محاسبه می‌کند. در این معادله $K_V$ کلید مخفی کنترل‌کننده است. هر کدام از این مقادیر در یک پنجرهٔ زمانی $\Delta_n$ معتبر است. درنهایت کنترل‌کنندهٔ زوج $\{t_n, \Delta_n\}$ را ذخیره می‌کند. پروتکل هنگامی‌که قرائت‌گر مقدار رمزگذاری‌شده $t_n$ را دریافت کند آغاز می‌شود. به‌طورکلی محاسبات برچسب‌ها به‌جز نخستین برچسب براساس مقادیر محاسبه‌شده توسط برچسب‌های قبلی شرکت‌کننده در اثبات است. قرائت‌گر با فرستادن مقدار $t_n$ برچسب نخست را به چالش می‌کشد. برچسب نخست دو مقدار تصادفی $(r_{T_A}, r'_{T_A})$ را تولید کرده و مقادیر زیر را محاسبه می کند:

$$M^1_{\mathrm{group}} = PRNG(ID_{group} \oplus r_{TA} \oplus PRNG(K_{group}) PRNG(t_n)) \quad (۲۱)$$

$$M_{T_A} = PRNG\left(ID_{T_A} \oplus r'_{T_A} \oplus PRNG(K_{T_A}) \oplus PRNG(t_n + 1)\right) \quad (۲۲)$$

برچسب نخست مقادیر $\{r_{T_A}, r'_{T_A}, M^1_{group}, M_{T_A}\}$ را برای قرائت‌گر می‌فرستد.

قرائت‌گر مقدار $r'_{T_A}$ را ذخیره کرده و مجموعه $\{t_n, r_{T_A}, M^1_{group}, M_{T_A}\}$ را برای برچسب B ارسال می‌کند. برچسب دوم تعلق‌داشتن برچسب نخست به گروه را بررسی می‌کند. به‌طور دقیق‌تر برچسب B مقدار زیر را محاسبه می‌کند:

$$M^{1*}_{\mathrm{group}} = PRNG\left(ID_{group} \oplus r_{T_A} \oplus PRNG(K_{group}) \oplus PRNG(t_i)\right) \quad (۲۳)$$

اگر رابطه $M^{1*}_{group} = M^1_{group}$ برقرار باشد، برچسب دوم مقادیر تصادفی $\{r_{t_B}, r'_{t_B}\}$ را تولید کرده و مقادیر زیر را محاسبه می‌کند:

$$M^2_{group} = PRNG(ID_{group} \oplus r_{T_B} \oplus PRNG(K_{group}) \oplus PRNG(M^1_{group})) \quad (۲۴)$$

$$M_{T_B} = PRNG(ID_{T_B} \oplus r'_{T_B} \oplus PRNG(K_{T_B}) \oplus PRNG(M_{T_A})) \quad (۲۵)$$

و درنهایت مجموعه $\{r_{T_B}, r'_{T_B}, M^2_{group}, M_{T_B}\}$ را برای قرائت‌گر می‌فرستد. قرائت‌گر مقدار $r'_{T_B}$ را ذخیره کرده و مجموعه $\{r_{T_B}, M^2_{group}, M_{T_B}\}$ را برای برچسب نخست می‌فرستد. برچسب نخست تعلق‌داشتن برچسب دوم به گروه را بررسی می‌کند. درحقیقت برچسب نخست مقدار زیر را محاسبه می‌کند:

$$M^{2*}_{group} = PRNG\begin{pmatrix} ID_{group} \oplus r_{T_B} PRNG(k_{group}) \\ PRNG(M^1_{group}) \end{pmatrix} \quad (۲۶)$$

اگر رابطه $M^{2*}_{group} = M^2_{group}$ برقرار باشد، برچسب نخست پیغام نهایی را محاسبه کرده و نتیجه را برای قرائت‌گر می فرستد.

$$M_{\mathrm{TAB}} = \mathrm{PRNG}(\mathrm{ID_{T_A}} \oplus \mathrm{M_{T_A}} \oplus \mathrm{PRNG}(\mathrm{M_{T_B}}) \oplus \mathrm{PRNG}(\mathrm{K_{T_A}} + 1)) \quad (۲۷)$$

قرائت‌گر مدرک زیر را تولید کرده و برای بررسی‌کننده می فرستد:

$$\mathrm{e_n^{T_{Ab}}} = \{\mathrm{ID_{T_A}}, \mathrm{ID_{T_B}}, \mathrm{t_n}, \mathrm{r'_{T_A}}, \mathrm{r'_{T_B}}, \mathrm{M_{TAB}}\} \quad (۲۸)$$

## ۱۸- حملهٔ معرفی‌شده برای پروتکل کازاهایا

در سال ۲۰۱۳، باقری و همکارانش به معرفی حمله‌ای به نام بازیابی کلید مخفی برای پروتکل ارائه‌شده در بخش قبل پرداختند[۱۸]. وجود این حمله به این دلیل است که برای تابع داده‌شده مانند $PRNG(x)$ و دانستن این واقعیت که $x \in \{0,1\}^{16}$ است، حمله‌کننده می‌تواند مقدار x رابا $2^{16}$ ارزیابی از این تابع به‌دست آورد. شرح این حمله به‌صورت زیر است:

۱. حمله‌کننده یک دور از پروتکل را شنود کرده و مقادیر زیر را ذخیره می‌کند.

$$\{t_n, r_{T_a}, r'_{T_a}, M^1_{group}, M_{T_a}, r_{T_b}, r'_{T_b}, M^2_{group}, M_{T_b}, M_{ab}\}$$

۲. برای $0 \le i \le 2^{16}-1$ مراحل زیر طی می‌شود:

- $N_i = PRNG(i)$ محاسبه می‌شود.
- اگر $N_i = M_{T_a}$ باشد، در این صورت رابطهٔ زیر برقرار است.

$$i = ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1) \quad (۳۰)$$

لازم به یادآوری است‌که $r'_{T_a}$ و $t_n + 1$ به‌طور واضح در کانال منتقل می‌شود؛ بنابراین به‌راحتی می‌توان مقدار $x$ را به‌دست آورد.

$$x = ID_{T_a} \oplus PRNG(K_{T_a}) = i \oplus PRNG(t_n + 1) \oplus r'_{T_a} \quad (۳۱)$$

۳. برای $0 \le j \le 2^{16}-1$ مراحل زیر طی می‌شود:

- $N_j = PRNG(j)$ محاسبه می‌شود.
- اگر $N_j = M_{T_{ab}}$ باشد در این صورت داریم:

$$j = PRNG \left(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1)\right) \quad (۳۲)$$

چون مقدار $M_{T_a}$ و $M_{T_b}$ در کانال منتقل می‌شود، بنابراین می‌توان به‌راحتی مقدار $PRNG(M_{T_b})$ را محاسبه کرد و به‌صورت زیر هم مقدار $y$ پیدا می شود:

$$y = ID_{T_a} \oplus PRNG(K_{T_a} + 1) = j \oplus M_{T_a} \oplus PRNG(M_{T_b}) \quad (۳۳)$$

۴. مقدار w براساس رابطهٔ زیر پیدا می‌شود:

$$w = x \oplus y = PRNG(K_{T_a}) \oplus PRNG(K_{T_a} + 1) \quad (۳۴)$$

۵. برای $0 \le z < 2^{16}-1$ رابطهٔ زیر بررسی می‌شود. درصورتی‌که رابطه برقرار باشد، مقدار $z$ را به‌عنوان $k_{T_a}$ در نظر می‌گیریم و $ID_{T_a}$ از رابطه زیر به‌دست می آید:

$$w = PRNG(z) \oplus PRNG(z+1) \quad (۳۵)$$

$$ID_{T_a} = x \oplus PRNG(Z) \quad (۳۶)$$

مراحل زیر طی می‌شود: $0 \le i \le 2^{16}-1$ برای

- ابتدا مقدار $N_i = PRNG(i)$ محاسبه می شود.
- اگر $N_i = M^1_{group}$ باشد، مقدار $i$ برابر رابطهٔ زیر خواهد بود:

$$i = ID_{group} \oplus r_{T_a} \oplus PRNG(K_{group}) \oplus PRNG(t_n) \quad (۳۷)$$

لازم به یادآوری است که $r_{T_a}$ و $t_n$ و متعاقباً $PRNG(t_n)$ از طریق کانال به دست می‌آید. بنابراین خواهیم داشت:

$$ID_{group} \oplus PRNG(K_{group}) = i \oplus r_{T_a} \oplus PRNG(t_n) \quad (۳۸)$$

- مقدار $ID_{group} \oplus PRNG(K_{group})$ را می‌توان به‌عنوان اطلاعات مخفی برچسب ذخیره کرد.

براساس روابط بالا می‌توان $k_{T_a}$، $ID_{T_a}$ و $ID_{group} \oplus PRNG(K_{group})$ را به‌دست آورد. دانستن این اطلاعات برای جعل برچسب، درست‌کردن یک اثبات جعلی و ردیابی یک برچسب کافی است.

در ادامه نشان داده شده که برای یک اثبات گروهی کازاهایا در زمان $t_n$ می‌توان یک اثبات گروهی جعلی در زمان دیگری مانند $t_j$ ساخت. اگر $r''_{T_a} = r'_{T_a} \oplus PRNG(t_j + 1) \oplus PRNG(t_n + 1)$ باشد، خروجی پروتکل به‌صورت زیر خواهد بود:

$$e_j^{T_{ab}} = \{ID_{t_a}, ID_{T_b}, t_j, r''_{T_a}, r'_{T_b}, M_{T_{ab}}\} \quad (۳۹)$$

در ادامه نشان می‌دهیم که چگونه اثبات جعلی ساختگی توسط بررسی‌کننده مورد تایید قرار می‌گیرد:

- $M'_{T_a} = PRNG(ID_{T_a} \oplus r''_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_j + 1) = PRNG(ID_{T_a} \oplus r'_{T_A} \oplus PRNG(t_n + 1) \oplus PRNG(t_j + 1) \oplus PRNG(K_{T_a}) \oplus PRNG(t_j + 1) = PRNG(ID_{T_a} \oplus r'_{T_a} \oplus PRNG(K_{T_a}) \oplus PRNG(t_n + 1) = M_{T_a}$ (۴۰)

- $M'_{T_b} = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(k_{T_b}) \oplus PRNG(M'_{T_a})) = PRNG(ID_{T_b} \oplus r'_{T_b} \oplus PRNG(K_{T_b}) \oplus PRNG(M_{T_a}) = M_{T_b}$ (۴۱)

- $M'_{T_{ab}} = PRNG\left(ID_{T_a} \oplus M'_{T_a} \oplus PRNG(M'_{T_b}) \oplus PRNG(K_{T_a} + 1)\right) = PRNG\left(ID_{T_a} \oplus M_{T_a} \oplus PRNG(M_{T_b}) \oplus PRNG(K_{T_a} + 1)\right)$ (۴۲)

پیچیدگی این حمله فقط شنودکردن یک دور موفقیت‌آمیز از پروتکل است. توسط این اثبات جعلی بیمار می‌تواند مدعی شود که هیچ دارویی از پرستار دریافت نکرده است و یا پرستار می‌تواند برای زمان دیگری غیر از $t_n$ اثبات تولید کند.

## ۱۹- سامانه‌های RFID در امنیت دارویی بیماران

در این بخش راه حلی برای افزایش امنیت دارویی بیماران معرفی شده است[۱]. با وجود این‌که بارکد در سامانه‌های بیمارستانی کاربرد دارد، اما استفاده از برچسب‌های RFID به‌دلیل امتیازات بالای آن حائز اهمیت است. در سامانهٔ پیشنهادی برچسب‌های RFID به بیماران و بسته‌های

دارویی متصل شده‌اند. قرائت‌گرهای RFID با استفاده از شناسه ایستای هر برچسب می‌توانند اطلاعات مربوط به بیمار یا دارو را به‌دست بیاورند. به همین دلیل قرائت‌گر باید از طریق یک کانال مطمئن به پایگاه داده متصل باشند. در این پروتکل برچسب‌ها یک رمزعبور ۳۲ بیتی دارند که از نوع غیرفعال هستند و تابع PRNG را پشتیبانی می‌کنند. در این بخش توضیح خواهیم داد که سامانهٔ پیشنهادی چگونه عمل می‌کند:

ابتدا پزشک بیمار را معاینه می‌کند. او با استفاده از یک دستیار دیجیتال[1] که شامل یک قرائت‌گر است اطلاعات اطلاعات موجود بر روی دستبند بیمار را می‌خواند؛ از این طریق پزشک از شناسهٔ ایستای بیمار($\text{inpatient}_i$) آگاه می‌شود. پزشک بعد از معاینه بیمار، داروهای مورد نیاز را تجویز می‌کند؛ و پس از معاینه تمام بیماران از طریق دستیار دیجیتال خود نسخه‌های تجویزشده را به سامانهٔ اطلاعات بیمارستان[2] تحویل می‌دهد؛ سپس این سامانه داروخانه را برای بسته‌بندی این داروها مطلع می‌کند. در داروخانه، نسخه‌پیچ خودکار داروها[3]، داروها را آماده می‌کند. هنگام آماده‌کردن داروها و قراردادن آن‌ها در پاکت‌های پلاستیکی یک اثبات گروهی مبنی بر این‌که یک سری از داروها به‌طور هم‌زمان دریک پاکت قرار داده شده‌اند تولید می‌شود؛ سپس نسخه‌پیچ یک شناسهٔ $U_{D_i}$ برای داروها تهیه کرده و توسط یک برچسب به آن‌ها متصل می‌کند؛ سپس نسخه‌پیچ سامانهٔ اطلاعات بیمارستان را مطلع کرده و شناساگر $U_{D_i}$ در این سامانه ثبت می‌شود؛ در این مرحله سامانهٔ اطلاعات بیمارستان، اطلاعات زیر را در مورد بیمار دارد:

| | | |
|---|---|---|
| $\text{Inpatient}_i$ | $\text{UD}_i$ | Additional_Information |

در قسمت اطلاعات اضافی، ممکن است اطلاعاتی مانند فاصلهٔ زمانی بین مصرف داروها ذکر شود.

**ایستگاه پرستاری:** در این ایستگاه، پرستار بسته‌های دارویی را دریافت کرده و مجموعه $\{UD_i, t_i, Additional\_information_i\}$ را برای N بیمار دریافت می‌کند. عنصر سوم $t_i$ نشان‌دهنده برچسب زمانی است که تنها در یک پنجرهٔ زمانی معتبر است. به این معنا که باید در یک پنجرهٔ زمانی دارو به بیمار داده شود. درنهایت پرستار این اطلاعات را به دستیار دیجیتال خود منتقل کرده و چرخهٔ دارودادن به بیماران شروع می‌شود. مرحله آخر از اهمیت بالایی دارد؛ زیرا به‌جزء این مرحله، مراحل بعدیبرون‌خط هستند و به‌عنوان نیازی به برقراری ارتباط با سامانه اطلاعاتی بیمارستان نیست.اطلاعات زیر در دستیار دیجیتال پرستار ثبت می شود:

| | | | | |
|---|---|---|---|---|
| $\text{Inpatient}_1$ | $\text{UD}_1$ | $t_1$ | ... | $\text{Additional\_information}_1$ |
| $\text{Inpatient}_i$ | $\text{UD}_i$ | $t_i$ | ... | $\text{Additional\_information}_i$ |
| $\text{Inpatient}_N$ | $\text{UD}_N$ | $t_N$ | ... | $\text{Additional\_information}_N$ |

**خوراندن دارو به بیماران:** پرستار به همراه بسته‌های دارویی برای دارودادن به بیماران به آن‌ها سر می‌زند. این مرحله خود دارای دو بخش است:

A: مرحلهٔ اثبات. در این مرحله هدف اطمینان از متناظربودن داروها با بیماران است.

B: مدرکی مبنی بر این‌که داروها به بیماران داده شده است توسط پرستار تولید می‌شود.

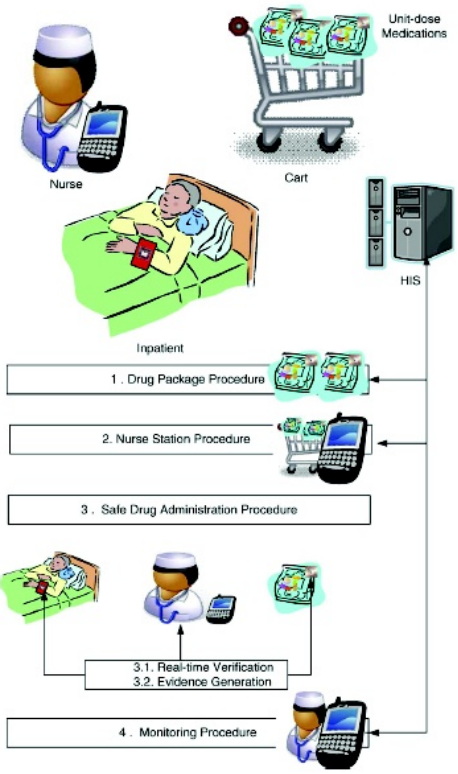
مراحل دقیق این دو چرخه در زیر آمده است:

1.A قرائت‌گر مقدار تصادفی $r_p$ و پیغام درخواست را تولید می‌کند و$\{\text{request}, r_p\}$ را برای برچسبی که بر روی دست بیمار قرار دارد و برچسب داروها می‌فرستد.

2.A با دریافت این مقادیر هرکدام از برچسب‌ها یک شناسهٔ مستعار را محاسبه کرده و برای قرائت‌گر می‌فرستد.$r_W$ و $r_M$ مقادیر تصادفی تولیدشده توسط برچسب بیمار و برچسب دارو هستند.برچسب مربوط به بیمار مجموعه $\{r_W, \text{PRNG}(\text{inpatient}_i, r_p, r_W)\}$ را تولید کرده و برای قرائت‌گر می‌فرستد.برچسب دارو نیز مجموعه $\{r_M, PRNG(UD_i, r_p, r_M)\}$ را ارسال می‌کند.

3.A جستجو بر روی مقادیر ذخیره‌شده در دستیار دیجیتال پرستار شروع می‌شود. مقادیر $(\text{inpatient}_1, \text{UD}_1)$ گرفته شده و مقادیر $\text{PRNG}(\text{inpatient}_1, r_p, r_W)$ و $\text{RNG}(\text{UD}_1, r_p, r_M)$ محاسبه می‌شوند. اگر این مقادیر با مقادیر به‌دست آمده از برچسب‌ها برابر باشند، یک پیغام تأیید بر روی صفحه دستیار دیجیتال ظاهر شده و پرستار با اطمینان می‌تواند دارو را به بیمار بدهد. در غیر این‌صورت پرستار همین محاسبات را با $(\text{inpatient}_2, \text{UD}_2)$ انجام داده و تا جایی که تطابق را بین مقادیری که خودش محاسبه می‌کند و مقادیری که از برچسب‌ها دریافت کرده پیدا کند، روند را ادامه می‌دهد. اگر تطابقی پیدا نشد پرستار هیچ دارویی به بیمار نمی‌دهد.

[1]Personal Digital Assistant
[2]Hospital information system
[3]Automatic medication dispenser




**شکل ۱۹: بخش نخست سامانه امنیت دارویی بیماران[۱]**

1.B قرائت‌گر مربوط به پرستار با فرستادن مقدار $t_i$، برچسب مربوط به بیمار را به چالش می‌کشد.

2.B این برچسب مقدار تصادفی $r'_W$ را تولید کرده و مقدار زیر را محاسبه می‌کند. سپس زوج $\{r'_W, m_T\}$ را برای قرائت‌گر می‌فرستد. در رابطهٔ زیر $k_{inpatient_i}$ کلید مخصوص برچسب بیمار است.

(۴۳)

$$m_T = PRNG\left(inpatient_i \oplus r'_W \oplus PRNG(t_i) \oplus PRNG\left(K_{inpatient_i}\right)\right)$$

3.B قرائت‌گر مقدار $r'_W$ را ذخیره کرده و مقدار $m_T$ را برای برچسب دارو می‌فرستد.

4.B برچسب بیمار یک مقدار تصادفی $r'_M$ را تولید کرده و مقدار زیر را محاسبه می‌کند؛ سپس زوج $\{r'_M, m_{UD}\}$ را برای قرائت‌گر می‌فرستد. در رابطهٔ زیر $k_{UD_i}$ برچسب مربوط به دارو است.

(۴۴)

$$m_{UD} = PRNG\left(UD_i \oplus r'_M \oplus PRNG(m_T) \oplus K_{UD_i}\right)$$

5.B قرائت‌گر مقدار $r'_M$ را ذخیره کرده و $m_{UD}$ را برای برچسب بیمار می‌فرستد.

6.B برچسب بیمار، مقدار زیر را محاسبه کرده و نتیجه را برای قرائت‌گر ارسال می‌کند.

(۴۵)

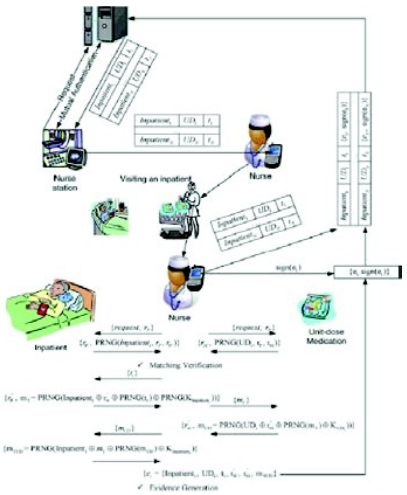
$$m_{TUD} = PRNG\left(inpatient_i \oplus m_T \oplus PRNG(m_{UD}) \oplus k_{inpatient_i}\right)$$

7.B قرائت‌گرمربوط به پرستار مدرک اثبات گروهی $e_i = \{inpatient_i, UD_i, t_i, r'_W, r'_M, m_{TUD}\}$ را تولید می‌کند.

8.B علاوه‌بر مدرک تولیدشده در قسمت قبل به امضای دیجیتالی پرستار $(sign(e_i))$ نیز نیازمند هستیم تا زوج $\{e_i, sign(e_i)\}$ در دستیار دیجیتال مربوط به پرستار ذخیره شود.

## ۲۰- تحلیل امنیتی سامانه امنیت دارویی ارائه‌شده

در سال ۲۰۱۳، چن[1] و همکارانش به تحلیل امنیتی این سامانه پرداختند[۱۹]. می‌دانیم پرستار تطابق بین دارو و بیمار را از طریق شناساگرهای آن‌ها یعنی $(inpatient_i, UD_i)$ انجام می‌دهد. اگرچه داروی بیمار ممکن است جای‌گزین شده و یک رونوشت از $UD_i$ بر روی آن چسبانده شود. اگرچه سامانهٔ اطلاعات بیمارستان می‌تواند این خطا را آشکار کند؛ اما ممکن است این کار بسیار دیر باشد و امنیت دارویی بیماران را به خطر بیندازد.



**شکل ۲۰: بخش دوم سامانه امنیت دارویی بیماران[۱]**

[1] Chen

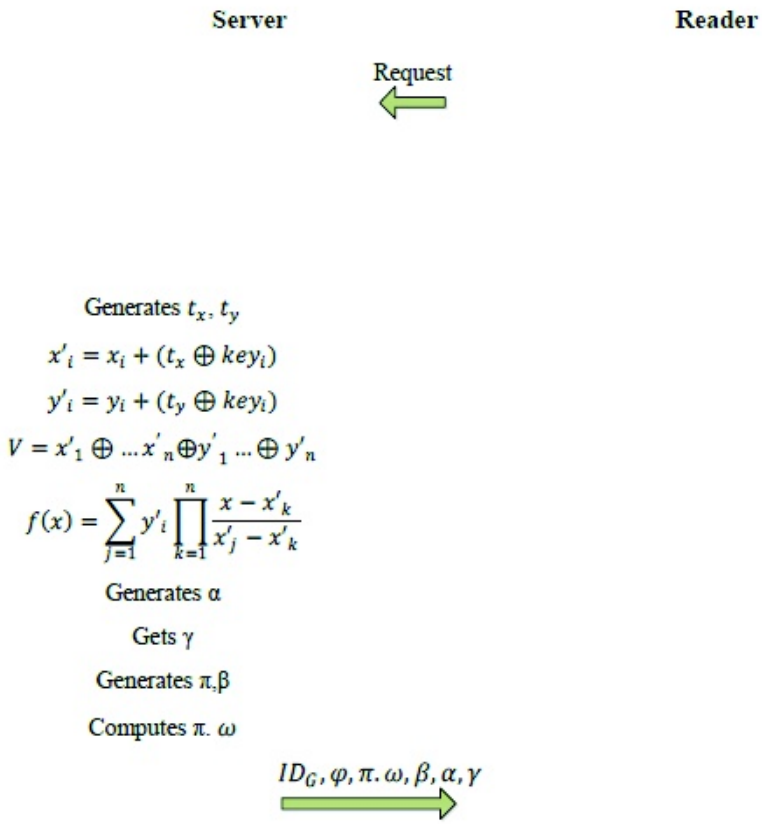
## ۲۱- پروتکل ارائه‌شده برای افزایش امنیت دارویی بیماران توسط چن

در این بخش به معرفی پروتکل ارائه‌شده توسط چن در سال ۲۰۱۳ می‌پردازیم[۱۹]. این پروتکل دارای سه بخش است: ۱) مرحلهٔ نخستین. ۲) تولید اثبات بلادرنگ. ۳) مرحلهٔ بازبینی و تایید اثبات. شرح کامل این پروتکل به‌صورت زیر است:

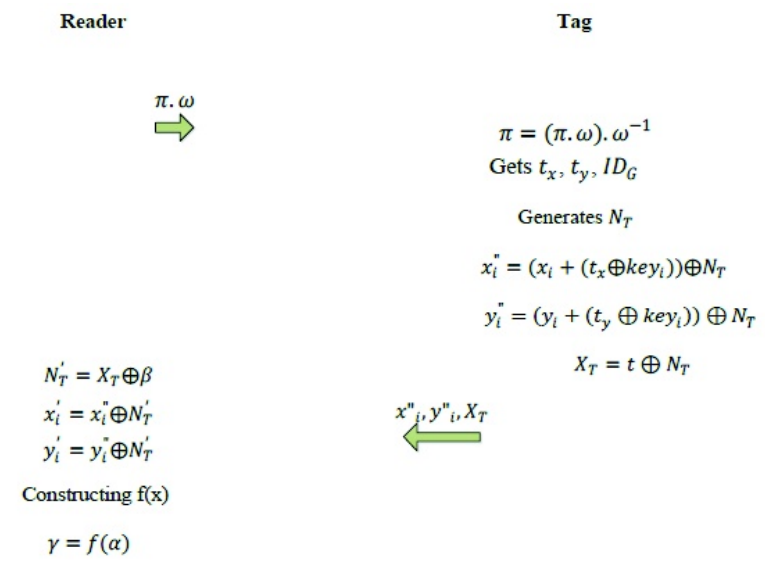
الف) مرحلهٔ نخستین: در این مرحله بین پایگاه داده و قرائت‌گر یک احراز هویت صورت می‌گیرد و به قرائت‌گر اجازهٔ تولید اثبات گروهی برای گروهی از برچسب‌ها داده می‌شود.

گام نخست: ابتدا قرائت‌گر ارتباط نخستین را با پایگاه داده برقرار می‌کند.

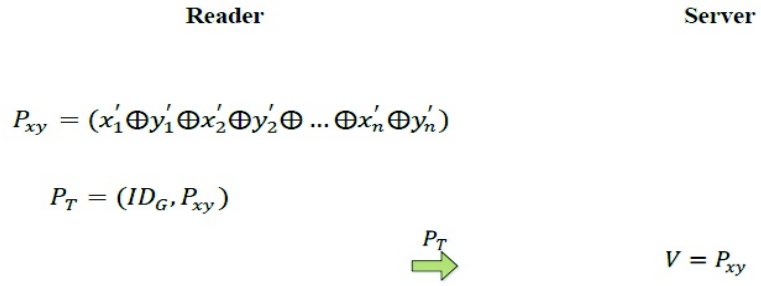
گام دوم: پایگاه داده یک مختصات مخفی برای برچسب i اُم به‌صورت $(x_i, y_i)$ذخیره کرده است. توسط این مختصات بررسی‌کننده یک تابع چندجمله‌ای پیش‌بینی شده را تولید می‌کند. لازم به ذکر است که هر برچسب یک کلید مخفی به نام $key_i$ دارد که در پایگاه داده نیز ذخیره شده است. ابتدا دو متغیر تصادفی $t_x$ و $t_y$ تولید می‌شود. براساس کلید و مختصات مخفی هر برچسب مقادیر زیر تولید می‌شوند:




**شکل ۲۱ : مرحله نخست پروتکل ارائه‌شده[ ۱۹]**




شکل ۲۲: مرحله دوم پروتکل ارائه‌شده[ ۱۹]




**شکل ۲۳: مرحله سوم پروتکل ارائه‌شده[ ۱۹]**

$$x_i' = x_i + (t_x \oplus key_i) \quad (۴۴)$$

$$y_i' = y_i + (t_y \oplus key_i) \quad (۴۵)$$

در این مرحله از تولید پیام، تأیید $V = x_1' \oplus x_2' \oplus \dots \oplus x_n' \oplus y_1' \oplus y_2' \oplus \dots \oplus y_n'$ محاسبه می‌شود. سپس تابع چندجمله‌ای پیش‌بینی‌شده $f(x)$ به روش درون‌یابی لاگرانژ محاسبه می‌شود:

$$f(x) = \sum_{j=1}^{n} y_i' \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k'}{x_j' - x_k'} \quad (۴۶)$$

با توجه به تابع به‌دست آمده، ابتدا مقدار $\alpha$ به‌صورت تصادفی تولیدشده و در رابطه بالا قرار داده می‌شود تا مقدار $\gamma$ به دست آید؛ سپس زوج $(\alpha, \gamma)$برای قرائت‌گر ارسال می‌شود. برای این‌که مقادیر $t_x$ و $t_y$ به‌صورت امن در کانال منتقل شود روند زیر توسط بررسی‌کننده انجام می‌شود: حالات مختلف و نامتناهی وجود دارد که دو خط متقاطع از نقطه $(t_x, t_y)$ عبور می‌کنند. بررسی‌کننده دو خط از این تعداد نامتناهی را انتخاب کرده و به‌طور تصادفی دو نقطه بر روی هر کدام از آن‌ها انتخاب می‌کند. مراحل این انتخاب به این صورت است که خط نخست توسط دو جفت $(t_x, t_y)$ و $(t1_x, t1_y)$ انتخاب می‌شود؛ سپس نقطهٔ دوم $(t2_x, t2_y)$ به‌صورت تصادفی بر روی این خط انتخاب می‌شود. به روش مشابه مقادیر $(t3_x, t4_y)$ و $(t4_x, t4_y)$ انتخاب می‌شوند. این مقادیر در ماتریس $\begin{bmatrix} t1_x & t1_y & t2_x \\ t2_y & t3_x & t3_y \\ t4_x & t4_y & ID_g \end{bmatrix}$ ذخیره شده و پیغام مخفی $\beta$ محاسبه می‌شود.

$$\beta = (t1_x \oplus t1_y \oplus t2_x \oplus t2_y \oplus t3_x \oplus t3_y \oplus t4_x \oplus t4_y) \quad (۴۷)$$

برای اهداف امنیتی ضرب ماتریسی $\pi.\omega$ منتقل می‌شود. در این رابطه $\omega$ ماتریس همانی است. درنهایت بررسی‌کننده مجموعه $(ID_G, \varphi, \pi.\omega, \beta, \alpha, \gamma)$ را برای قرائت‌گر می‌فرستد. در این رابطه $\varphi$ تعداد برچسب‌های گروه است. مرحلهٔ نخستیه این پروتکل در زیر آمده است:

ب) مرحله اثبات بدون درنگ اثبات گروهی: در این مرحله قرائت‌گر چالشی را برای تمامی برچسب‌ها می‌فرستد که تنها برچسب‌های متعلق به یک گروه پاسخ می‌دهند.

گام نخست: ماتریس $\pi.\omega$ برای برچسب‌ها فرستاده می‌شود.

گام دوم: برچسب‌ها ماتریس $\pi$ را از رابطه $\pi = (\pi.\omega).\omega^{-1}$ محاسبه می‌کنند؛ سپس زوج‌های $(t1_x, t2_x)$، $(t2_x, t2_y)$، $(t3_x, t3_y)$ و $(t4_x, t4_y)$ به همراه $ID_G$ استخراج می شود. با این نقاط می‌توان دو خط متقاطع رسم کرد و نقطه $(t_x, t_y)$ را به‌دست آورد؛ سپس پیغام $t$ به‌صورت زیر محاسبه می‌شود:

(۴۸)

$$t = t1_x \oplus t1_y \oplus t2_x \oplus t2_y \oplus t3_x \oplus t3_y \oplus t4_x \oplus t4_y)$$

برچسب متعلق به گروه با شناساگر $ID_G$ باید با تولید یک مدرک؛ قرائت‌گر را از این‌که توانسته $(t_x, t_y)$ را پیدا کند، آگاه سازد. بنابراین یک مقدار تصادفی مانند $N_T$را تولید و مختصات زیر را محاسبه می‌کند:

$$x_i^{"} = \left(x_i + (t_x \oplus key_i)\right) \oplus N_T \quad (۴۹)$$

$$y_i^{"} = \left(y_i + (t_y \oplus key_i)\right) \oplus N_T \quad (۵۰)$$

برای حفظ امنیت پروتکل مقدار $N_T$ در پیام $t$ به‌صورت $X_T = t \oplus N_T$ جاسازی شده است؛ سپس پیام $X_T$ به همراه مختصات $(x_i^{"}, y_i^{"})$ برای قرائت‌گر فرستاده می‌شود.

گام سوم: قرائت‌گر براساس پیغام $X_T$ و $\beta$ مقدار $N_T' = X_T \oplus \beta$ را محاسبه کرده و مختصات $(x_i', y_i')$ به‌صورت زیر محاسبه می‌شود.

$$y_i' = y_i^{"} \oplus N_T' \quad (۵۱)$$

$$x_i' = x_i^{"} \oplus N_T' \quad (۵۲)$$

وقتی قرائت‌گر زوج‌های $(x_i', y_i')$ را از تمامی برچسب‌ها دریافت می‌کند، تعداد برچسب‌ها را مورد ارزیابی قرار می‌دهد؛ سپس تابع چندجمله‌ای $f(x)$ را به‌صورت زیر محاسبه می‌کند:

$$f(x) = \sum_{j=1}^{n} y_i' \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k'}{x_j' - x_k'} \quad (۵۳)$$

با استفاده از تابع به دست آمده، قرائت‌گر $\gamma = f(\alpha)$ را به‌دست می‌آورد. اگر این مقدار با مقداری که خود قرائت‌گر در مرحلهٔ قبل به‌دست آورده، یکی باشد، در این صورت وجود هم‌زمان برچسب‌ها ثابت می‌شود؛ در غیر این صورت یک پیغام خطا از سوی قرائت‌گر فرستاده می‌شود تا بسته‌بندی دارو متوقف شود. مراحل انجام‌شده در این مرحله در شکل ۲۲ آورده شده است.

پ) مرحلهٔ بازنگری اثبات: بعد از این‌که وجود هم‌زمان برچسب‌ها از سوی قرائت‌گر تأیید شد، مدرک اثبات وجود هم‌زمان برچسب‌ها باید تولیدشده و از سوی قرائت‌گر برای کنترل‌کننده ارسال شود.

گام نخست: اثبات جمعی $P_T$ تولیدشده و برای کنترل‌کننده فرستاده می‌شود.

$$P_{xy} = (x_1' \oplus y_1' \oplus x_2' \oplus y_2' \oplus \quad \dots \oplus x_n' \oplus y_n')(ID_G, P_{xy}) \quad (۵۴)$$

$$P_T = (ID_G, P_{xy}) \quad (۵۵)$$

گام دوم: بررسی‌کننده درستی رابطهٔ $V = P_{xy}$ را چک می‌کند. مراحل انجام‌شده در این مرحله در شکل ۲۳ آورده شده است.

## ۲۳- نتیجه‌گیری و کارهای آینده

این مقاله، به بررسی کلی پروتکل‌های اثبات گروهی پرداخته است و با نگاهی جامع و دقیق به این پروتکل‌ها چالش‌های امنیتی آن‌ها را بیان کرده است.

در زمینه اثبات گروهی، ابتدا ایده تولید اثبات گروهی برای دو برچسب مطرح شد و به‌دنبال آن مشکلات موجود بر روی این پروتکل مورد بررسی قرار گرفت. در ادامه، اثبات گروهی برای چند برچسب معرفی شد که مورد توجه بسیاری از پژوهش‌گران قرار گرفت و هر کدام از آن‌ها به معرفی پروتکلی در این زمینه پرداختند. در بیش‌تر این پروتکل‌ها حملهٔ غالب، حملهٔ تکرار است. برای افزایش امنیت این پروتکل‌ها استفاده از برچسب‌های زمانی مطرح شد. در طراحی پروتکل‌های بعدی برای جلوگیری از این‌که دشمن بتواند برچسب‌های زمانی جعلی تولید کند، بررسی‌کننده مقدار رمزشدهٔ برچسب زمانی را برای قرائت‌گر می‌فرستاد. درنهایت اثبات‌های گروهی جای خود را در سامانه‌های امنیت دارویی بیماران باز کردند. وجود خطاهای دارویی می‌تواند جان بیماران را به خطر اندازد. برای جلوگیری از این خطرات پروتکل‌های اثبات گروهی وارد این

سامانه‌ها شدند که متأسفانه هر کدام از آن‌ها در مقابل یک یا چند حملهٔ آسیب‌پذیرهستند.

در بسیاری از پروتکل‌های اثبات گروهی موجود از تابع کد افزونگی چرخشی یا CRC استفاده شده که دارای خاصیت خطی است که این خاصیت امکان حملهٔ جعل هویت را امکان‌پذیر می‌سازد. بنابراین برای طراحی یک پروتکل امن می‌توان از توابع دیگری به‌عنوان مثال توابع حساب پیمانه‌ای که خاصیت غیر خطی دارند، استفاده کرد. هم‌چنین برای امنیت بیش‌تر در پروتکل‌هایی که در آن‌ها از تابع مولد اعداد شبه‌تصادفی استفاده شده، بهتر است از توابع مولد اعداد شبه‌تصادفی ۶۴ بیتی یا بیش‌تر استفاده کرد تا امکان ارزیابی برون‌خط این تابع و پیداکردن اطلاعات مخفی برچسب وجود نداشته باشد.

امید است در آینده نیز همانند گذشته پژوهش‌گران این دسته از مشکلات را در نظر داشته باشند و به طراحی یک پروتکل اثبات گروهی جامع و امن بپردازند.

## ۲۴- مراجع


[1] P. Peris-Lopez, A. Agustin, J. Hernandez-Castro, and J. Van der Lubbe.Flawson RFID Grouping-Proofs.Guidelines for Future Sound Protocols.In *Journal of Network andComputer Applications*. 34(3): 833-845, May 2011.

[2] C. Ma, J. Lin, Y. Wang, M. Shang. Offline RFID Grouping Proofs with Trusted Timestamps. In *11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Pages 674-681, Liverpool, June 2012.

[3] A. Jules. Yoking-proofs for RFID tags. In *Proceedings of the Second IEEE Annual Conference onPervasive Computing and Communications Workshops*, Pages 138-143, March 2004.

[4] D. Moriyama.A Provably Secure Offline RFID Yoking-Proof Protocol with Anonymity.In *Lightweight Cryptography for Security and Privacy*, volume 8898, pages 155-167, March 2015.

[5] J. Saito, K. Sakurai. Grouping proof for RFID tags. In *19th International Conference on Advanced Information Networking and Applications*, Volume 2, Pages 621-624,Taiwan,March 2005.

[6] S. Piramuthu.On Existence Proofs for Multiple RFID Tags. In *International Conference on Pervasive Services*, pages 317-320, Lyon, June 2006.

[7] J. Shen, H. Tan, Y. Wang, S. Ji and J. Wang.An Enhanced Grouping Proof for Multiple RFID Readers and Tag Groups.In *International Journal of Control and Automation*. 12(7): 239-246, 2014.

[8] P. Peris-Lopez, J.C Hernandez-Castro, J.M Estevez-Tapiador, and A. Ribagorda.Solving the simultaneous scanning problem anonymously: clumping proofs for RFID tags. In *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*. Pages 55-60, Istanbul, July 2007.

[9] C.C. Lin, Y.C Lai, J.D Tygar, C.K Yang, and C.L Chiang. Coexistence Proof Using Chain of Timestamps for Multiple RFID Tags. In *Advances in Web and Network Technologies, and Information ManagementLecture Notes in Computer Science*, volume 4537, pages 634-643, 2007.

[10] M. Lata, A. Kumar. Survey on Lightweight Primitives and Protocols for RFID in Wireless Sensor Networks. In *International Journal of Communication NetworksandInformation Security*, 6(1): 26-43, April 2014.

[11] J. M. de Fuentes, P. Peris-Lopez, J. E. Tapiador, and S. Pastrana.Probabilistic yoking proofs for large scale IOT systems. In *Ad Hoc Networks*, January 2015.

[12] M. Burmester, M. Ramos. Provably secure grouping-proofs for RFID Tags. In *proceeding of 8th smart card research and advance applications CARDIS*, Pages 176-190, 2008.

[13] H.Y. Chien, S.B. Liu. Tree based RFID yoking proof.In *International conference on network security, wireless communication* and trusted computing, volume 1, Pages 550-553, Wuhan, April 2009.

[14] Y.U. Yaochang, H.S.U. Jenming, and H.O.U. Tingwei.A Heterogeneous RFID System to Improve Inpatient Medication Safety. In Journal of Computational Information Systems, 11(1): 177-184, 2015.

[15] H.H. Huang, C.Y. Ku. A RFID grouping-proofprotocol for medication safety of inpatient. In Journal of Medical Systems, 33(6): 467–74, December 2009.

[16] C. Jin, C. XU, X. Zhang, and J. Zhao.A Secure RFID Mutual Authentication Protocol for Healthcare Environments Using Elliptic Curve Cryptography.In Journal of Medical Systems, 39(3), February 2015.

[17] H.Y. Chien, C.C. Yang, T.C. Wu, and C.F. Lee.Two RFID-based solutions to enhance inpatient medication safety. In Journal of Medical Systems, 35(3): 369-375, June 2011.

[18] M. Safkhani, N. Bagheri, and M. Naderi.Secret Disclosure attack on Kazahaya, a Yoking-Proof For Low-Cost RFID Tags.In Cryptology ePrint Archive, Report 2013/453, Pages 1-7, 2013.

[19] Y.Y.Chen, M.L Tsai. An RFID solution for enhancing inpatient medication safety with real time verifiable grouping proof. In Medical Informatics,83(1): 70-81, January 2014.

**نصور باقری** هم‌اکنون به‌عنوان استادیار در دانشکده مهندسی برق دانشگاه تربیت دبیر شهید رجایی فعالیت می‌کند. ایشان کارشناسی خود را در رشتهٔ مهندسی برق از دانشگاه مازندران و کارشناسی ارشد و دکترای خود را در همین رشته از دانشگاه علم و صنعت دریافت کرده است. علایق پژوهشی ایشان شامل تحلیل و طراحی طرح‌های رمزنگاری متقارن، فناوریRFID و پروتکل‌های امنیتی است.

**سارا مجیدی** مدرک‌های کارشناسی و کارشناسی ارشد خود را به‌ترتیب از دانشگاه گیلان، رشت، ایران و دانشگاه تربیت دبیری شهید رجایی، تهران، ایران در سال‌های ۹۰ و ۹۴ دریافت کرده است. زمینهٔ پژوهشی وی سامانه‌های شناسایی به‌وسیلهٔ امواج رادیویی و بررسی و تحلیل امنیتی انواع پروتکل‌های اثبات گروهی در سامانه‌های RFID است.