ششمین همایش بین‌المللی افق های نویـن در
مهندسی برق، کامپیوتـر و مکانیک
6th International Conference on the New Horizons in
Electrical Engineering, Computer and Mechanical
www.mhconf.ir

پروتکل احراز هویت متقابل امن و سبک برای شبکه های حسگر بی سیم

دانشجوی کارشناسی فناوری اطلاعات

مهدی غلامی وثیق www.mahdigholamivaisgh@gmail.com

دانشگاه جامع علمی کاربردی خانه کارگر کرج

چکیده:

شبکه‌های حسگر بیسیم (WSN) به طور گسترده‌ای برای ارائه سرویس‌های مناسب مانند مراقبت‌های بهداشتی، خانه‌های هوشمند و... استفاده می‌شوند. برای ارائه سرویس‌های مناسب، گره‌های حسگر در محیط‌های WSN، داده‌های سنجش را جمع‌آوری و به دروازه می‌فرستند. هرچند که، دارای یکسری مشکلات امنیتی نیز می‌باشند، چراکه پیام‌های حساس را از طریق کانالی ناامن رد و بدل می‌کنند. بنابراین، پروتکل‌های اعتبارسنجی امن برای جلوگیری از نقص امنیتی در WSNها ضروری هستند. در سال ۲۰۲۰، مقدم و همکاران طرحی احراز هویتی کارآمد و کلیدی در WSN را پیشنهاد دادند. متأسفانه، متوجه شدیم که طرح مقدم و همکاران نمی تواند از حملات نشت اعداد تصادفی داخلی و خاص جلوگیری کند. ما همچنین ثابت می‌کنیم که طرح مقدم و همکاران از محرمانگی پیشرو کامل اطمینانی حاصل نمی‌کند. برای جلوگیری از آسیب پذیری‌های امنیتی طرح مقدم و همکاران، ما یک پروتکل احراز هویت متقابل امن و سبک را برای WSNها (WSN-SLAP) پیشنهاد می‌کنیم. WSN-SLAP در برابر مشکلات مختلف امنیتی مقاومت و همچنین محرمانگی کامل و احراز هویت متقابل را ایجاد می‌کند. با استفاده از منطق Burrows-Abadi-Needham (BAN) ، مدل Real–یا–Random (ROR) و تأیید خودکار پروتکل‌ها و برنامه‌های امنیتی اینترنت (AVISPA) ، امنیت WSN-SLAP را ثابت می‌کنیم. علاوه بر این، عملکرد WSN-SLAP را در مقایسه با پروتکل‌های مرتبط موجود ارزیابی می‌کنیم و نشان می‌دهیم که WSN-SLAP نسبت به پروتکل‌های قبلی برای محیط‌های WSN امن تر و مناسب تر هستند.

کلمات کلیدی: احراز هویت متقابل؛ شبکه های حسگر بی سیم؛ BANمنطق، مدلROR؛ AVISPA.

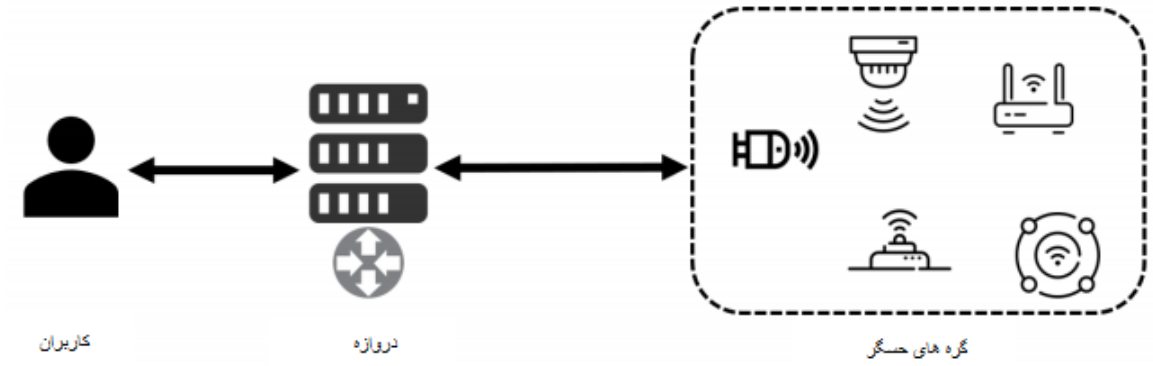
۱. مقدمه :

نمونه‌ای از توسعه سریع فن آوری ارتباطات بیسیم ، شبکه‌های حسگر بیسیم (WSN) می‌باشد که می توان آن را در محیط‌های مختلف مانند شبکه‌های هوشمند، خانه‌های هوشمند، صنعت کشاورزی، اینترنت اشیا (IoT) و بهداشت و درمان استفاده کرد[۱-۵]. افراد می‌توانند با بهره گیری از محیط های WSN به زندگی بهتری دست پیدا کنند. به طور کلی، محیط‌های WSN از گره‌های حسگر، یک دروازه و کاربران تشکیل می‌شود، همانطور که در شکل ۱ نشان داده شده است. گره های حسگر محیط اطراف خود را شناسایی و کنترل می‌کنند. سپس، گره های حسگر داده های نظارت شده را به دروازه انتقال می‌دهند. دروازه پیام بین گره های حسگر و کاربران رله و تحلیل می‌کند. همچنین این درگاه اطلاعات خصوصی گره های حسگر و کاربران را برای ارائه خدمات ایمن مدیریت می‌کند. کاربران می توانند از طریق دروازه به داده های جمع آوری شده توسط گره های حسگر دسترسی پیدا کنند.

نمونه ای از محیط برنامه در WSN خدمات مراقبت های بهداشتی است. حسگرهای پوشیدنی متصل به بیمار، وضعیت سلامتی بیمار را تجزیه و تحلیل و سپس، این حسگرها اطلاعات جمع آوری شده را برای پزشک ارسال می‌کنند. با این حال، این سرویس

ها می توانند در معرض حملات امنیتی مختلف قرار بگیرند زیرا هر موجودیت از طریق یک کانال عمومی اطلاعات را رد و بدل می‌کند. اگر دشمنی پیام‌ها را در WSN رهگیری کند، می تواند خود را به عنوان یک کاربر قانونی بدل کرده و پیام نادرستی را به گره سنسور ارسال کند. علاوه بر این، اگر یک دشمن به عنوان یک شخص حقوقی در دروازه ثبت نام کند، می تواند سعی کند اطلاعات حساس کاربر قانونی دیگر را بدست آورد. بنابراین، ما به یک پروتکل احراز هویت نیاز داریم که بتواند خدمات ایمنی را ارائه دهد و از حملات مختلف در محیط WSN جلوگیری کند.

در سال ۲۰۲۰، مقدم و همکاران [٦] طرح احراز هویت و توافق نامه کلیدی را برای محیط‌های WSN با استفاده از منحنی بیضوی Diffie-Hellman (ECDH) پیشنهاد کردند [٧]. آنها نشان دادند که طرح آنها در برابر حملات امنیتی مختلف مانند تکرار، حدس رمز عبور، تأیید کننده سرقت، و حملات (MITM) کارآمد و امن می‌باشد. با این حال، متوجه می شویم که طرح مقدم و همکاران امنیت در برابر حملات داخلی و حملات نشت اعداد تصادفی خاص–جلسه را تأمین نمی‌کند. و همچنین ثابت می‌کنیم که طرح مقدم و همکاران از محرمانگی کامل پیشرو پشتیبانی نمی کند. علاوه بر این، هر موجودیت برای محاسبه کلید جلسه در طرح مقدم و همکاران، عملیات ضرب Elliptic Curve Cryptography (ECC) را انجام می‌دهد. با این وجود ECC به هزینه‌های محاسباتی سنگینی نیاز دارد. از آنجا که گره های حسگر دارای قابلیت محاسبه و ذخیره سازی کم در یک محیط WSN هستند ، ما نمی توانیم با استفاده از ECC در محیط WSN از ارتباطات در زمان واقعی اطمینان حاصل کنیم. بنابراین، استفاده از طرح مقدم و همکاران، ارائه خدمات کارآمد را دشوار می‌کند. برای بهبود آسیب پذیری‌های امنیتی و کاهش هزینه محاسباتی طرح مقدم و همکاران ، ما یک پروتکل احراز هویت متقابل امن و سبک (WSN-SLAP) را با توجه به ویژگی‌های امنیتی و کارایی با استفاده از توابع hash و عملیات XOR پیشنهاد می‌کنیم.



شکل ۱. مدل سیستم در شبکه های حسگر بی سیم (WSN).

۱,۱ مشارکت ها:

مشارکتهای مقاله ما به شرح زیر است.

• ما آسیب پذیری های امنیتی طرح مقدم و همکاران را تحلیل و اثبات می‌کنیم.

سپس، WSN-SLAP را برای حل آسیب پذیری های امنیتی طرح مقدم و همکاران پیشنهاد می‌کنیم.

• احراز هویت متقابل WSN-SLAP را با استفاده از منطق Burrows – Abadi – Needham (BAN) نشان می‌دهیم [۸].

• امنیت کلید جلسه WSN-SLAP را با استفاده از مدل Real-or-Random (ROR) اثبات می‌کنیم [۹]

• برای اثبات ویژگی‌های امنیتی WSN-SLAP در برابر حملات پخش و MITM، از تأیید خودکار پروتکل‌ها و برنامه‌های امنیتی اینترنت AVISPA [10]،[11]استفاده می‌کنیم.

• هزینه ارتباطات، هزینه محاسباتی و خصوصیات امنیتی WSN-SLAP را در مقایسه با طرح‌های مربوطه تحلیل می‌کنیم.

۲,۱. مدل دشمن:

WSN-SLAP از یک مدل معروف دشمن استفاده می کند به نام مدل Dolev-Yao (DY) [12]. از طریق مدل DY ، دشمن می تواند شنود ، حذف ، رهگیری و درج پیام های رد و بدل شده از طریق یک کانال عمومی را انجام دهد. علاوه بر این ، دشمن می تواند پارامترهای زودگذر مخصوص جلسه را که مبتنی بر مدل دشمن Canetti-Krawczyk (CK) است ، در معرض دید قرار دهد [۱۳]. دشمن می تواند حملات امنیتی مختلفی را با مدل DY و مدل CK انجام دهد. مفروضات دقیق مدل دشمن به روش زیر تعریف شده است.

• اگر یک دشمن به عنوان یک کاربر قانونی در دروازه ثبت نام کند ، دشمن می تواند با نهادهای دیگر احراز هویت کند.

• یک دشمن می تواند کارت هوشمند گمشده / سرقت شده کاربر را بدست آورد. دشمن می تواند حمله تجزیه و تحلیل قدرت [۱٤] را برای دریافت پارامترهای ذخیره شده کارت هوشمند انجام دهد.

• یک دشمن می تواند حملات مختلفی مانند پخش مجدد ، ضبط گره سنسور ، تأیید کننده سرقت و حملات حدس رمز عبور خارج از خط را امتحان کند.

۳,۱ سازماندهی:

در بخش ۲ ، ما کارهای مرتبط را برای محیط WSN شرح می دهیم. سپس ، ما مجدداً از طرح مقدم و همکاران در بخش ۳ بازدید کرده و نقایص امنیتی طرح مقدم و همکاران را در بخش ٤ اثبات می کنیم. بخش ٥ WSN-SLAP را نشان می دهد. در بخش ٦ ، ما با استفاده از منطق BAN ، مدل ROR و ابزار شبیه سازی AVISPA ، تجزیه و تحلیل امنیتی غیررسمی و رسمی WSN-SLAP را انجام می دهیم. در بخش ۷ ، ما عملکرد WSN-SLAP را در مقایسه با پروتکل های مربوطه تجزیه و تحلیل می کنیم. در بخش ۸ ، مقاله خود را جمع بندی و خلاصه می کنیم.

۲.کارهای مرتبط:

در چند دهه گذشته ، چندین طرح احراز هویت مبتنی بر رمز عبور برای تأمین امنیت و کارایی در محیط های WSN ارائه شده است [۱۹–۱۵]. در سال ۱۹۸۱ ، لمپورت [۲۰] مکانیزم احراز هویت را بر اساس رمز عبور پیشنهاد کرد. لامپورت از توابع هش یک طرفه برای رمزگذاری رمز استفاده کرده و رمز عبور هش شده را در داخل سیستم ذخیره می کند. در سال ۲۰۰٦ ، وونگ و همکاران [۲۱] طرح احراز هویت مبتنی بر رمز عبور را در محیط های WSN پیشنهاد کرد. متاسفانه ، Tseng و همکاران [۲۲] ثابت کرد که برنامه وونگ و همکاران در برابر حملات جعل و پخش ناامن است. تسنگ و همکاران یک طرح احراز هویت کاربر پویا را برای بهبود آسیب پذیری های امنیتی Wong و همکاران نشان داد. [۲۱] با این حال ، این طرح ها [۲۰–۲۲] می توانند از حملات حدس زدن رمز عبور در حالت آفلاین یا آفلاین رنج ببرند زیرا آنها فقط از رمز عبور به عنوان عاملی برای ورود به سیستم و تأیید اعتبار با نهادهای دیگر استفاده می کنند.

در چند دهه گذشته ، طرح های احراز هویت مبتنی بر دو عامل [۲۳–۲۵] با استفاده از توابع هش و عملیات XOR برای بهبود ضعف های امنیتی تک عامل ارائه شده است. در سال ۲۰۰۹ ، داس و همکاران [۲۳] طرح احراز هویت دو عاملی مبتنی بر کارت هوشمند در شبکه های WSN را پیشنهاد کرد. آنها نشان دادند که طرح آنها می تواند از حملات مختلف مانند پخش مجدد ، تأیید کننده سرقت شده و حملات حدس رمز عبور خارج از خط جلوگیری کند. با این حال ، خان و همکاران [۲٤] تحلیل کرد که داس و همکارانش. [۲۳] طرح در معرض حمله خودی ممتاز است. او و دیگران [۲۵] دریافت که داس و همکاران [۲۳] طرح در برابر حملات خودی و جعل هویت آسیب پذیر است. برای بهبود آسیب پذیری های امنیتی طرح Das و همکاران ، او و همکاران [۲۵]

طرح تأیید اعتبار کاربر دو عاملی را برای شبکه های WSN پیشنهاد داده است. با این حال ، این طرح ها [۲۳–۲۵] می توانند از حملات مختلف مانند استفاده از کارت های هوشمند و دستگاه های تلفن همراه دزدیده شده رنج ببرند.

برای رفع اشکالات امنیتی مرتبط با طرح های احراز هویت مبتنی بر دو عامل و بهبود سطح امنیت در محیط های WSN ، محققان بسیاری از طرح های احراز هویت مبتنی بر ECC را ارائه داده اند [۲۶–۳۱]. در سال ۲۰۱۱ ، Yeh و همکاران [۲۶] یک پروتکل احراز هویت برای محیط های WSN با استفاده از ECC ارائه داد. طرح Yeh و همکاران از کارت هوشمند و ECC برای جلوگیری از مشکلات امنیتی مختلف مانند حمله داخلی و حملات نقاب استفاده کرده است. چوی و همکاران [۲۷] طرح تأیید اعتبار کاربر مبتنی بر ECC را برای WSN پیشنهاد کرد. با این حال ، وو و همکاران [۲۸] اشاره کرد که پروتکل چوی و همکاران در برابر حمله جعل امنیت ایجاد نمی کند. نام و همکاران [۲۹] یک پروتکل احراز هویت ایمن را برای WSN براساس ECC پیشنهاد کرد. طرح Nam و همکاران یک پروتکل ایمن را بر اساس مشکل محاسبه منحنی بیضوی Diffie-Hellman (ECCDH)) ارائه می دهد. در سال ۲۰۱۶ ، جیانگ و همکاران [۳۰] طرح احراز هویت مبتنی بر ECC را پیشنهاد کرد. طرح جیانگ و همکاران ارتباطات ایمن و غیرقابل ردیابی را در محیط WSN فراهم می کند. در سال ۲۰۱۷ ، وو و همکاران [۳۱] طرح احراز هویت کاربر را با استفاده از ECC پیشنهاد داده است. طرح Wu و همکاران می تواند حریم خصوصی کاربران را در محیط WSN حفظ کند. با این حال ، گره های حسگر در WSN قدرت و منابع محاسباتی کمی دارند. بنابراین ، تهیه کارایی در محیط WSN با استفاده از این طرح ها دشوار است [۲۶–۳۱] زیرا ECC به منابع محاسباتی زیادی احتیاج دارد.

در سال ۲۰۲۰ ، مقدم و همکاران [۶] طرح احراز هویت و توافق نامه کلیدی را با استفاده از ECDH پیشنهاد کرد. آنها ادعا کردند که طرح آنها مقاومت در برابر حملات مختلف مانند پخش مجدد ، MITM ، حدس رمز عبور خارج از خط و حملات تأیید کننده سرقت را فراهم می کند. با این حال ، متوجه می شویم که طرح مقدم و همکاران در برابر حملات نشتی اعداد تصادفی خاصجلسه، و محرمانگی کامل پیشرو آسیب پذیر است. علاوه بر این ، طرح مقدم و دیگران هزینه محاسباتی سنگینی دارد زیرا شامل یک محاسبه مبتنی بر ECC می شود. بنابراین، ما WSN-SLAP را پیشنهاد می‌کنیم، که در برابر مشکلات امنیتی مختلف مقاوم می‌باشد.

۳. بررسی طرح مقدم و همکاران:

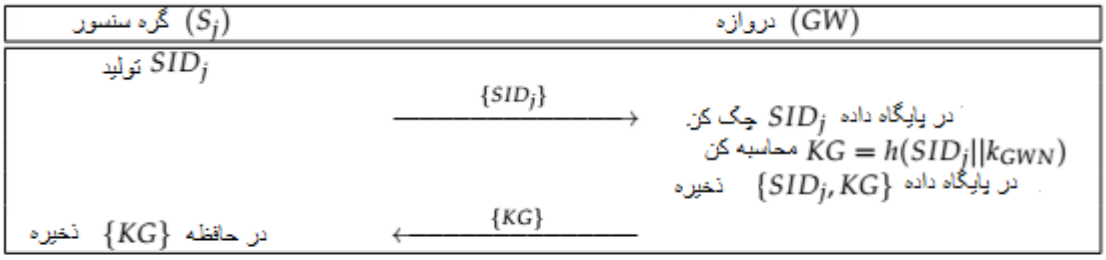
مقدم و همکاران طرح احراز هویت مبتنی بر ECDH را در WSN پیشنهاد کرد [۶]. طرح Moghadam و دیگران از مراحل ثبت گره حسگر، ثبت نام کاربر و مراحل ورود و احراز هویت تشکیل شده است. جدول ۱ علائم طرح مقدم و همکاران و WSN-SLAP را نشان می‌دهد.

.جدول 1. علائم نویسی

Notation	Description
U_i	دروازه
GW	گره سنسور
S_j	هویتواقعی کاربر
ID_i	رمز کاربر
PW_i	هویت شیه کاربر
PID_i	هویت واقعی گره حسگر
SID_j	کلید اصلی دروازه
k_{GWN}	کلید مخفی مشترک بین دروازه و گره
KG	حسگر
X	کلید عمومی دروازه
G	گروه منحنی بیضوی
P	ژنراتور G
$R_k, N_k, z_i, a_i, f_i, g_i, q_i$	اعداد تصادفی
T_k	مهر زمان
SK	کلید جلسه
E_k / D_k	رمزگذاری / رمزگشایی متقارن
$h(.)$	عملکرد هش
$\|\|$	تابع اتصال
$\oplus$	منحصر به فرد یا تایم

۱،۳ مرحله ثبت گره سنسور:

در این مرحله، یک گره حسگر Sj هویت خود را به GW دروازه می فرستد. سپس، GW یک پارامتر مخفی مشترک بین GW و Sj را محاسبه می‌کند. در شکل ۲، مرحله ثبت گره سنسور را نشان می‌دهیم و جزئیات به شرح زیر است.



.شکل 2. مرحله ثبت گره سنسور طرح مقدم و همکاران

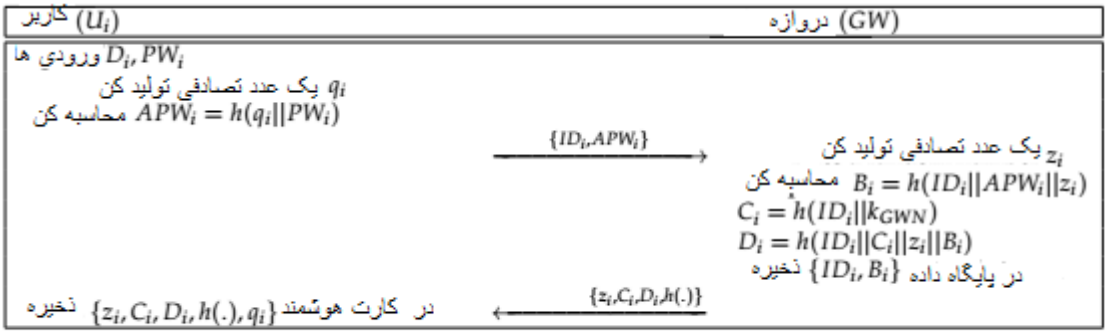
مرحله ۱: S_j هویت خود را SID_j تولید کرده و از طریق یک کانال امن به GW می فرستد.

مرحله ۲: GW SID_j را دریافت می کند و اعتبار SID_j را بررسی می کند. پس از آن ، (GW kG = h (SIDj || k_{GWN} را محاسبه می کند و {KG ،SID_j} را در پایگاه داده امن خود ذخیره می کند ، جایی که kGWN کلید اصلی GW است. سرانجام ، {KG} GW را برای Sj ارسال می کند.

مرحله ۳: S_j ، {KG} را در پایگاه داده خود دریافت و ذخیره می‌کند.

۲,۳ مرحله ثبت نام کاربر:

یک کاربر Ui با ارسال یک شناسه و یک مقدار رمز عبور مخفی ، به GW دروازه ثبت می شود. سپس ، GW کارت هوشمندی را برای Ui صادر می کند. در شکل ۳ ، مرحله ثبت نام کاربر را شرح می دهیم و جزئیات به صورت زیر نشان داده شده است.



شکل 3. مرحله ثبت نام کاربر از طرح مقدم و همکاران

مرحله ۱: Ui شناسه ID و رمز عبور PWi را وارد می کند و سپس یک عدد qi تصادفی ایجاد می کند. پس از آن ، Ui APWi (= h (qi || PWi را محاسبه می کند و پیام درخواست ثبت نام {APWi ،IDi} را به GW دروازه از طریق یک کانال امن می فرستد.

مرحله ۲: GW IDi ،APWi} را از Ui دریافت می کند و سپس یک عدد zi تصادفی ایجاد می کند. پس از آن ، GW Bi = (Ci = h (IDi || kGWN ،h (IDi || APWi || zi) و (Di = h (IDi || Ci || zi || Bi را محاسبه می کند. سرانجام ، GW، zi ،Ci ،Di ،h (.)را در یک کارت هوشمند ذخیره می کند و آن را از طریق یک کانال امن به Ui صادر می کند.

مرحله ۳: Ui کارت هوشمند را دریافت می کند و qi را در کارت هوشمند ذخیره می کند. در آخر ، پارامترهای {zi، Ci، Di، h (.)، qi} در کارت هوشمند ذخیره می شوند.

۳,۳ مرحله ورود و اعتبارسنجی:

پس از مرحله ثبت نام ، کاربر Ui دروازه GW را تأیید می کند. در شکل ٤ ، مرحله ورود و تأیید اعتبار را شرح می دهیم و مراحل دقیق مرحله به شرح زیر نشان داده شده است.

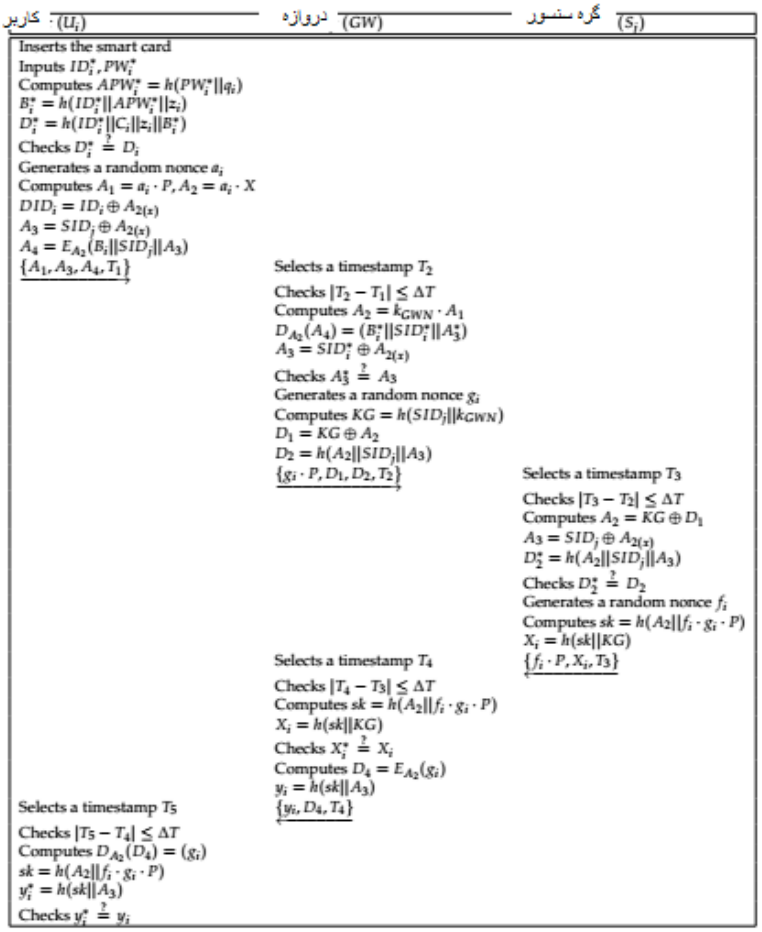
مرحله ۱: پس از قرار دادن کارت هوشمند ، Ui شناسه شناسه * i و رمز ورود PW * i را وارد می کند. کارت هوشمند APW * i = h (PW * i || qi)، B * i = h (ID * i || APW * i || zi)، D * i = h (ID * i || Ci || zi) را محاسبه می کند. (B * i و تأیید D verif i =؟ Di. اگر روند تأیید موفقیت آمیز باشد ، کارت هوشمند یک T1 غیرمستقیم و تایم تایم تصادفی ایجاد می کند. با کلید عمومی دروازه X ، کارت هوشمند A1 = ai · P، A2 = ai · X، D IDi = IDi ⊕ A2 (x)، A3 = SIDj ⊕ A2 (x) و A4 = EA2 (Bi || SIDj || A3). سرانجام ، کارت هوشمند از طریق یک کانال عمومی { A1، A3 ، A4، T1} را به GW ارسال می کند.

مرحله ۲: {A1، GW، A3، A4، T1} را از Ui دریافت می کند و یک مهر زمان T2 را انتخاب می کند و اعتبار T1 را بررسی می کند. اگر مهر زمان مشخص باشد ، A1 · GW A2 = kGWN، DA2 (A4) = (B * i || SID * i || A * 3)، A3 = (SID * i ⊕ A2 (x) را محاسبه می کند و A * 3 را تأیید می کند ؟ = A3 اگر برابری برقرار باشد ، GW یک gi غیر تصادفی تولید می کند و KG = h (SIDj || kGWN)، D1 = KG-A2، D2 = h (A2 || SIDj || A3) را محاسبه می کند. سرانجام ، gi · P، {GW، D1، D2، T2} را به گره سنسور Sj از طریق یک کانال عمومی ارسال می کند.

مرحله ۳: پس از دریافت پیام {gi · P، D1، D2، T2} از GW، Sj یک مهر زمان T3 را انتخاب می کند و اعتبار T2 را بررسی می کند. سپس ، Sj A2 = KG ⊕ D1، A3 = SIDj ⊕A2 (x)، D * 2 = h (A2 || SIDj || A3) را محاسبه می کند و D2 = ؟D * 2 را تأیید می کند. اگر تأیید صحت داشته باشد ، Sj یک nonce fi تصادفی ایجاد می کند و sk = h (A2 || fi · gi · P)، Xi = h (sk || KG) را محاسبه می کند. سرانجام ، {fi · P، Sj، Xi، T3} را به GW ارسال می کند.

مرحله ٤: GW پس از دریافت {T3، Xi، fi · P} از Sj ، مهر تایم T4 را انتخاب می کند و اعتبار T3 را بررسی می کند. سپس ، sk = h (A2 || fi · gi · P)، Xi = h (sk || KG) را محاسبه می کند و Xi = ؟X * i را تأیید می کند. اگر برابر باشد ، D4 = EA2 (gi)، yi = h (sk || A3) را محاسبه می کند و {T4، D4، yi} را به Ui می فرستد.

مرحله ۵: Ui پیام {yi، D4، T4} را دریافت می کند و یک مهر زمان T5 را انتخاب می کند و اعتبار T4 را بررسی می کند. سرانجام ، Ui DA2 (D4) = (gi)، sk = h (A2 || fi · gi · P) ، y * i = h (sk || A3) (yi = ?y * i را محاسبه می کند و را تأیید می کند. اگر برابر باشد ، پذیرش کلید موفقیت آمیز است.



شکل 4. مرحله ورود و تأیید اعتبار طرح مقدم و همکاران

٤. تحلیل رمز طرح مقدم و همکاران:

در این بخش، آسیب پذیری های امنیتی طرح مقدم و همکاران [٦] مانند حملات داخلی و حملات نشت اعداد تصادفی خاص-جلسه را نشان می دهیم. طرح مقدم و همکاران به یک محرمانگی کامل پیشرو دست نمی یابد.

۱,٤ حمله داخلی:

اگر یک دشمن A معمولی به عنوان یک کاربر قانونی Ui ثبت نام کند، A می تواند با درگاه GW و گره سنسور Sj با مبادله پیام ها احراز هویت کند. با استفاده از این اطلاعات ، A می تواند کلید جلسه Uli کاربر قانونی دیگری را محاسبه کند. جزئیات به صورت زیر نشان داده شده است.

مرحله ۱: A کارت هوشمند را وارد می کند و شناسه ID و رمز عبور PWi را وارد می کند. سپس کارت هوشمند اعتبار A را بررسی می کند و یک پیام درخواست ورود به سیستم {A1، A3، A4، T1} را به GW ارسال می کند. پس از احراز هویت A ، {gi · P، GW، D1، D2، T2} را به Sj ارسال می کند. با دریافت پیام {gi · P، D1، D2، T2}، Sj یک کلید جلسه sk را محاسبه می کند. سپس ، Sj پیام پاسخ تأیید اعتبار {fi · P، Xi، T3} را به GW ارسال می کند. GW کلید جلسه را محاسبه می کند و {yi، D4، T4} را برای A ارسال می کند. A کلید جلسه را محاسبه می کند و پیام های ارتباطی را در مرحله ورود و تأیید اعتبار به دست می آورد.

مرحله ۲: پس از به دست آوردن پیام {gi · P، D1، D2، T2} A KG = D1 ⊕ A2 را محاسبه می کند ، جایی که A2 کلید مخفی A با استفاده از ECC است و KG یک کلید مخفی مشترک بین GW و Sj است.

مرحله ۳: یک پیام $\{g^l_i \cdot P, D^l_1, D^l_2, T^l_2\}$ را از پیام کاربر قانونی دیگری Uli رهگیری می کند. از آنجا که A KG را می شناسد ، می تواند Al2 = Dl1 ⊕ KG را محاسبه کند ، جایی که A^l_2 کلید مخفی U^l_i است.

مرحله ٤: A با استفاده از کلید مخفی A^l_2 از U^l_i پیام $\{y^l_i, D^l_4, T^l_4\}$ را رمزگشایی کرده و D^l_4 را رمزگشایی می کند. سپس ، A می تواند به صورت تصادفی غیر مخفی g^l_i گره حسگر را بدست آورد. A می تواند با استفاده از پیام {fli · P، Xli، fli، Tl3} gli · P را محاسبه کند. سرانجام ، A کلید نشست $sk^l = h$ را محاسبه کنید $(A^l_2 \| f^l_i \cdot g^l_i \cdot P)$.

بنابراین ، طرح مقدم و همکاران نمی تواند از حملات خودی جلوگیری کند.

۲,٤ محرمانگی کامل روبه جلو:

مقدم و همکاران نشان داد که طرح آنها می تواند از ویژگی امنیتی محرمانگی کامل روبه جلو اطمینان حاصل کند. با این وجود ، اگر دشمن A کلید اصلی kGWN دروازه GW را بدست آورد ، دشمن می تواند کلید قانونی جلسه Ui کاربر sk را محاسبه کند. جزئیات در مراحل زیر نشان داده شده است.

مرحله ۱: اگر A کلید اصلی kGWN را بدست آورد ، A می تواند با استفاده از پیام درخواست ورود {A1، A3، A4، T1} کلید محرمانه A2 = kGWN · A1 Ui را محاسبه کند.

مرحله ۲: وقتی A پیام {yi، D4، T4} را قطع می کند ، A می تواند EA2 (gi) را رمزگشایی کند زیرا A2 کلید متقارن بین UI و GW دروازه است.

مرحله ۳: بعد از اینکه A پیام { fi · P، Xi، T3} را بدست آورد ، A می تواند (A2، gi) و (fi · P) را بدست آورد. سرانجام ، A کلید جلسه Ui را برای sk = h محاسبه می کند (A2 || fi · g · P). در نتیجه ، طرح مقدم و همکاران از پنهان کاری کامل رو به جلو اطمینان حاصل نمی کند.

۳,٤ حمله نشت شماره تصادفی مختص جلسه:

فرض کنید که یک nonce تصادفی برای یک دشمن مشخص شود. با استفاده از کلید عمومی X از GW دروازه ، A می تواند A2 = ai · X را محاسبه کند. سپس ، A می تواند کلید جلسه sk را محاسبه کند. جزئیات به شرح زیر است.

مرحله ۱: A پس از دریافت پارامتر A2 ، پیام {yi، D4، T4} را ضبط می کند. سپس ، Adecrypts D4 = EA2

(gi) با استفاده از کلید متقارن A2 و gi بدست می آید.

مرحله ۲: پیام پیام گره حسگر {fi · P Sj، Xi، T3} را استراق سمع می کند. سرانجام ، A با استفاده از fi · P در پیام Sj ، کلید جلسه sk = h (A2 || fi · gi · P) را محاسبه می کند.

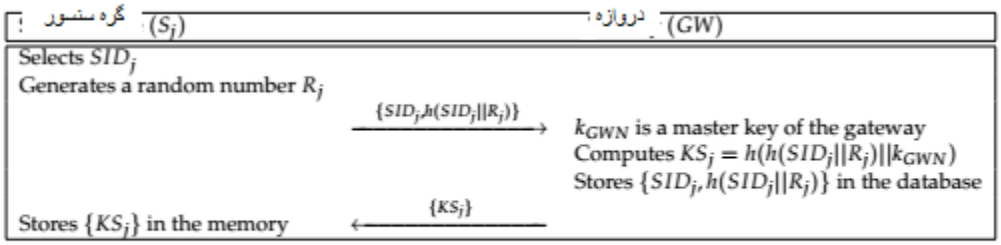
بنابراین، طرح مقدم و همکاران نمی تواند ازتعداد تصادفی حملات نشت جلسه-بخصوص جلوگیری کند.

۵. طرح پیشنهادی:

ما یک پروتکل احراز هویت متقابل ایمن و سبک را برای محیط WSN پیشنهاد می دهیم تا نقاط ضعف امنیتی طرح مقدم و دیگران را برطرف کنیم [٦]. برای در نظر گرفتن گره های حسگر محدود به منابع ، WSN-SLAP از توابع هش و عملیات XOR استفاده می کند که هزینه های محاسباتی کمتری را ایجاد می کند. WSN-SLAP از مراحل ثبت گره سنسور ، ثبت نام کاربر ، ورود به سیستم و احراز هویت ، به روزرسانی رمز عبور و مراحل اضافه شدن گره حسگر تشکیل شده است.

۱,۵ مرحله ثبت گره سنسور:

اگر گره حسگر Sj پیام درخواست ثبت نام را ارسال می کند ، GW دروازه یک پارامتر مخفی برای گره سنسور محاسبه می کند. سپس ، Sj پارامتر را ذخیره می کند. ما مرحله ثبت گره سنسور را در شکل ۵ نشان می دهیم و جزئیات به صورت زیر ارائه شده است.



شکل 5. مرحله ثبت گره سنسور یک پروتکل احراز هویت متقابل امن و سبک (WSN-SLAP)

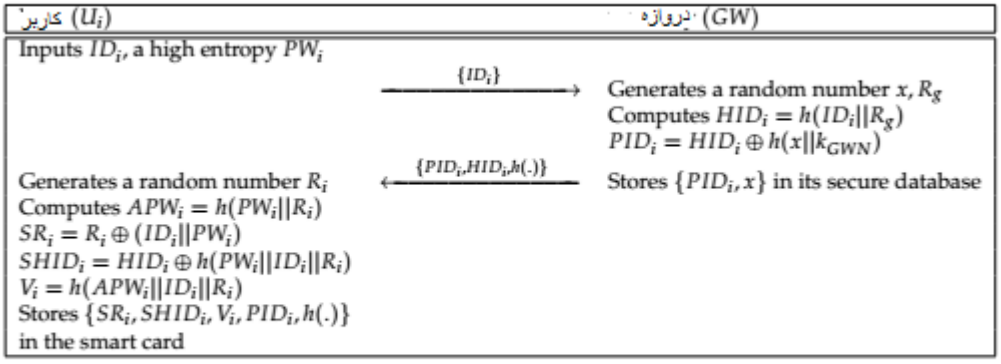
مرحله ۱: Sj هویت خود را SIDj انتخاب می کند و یک عدد تصادفی Rj تولید می کند. سپس ، Sj h (SIDj || Rj) را محاسبه کرده و {SIDj، h (SIDj || Rj)} را از طریق یک کانال امن به GW می فرستد.

مرحله ۲: {SIDj ،GW ،h (SIDj || Rj)} را دریافت می کند و KSj = h (h (SIDj || Rj) || kGWN) را محاسبه می کند ، جایی که kGWN کلید اصلی GW است. {SIDj ،GW ،h (SIDj || Rj)} را در پایگاه داده امن ذخیره می کند و {KSj} را برای Sj ارسال می کند.

مرحله ۳: سرانجام ، {KSj ،Sj} را در حافظه خود ذخیره می کند.

۲,۵ مرحله ثبت نام کاربر:

کاربر Ui یک پیام درخواست ثبت نام را به GW دروازه ارسال می کند. سپس، GW پارامترهای مخفی را محاسبه کرده و کارت هوشمندی را برای کاربر صادر می کند. در شکل ٦ ، مرحله ثبت نام کاربر را شرح می دهیم و مراحل دقیق به صورت زیر نشان داده شده است.



شکل 6. مرحله ثبت نام کاربر WSN-SLAP.

مرحله ۱: Ui یک شناسه هویت و یک رمز عبور آنتروپی بالا PWi را وارد می کند. پس از آن ، Ui از طریق یک کانال امن {IDi} را به GW منتقل می کند.

مرحله ۲: GW اعداد تصادفی x و Rg تولید می کند و (HIDi = h (IDi || Rg ،PIDi = H IDi ⊕ h (x || kGWN را محاسبه می کند. {PIDi ،GW ،x} را در پایگاه داده امن خود ذخیره می کند و پیام {PIDi ،H IDi ،h (.)} را به Ui می فرستد.

مرحله ۳: Ui یک عدد تصادفی Ri تولید می کند. با Ri ،Ui APWi = h (PWi || Ri) ،SRi = Ri ⊕ (IDi || PWi) ،SH IDi = H IDi ⊕ h (PWi || IDi || Ri) و Vi = h را محاسبه می کند (APWi | | IDi || Ri).

سرانجام ، {SRi ،Ui ،SH IDi ،Vi ،PIDi ،h (.)} را در کارت هوشمند ذخیره می کند.

۳,۵مرحله ورود و اعتبارسنجی:

برای دسترسی به اطلاعات سنسور Sj ، کاربر Ui یک پیام درخواست ورود به درگاه GW ارسال می کند. در شکل ۷ ، مرحله ورود و احراز هویت را شرح می دهیم و جزئیات در زیر ارائه شده است.

شکل 7. مرحله ورود و احراز هویت WSN-SLAP.

مرحله ۱: پس از قرار دادن کارت هوشمند ، شناسه ID و رمز عبور PWi را وارد می کند. کارت هوشمند R * 1 = SRi ⊕ h (IDi || PWi)، APW * i = h (PWi || Ri) و V * i = h (APW * i || IDi || R * 1) را محاسبه می کند. سپس ، کارت هوشمند اعتبار V * i را در مقایسه با Vi ذخیره شده در کارت هوشمند بررسی می کند. در صورت تأیید اعتبار ، کارت هوشمند غیر تصادفی N1 تولید می کند و H IDi = SH IDi ⊕ h (PWi || IDi || Ri)، Si = SIDj ⊕h (PIDi || H IDi)، M1 = N1 comp h (H IDi || PIDi) و V1 = h (SIDj || PIDi || N1 || H IDi) را محاسبه می کند. سرانجام ، Ui از طریق یک کانال عمومی { PIDi، Si، M1، V1} را به GW ارسال می کند.

مرحله ۲: وقتی {PIDi ،GW ،Si ،M1 ،V1} را از Ui دریافت می کند ، GW PIDi و مقدار مخفی مشترک x را از پایگاه داده GW بازیابی می کند. سپس ، SID * j = Si ⊕ h (PIDi || H ID ،GW H ID * i = PIDi ⊕ h (x || kGWN) (i *، M1 = N * 1 ⊕ h (H ID * i || PIDi || N * 1 || H ID) و PIDi) را محاسبه می کند. V * 1 = h (SID * j || PIDi || N * 1 || H ID i *) ، و اعتبار V * 1 را در مقایسه با V1 بررسی می کند. در صورت تأیید اعتبار ، GW SIDj و h (SIDj || Rj) را از پایگاه داده GW بازیابی می کند. GW KSj = h (h (SIDj || Rj) || kGWN) ،M2 = h (N2 || H IDi) ⊕ h (KSj || PIDi) ، M3 = N1 ⊕ ساعت (h (N2 || H IDi) را محاسبه می کند) || KSj) و V2 = h (PIDi || SIDj || ساعت (N2 || H IDi) || N1). سرانجام ، GW از طریق یک کانال عمومی { PIDi ،M2 ،M3 ،V2} را به Sj ارسال می کند.

مرحله ۳: اگر {PIDi ،Sj ،M2 ،M3 ،V2} را دریافت کرد ، Sj محاسبه M2 = * (N2 || H IDi) ⊕ h ساعت (KSj || N2 || H IDi) || N ساعت (V * 2 = h (PIDi || SIDj ،N * 1 = M3 (h (h (N2 | | H IDi) * || PIDi) ،PIDi) 1 *) و اعتبار V V 2 را در مقایسه با پارامتر V2 بررسی می کند. اگر اعتبار تأیید شود ، Sj محاسبه SK = h (h (N2 || H IDi) || N3 || N1) ،M4 = N3⊕h (KSj || N2) ،SIDj (V3 = h (SK || N3 ||) ، جایی که SK کلید جلسه است. سرانجام ، {M4 ،Sj ،V3} را به GW ارسال می کند.

مرحله ٤: GW پس از دریافت پیام {M4، V3} از Sj، SK * = h (h (N2 || H ،N * 3 = M4 ⊕ h (KSj || N2) (V * 3 = h (SK * || N * 3 || SIDj ،IDi) || N * 3 | | N1) و برابری V * 3 و V3 را تأیید می کند. اگر تأیید موفقیت آمیز باشد ، GW یک N2 غیر تصادفی تولید می کند و xnew = h (x || N2) ،PIDnewi = H IDi ⊕ h (xnew || Pi = PIDnewi (h (N1 || H IDi ،kGWN) را محاسبه می کند) M6 = ،M5 = N2 ⊕ h (H IDi || SIDj || N1) ساعت (N2 || H IDi || PIDnewi) ⊕ N3 و V4 = h (N2 || N3 || PIDnewi || SK). سرانجام ، {Pi ،GW ،M5 ،M6 ،V4} را به Ui ارسال می کند و در صورت موفقیت در توافق نامه ، {x ،PIDi} را به {xnew ،PIDnewi} به روز می کند.

مرحله ۵: وقتی Ui از GW پیام {Pi، M5، M6، V4} را دریافت می کند ، Ui محاسبه می کند PIDnewi = Pi ⊕ h (N1 || H IDi) ،N * 2 = M5 ⊕ h (H IDi || SIDj || N1) ،N * 3 = M6 ⊕ ساعت (N * 2 || H IDi || PIDnewi) ساعت (ساعت (N * 3 || N1) ،N * 2 || H IDi) = V 4 ،ساعت (SK || PIDnewi || N * 3 || N * 2 *) و اعتبار V * 4 را در مقایسه با V4 بررسی می کند. در صورت تأیید اعتبار ، Ui جای کارت {PIDi} را به {PIDnewi} کارت هوشمند می دهد.

٤,٥ مرحله به روزرسانی رمز عبور:

در WSN-SLAP ، کاربران می توانند به راحتی رمز ورود خود را تغییر دهند. جزئیات به صورت زیر نشان داده شده است.

مرحله ۱: پس از قرار دادن کارت هوشمند ، کاربر Ui شناسه IDi و رمز عبور PWi را وارد می کند. کارت هوشمند $R * i =$ $SRi \oplus h (IDi \| PWi)$ ، $APW * i = h (PWi \| Ri)$ ، $V * i = h (APWi \| IDi \| R * i)$ را محاسبه می کند و برابری را تأیید می کند از $V * i$ و Vi. اگر تأیید موفقیت آمیز باشد ، کارت هوشمند رمز عبور جدیدی را به Ui درخواست می کند.

مرحله ۲: رمز عبور جدید PWnewi را وارد می کند. کارت هوشمند یک عدد Rnewi تصادفی انتخاب می کند و APWnewi $= h (PWnewi \| Rnewi)$، $SRnewi = Rnewi \oplus (IDi \| PWnewi)$ ، $SH IDnewi = H IDi \oplus h (PWnewi \| IDi \| Rnewi)$، $Vnewi =$ $h (APWnewi \| IDi \| Rnewi)$ را محاسبه می کند . سرانجام ، کارت های هوشمند {SRnewi، SH IDnewi، Vnewi، PIDi، h (.)} را ذخیره می کند.

٥,٥ مرحله افزودن گره سنسور:

برای افزودن گره حسگر جدید Snewj به WSN-SLAP، Snewj در GW دروازه ثبت می شود. مراحل دقیق به شرح زیر شرح داده شده است.

مرحله ۱: Snewj هویت خود را SIDnewj انتخاب می کند. سپس ، Snewj یک عدد تصادفی Rnewj تولید می کند. با SIDnewj و Rnewj، $Snewj\ h (SIDnewj \| Rnewj$) را محاسبه می کند و {SIDnewj، Rnewj، $h (SIDnewj \| Rnewj)$} را از طریق یک کانال امن به GW می فرستد.

مرحله ۲: GW پس از دریافت {SIDnewj، Rnewj، $h (SIDnewj \|$ از Snewj، $KSnewj = h (h (SIDnewj \| kGWN \| (Rnewj)$ را محاسبه می کند و {SIDnewj، Rnewj، $h (SIDnewj \| Rnewj)$} را در پایگاه داده GW. سرانجام ، GW {KSnewj} را به Snewj ارسال می کند.

مرحله ۳: Snewj پیام {KSnewj} را از GW دریافت می کند و {KSnewj} را در حافظه Snewj ذخیره می کند.

٦. تحلیل امنیت:

WSN-SLAP نه تنها ویژگی های سبک را با استفاده از توابع هش و عملیات XOR در نظر می گیرد ، بلکه سطح امنیتی بالاتری را در مقایسه با طرح های مربوطه تضمین می کند. برای ارزیابی امنیت WSN-SLAP ، ما تجزیه و تحلیل امنیت غیر رسمی و تجزیه و تحلیل رسمی امنیتی مانند منطق BAN ، مدل ROR و ابزار شبیه سازی AVISPA را انجام می دهیم. ما نشان می دهیم که WSN-SLAP با استفاده از تجزیه و تحلیل غیررسمی از انواع حملات جلوگیری می کند. ما احراز هویت متقابل WSN-SLAP را با استفاده از منطق BAN نشان می دهیم و همچنین امنیت کلیدی جلسه WSN-SLAP را با استفاده از مدل ROR اثبات می کنیم. ما برای اثبات ویژگی های امنیتی WSN-SLAP در برابر حملات پخش و MITM از ابزار شبیه سازی AVISPA استفاده می کنیم.

۱,٦ تحلیل امنیت غیر رسمی:

WSN-SLAP امنیت را در برابر حملات مختلف از جمله خودی ، کارت هوشمند دزدیده شده ، پخش مجدد ، ضبط گره حسگر ، حدس رمز عبور خارج از خط ، خودی ممتاز ، تأیید کننده سرقت و حملات MITM فراهم می کند. علاوه بر این ، WSN-SLAP رازداری کامل و احراز هویت متقابل را تضمین می کند.

۱,۱,٦ حمله داخلی:

اگر یک دشمن A به عنوان یک کاربر قانونی در GW Geway ثبت نام کند ، A می تواند برای GW و گره سنسور Sj احراز هویت کند. A پیام های { PIDi، M2، M3، {V2، M4}، V3} و { Pi، M5، M6، V4} را ضبط می کند. سپس ، A ساعت (N2 || H IDi) ⊕ h (KSj || PIDi) = M2 و h (h (N2 || H IDi) || KSj) = M3 ⊕ N1 را محاسبه می کند. برای به خطر انداختن سایر جلسات کاربر قانونی ، A برای محاسبه کلید جلسه باید به KSj نیاز داشته باشد. از آنجا که توابع هش nonce تصادفی N2 و پارامتر مخفی کاربر H IDi مانند h (h (N2 || H IDi) || KSj) را می پوشاند ، A نمی تواند پارامتر مخفی مشترک KSj را بین GW و Sj محاسبه کند. بنابراین ، WSN-SLAP در برابر حملات خودی ایمن است.

۲,۱,٦ حمله کارت هوشمند سرقت شده:

فرض کنید یک دشمن کارت هوشمند کاربر Ui را ضبط کند. سپس ، A از حمله تجزیه و تحلیل قدرت برای استخراج پارامترهای ذخیره شده در کارت هوشمند استفاده می کند. با پارامترهای کارت هوشمند Ui، A سعی می کند با درگاه GW و گره سنسور Sj احراز هویت کند. با این حال ، A نمی تواند پیام درخواست ورود به سیستم { PIDi، Si، M1، V1} را محاسبه کند زیرا

H IDi توسط (SH IDi = H IDi ed h (PWi || IDi || Ri پوشانده می شود. برای محاسبه H IDi، A باید همزمان IDi و PWi را حدس بزنند. از آنجا که این وظایف از نظر محاسباتی غیرقابل انجام است ، به دست آوردن IDi و PWi دشوار است. به همین دلایل ، WSN-SLAP در برابر حملات کارت های هوشمند دزدیده شده ایمن است.

۳،۱،٦پخش مجدد حمله:

اگر یک دشمن A پیام های PIDi، M2، M3، V2} و { IDi، Si، M1، V1} را از یک کاربر حقوقی Ui رهگیری کند ، A با ارسال پیام های رهگیری در جلسات دیگر سعی در تأیید اعتبار با GW دروازه دارد. در WSN-SLAP، GW و گره سنسور تازگی موارد غیر تصادفی N1، N2 و N3 را بررسی می کنند. بنابراین ، WSN-SLAP می تواند امنیت را در برابر حملات پخش مجدد تأمین کند.

٤،١،٦ حمله تسخیر گره سنسور:

فرض می کنیم که یک دشمن A گره حسگر خاص Sj را گرفته و با استفاده از حمله تحلیلی قدرت ، پارامترهای {KSj، SIDj} را از حافظه Sj بدست می آورد. سپس ، A می تواند با GW دروازه و کاربر UI احراز هویت کند. با این حال ، A نمی تواند گره های حسگر دیگر را تهدید کند. از آنجا که پارامتر مخفی مشترک (KSj = h (h (SIDj || Rj) || kGWN، A فقط می تواند با گره حسگر خاص Sj احراز هویت شود. A نمی تواند هیچ اطلاعاتی درباره گره های حسگر دیگر محاسبه کند. بنابراین ، WSN-SLAP در برابر حملات جذب گره سنسور ایمن است.

۵،۱،٦حمله حدس رمز خارج خط–خاموش:

مطابق بخش ۱،۲ ، یک دشمن A می تواند رمز ورود کاربر قانونی UI PWi را حدس بزند. A همچنین می تواند پارامترهای ذخیره شده { SRi، SH IDi، Vi، PIDi، h (.)} را از کارت هوشمند مجاز Ui استخراج کند. سپس ، A سعی می کند خود را به عنوان Ui معرفی کند. با این حال ، A نمی تواند (Ri = SRi (h (IDi || PWi را برای بدست آوردن H IDi = SH IDi (⊕ h (PWi || IDi || Ri بدون دانستن شناسه IDi محاسبه کند. بنابراین ، A نمی تواند پیام قانونی {PIDi، M2، M3، V2} را محاسبه کند. بر این اساس ، WSN-SLAP در برابر حملات حدس رمز خط–خاموش مقاومت دارد.

٦,١,٦. حمله خودی خاص:

اگر یک دشمن داخلی مخفی A پیام ثبت نام Ui کاربر قانونی {IDi} را رهگیری کند ، A سعی می کند با استفاده از پیام های بخش ٥,٣ ، کلید جلسه Ui را محاسبه کند. با این حال ، A نمی تواند کلید جلسه Ui را محاسبه کند. برای محاسبه SK = h (h (N2 || H IDi) || N3 || N1)، A باید H IDi را محاسبه کند که پارامتر مخفی مشترک بین Ui و GW دروازه است. با این حال ، A نمی تواند (H IDi = SH IDi ⊕ h (PWi || IDi || Ri را از پیام درخواست ورود { PIDi، Si، M1، V1} بدون رمز ورود Ui و عدد تصادفی Ri محاسبه کند. در نتیجه ، WSN-SLAP امنیت در برابر حملات خودی ممتاز را تضمین می کند.

٧,١,٦ حمله تایید کننده سرقت :

با فرض اینکه یک حریف A جدول درگاه ورودی GW را شامل { SIDj، Rj || SIDj) h)} و (PIDi، x) می دزدد. با این حال ، A نمی تواند کلید جلسه کاربر قانونی Ui را با این پارامترها محاسبه کند. برای محاسبه کلید جلسه SK = h (h (N2 || HIDi)، A باید با استفاده از PIDi = H IDi ⊕ h (x || kGWN) H IDi را محاسبه کند. از آنجا که پارامتر kGWN کلید اصلی GW است ، A نمی تواند H IDi را محاسبه کند. بنابراین ، WSN-SLAP در برابر حملات تأیید کننده سرقت مقاومت می کند.

٨,١,٦حمله MITM:

در مرحله ورود و تأیید اعتبار ، یک دشمن A رهگیری می کند و سعی می کند پیام درخواست ورود به سیستم {PIDi، Si، M1، V1} را اصلاح کند. با این حال ، GW دروازه با استفاده از جدول تأیید می تواند پیام تغییر یافته را به راحتی تشخیص دهد. علاوه بر این ، اصلاح همه پیام ها غیرممکن است زیرا آنها شامل پارامترهای تصادفی هستند. بنابراین ، WSN-SLAP می تواند از حملات MITM جلوگیری کند.

٩,١,٦حمله نشت تعداد تصادفی ویژه-جلسه:

فرض کنید که یک دشمن A تمام پارامترهای تصادفی N1، N2 و N3 را بدست آورد. سپس ، A سعی می کند کلید جلسه SK را محاسبه کند. با این وجود محاسبه کلید جلسه بدون دانستن H IDi غیرممکن است. H IDi در طول جلسه با کلید مخفی x و کلید اصلی kGWN پوشانده می شود. بر این اساس ، WSN-SLAP در برابر حملات نشت تعداد تصادفی ویژه–جلسه ایمن است.

۱۰،۱،٦محرمانگی پیشرو کامل:

ما فرض می کنیم که یک دشمن A کلید اصلی GW kGWN را بدست می آورد. سپس ، A سعی می کند کلید جلسه SK = h (h (h (N2 || HIDi) || N3 || N1 کاربر Ui را محاسبه کند. با این حال ، از کلید اصلی kGWN استفاده می شود ، یعنی h (x || kGWN)) و h (h (SIDj || Rj) || kGWN). بنابراین ، A برای تحلیل پارامتر مخفی به پارامتر مخفی مشترک x یا h (SIDj || Rj)) نیاز دارد. به همین دلیل ، WSN-SLAP محرمانگی پیشرو کامل را فراهم می کند.

۱۱،۱،٦. احراز هویت متقابل:

برای احراز هویت با یکدیگر ، هر یک از شرکت کنندگان WSN-SLAP فرایندهای تأیید را انجام می دهد. GW دروازه اعتبار V1؟ = 1 * V و V3؟ = 3 * V را بررسی می کند ، گره سنسور Sj V2؟ = 2 * V را تأیید می کند و Ui V4؟ = 4 * V را بررسی می کند. اگر کل روند تأیید موفقیت آمیز باشد ، می توانیم نتیجه بگیریم که هر شرکت کننده با یکدیگر احراز هویت می شود. بنابراین ، WSN-SLAP احراز هویت متقابل را تضمین می کند.

۲،٦منطق BAN:

در این بخش ، ما احراز هویت متقابل WSN-SLAP را با استفاده از تحلیل منطق BAN اثبات می کنیم [۸]. منطق BAN به طور گسترده ای برای تحلیل احراز هویت متقابل در طرح های مختلف احراز هویت استفاده شده است [۳۲،۳۳]. در WSN-SLAP ، شرکت کنندگان با احراز هویت یکدیگر یک کلید جلسه SK را در میان U، GW و SN ایجاد می کنند. جدول ۲ نشانه های اساسی منطق BAN را در این اثبات ارائه می دهد.

جدول2. عبارات اصلی

عبارت	تعریف
P_1, P_2	دو اصل
S_1, S_2	دو جمله
SK	کلید جلسه
$P_1 \mid\equiv S_1$	S1 معتقد است P1
$P_1 \mid\sim S_1$	S1 یک بار گفت P1
$P_1 \Rightarrow S_1$	S1کنترل می کند P1
$P_1 \lhd S_1$	P1 را دریافت می کند S1
$\#S_1$	S1 تازه است
$\{S_1\}_{Key}$	S با1 رمزگذاری شده Key است
$P_1 \overset{Key}{\longleftrightarrow} P_2$	P1 و P2 کلید مشترک را دارند

۱,۲,٦قوانین:

قوانین منطقی منطق BAN به شرح زیر است.

۱. قانون معنی پیام (MMR):

$$\frac{P_1 \mid\equiv P_1 \overset{Key}{\leftrightarrow} P_2, \quad P_1 \lhd (S_1)_{Key}}{P_1 \mid\equiv P_2 \mid\sim S_1}$$

۲. قانون اعتبارسنجی غیر رسمی (NVR):

$$\frac{P_1 \mid\equiv \#(S_1), \quad P_1 \mid\equiv P_2 \mid\sim S_1}{P_1 \mid\equiv P_2 \mid\equiv S_1}$$

۳. قانون صلاحیت (JR):

$$\frac{P_1 \mid\equiv P_2 \mid\Longrightarrow S_1, \quad P_1 \mid\equiv P_2 \mid\equiv S_1}{P_1 \mid\equiv S_1}$$

٤. قانون اعتقاد (BR):

$$\frac{P_1 \mid \equiv (S_1, S_2)}{P_1 \mid \equiv S_1}$$

۵. قانون تازگی (FR):

$$\frac{P_1 \mid \equiv \#(S_1)}{P_1 \mid \equiv \#(S_1, S_2)}$$

۲,۲,٦. اهداف:

در WSN-SLAP ، اهداف اساسی منطق BAN این است که هر مدیر اصلی یک کلید جلسه ایجاد می کند و به تأیید اعتبار متقابل می رسد. اهداف اثبات احراز هویت متقابل WSN-SLAP به شرح زیر است:

Goal 1: $U| \equiv U \overset{SK}{\longleftrightarrow} GW$

Goal 2: $U| \equiv GW| \equiv U \overset{SK}{\longleftrightarrow} GW$

Goal 3: $GW| \equiv U \overset{SK}{\longleftrightarrow} GW$

Goal 4: $GW| \equiv U| \equiv U \overset{SK}{\longleftrightarrow} GW$

Goal 5: $SN| \equiv SN \overset{SK}{\longleftrightarrow} GW$

Goal 6: $SN| \equiv GW| \equiv SN \overset{SK}{\longleftrightarrow} GW$

Goal 7: $GW| \equiv SN \overset{SK}{\longleftrightarrow} GW$

Goal 8: $GW| \equiv SN| \equiv SN \overset{SK}{\longleftrightarrow} GW$

۳,۲,٦ فرم های ایده آل:

در WSN-SLAP ، درخواست احراز هویت و پیام های پاسخ { PIDi، Si، M1، {V1، PIDi}، M2، M3، {V2، M4}، V3} و { Pi، M5، M6، V4} از طریق یک کانال عمومی ما این پیام ها را به فرم ایده آل منتقل خواهیم کرد و پیام های دیگر را حذف خواهیم کرد زیرا آنها نمی توانند به طور کارآمد خصوصیات منطقی منطق BAN را ارائه دهند. پیام های فرم ایده آل WSN-SLAP به صورت زیر نشان داده شده است:

$$Msg_1 : U \rightarrow GW : \{N_1, SID_j\}_{HID_i}$$

$$Msg_2 : GW \rightarrow SN : \{h(N_2||HID_i), N_1\}_{KS_j}$$

$$Msg_3 : SN \rightarrow GW : \{N_3\}_{KS_j}$$

$$Msg_4 : GW \rightarrow U : \{N_2, N_3\}_{HID_1}$$

٤,٢,٦فرضیات:

پس از مرحله ثبت نام ، هر مدیر اصلی معتقد است که کلیدهای مخفی دارد که بین یکدیگر به اشتراک گذاشته شده است. مدیر اصلی همچنین اعتماد دارد که اعداد تصادفی و هویت شبه تازه است. علاوه بر این ، مدیر اصلی معتقد است که یک مدیر قانونی می تواند اجزا و ارزش های تحت کنترل را کنترل کند. فرضیات منطق BAN در WSN-SLAP به شرح زیر است:

$$A_1: GW| \equiv \#(N_1)$$

$$A_2: GW| \equiv \#(N_3)$$

$$A_3: SN| \equiv \#(h(N_2||HID_i))$$

$$A_4: U| \equiv \#(N_2)$$

$$A_5: U| \equiv GW \Rightarrow (U \overset{SK}{\longleftrightarrow} GW)$$

$$A_6: GW| \equiv U \Rightarrow (U \overset{SK}{\longleftrightarrow} GW)$$

$$A_7: SN| \equiv GW \Rightarrow (SN \overset{SK}{\longleftrightarrow} GW)$$

$$A_8: GW| \equiv SN \Rightarrow (SN \overset{SK}{\longleftrightarrow} GW)$$

$$A_9: U| \equiv U \overset{HID_i}{\longleftrightarrow} GW$$

$$A_{10}: GW| \equiv U \overset{HID_i}{\longleftrightarrow} GW$$

$$A_{11}: SN| \equiv SN \overset{KS_j}{\longleftrightarrow} GW$$

$$A_{12}: GW| \equiv SN \overset{KS_j}{\longleftrightarrow} GW$$

۵،۲،۶ اثبات منطق **BAN**:

ما تحلیل منطق BAN از WSN-SLAP را به صورت زیر انجام می دهیم:

مرحله ۱: S1 را می توان از Msg1 بدست آورد.

$$S_1 : GW \lhd \{N_1, SID_j\}_{HID_i}$$

مرحله ۲: با استفاده از MMR با استفاده از S1 و A10 می توان S2 را القا کرد.

$$S_2 : GW| \equiv U| \sim (N_1, SID_j)$$

مرحله ۳: S3 را می توان با استفاده از FR با استفاده از S2 و A1 القا کرد.

$$S_3 : GW| \equiv \#(N_1, SID_j)$$

مرحله ٤: S4 را می توان با استفاده از NVR با استفاده از S2 و S3 القا کرد.

$$S_4 : GW| \equiv U| \equiv (N_1, SID_j)$$

مرحله ٥: S5 توسط S4 و BR القا می شود.

$$S_5 : GW| \equiv U| \equiv (N_1)$$

مرحله ٦: S6 از Msg2 بدست می آید.

$$S_6 : SN \lhd \{h(N_2||HID_i), N_1\}_{KS_j}$$

مرحله ۷: با استفاده از MMR با استفاده از S6 و A13 می توان S7 را القا کرد.

$$S_7 : SN| \equiv GW| \sim (h(N_2||HID_i), N_1)$$

مرحله ۸: با استفاده از FR با استفاده از S7 و A3 می توان S8 را القا کرد.

$$S_8 : SN| \equiv \#(h(N_2||HID_i), N_1)$$

مرحله ۹: با استفاده از NVR با استفاده از S7 و S8 می توان S9 را القا کرد.

$$S_9 : SN| \equiv GW| \equiv (h(N_2||HID_i), N_1)$$

مرحله ۱۰: S10 از Msg3 بدست می آید.

$$S_{10} : GW \lhd \{N_3\}_{KS_j}$$

مرحله ۱۱: با استفاده از MMR با استفاده از A5 و S8 می توان S11 را القا کرد.

$$S_{11} : GW| \equiv SN| \sim (N_3)$$

مرحله ۱۲: با استفاده از NVR با استفاده از S9 و S10 می توان S12 را القا کرد.

$$S_{12} : GW| \equiv SN| \equiv (N_3)$$

مرحله ۱۳: S13 و S14 می توانند توسط S9 و S12 القا شوند. SN و GW می توانند کلید جلسه SK = h (h (N2 || H (IDi) || N3 || N1 را محاسبه کنند.

$$S_{13} : GW| \equiv SN| \equiv (SN \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 8)}$$
$$S_{14} : SN| \equiv GW| \equiv (SN \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 6)}$$

مرحله ١٤: S15 و S16 را می توان با استفاده از JR به ترتیب با استفاده از S13 و A8 و S14 و A7 القا کرد.

$$S_{15} : GW| \equiv (SN \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 7)}$$
$$S_{16} : SN| \equiv (SN \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 5)}$$

مرحله ۱۵: S17 از Msg4 بدست می آید.

$$S_{17} : U \lhd \{N_2, N_3\}_{HID_i}$$

مرحله ١٦: S18 را می توان توسط A9، S17 و MMR القا کرد.

$$S_{18} : U| \equiv GW| \sim (N_2, N_3)$$

مرحله ۱۷: با استفاده از FR با استفاده از S18 و A4 می توان S19 را القا کرد.

$$S_{19} : U| \equiv \#(N_2, N_3)$$

مرحله ۱۸: S20 توسط S16، S17 و NVR القا می شود.

$$S_{20} : U| \equiv GW| \equiv (N_2, N_3)$$

مرحله ۱۹: S21 و S22 را می‌شود توسط S5، S18 القا کرد. U و GW می توانند کلید جلسه SK = h (h (N2 || H IDi) || N3 || N1) را محاسبه کنند.

$$S_{21} : U| \equiv GW| \equiv (U \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 2)}$$
$$S_{22} : GW| \equiv U| \equiv (U \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 4)}$$

مرحله ۲۰: S23 و S24 را می توان با استفاده از JR به ترتیب با استفاده از S21 و A5، S22 و A6 القا کرد.

$$S_{23} : U| \equiv (U \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 1)}$$
$$S_{24} : GW| \equiv (U \overset{SK}{\longleftrightarrow} GW) \quad \text{(Goal 3)}$$

۳,٦ مدل ROR:

این بخش با استفاده از مدل شناخته شده Real-Or-Random (ROR) امنیت کلید جلسه WSN-SLAP را اثبات می کند [۹]. در WSN-SLAP ، سه شرکت کننده وجود دارد. Pt1U یک کاربر ، Pt2GW یک دروازه است و Pt2GW یک گره حسگر است. در مدل ROR ، شبکه تحت یک دشمن A قرار دارد که می تواند پیام ها را شنود ، ضبط ، درج و حذف کند. با این توانایی ها ، A حملات مختلفی را با استفاده از پرس و جوهای Execute، CorruptSC، Reveal، Send و Test انجام می دهد.

• Execute: این کوئری یک حمله منفعل است که A می تواند پیام شخص حقوقی را استراق سمع کند.

• CorruptSC: این سeryال به این معنی است که A پارامترهای ذخیره شده را از کارت هوشمند کاربر به دست می آورد.

• Reveal: این عبارت به معنای A است که کلید جلسه SK را نشان می دهد.

• Send: این کوئری یک حمله فعال است که A برای دریافت پیام پاسخ پیامی ارسال می کند.

• تست: یک حریف A قبل از شروع بازی سکه بی طرفانه بدست می آورد. اگر A c = 1 بدست آورد ، به این معنی است که کلید جلسه SK تازه است. اگر A c = 0 بدست آورد ، به این معنی است که کلید جلسه تازه نیست. در غیر این صورت ، A مقدار NULL بدست می آورد. برای اطمینان از امنیت کلید جلسه ، لازم است که A نتواند مقدار نتیجه را بین یک عدد تصادفی و کلید جلسه تشخیص دهد.

اثبات امنیت:

قضیه ۱. اجازه دهید تلاش برای بدست آوردن کلید جلسه WSN-SLAP در زمان چند جمله ای به شرح زیر باشد. AdvA (Poly)) به ترتیب احتمال شکسته شدن کلید جلسه توسط A. q2h، HASH و qsend به ترتیب تعداد پرسشهای هش ، فضای محدوده عملکرد هش و تعداد پرسشهای ارسال است. s0 و C0 پارامترهای Zipf هستند [۳٤].

$$Adv_{\mathcal{A}}(Poly) \leq \frac{q_h^2}{|HASH|} + 2\{C' q_{send}^{s'}\}$$

ما طبق روش [۳۵،۳٦] از اثبات پیروی می کنیم. ما چهار بازی Gamek را انجام می دهیم ، جایی که k ∈ [0، SuccA .]3، Gamek رویدادی است که A می تواند یک بیت صحیح c را در Gamek حدس بزند ، و Gamek ،Pr [SuccA] احتمال SuccA، Gamek است. با این پارامترها می توانیم Gamek را به صورت زیر انجام دهیم.

- Game0: این بازی حمله واقعی A در WSN-SLAP تحت مدل ROR را توصیف می کند. قبل از شروع بازی باید بیت تصادفی c انتخاب شود. بنابراین ، می توانیم به شرح زیر استنتاج کنیم.

$$Adv_A(Poly) = |2Pr[Succ_{\mathcal{A},Game_0}] - 1| \qquad (1)$$

- Game1: در Game1، A با استفاده از Execute پیام های موجودیت {PIDi، Si، M1، {PIDi، V1}، M2، M3، {V2، {M4، V3} و {Pi، M5، M6، V4 را به دست می آورد. پرس و جو. سپس ، A پرسش های Test and Reveal را برای به دست آوردن کلید جلسه SK انجام می دهد. از آنجا که SK = h (h (N2 || H IDi) || N3 || N1)، A باید موارد غیر تصادفی N1، N2 و N3 بدست آورد. علاوه بر این ، A به هویت پوشیده کاربر H IDi نیاز دارد. به همین دلایل ، A نمی تواند SK را محاسبه کند. این بدان معنی است که Game0 و Game1 قابل تشخیص نیستند. بنابراین ، می توانیم معادل زیر را بدست آوریم.

$$Pr[Succ_{\mathcal{A},Game_1}] = Pr[Succ_{\mathcal{A},Game_0}] \qquad (2)$$

- Game2: در این بازی ، A جستجوی Send را انجام می دهد که یک حمله فعال است. A با استفاده از { PIDi، Si، M1، {PIDi ،V1}، M2، M3، {V2، {M4، V3} ، و { Pi، M5، M6، V4} برای دریافت کلید جلسه SK. پارامترهای V1، V2، V3 و V4 توسط پرس و جو HASH پوشانده می شوند. علاوه بر این ، پارامترهای PIDi، M1، M2، M3، M4، M5، M6 و Pi حاوی موارد غیر تصادفی N1، N2 و N3 هستند. با استفاده از ناچ های تصادفی ، می توانیم از جلسات دیگر جلوگیری کنیم. با توجه به پارادوکس تولد [۳۷] ، می توانیم نابرابری زیر را بدست آوریم.

$$|Pr[Succ_{\mathcal{A},Game_2}] - Pr[Succ_{\mathcal{A},Game_1}]| \leq \frac{q_h^2}{|HASH|} \qquad (3)$$

- Game3: در Game3، A با استفاده از حمله تحلیل قدرت ، پارامترهای ذخیره شده کارت هوشمند {SRi، SH IDi، Vi، PIDi} را به دست می آورد ، جایی که SRi = Ri ⊕ h (IDi || PWi)، SH IDi = H IDi ⊕ h (PWi || IDi || Ri)، Vi = h (APWi || IDi || Ri) و PIDi = H IDi ⊕ h (x || kGWN). برای به دست آوردن Ri و H IDi، A به شناسه هویت و رمز عبور PWi نیاز دارد. بنابراین ، A نمی تواند با Game2 و Game3 تشخیص دهد که آیا حدس زدن PWi از نظر محاسباتی یک کار غیرقابل اجرا است. سپس ، می توانیم با استفاده از قانون Zipf نتیجه را بدست آوریم [۳٤].

$$|Pr[Succ_{\mathcal{A},Game_3}] - Pr[Succ_{\mathcal{A},Game_2}]| \leq C' q_{send}^{s'} \qquad (4)$$

سرانجام ، A بیت حدس زده شده c را بدست می آورد زیرا بازی ها تمام شده اند.

$$Pr[Succ_{\mathcal{A},Game_3}] = \frac{1}{2} \qquad (5)$$

علاوه بر این ، با استفاده از (۱) و (۲) می توانیم نتیجه زیر را بدست آوریم.

$$\frac{1}{2}Adv_{\mathcal{A}}(Poly) = |Pr[Succ_{\mathcal{A},Game_0}] - \frac{1}{2}| = |Pr[Succ_{\mathcal{A},Game_1}] - \frac{1}{2}| \qquad (6)$$

با استفاده از (۵) و (٦) معادله زیر را بدست می آوریم.

$$\frac{1}{2}Adv_{\mathcal{A}}(Poly) = |Pr[Succ_{\mathcal{A},Game_1}] - Pr[Succ_{\mathcal{A},Game_3}]| \qquad (7)$$

با استفاده از نابرابری مثلثی نتیجه زیر را بدست می آوریم.

$$\begin{aligned}\frac{1}{2}Adv_{\mathcal{A}}(Poly) &= |Pr[Succ_{\mathcal{A},Game_1}] - Pr[Succ_{\mathcal{A},Game_3}]| \\ &\leq |Pr[Succ_{\mathcal{A},Game_1}] - Pr[Succ_{\mathcal{A},Game_2}]| \\ &\quad + |Pr[Succ_{\mathcal{A},Game_2}] - Pr[Succ_{\mathcal{A},Game_3}]| \\ &\leq \frac{q_h^2}{2|HASH|} + C' q_{send}^{s'} \qquad (8)\end{aligned}$$

با ضرب (۸) در ۲ ، نتیجه زیر را بدست می آوریم.

$$Adv_A(Poly) \leq \frac{q_h^2}{|HASH|} + 2\{C' q_{send}^{s'}\}$$

٤,٦ شبیه سازی AVISPA:

در این بخش ، ما با استفاده از AVISPA [10،11] ویژگی های امنیتی WSN-SLAP را تجزیه و تحلیل می کنیم. AVISPA یک ابزار رسمی تأیید امنیت است که MITM را شناسایی کرده و حملات را بر علیه پروتکل احراز هویت پخش می کند. AVISPA از زبان مشخصات پروتکل های سطح بالا (HLPSL) استفاده می کند. پس از دریافت پروتکل نوشته شده در HLPSL ، مترجم پروتکل مبتنی بر HLPSL را به قالب متوسط (IF) تبدیل می کند. سپس ، مترجم ورودی IF را به چهار قسمت انتهایی وارد می کند که عبارتند از: (CL-AtSe) Contraint Logic-Attack Searcher، Tree Automata براساس تقریب خودکار برای تجزیه و تحلیل پروتکل امنیتی ((SATMC ،TA4SP) SAT-based Model Checker) و On-fly Model-Checker (OFMC) به ترتیب. در نتیجه ، IF به یک قالب خروجی (OF) تبدیل می شود. اگر خلاصه OF SAFE باشد ، به این معنی است که پروتکل در برابر حملات مجدد و MITM مقاومت دارد. به طور خاص ، OFMC back-end می تواند از عملیات XOR استفاده کند. بنابراین ، ما از این نتیجه نهایی در مقاله خود استفاده می کنیم.

١,٤,٦. مشخصات HLPSL:

در HLPSL، WSN-SLAP از کاربران UA، GWN دروازه و گره های حسگر SN تشکیل شده است. این موجودیت ها به عنوان نقش نوشته می شوند. همچنین دو نقش ترکیب به نام جلسه و محیط وجود دارد که شامل اهداف امنیتی است. شکل ۸ اهداف و نقش جلسه و محیط WSN-SLAP را نشان می دهد. شکل ۹ کل فرآیند UA کاربر را نشان می دهد. در حالت ۱ ، کاربر UA در GWN ثبت نام می کند. برای شروع جلسه ، UA پیام شروع را دریافت می کند. سپس ، UA از طریق یک کانال امن یک پیام درخواست ثبت نام {IDi} را به GWN دروازه می فرستد. در حالت ۲ ، UA یک کارت هوشمند از GWN دریافت می کند و { Ri، SRi، SH IDi، Vi} را در کارت هوشمند ذخیره می کند. در مرحله ورود و تأیید اعتبار ، UA از طریق یک کانال عمومی { PIDi، Si، M1، V1} را به GWN ارسال می کند. شاهد عملکرد (UA، GWN، ua_gw_n1، N10) نشان دهنده طراوت N1 تولید شده توسط UA است. در حالت ۳ ، {Pi، UA، M5، M6، V4} را از GWN دریافت می کند. سپس ، UA با استفاده از N02 درخواستی (GWN، UA، gw_ua_n3، N02) با GWN احراز هویت می کند.

```
role session(UA, SN, GWN : agent, SKuagwn, SKsngwn : symmetric_key, H: hash_func)
def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
user(UA, SN, GWN, SKuagwn, SKsngwn, H, SN1, RV1)
/\ sen(UA, SN, GWN, SKuagwn, SKsngwn, H, SN2, RV2)
/\ gate(UA, SN, GWN, SKuagwn, SKsngwn, H, SN3, RV3)
end role

role environment()
def=
const ua, sn, gwn : agent,
skuagwn, sksngwn: symmetric_key,
h: hash_func,
idi, pidi, sidj: text,
ua_gw_n1, gw_sn_n2, sn_gw_n3, gw_ua_n3: protocol_id,
sp1, sp2, sp3, sp4, sp5, sp6: protocol_id
intruder_knowledge = {idi, pidi, sidj, h}
composition
session(ua, sn, gwn, skuagwn, sksngwn, h)/\session(i, sn, gwn, skuagwn, sksngwn, h)
/\session(ua, i, gwn, skuagwn, sksngwn, h)
/\session(ua, sn, i, skuagwn, sksngwn, h)
end role

goal
secrecy_of sp1, sp2, sp3, sp4,sp5, sp6
authentication_on ua_gw_n1
authentication_on gw_sn_n2
authentication_on sn_gw_n3
authentication_on gw_ua_n3
end goal

environment()
```

شکل 8. نقش جلسه ، محیط و هدف

```
role user(UA, SN, GWN: agent, SKuagwn, SKsngwn : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by UA
def=
local State: nat,
   IDi, PWi, X, Rg, PIDi, APWi, Ri, SRi, SHIDi, Vi, HIDi, Kgwn, SIDj, Rj, KSj: text,
   N1, Si, M1, V1, N2, M2, M3, V2, N3, SK, M4, V3, Xnew, PIDinew, Mpsid, M5, M6, V4: text

const sp1, sp2, sp3, sp4, sp5, sp6, ua_gw_n1, gw_sn_n2, sn_gw_n3, gw_ua_n3: protocol_id
init State := 0
transition

%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ SND({IDi}_SKuagwn)
   /\ secret({IDi}, sp1, {UA,GWN})

%%%%%%%%%%Recieve smartcard
2. State = 1 /\ RCV ({xor(H(IDi.Rg'),H(X'.Kgwn)).H(IDi.Rg')}_SKuagwn)=|>
 State' := 2  /\ Ri' := new() /\ APWi' := H(PWi.Ri')
      /\ SRi' := xor(Ri', H(IDi.PWi))
      /\ SHIDi' := xor(H(IDi.Rg'), H(PWi.IDi.Ri'))
      /\ Vi' := H(APWi'.IDi.Ri')
      /\ secret({Ri',PWi}, sp2, UA)
%%%%%%%%%%Login & Authentication phase
      /\ N1' := new() /\ Si' := xor(SIDj, H(xor(H(IDi.Rg'),H(X'.Kgwn)).H(IDi.Rg')))
   /\ M1' := xor(N1', H(H(IDi.Rg').xor(H(IDi.Rg'),H(X'.Kgwn))))
   /\ V1' := H(SIDj.xor(H(IDi.Rg'),H(X'.Kgwn)).N1'.H(IDi.Rg'))
      /\ SND(xor(H(IDi.Rg'),H(X'.Kgwn)).Si'.M1'.V1')
      /\ witness(UA,GWN,ua_gw_n1,N1')


3. State = 2 /\ RCV(xor(xor(H(IDi.Rg'),H(H(X.N2').Kgwn)), H(N1'.H(IDi.Rg'))).xor(N2', H(H(IDi.Rg').SIDj.N1')).xor(N3',
H(N2'.H(IDi.Rg').xor(H(IDi.Rg'),H(H(X.N2').Kgwn)))).H(N2'.N3'.xor(H(IDi.Rg'),H(H(X.N2').Kgwn)).H(H(N2'.H(IDi.Rg')).N3'.N1'))) =|>
State' := 3 /\ request(GWN,UA,gw_ua_n3, N2')
end role
```

شکل 9. نقش کاربر

۲,٤,٦ نتیجه شبیه سازی:

اگر خلاصه نتیجه پروتکل در شبیه سازی OFMC SAFE باشد ، پروتکل در برابر حملات پخش و MITM مقاومت دارد. نتیجه ابزار شبیه سازی AVISPA WSN-SLAP با استفاده از OFMC back-end در شکل ۱۰ نشان داده شده است. بنابراین ، WSN-SLAP می تواند از حملات پخش و MITM جلوگیری کند.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/dk.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 26.60s
  visitedNodes: 2704 nodes
  depth: 12 plies
```

شکل 10. نتیجه تأیید خودکار پروتکل ها و برنامه های امنیتی اینترنت (AVISPA) .شبیه سازی

۷. تحلیل عملکرد:

در این بخش ، ما هزینه های محاسباتی ، هزینه های ارتباطی و خصوصیات امنیتی WSN-SLAP را در مقایسه با طرح های مرتبط موجود تخمین می زنیم [۶،۲۷،۲۸،۳۱].

۷,۱ هزینه های محاسباتی

ما هزینه محاسباتی WSN-SLAP را در مقایسه با عملکرد طرح های مربوطه تجزیه و تحلیل می کنیم [۶،۲۷،۲۸،۳۱]. طبق [۶،۳۸] ، زمان اجرای هر عملیات در رایانه ای با پردازنده چهار هسته ای ۳,۲ گیگاهرتز و ۸ گیگابایت حافظه بدست می‌آید. ما تخمین می زنیم که Th، Tecm و Tsym به ترتیب زمان اجرای تابع هش (۰،۰۰۰۳۲ s) ، ضرب نقطه (0170 s s 0 ECC) و رمزگذاری / رمزگشایی متقارن (≈۰,۰۰۵۶ ثانیه) است. ما زمان اجرای عملیات XOR را در نظر نمی گیریم زیرا قابل اغماض است.

جدول ۳ نتیجه هزینه های محاسباتی را نشان می دهد. بر این اساس ، WSN-SLAP هزینه محاسباتی کارآمدتری نسبت به طرح های مرتبط دارد [۶،۲۷،۲۸،۳۱].

Table 3. Computational costs comparison.

Schemes	User	Gateway	Sensor Node	Total	Total Cost (s)
Choi et al. [27]	$9T_h + 3T_{ecm}$	$6T_h + 2T_{ecm}$	$5T_h + 1T_{ecm}$	$20T_h + 6T_{ecm}$	0.109
Wu et al. [28]	$12T_h + 2T_{ecm} + 1T_{sym}$	$11T_h + 2T_{sym}$	$4T_h + 2T_{ecm} + 1T_{sym}$	$27T_h + 4T_{ecm} + 4T_{sym}$	0.09944
Wu et al. [31]	$13T_h + 2T_{ecm}$	$13T_h$	$4T_h + 2T_{ecm}$	$30T_h + 4T_{ecm}$	0.078
Moghadam et al. [6]	$5T_h + 3T_{ecm} + 2T_{sym}$	$5T_h + 3T_{ecm} + 2T_{sym}$	$3T_h + 2T_{ecm}$	$13T_h + 8T_{ecm} + 4T_{sym}$	0.16336
Ours	$13T_h$	$18T_h$	$6T_h$	$37T_h$	0.01184

۲،۷ هزینه‌های ارتباطی:

ما هزینه ارتباط WSN-SLAP را در مقایسه با طرح‌های مرتبط [۶،۲۷،۲۸،۳۱] در این بخش ارزیابی می کنیم. طبق [۶] ، تعریف می‌کنیم که هویت کاربر، هویت گره حسگر، عدد تصادفی، زمان سنجی، هش SHA-1 و نقطه ECC به ترتیب ۱۲۸ ، ۱۶ ، ۱۲۸ ، ۳۲ ، ۱۶۰ و ۳۲۰ بیت است. در WSN-SLAP، پیام درخواست ورود به سیستم { PIDi ، Si ، M1 ، + (160 + 160 V1} 160 + 160 = 640 بیت) و پیام های احراز هویت ارسال شده {PIDi ،M2 ،M3 ، {M4 ،V2}، V3} و {Pi ،M5 ،M6 ،V4} نیاز به (۱۶۰ + ۱۶۰ + ۱۶۰ + ۱۶۰ = ۶۴۰ بیت) ، (۱۶۰ + ۱۶۰ = ۳۲۰ بیت) و (۱۶۰ + ۱۶۰ + ۱۶۰ + ۱۶۰ = ۶۴۰ بیت). در نتیجه، کل هزینه‌های ارتباطی WSL-SLAP و طرح‌های مربوطه [۶،۲۷،۲۸،۳۱] در جدول ٤ نشان داده شده است. بنابراین، WSN-SLAP هزینه ارتباطی کارآمدتری را نسبت به طرح‌های مرتبط فراهم می‌کند [۶،۲۷،۲۸، ۳۱].

جدول 4. مقایسه هزینه های ارتباطی.

Schemes	Communication Costs	Number of Messages
Choi et al. [27]	3200 bits	4 messages
Wu et al. [28]	3296 bits	4 messages
Wu et al. [31]	3392 bits	4 messages
Moghadam et al. [6]	2512 bits	4 messages
Ours	2240 bits	4 messages

۳،۷ ویژگی‌های امنیتی:

در جدول ۵، ویژگی‌های امنیتی WSN-SLAP را با طرح‌های مرتبط ارائه دادیم [۶،۲۷،۲۸،۳۱]. ما نشان می‌دهیم که پروتکل‌های موجود [۶،۲۷،۲۸،۳۱] از حملات مختلفی رنج می‌برند، از جمله درونی، کارت هوشمند دزدیده شده و حملات نشت تعداد تصادفی خاص–جلسه. در پایان، WSN-SLAP عملکرد و ویژگی‌های امنیتی بهتری را در مقایسه با برنامه‌های مربوطه ارائه کرد [۶،۲۷،۲۸،۳۱].

جدول 5. ویژگی های امنیتی.

اجزا امنیتی	Choi et al. [27]	Wu et al. [28]	Wu et al. [31]	Moghadam et al. [6]	کار ما
حمله داخلی	o	o	×	×	o
حمله کارت هوشمند سرقت شده	×	×	×	o	o
پخش مجدد حمله	o	o	o	o	o
حمله تسخیر گره سنسور	o	o	o	o	o
حمله حدس رمز خط–خاموش	×	×	o	o	o
حمله خودی خاص	o	o	×	o	o
حمله تأیید کننده سرقت	×	o	o	o	o
حمله MITM	o	o	×	o	o
شماره تصادفی خاص جلسه حمله نشتی	×	×	×	×	o
محرمانگی پیشرو کامل	o	o	o	×	o
احراز هویت متقابل	o	o	o	o	o

o: ایمن در برابر حمله ×: ناامن در برابر حمله

۸. نتیجه گیری:

در این مقاله، یافتیم که طرح مقدم و همکاران دارای آسیب پذیری‌هایی در برابر حملات نشتی تعداد تصادفی خاص–جلسه است. همچنین ثابت کردیم که طرح مقدم و همکاران محرمانگی کامل را تضمین نمی‌کند. برای رفع نقاط ضعف امنیتی طرح مقدم و همکاران، یک پروتکل احراز هویت متقابل امن و سبک را برای محیط‌های WSN ارائه کردیم. WSN-SLAP در برابر حملات مختلف از جمله خودی، کارت هوشمند دزدیده شده، حدس رمز خط–خاموش، تأیید کننده سرقت و حملات نشت تعداد تصادفی خاص–جلسه مقاومت نشان می‌دهد. همچنین نشان دادیم که WSN-SLAP محرمانگی کامل و احراز هویت متقابل را برایمان فراهم می‌کند. امنیت WSN-SLAP را با استفاده از تحلیل رسمی امنیتی AVISPA، منطق BAN و مدل ROR، ثابت کردیم. علاوه بر این، WSN-SLAP دارای هزینه‌های محاسباتی و ارتباطی سبک است زیرا شامل عملیات XOR و توابع hash می‌باشد. بنابراین، WSN-SLAP پیشنهادی خدمات ارتباطی امن‌تر و کارآمدتری را در مقایسه با پروتکل‌های مرتبط موجود ارائه می‌دهد

و برای محیط‌های WSN مناسب است. در کارهای آینده، ما شبکه‌ای کامل با پروتکل‌های امنیتی‌اش را برای اجرا و پیاده سازی در WSN طراحی خواهیم کرد.

منابع :

1. Mandal, S.; Bera, B.; Sutrala, A.K.; Das, A.K.; Choo, K.K.R.; Park, Y. Certificateless-signcryption-based three-factor user access control scheme for IoT environment. IEEE Internet Things J. 2020, 7, 3184–3197.

2. Yu, S.; Park, Y. SLUA-WSN: Secure and lightweight three-factor-based user authentication protocol for wireless sensor networks. Sensors 2020, 20, 4143. [CrossRef]

3. Ghahramani, M.; Javidan, R.; Shojafar, M.; Taheri, R.; Alazab, M.; Tafazolli, R. RSS: An energy-efficient approach for securing IoT service protocols against the DoS attack. IEEE Internet Things J. 2020, doi:10.1109/JIOT.2020.3023102.

4. Park, K.; Noh, S.; Lee, H.; Das, A.K.; Kim, M.; Park, Y.; Wazid, M. LAKS-NVT: Provably secure and lightweight authentication and key agreement scheme without verification table in medical internet of things. IEEE Access 2020, 8, 119387–119404. [CrossRef]

5. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. Sensors 2019, 19, 2358. [CrossRef] [PubMed]

6. Moghadam, M.F.; Nikooghadam, M.; Al Jabban, M.A.B.; Alishahi, M.; Mortazavi, L.; Mohajerzadeh, A. An efficient authentication and key agreement scheme based on ECDH for wireless sensor network. IEEE Access 2020, 8, 73182–73192. [CrossRef]

7. Coron, J.S. Resistance against differential power analysis for elliptic curve cryptosystems. In Proceedings of the 1st International Workshop on Cryptographic Hardware and Embedded Systems, Worcester, MA, USA, 12–13 August 1999; pp. 292–302.

8. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. ACM Trans. Comput. Syst. 1990, 8, 18–36. [CrossRef]

9. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science (LNCS), Les Diablerets, Switzerland, 23–26 January 2005; pp. 65–84.

10. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 4 December 2020).

11. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 4 December 2020).

12. Dolev, D.; Yao, A. On the security of public key protocols. IEEE Trans. Inf. Theory 1983, 29, 198–208. [CrossRef]

13. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT'02), Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 337–351.

14. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397.

15. Yu, S.; Lee, J.; Lee, K.; Park, K.; Park, Y. Secure authentication protocol for wireless sensor networks in vehicular communications. Sensors 2018, 18, 3191. [CrossRef] [PubMed]

16. Fu, X.; Fortino, G.; Li, W.; Pace, P.; Yang, Y. WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings. Future Gener. Comput. Syst. 2019, 91, 223–237. [CrossRef]

17. Fu, X.; Fortino, G.; Pace, P.; Aloi, G.; Li, W. Environment-fusion multipath routing protocol for wireless sensor networks. Inf. Fusion 2020, 53, 4–19. [CrossRef]

18. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks. IEEE Access 2020, 8, 107046–107062. [CrossRef]

19. Fu, X.; Pace, P.; Aloi, G.; Yang, L.; Fortino, G. Topology optimization against cascading failures on wireless sensor networks using a memetic algorithm. Comput. Netw. 2020, 177, 107327. [CrossRef]Sensors 2021, 21, 936 23 of 23

20. Lamport, L. Password authentication with insecure communication. Commun. ACM 1981, 24, 770–772. [CrossRef]

21. Wong, K.H.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), Taichung, Taiwan, 5–7 June 2006; pp. 1–8.

22. Tseng, H.R.; Jan, R.H.; Yang, W. An improved dynamic user authentication scheme for wireless sensor networks. In Proceedings of the IEEE Globecom, Washington, DC, USA, 26–30 November 2007; pp. 986–990.

23. Das, M.L. Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. 2009, 8, 1086–1090. [CrossRef]

24. Khan, M.K.; Alghathbar, K. Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'. Sensors 2010, 10, 2450–2459. [CrossRef]

25. He, D.; Gao, Y.; Chan, S.; Chen, C.; Bu, J. An enhanced two-factor user authentication scheme in wireless sensor networks. Ad Hoc Sens. Wirel. Netw. 2010, 10, 361–371.

26. Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2011, 11, 4767–4779. [CrossRef]

27. Choi, Y.; Lee, D.; Kim, J.; Jung, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2014, 14, 10081–10106. [CrossRef]

28. Wu, F.; Xu, L.; Kumari, S.; Li, X. A new and secure authentication scheme for wireless sensor networks with formal proof. Peer-to-Peer Netw. Appl. 2017, 10, 16–30. [CrossRef]

29. Nam, J.; Kim, M.; Paik, J.; Lee, Y.; Won, D. A provably-secure ECC-based authentication scheme for wireless sensor networks. Sensors 2014, 14, 21023–21044. [CrossRef] [PubMed]

30. Jiang, Q.; Ma, J.; Wei, F.; Tian, Y.; Shen, J.; Yang, Y. An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks. J. Netw. Comput. Appl. 2016, 76, 37–48. [CrossRef]

31. Wu, F.; Xu, L.; Kumari, S.; Li, X. A privacy-preserving and provable user authentication scheme for wireless sensor networks based on Internet of Things security. J. Ambient. Intell. Humaniz. Comput. 2017, 8, 101–116. [CrossRef]

32. Ghahramani, M.; Javidan, R.; Shojafar, M. A secure biometric-based authentication protocol for global mobility networks in smart cities. J. Supercomput. 2020, 76, 8729–8755. [CrossRef]

33. Yu, S.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. A secure and efficient three-factor authentication protocol in global mobility networks. Appl. Sci. 2020, 10, 3565. [CrossRef]

34. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. IEEE Trans. Inf. Forensics Secur. 2017, 12, 2776–2791. [CrossRef]

35. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment. IEEE Internet Things J. 2019, 6, 8804–8817. [CrossRef]

36. Yu, S.; Lee, J.; Park, K.; Das, A.K.; Park, Y. IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. IEEE Access 2020, 8, 167875–167886. [CrossRef]

37. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.

38. Lee, C.C.; Chen, C.T.; Wu, P.H.; Chen, T.Y. Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices. IET Comput. Digit. Tech. 2013, 7, 48–55.