چهارمین کنفرانس ملی ایده های نوین مهندسی برق
۲۰و۲۱ آبان ماه ۱۳۹۴ - دانشگاه آزاد اسلامی واحد اصفهان (خوراسگان)

تحلیل و بررسی پروتکل سبک وزن HB و تعدیل آن

محسن شریفی ۱، سید مهدی سجادیه ۲
۱کارشناسی ارشد دانشگاه آزاد اسلامی واحد خوراسگان، Mohsen mola1370@yahoo.com
۲ استادیار دانشگاه آزاد اسلامی واحد خوراسگان

چکیده

بهره گیری از پروتکل های احراز هویت سبک وزن، به لحاظ محدودیت های ناشی از سطوح برچسب زنی (تگ)، به عنوان امری ضروری در کاربردهای RFID تلقی می شود. در طول چند سال اخیر، چندین نوع از اینگونه پروتکل ها ارائه شده است و مورد آنالیز قرار گرفته است. در این مقاله، بر روی پروتکل HB و نسخه های گوناگون آن، متمرکز می‌شویم. در اینجا، آسیب پذیری برخی از این پروتکل ها را نسبت به حملات مبتنی بر تگ ها (که در آن، عامل متخاصم، خود را به عنوان یک سیستم اطلاعـات خوان معتبر وانمود می کند)، نشان می‌دهیم، و یک پروتکل تعدیل و اصلاح شده را به منظور اجتناب از این نوع حملات، ارائه می‌کنیم.

کلید واژه: RFID_ HB

۱- مقدمه

در آینده بسیار نزدیک، تگ های RFID، آماده اند تا جایگزین بارکدها شوند. مزیت های تگ های RFID در مقایسه با بارکدها، بسیار زیاد است که شامل موارد زیر می شود: ظرفیت آنها در خصوص ذخیره سازی اطلاعاتِ بیشتر؛ سهولت این که می‌توان اطلاعات این تگ ها را بدون نیاز به برخورداری از خط دید مستقیم، خواند.

مانع اصلی در استفاده گسترده از این تگ ها، هزینه آنها می‌باشد. همچنین موضوعات محرمانگی و امنیت، نقش عمده ای را در موفقیت پیاده سازی تگ RFID (به لحاظ اینکه عامل متخاصم می تواند اشیاء برخوردار از اینگونه تگ ها را به سهولت شناسایی و یا مورد حمله قرار دهد)، ایفا می کند. به هنگام مواجهه با موضوعات محرمانگی یا امنیتی در پیاده سازی های تگ RFID، محدودیت های ناشی از فضای حافظه و قدرت پردازش آنها، موضوع بهره گیری از پروتکل های احراز هویت سبک وزن را اجبار می سازد. تعدادی از محققین، پروتکل هایی را ارائه و ارزیابی کرده اند که التزام سبک وزن بودن را برآورده می سازد و در آن واحد، تا حد معقولی، امن می باشد.

یک مورد از اینگونه پروتکل ها، پروتکل HB می باشد که توسط هاپر و بلوم ، ارائه شده است. با وجودی که این پروتکل می‌تواند تحت بسیاری از شرایط (که در آن عامل های متخاصم غیر فعال وجود دارد)، به خوبی کار کند، یک عامل متخاصم فعال می تواند امنیت آن را با شکست مواجه سازد. ژولز و ویس، پروتکل HB را به منظور حفاظت در برابر حملات فعال ناشی از عامل های متخاصم، تعدیل و اصلاح نمودند. همچنین، این پروتکلِ تعدیل و اصلاح شده (HB^{+})، نیز به طور کامل برای شرایط خاص، امن نشده است . پس از آن، برینگر و همکارانش، پروتکل HB^{+} را به منظور امن سازی آن در برابر حملات فعال ناشی از عامل های متخاصم، تعدیل و اصلاح نمودند. در اینجا نشان خواهیم داد که پروتکل های HB^{++} و HB^{++} (اولین نسخه)، که توسط برینگر و همکارانش ارائه شده است، در برابر

حملات فعال، آسیب پذیر می باشد و یک نسخه تعدیل و اصلاح شده از آن را ارائه می کنیم.

این مقاله به صورت زیر سازماندهی شده است: در بخش بعدی، مروری اجمالی از پروتکل HB و نسخه های گوناگون آن، ارائه شده است. این بخش، با ارائه پیشنهاداتی در خصوص تعدیل و اصلاح نسخه اخیرا ارائه شده پروتکل HB (یعنی HB^{++})، دنبال شده است. در بخش ۳، آنالیز امنیتی مختصری در خصوص اصلاحات پیشنهادی، صورت پذیرفته است و در بخش ۴، نتیجه گیری مقاله ارائه شده است.

۲- پروتکل HB و نسخه های گوناگون آن

در این بخش، پروتکل HB و نسخه های گوناگون آن را به طور خلاصه تشریح و ارزیابی می کنیم. پس از معرفی مختصر این پروتکل ها، برخی از موارد نقض امنیتی را که ممکن است رخ دهد، مدنظر قرار می دهیم و در خصوص راهکارهای ارائه شده در مقالات، بحث خواهیم نمود. سپس، پیشنهاداتی را در خصوص تعدیل و اصلاح پروتکل HB^{++} (نسخه اخیر پروتکل HB)، ارائه می کنیم.

در این مقاله، از نمادگذاری های زیر استفاده شده است:

- پارامترهای a و b: بردارهای باینری تصادفی k بیتی
- پارامترهای x، x'، y، y': بردارهای کلید رمز k بیتی
- پارامتر ν: بیت نویز (برابر با ۱، با احتمال $\eta \in [0,1/2]$)

۲-۱ پروتکل HB

در تصویر ۱، یک چشم انداز کلی از دور تبادل اطلاعات مربوط به پروتکل HB، ارائه شده است. در اینجا، عبارات $(a \cdot x)$ و $(a \oplus x)$، به ترتیب بیانگر ضرب اسکالر و عملیات منطقی XOR (مبتنی بر بردارهای باینری k بیتی a و x)، می باشد. پروتکل HB، بر میزان سختی محاسباتی مسئله LPN، تمرکز دارد و بر روی راه حل های کلاسیک رمزنگاری کلید متقارن، متمرکز نمی باشد. این بدان معنی است که این پروتکل، فقط در برابر حملات غیر فعال، امن می باشد و در برابر حملات فعال، امن نیست. دور تبادل اطلاعات ارائه شده در تصویر ۱، به تعداد r دفعه، تکرار می شود و چنانچه بررسی های مبتنی بر سمت سیستم اطلاعات خوان، حداکثر به تعداد ηr مرتبه، با شکست مواجه شود، تگ مربوطه احراز هویت می شود. یک حمله فعال ساده که در آن، یک عامل متخاصم خود را به عنوان سیستم اطلاعات خوان وانمود می کند (که یک پارامتر ثابت a را برای چندین مرتبه به سمت تگ مربوطه ارسال می‌نماید)، می‌تواند مقدار x را بازیابی کند.

Tag (secret x)		Reader (secret x)
$\nu \in \{0,1 \mid \mathcal{P}(\nu=1)=\eta\}$		
	$\xleftarrow{a}$	Generate challenge $a \in_R \{0,1\}^k$
Compute $z = a \cdot x \oplus \nu$	$\xrightarrow{z}$	Check $a \cdot x \approx z$

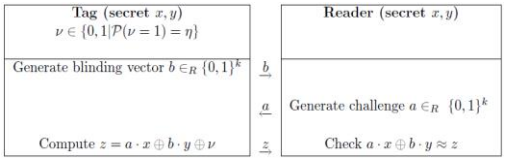
تصویر ۱ – دور تبادل اطلاعات مربوط به پروتکل HB

۲-۲ پروتکل HB^+

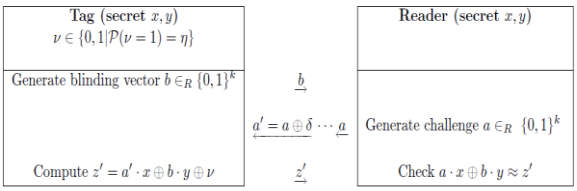
ژولز و ویس (۲۰۰۵)، پروتکل HB را تعدیل و اصلاح نمودند و نشان دادند که پروتکل اصلاح شده (HB^+)، در برابر حملات فعال، امن می باشد. دور تبادل اطلاعات مربوط به پروتکل HB^+، در تصویر ۲ ارائه شده است. آنها، کلید رمز بردار k بیتی (y) را ارائه نمودند که بین سیستم اطلاعات خوان و تگ مربوطه، به اشتراک گذاشته می شود. آنها همچنین پروتکل HB را به طریقی تعدیل و اصلاح نمودند که تگ مربوطه (و نه سیستم اطلاعات خوان)، فرایند احراز هویت را راه اندازی می نماید. ابتدا تگ مربوطه، یک بردار جعل k بتی را به سمت سیستم اطلاعات خوان، ارسال می کند. اصلاح دیگری که در این خصوص صورت گرفته است، نحوه محاسبه پارامتر z می باشد. حاصلضرب اسکالر کلید رمز جدید ارائه شده (y) و بردار جعل (b)، با پارامتر z واقع در پروتکل HB، تحت عملیات XOR قرار می گیرد.

با وجودیکه ژولز و ویس (۲۰۰۵)، نشان داده اند که پروتکل HB^+ در برابر حملات فعال، امن می باشد، گیلبر و همکارانش نشان داده اند که پروتکل HB^+ در برابر حمله man-in-the-middle، امن نمی باشد (که در نسخه قبلی آن، مدنظر قرار نگرفته است). توصیفی از این حمله، در تصویر ۳ ارائه شده است. در اینجا، فرض بر آن است که عامل متخاصم، قادر به دستکاری

کدهای چالش ارسال شده توسط یک سیستم اطلاعات خوان موثق به یک تگ موثق (در حین فرایند احراز هویت)، می باشد. همچنین، فرض بر آن است که عامل متخاصم، زمانیکه یک رَویه احراز هویت با شکست یا موفقیت مواجه می شود، قادر به تشخیص آن می باشد. عملیات اصلی این حمله، شامل دستکاری کد چالش ارسال شده توسط سیستم اطلاعات خوان (a) می باشد که این کار از طریق ارسال نتیجه عملیات XOR (مابین بردار a و بردار ثابت k بیتی δ) به سمت تگ مربوط (در تمام r دور تبادل اطلاعات مربوط به فرایند احراز هویت)، صورت می پذیرد. چنانچه فرایند احراز هویت با موفقیت مواجه شود، به احتمال زیاد، نتیجه ضرب اسکالر $\delta \cdot x$ برابر با صفر خواهد شد. چنانچه فرایند احراز هویت با شکست مواجه شود، به احتمال زیاد، نتیجه ضرب اسکالر $\delta \cdot x$ برابر با یک خواهد شد. در اینجا، می توان δ را به منظور آشکارسازی هر یک از بیت های کلید رمز x (یکی پس از دیگری)، دستکاری نمود. این پروتکل را می توان به تعداد k مرتبه تکرار نمود تا اینکه تمامی بیت های کلید رمز k، بازیابی شود.



تصویر ۲ – دور تبادل اطلاعات مربوط به پروتکل HB^+



تصویر ۳ – حمله به دور تبادل اطلاعات مربوط به پروتکل HB^+

به محض شناسایی x، عامل متخاصم می تواند تگ مربوطه را در اختیار گرفته و جعل نماید و بردار جعل مفروض b را به سمت سیستم اطلاعات خوان، ارسال نماید. در عوض، عامل متخاصم در پاسخ به سیستم اطلاعات خوان می تواند $a \cdot x$ را به سمت سیستم اطلاعات خوان، ارسال نماید. چنانچه فرایند احراز هویت با موفقیت مواجه شود، به احتمال زیاد، عامل متخاصم می داند که نتیجه ضرب اسکالر $b \cdot y$ برابر با صفر خواهد شد. از سوی دیگر، چنانچه فرایند احراز هویت با شکست مواجه شود، به احتمال زیاد، عامل متخاصم می داند که نتیجه ضرب اسکالر $b \cdot y$ برابر با یک خواهد شد. عامل متخاصم با بهره گیری از این اطلاعات، قادر به استخراج دیگر کلید رمز y خواهد بود. زمانیکه هر دو کلید رمز x و y، برای عامل متخاصم آشکار گردید، محرمانگی این تگ، با تهدید رو به رو می شود.

پروتکل HB^+، در برابر شکل دیگری از حمله انجام شده توسط عامل متخاصمی که خود را وانمود به یک سیستم معتبری می کند که اطلاعات تگ را می خواند، امن نمی باشد. در اینجا فرض بر آن است که عامل متخاصم، قادر به شنود تمامی ارتباطات مابین یک تگ و یک سیستم اطلاعات خوان می باشد و همچنین می تواند داده های ارسالی از هر گونه ارتباطات ناشی از سمت تگ به سمت سیستم اطلاعات خوان را، بلوکه کند. همچنین فرض بر آن است که عامل متخاصم قادر به ارسال a به سمت تگ مربوطه می باشد (قبل از آنکه تگ مربوط بتواند دور بعدی تبادل اطلاعات را با یک مقدار جدید b، آغاز نماید)؛ یعنی، سیستم اطلاعات خوان، مکررا a را به سمت تگ مربوطه ارسال می کند، که آن را همچنان به محاسبه zs، مشغول می سازد. همچنین، مورد پی آیند نیز می بایستی با bs های مختلف، طی دورهای تبادلاتی مختلف پروتکل مربوطه، کار کند.

زمانیکه فرایند احراز هویت به محض ارسال b توسط تگ مربوطه، آغاز می گردد، عامل متخاصم آن را شنود می کند و مقدار a=0 را به تگ مربوطه، ارسال می کند. سپس، تگ مربوطه مقدار $z(= b \cdot y \oplus \nu)$ را محاسبه می کند و آن را به سمت سیستم اطلاعات خوان، ارسال می کند(که در این مورد، عامل متخاصم نقش سیستم اطلاعات خوان را بازی می کند). این فرایند می تواند به تعداد کافی تکرار شود تا زمانیکه مقدار $b \cdot y$، مجددا حاصل گردد. از آنجا که مقدار b برای عامل متخاصم مشخص می باشد، لذا مقدار y را می توان استنباط نمود. با دانستن مقادیر a، b و y، می توان این فرایند را تا زمانیکه مقدار x شناسایی شود، تکرار نمود. عامل متخاصم از این

حقیقت که ارتباطات مابین تگ و سیستم اطلاعات خوان توسط خود سیستم اطلاعات خوان کنترل می شود، بهره می برد؛ یعنی، زمانیکه فرایند احراز هویت با موفقیت به انجام می رسد، وظیفه پروتکل مربوطه خاتمه می یابد. از آنجا که عامل متخاصم، در طول این حمله به عنوان سیستم اطلاعات خوان محسوب می شود، می تواند تا آنجا که مورد نیاز است، همچنان به تعداد زیادی از دورهای تبادلاتی پروتکل مربوطه، به کار خود ادامه دهد تا زمانیکه در شناسایی کلیدهای رمز (x,y) موفق شود.

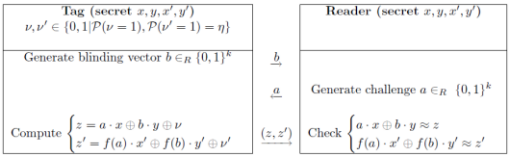
در حمله موثر علیه پروتکل HB^{+}، در وهله ای که کلیدهای رمز برای عامل متخاصم مشخص می گردد، تگ مربوطه و سیستم اطلاعات خوان، احراز هویت می شود. این امکان وجود دارد که زمانیکه تگ مربوطه، احراز هویت شد، احتمالا سیستم را ترک نماید و دیگر هیچ تعاملی با هیچ سیستم اطلاعات خوانی، نداشته باشد. تحت این شرایط، حمله ارائه شده در دو پاراگراف قبلی، از یک حاشیه برخوردار است، چرا که رویداد آن بدون هیچ تعاملی مابین تگ مربوطه و سیستم اطلاعات خوان، صورت می پذیرد.

۳-۲ پروتکل HB^{++} (نسخه اولیه) و پروتکل HB^{++} (نسخه اخیر)

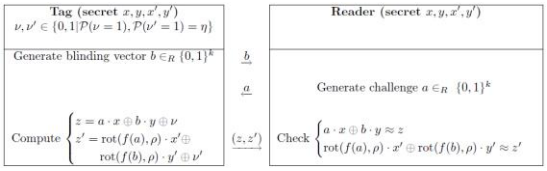
در پاسخ به سناریوی حمله گیلبرت و همکارانش به پروتکل HB^{+}، برینگر و همکارانش دو پروتکل HB^{++} (نسخه اولیه؛ تصویر ۴) و پروتکل HB^{++} (نسخه اخیر؛ تصویر ۵) را ارائه نمودند که در برابر چنین حملات man-in-the-middle ای، امن می باشد. همچنین، این پروتکل ها همچنان در برابر حملات ناشی از عامل متخاصمی که خود را به عنوان یک سیستم اطلاعات خوان موثق، وانمود ساخته است، مصون نمی باشد. مشخص شده است که پروتکل HB^{++} (نسخه اولیه)، در برابر حملات طرح ریزی شده در مقاله برینگر و همکارانش (۲۰۰۶)، مصون نمی باشد.

آسیب پذیری دیگر، از این حقیقت ناشی می شود که پروتکل های HB^{++} (نسخه اولیه) و HB^{++} (نسخه اخیر)، همانند پروتکل HB^{+}، حاوی بردار z هستند، و از این بردار z می توان برای شناسایی کلیدهای رمز x و y، بهره گرفت. از آنجا که عامل متخاصم می تواند تگ ای را که فقط از مقدار z آگاهی دارد، رد گیری نماید، این موضوع به عنوان یک آسیب پذیری محسوب می شود که سبب ایجاد حمله "کانال جانبی"، می‌شود.

همچنانکه پیش تر در این بخش ذکر گردید، عامل متخاصم می‌تواند به راحتی مقدار z را با استفاده از حمله man-in-the-midlle، یا از طریق وانمود کردن خود به عنوان یک سیستم اطلاعات خوان معتبر، محاسبه نماید.



تصویر ۴ - یک دور تبادل اطلاعاتی مربوط به پروتکل HB^{++} (نسخه اولیه)



تصویر ۵ - یک دور تبادل اطلاعاتی مربوط به پروتکل HB^{++}

با نادیده گرفتن امکان این حمله "کانال جانبی"، عامل متخاصم همچنان نیازمند شناسایی دو کلید رمز دیگر $(x`,y`)$ می باشد. در اینجا، فرض بر آن است که عامل متخاصم قادر به دریافت اطلاعات ارسال شده از سمت تگ می باشد و می تواند از رسیدن این اطلاعات به سمت سیستم اطلاعات خوان، ممانعت نماید. عامل متخاصم می تواند مقدار a را به نفع خود، دستکاری نماید تا کلیدهای رمز $(x`,y`)$ را بازیابی نماید. ابتدا به امر، عامل متخاصم می تواند مقدار a=0 را جهت تعیین $f(b) \cdot y'$، ارسال نماید. به محض تکمیل این فرایند، عامل متخاصم می تواند مقدار a=1 را جهت تعیین $1 \cdot x' \oplus f(b) \cdot y'$، تنظیم نماید. سپس، عامل متخاصم می تواند کلید رمز x' را از $f(b) \cdot y'$ و $1 \cdot x' \oplus f(b) \cdot y'$، شناسایی کند.

در طول چندین دور تبادلاتی بعدی این پروتکل، عامل متخاصم می تواند مقدار a=b را تخصیص دهد و

$f(b)\cdot(x'\oplus y')$ را تعیین نماید. عامل متخاصم با دانستن مقادیر $f(b)\cdot y'$ و x'، می تواند مقدار y' را مشخص نماید.

برینگر و همکارانش ، پروتکل HB^{++} (نسخه اولیه) را به منظور جبران این آسیب پذیری، تعدیل و اصلاح نمودند و پروتکل HB^{++} (نسخه اخیر) را که در برابر این آسیب پذیری ها، از محافظت برخوردار می باشد، تعیین و ارائه نمودند. همچنین، در جائیکه عامل متخاصم خود را به عنوان یک سیستم اطلاعات خوان معتبر وانمود می سازد، پروتکل HB^{++} (نسخه اخیر) نیز مستعد حملات مشابه می باشد. مجددا در اینجا، همانند پروتکل HB^{+} می توان از z، جهت بازیابی کلیدهای رمز x و y، بهره گرفت. به منظور بازیابی دو کلید رمز دیگر (x`,y`)، مفروضات زیر را در نظر می گیریم: مقدار ρ، تنها برای یکبار در طول یک دور تبادلاتی، بهنگام می شود و بدین منظور، به هنگام شروع ارسال b توسط تگ مربوطه و پایان بررسی z و z' توسط سیستم اطلاعات خوان، یک دور تبادلاتی، تعیین می گردد. احتمالا، فرایندهای بهنگام سازی ρ، در شروع هر دور تبادلاتی، رخ می دهد .

همچنین، یک عامل متخاصم سریع، می تواند مابین دو ارسال متوالی b (توسط تگ مربوطه)، برای چندین مرتبه با آن تگ ارتباط برقرار کند.

مادامیکه مقادیر b و ρ (زمانیکه عامل متخاصم در حال ارتباط با تگ مربوطه می باشد)، به طور ثابت باقی بماند، فرایند زیر می تواند در آشکارسازی کلیدهای رمز (x`,y`) کمک نماید. فرض بر آن است که عامل متخاصم، قادر به شنود b (که توسط تگ مربوطه ارسال شده است)، می باشد و همچنین می تواند از رسیدن b به سیستم اطلاعات خوان، ممانعت نماید. عامل متخاصم می تواند مقدار a را به نفع خود دستکاری نماید تا کلیدهای رمز (x`,y`) را بازیابی نماید. ابتدا به امر، عامل متخاصم می تواند مقدار a=0 را به منظور تعیین مقدار $rot(f(b)\cdot\rho)\cdot y'$، ارسال نماید. به محض تکمیل این فرایند، عامل متخاصم می تواند مقدار a=1 را به منظور تعیین مقدار $1\cdot x'\oplus rot(f(b)\cdot\rho)\cdot y'$، تنظیم نماید. عامل متخاصم می تواند کلید رمز x' را از $rot(f(b)\cdot\rho)\cdot y'$ و $1\cdot x'\oplus rot(f(b)\cdot\rho)\cdot y'$، شناسایی نماید.

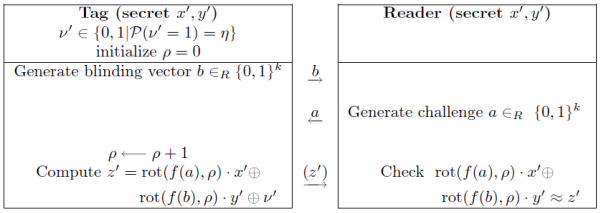
در طول چندین دور تبادلاتی بعدی این پروتکل، عامل متخاصم می تواند مقدار a=b را تخصیص دهد و $rot(f(b)\cdot\rho)\cdot(x'\oplus y')$ را تعیین نماید. عامل متخاصم با دانستن مقادیر $rot(f(b)\cdot\rho)\cdot y'$ و x'، می تواند مقدار y' را مشخص نماید.

۲-۴ پیشنهادات مربوط به تعدیل و اصلاح پروتکل HB^{++}

به منظور حفظ مقاومت امنیتی در برابر مدل متخاصم ارائه شده توسط ژولز و ویس ، پیشنهادات اصلاحی، مسئله غامض پروتکل HB^{+} را دست نخورده باقی نگه می دارد. اصلاحات اصلی، به صورت زیر می باشد:

- حذف z و بردارهای مربوطه (x,y) و نیز ν. این اقدام به منظور جلوگیری از حمله "کانال جانبی"، که قبلا ذکر شده است، می باشد. اثر جانبی آن باعث می شود تا این پروتکل، به صورت یک پروتکل "سبک وزن" در آید.

- بهنگام سازی ρ، در هر زمان که مقدار z محاسبه شود. این فرایند به منظور جلوگیری از کاربرد ρ مشابه (قبل از آغاز دور تبادلاتی جدید از انتهای تگ مربوطه)، می باشد.



تصویر ۶ – یک دور تبادلاتی مربوط به پروتکل اصلاح شده HB^{++}

۳- آنالیز امنیت

به دنباله مقاله دیمیتریو ، آنالیز امنیتی در خصوص اصلاحات پیشنهادی مربوط به پروتکل HB^{++} را به طور خلاصه ارائه می‌دهیم.

- حمله به تگ مربوطه. این نوع حمله، به سناریویی که در آن عامل متخاصم خود را به عنوان سیستم اطلاعات خوان وانمود می کند، اطلاق می گردد. از آنجا که این پروتکل دقیقا بر اساس

اجتناب از این نوع حمله، بنا شده است، امید می رود که عامل متخاصم در حملات خود موفق نباشد.

- حمله به سیستم اطلاعات خوان. در اینجا، عامل متخاصم خود را به عنوان یک تگ معتبر وانمود می کند. به لحاظ کلیدهای رمز مشترک $(x`,y`)$ این نوع حمله موفق نخواهد بود.

- حمله به ارتباطات مابین تگ و سیستم اطلاعات خوان. عامل متخاصم می تواند پیام های مابین سیستم اطلاعات خوان و تگ مربوطه را بلوکه نماید. زمانیکه این اتفاق می افتد، فرایند احراز هویت متوقف و دچار شکست می شود. تعاملات تکراری (چه از طریق بلوکه کردن ارتباطات، چه از طریق حملات man-in-the-middle)، زمینه را برای عامل متخاصم جهت شناسایی کلیدهای رمز، مهیا می سازد.

- حمله به اطلاعات محرمانگی کاربر. از آنجا که هیچ نوع اطلاعات محرمانگی در طول فرایند اعتبار سنجی ارسال نمی گردد، لذا این موضوع در اینجا مدنظر قرار نگرفته است.

- حمله به محرمانگی موقعیت مکانی. از آنجا که کلیدهای رمز در طول اجراهای مختلف پروتکل، تغییر نمی کند، این موضوع مدنظر قرار می گیرد.

- حمله به کلید. این حمله زمانی اتفاق می افتد که عامل مهاجم، در حال شنود تعاملات باشد و سعی در شناسایی مقادیر کلید، داشته باشد. چنانچه کلیدها به طور مناسب انتخاب شده باشند، مجددا این موضوع مدنظر قرار نخواهد گرفت .

- حمله به ساختار پیاده سازی. به شرطی که کلیدها و اعداد تصادفی با دقت تولید شده باشند، این موضوع مدنظر قرار نخواهد گرفت.

- منفک سازی تگ ها. واضح است که این کلید ها، در برابر مداخلات ساختاری، مقاوم نیستند و می توان آنها را به منظور بازیابی ساختار z' و نیز کلیدهای رمز (x', y')، از یکدیگر منفک نمود.

۴- نتیجه گیری

در این مقاله، پروتکل HB و نسخه های گوناگون آن را به منظور احراز هویت ، سامانه های RFID، مورد بحث و ارزیابی قرار داده شد. علاوه بر مصالحات امنیتی که در مقالات مربوطه ذکر شده است، همچنان مصالحه امنیتی دیگری ارائه شد که می‌تواند به لحاظ وانمودسازی عامل متخاصم، به عنوان یک سیستم اطلاعات خوان معتبر، روی دهد. به محض اینکه سیستم اطلاعات خوان معتبر (زمانیکه عامل متخاصم با تگ مربوطه جهت شناسایی کلیدهای رمز، تعامل برقرار می کند)، کنار گذاشته می شود، این تهدید بدتر نیز خواهد شد. همچنین، آسیب پذیری نسخه اخیر پروتکل HB (یعنی HB^{++}) نشان داده شد و روشی را به منظور اجتناب از این آسیب پذیری، ارائه شد. با وجودیکه روش ارائه شده در برابر تمامی انواع حملات صورت گرفته توسط عامل متخاصم، امن نمی باشد، به طور معقول در برابر آن دسته از حملاتی که مدنظر قرار گرفته است، امن می باشد (در حالیکه خصوصیت سبک وزنی پروتکل را حفظ می نماید).

مراجع

[1] G. Avoine and P. Oechslin. \RFID Traceability: A Multilayer Problem,"
Financial Cryptography - FC'05, LNCS, Springer, 2005.

[2] J. Bringer, H. Chabanne, and E. Dottax. \HB++: a Lightweight Authentication
Protocol Secure Against Some Attacks," IEEE International Conference
on Pervasive Services, Workshop on Security, Privacy and Trust in
Pervasive and Ubiquitous Computing { SecPerU, 2006.

[3] T. Dimitriou. \A Lightweight RFID Protocol to Protect Against Traceability
and Cloning Attacks," Proceedings of the IEEE International Conference
on Security and Privacy for Emerging Areas in Communication Networks -
SECURECOMM, 2005.

[4] K. Finkenzeller. RFID Handbook, second edition, Wiley & Sons, 2002.

[5] H. Gilbert, M. Robshaw, and H. Sibert. \An Active Attack Against HB+ - A
Provably Secure Lightweight Protocol." Cryptology ePrint Archive, Report
2005/237, 2005. http://eprint.iacr.org.

[6] N.J. Hopper and M. Blum. \Secure Human Identi_cation Protocols." in
C. Boyd (ed.) Advances in Cryptology - ASIACRYPT 2001, Volume 2248,

Lecture Notes in Computer Science, pp. 52-66, Springer-Verlag, 2001.

[7] A. Juels and S. Weis. \Authenticating Pervasive Devices with Human Protocols,"
in V. Shoup (ed.) Advanced in Cryptology - CRYPTO'05, Volume
3126, Lecture Notes in Computer Science, pp. 293-308, Springer-Verlag,
2005.

[8] A. Lenstra and E. Verheul. \Selecting Cryptographic Key Sizes," Journal of
Cryptography, 14(4), pp. 255-293, 2001.

[9] R. C.-W. Phan and S.-M. Yen. \Amplifying Side-Channel Attacks with Techniques
from Block Cipher Cryptanalysis," Proceedings of the 7th Smart Card
Research and Advanced Application IFIP Conference - CARDIS'06, 2006.

[10] S. Weis, S. Sarma, R. Rivest, and D. Engels. \Security and Privacy Aspects