

یک روش احراز هویت و توافق کلید نشست امن در شبکه‌های سیار سراسری با حفظ گمنامی کاربر

فهیمة احمدی^۱، دانشجوی کارشناسی ارشد؛ مرتضی نیکوقدم^۲، استادیار

۱- دانشکده مهندسی کامپیوتر- دانشگاه بین‌المللی امام رضا (ع)- مشهد- ایران - fahimeh.ahmadi@imamreza.ac.ir

۲- دانشکده مهندسی کامپیوتر- دانشگاه بین‌المللی امام رضا (ع)- مشهد- ایران - m.nikooghadam@imamreza.ac.ir

چکیده: در سال‌های اخیر شبکه‌های سیار سراسری رشد سریع و چشم‌گیری را به خود اختصاص داده‌اند و دستگاه‌های تلفن همراه هوشمند، به ابزاری کاربردی و حتی حیاتی برای کاربران تبدیل شده است. همچنین، خدمات رومینگ دستگاه‌های تلفن همراه هوشمند این امکان را فراهم می‌آورد تا کاربران بتوانند در خارج از محدوده جغرافیایی تحت پوشش شبکه اپراتور مرجع و با استفاده از شبکه اپراتورهای کمکی، اطلاعات خود را با دیگران به اشتراک بگذارند. در این مقاله، ابتدا اثبات می‌شود که روش‌هایی که تاکنون در این زمینه پیشنهاد شده‌اند نه تنها در مقابل برخی از حملات از جمله حمله تکرار، حمله داخلی، حمله جعل هویت کاربر، اپراتور مرجع و اپراتور کمکی و حمله منع سرویس آسیب‌پذیرند، بلکه برخی ویژگی‌های امنیتی از جمله گمنامی و عدم ردیابی کاربر، احراز هویت متقابل، محرمانگی کامل روبه‌جلو و امنیت کلید نشست را فراهم نمی‌آورند. سپس، یک طرح احراز هویت مبتنی بر کارت هوشمند برای شبکه‌های سیار سراسری ارائه می‌شود که نه تنها ضعف‌های امنیتی موجود در طرح‌های پیشین را برطرف می‌سازد، بلکه احراز هویت متقابل میان هر سه موجودیت (کاربر، اپراتور مرجع و اپراتور کمکی) را به همراه حفظ گمنامی کاربر نیز فراهم می‌کند. در نهایت، به مقایسه امنیت و کارایی طرح پیشنهادی با طرح‌های پیشین پرداخته شده است و نشان داده می‌شود که طرح پیشنهادی از امنیت و کارایی قابل قبولی برخوردار است.

واژه‌های کلیدی: شبکه‌های سیار سراسری، خدمات رومینگ، توافق کلید، احراز هویت متقابل، گمنامی، غیرقابل ردیابی، کارت هوشمند، بیومتریک، رمزنگاری منحنی بیضوی.

A secure authentication and session key agreement scheme in global mobile networks preserving user anonymity

F. Ahmadi¹, MSc Student; M. Nikooghadam², Assistant professor

1- Faculty of Computer Engineering, Imam Reza International University, Mashhad, Iran, Email: fahimeh.ahmadi@imamreza.ac.ir

2- Faculty of Computer Engineering, Imam Reza International University, Mashhad, Iran, Email: m.nikooghadam@imamreza.ac.ir

Abstract: In recent years, the global mobility networks have grown rapidly and significantly and the smart phones have become practical and even vital tools for users. Furthermore, the roaming service of smart phones provides a possibility for users to share their information with others outside of the geographical region of home agent with the aid of foreign agent. In this paper, first, we prove that the previously-published schemes in this field not only are vulnerable to some known attacks, such as the reply attack, insider attack, user, home agent, and foreign agent impersonation attacks, and Denial of Service attack, but also some security features such as user anonymity, untraceability, mutual authentication, perfect forward secrecy, and session key security are not provided. Second, an authentication scheme based on smart card is presented for the global mobility networks, which not only can solve the security weaknesses of the previous schemes, but also can provide the anonymity and mutual authentication between the three entities of user, home agent, and foreign agent. Finally, security and efficiency of the proposed scheme are compared with the previously-proposed schemes. The results demonstrate that the proposed scheme provides a proper level of both security and efficiency.

Key words: Global mobility networks, Roaming, Key agreement, Mutual authentication, Anonymity, Untraceability, Smart card, Biometric, Elliptic curve cryptography

تاریخ ارسال مقاله: ۱۳۹۶/۰۳/۲۴

تاریخ اصلاح مقاله: ۱۳۹۶/۰۸/۱۶

تاریخ پذیرش مقاله: ۱۳۹۶/۱۲/۱۳

نام نویسنده مسئول: مرتضی نیکوقدم

نشانی نویسنده مسئول: ایران - مشهد - خیابان دانشگاه - خیابان اسرار - دانشگاه بین‌المللی امام رضا (ع) - دانشکده مهندسی کامپیوتر.

۱ - مقدمه

در سال‌های اخیر بهبود انواع شبکه‌های مخابراتی، از جمله شبکه مخابرات بی‌سیم^۱ و نوری [۲،۱]، موجب کمینه‌سازی تأخیر ارسال اطلاعات و افزایش سرعت انتقال اطلاعات شده است؛ بنابراین، کاربران تمایل بسیار زیادی به استفاده از خدمات این شبکه‌های مخابراتی، از جمله شبکه‌های تلفن همراه^۲، داشته‌اند. به موازات آن، نگرانی‌های کاربران در مورد امنیت^۳ و حفظ حریم خصوصی^۴ خودشان روز به روز در حال افزایش است. به منظور حفظ گمنامی کاربران^۵ و جلوگیری از افشاء اطلاعات خصوصی آن‌ها، محققان طرح‌هایی جهت احراز هویت^۶، به صورت گمنام، در شبکه‌های سیار^۷ ارائه داده‌اند. در این طرح‌ها، کاربر تلفن همراه ابتدا در سیستم اپراتور مرجع^۸ ثبت نام می‌کند؛ سپس، به راحتی می‌تواند با کمک اپراتور مرجع برای هر اپراتور کمکی^۹ معتبر احراز هویت شود، کلید نشست را با اپراتور کمکی به اشتراک بگذارد و از انواع مختلف داده‌های اپراتور کمکی (صوت، تصویر و پیامک) در بستر اینترنت و یا شبکه مخابرات استفاده کند. این در حالی است که هویت کاربر در طول اجرای طرح گمنام می‌ماند.

در زمینه شبکه‌های سیار سراسری، برای اولین بار در سال ۲۰۰۴، یک طرح احراز هویت گمنام ارائه شد [۳]. سپس، نشان داده شد که این روش، گمنامی کاربران، محرمانگی رو به عقب کلید نشست^{۱۰} و احراز هویت متقابل^{۱۱} را فراهم نمی‌آورد و مورد حمله جعل^{۱۲} قرار می‌گیرد [۴]. جهت رفع مشکلات ذکر شده، یک طرح احراز هویت بهبود یافته ارائه شد [۴]. در ادامه، محققان نشان دادند که در این روش گمنامی کاربران با حمله حدس برون خط^{۱۳} و به دست آوردن هویت کاربر به خطر می‌افتد و یک طرح احراز هویت بهبود یافته ارائه دادند [۵]. پس از آن، در برخی از مقالات دیگر نه تنها گمنامی کاربر در این روش مورد تردید قرار گرفت [۶-۹]، بلکه، اثبات شد که این روش محرمانگی کامل رو به جلو^{۱۴} را فراهم نمی‌آورد و در مقابل حمله جعل و حمله تکرار^{۱۵} آسیب پذیر است [۱۱،۱۰]. برای رفع این ایرادات امنیتی یک چارچوب جدید از احراز هویت گمنام و امن برای خدمات رومینگ^{۱۶} ارائه شد [۱۱]. در ادامه مقاله برای راحتی، روش ذکر شده SARS^{۱۷} نامیده می‌شود. گروهی دیگر از محققان اثبات کردند که طرح SARS [۱۱] در مقابل حمله تکرار، حمله شخص میانی^{۱۸}، حمله جعل هویت اپراتور مرجع/ اپراتور کمکی/ کاربر تلفن همراه^{۱۹} و حمله داخلی^{۲۰} آسیب پذیر است و نه تنها گمنامی کاربر و احراز هویت متقابل را فراهم نمی‌آورد، بلکه احراز هویت اولیه با استفاده از کارت هوشمند^{۲۱} در این طرح انجام نمی‌شود [۱۲،۱۳].

در سال ۲۰۱۴، یک طرح احراز هویت گمنام و کارآمد در شبکه‌های سیار ارائه شد که در ادامه مقاله از آن با عنوان SARS^{۲۲} یاد می‌شود [۱۴]. در سال ۲۰۱۵، برای اولین بار اثبات شد که طرح SARS [۱۴] در مقابل حمله فرد میانی و جعل هویت کاربر در الگوریتم احراز هویت و در الگوریتم به روز رسانی^{۲۳} کلید نشست، حمله جعل هویت اپراتور کمکی در الگوریتم احراز هویت، حمله به پروتکل در صورت سرقت تصدیق کننده‌ها^{۲۴} و حمله منع سرویس^{۲۵} آسیب پذیر

است و برای رفع این ایرادات یک روش احراز هویت جدید و غیرقابل ردیابی^{۲۶} برای خدمات رومینگ ارائه شد که در ادامه NUA^{۲۷} نامیده می‌شود [۱۵]. در همان سال، همچنین اثبات شد که طرح SAMA [۱۴] در مقابل حمله داخلی و حمله به پروتکل در صورت سرقت تصدیق کننده‌ها آسیب پذیر است و تصدیق محلی^{۲۸} و الگوریتم تغییر رمز عبور کاربر پسند^{۲۹} و مستقل را فراهم نمی‌آورد [۱۶]. در ادامه، یک طرح احراز هویت گمنام و امن برای ارتباطات بی‌سیم با استفاده از کارت هوشمند نیز ارائه شد که در این مقاله AWC^{۳۰} نامیده می‌شود. با هدف رفع مشکل توافق کلید، آسیب پذیری در مقابل حمله جعل و رعایت گمنامی کاربر در طرح‌های گذشته [۱۰]، یک روش احراز هویت متقابل و توافق کلید کارآمد همراه با حفظ گمنامی کاربر توسط محققین پیشنهاد شد که در ادامه مقاله MAKA^{۳۱} نامیده می‌شود [۱۷]. باین وجود، برخی آسیب پذیری‌ها در روش MAKA [۱۷]، از جمله ضعف در مقابل حمله حدس برون خط، امنیت کلید نشست^{۳۲} و محرمانگی رو به جلو باعث شد یک طرح احراز هویت و توافق کلید امن، گمنام و جدید به عنوان روشی جایگزین و بهبود یافته ارائه شود که در ادامه PAKA^{۳۳} نامیده می‌شود [۱۸]. سپس، محققان اثبات کردند که طرح SARS [۱۱] محرمانگی کامل رو به عقب را فراهم نمی‌آورد [۱۹]. همچنین، برخی از محققان نشان دادند که در طرح MAKA [۱۷] گمنامی کاربر حفظ نمی‌شود و در مقابل حمله داخلی، حمله تکرار، حمله جعل، حمله منع سرویس و حمله به پروتکل در صورت سرقت کارت هوشمند^{۳۴} آسیب پذیر است [۲۰،۲۱].

در ادامه ساختار مقاله به این گونه است که بخش ۲ به معرفی رمزنگاری منحنی بیضوی^{۳۵} و استخراج کننده فازی^{۳۶} می‌پردازد. بخش ۳ مروری بر طرح SAMA [۱۴] دارد و به تحلیل امنیت طرح‌های گذشته می‌پردازد. سپس، یک طرح احراز هویت متقابل بهبود یافته، مبتنی بر کارت هوشمند و همراه با حفظ گمنامی کاربر در بخش ۴ ارائه می‌شود. بخش ۵، به تحلیل و مقایسه امنیت طرح پیشنهادی با طرح‌های گذشته اختصاص یافته است. در بخش ۶، کارایی طرح پیشنهادی محاسبه و با طرح‌های گذشته مقایسه می‌شود. در نهایت، در بخش ۷، نتیجه گیری و جمع بندی مقاله ارائه می‌گردد.

۲ - معرفی رمزنگاری منحنی بیضوی و استخراج کننده فازی

۲-۱ - رمزنگاری منحنی بیضوی

رمزنگاری منحنی بیضوی به عنوان یکی از روش‌های رمزنگاری کلید عمومی^{۳۷} مبتنی بر دشواری حل مسئله لگاریتم گسسته روی منحنی بیضوی^{۳۸} است. در یک منحنی بیضوی فرض کنید مجموعه‌ای از نقاط روی یک میدان گالوای^{۳۹} اول F_p است که توسط رابطه (۱) تولید می‌شوند و تشکیل یک گروه آبلین^{۴۰} می‌دهند [۲۲]:

$$y^2 \bmod q = (x^3 + ax + b) \bmod q$$

$$x, y, a, b \in F_q, \quad (4a^3 + 27b^2) \bmod q \neq 0 \quad (1)$$

جدول ۱: علائم و نمادهای مورد استفاده در طرح SAMA

نماد	تعریف
MU	کاربر تلفن همراه
FA	اپراتور کمکی
HA	اپراتور مرجع
PW_{MU}	رمز عبور کاربر تلفن همراه
ID_A	هویت یک موجودیت A
$h(.)$	تابع درهم‌ساز یک‌طرفه
P_{MU}	کلید مخفی ^{۴۱} انتخاب‌شده توسط MU
\oplus	عملیات XOR
l	عملیات الحاق رشته‌ای
N_A	عدد Nonce و تصادفی انتخاب‌شده توسط موجودیت A
P	یک نقطه روی منحنی بیضوی $E_p(a,b)$
$P.x$	مقدار بعد x از نقطه P
P_{HA-MU_i}	کلید مخفی از HA برای MU_i

در این رابطه، B_i به‌عنوان ورودی به تابع $Gen(.)$ داده می‌شود و به‌صورت تصادفی کلید محرمانه کاربر با نماد σ_i و با طول L (مورد استفاده برای رمزنگاری و کلید احراز هویت) و یک پارامتر کمکی عمومی به‌صورت تصادفی و با نماد τ_i (مورد استفاده برای بازبازی پارامتر محرمانه σ_i) در خروجی ظاهر می‌شود. (یک پارامتر عمومی است اما به‌هیچ‌وجه امنیت پارامتر σ_i را به خطر نمی‌اندازد).

(۲) $Rep(.)$: اگر بیومتریک پارازیت دار کاربر را با نماد B'_i نشان دهیم، تابع $Rep(.)$ مطابق رابطه (۵) تعریف می‌شود.

$$\text{if } d(B_i, B'_i) < t \text{ then } Rep(B'_i, \tau_i) = \sigma_i \quad (5)$$

در این رابطه، B'_i و پارامتر عمومی τ_i به‌عنوان ورودی به تابع $Rep(.)$ داده می‌شود. اگر فاصله همینگ میان B_i و B'_i از مقدار آستانه t کمتر باشد، آنگاه، تابع $Rep(.)$ می‌تواند مقدار تصادفی کلید محرمانه کاربر (σ_i) را بازبازی کند.

۳ - مروری بر طرح SAMA و تحلیل امنیت طرح‌های گذشته

۳ ۴ - مروری بر طرح SAMA

علائم و نمادهایی که در طرح SAMA [۱۴] مورد استفاده قرار می‌گیرد در جدول (۱) نشان داده می‌شود:

۳ ۴ ۳ - الگوریتم ثبت‌نام

در این الگوریتم MU باید قبل از استفاده از خدمات اپراتور کمکی (FA)، در سیستم اپراتور مرجع (HA) ثبت‌نام کرده باشد. مراحل اجرای این الگوریتم به‌صورت زیر است:

- ابتدا MU کلید مخفی p_{MU} و هویت ID_{MU} را انتخاب می‌کند. سپس، رمز عبور خود را مطابق رابطه (۶) محاسبه کرده و پیام $\langle ID_{MU}, PW_{MU} \rangle$ را از طریق کانال امن^{۴۶} برای HA ارسال می‌کند.

$$PW_{MU} = h(ID_{MU} \parallel p_{MU}) \quad (6)$$

- HA ابتدا چک می‌کند که آیا ID_{MU} وجود دارد. اگر وجود نداشته باشد، HA یک عدد Nonce تصادفی N_{MU_i} و مقدار P_{HA-MU_i} را تولید می‌کند. سپس، مقدارهای U, W_i, V_i را، به ترتیب، مطابق رابطه‌های (۷)، (۸) و (۹) محاسبه می‌کند. پس از آن، HA مقدارهای $ID_{HA}, W_i, V_i, h(.)$ را روی کارت هوشمند ذخیره می‌کند و برای MU از طریق کانال امن ارسال می‌کند. سپس، مقدارهای U, PW_{MU} و P_{HA-MU_i} را در پایگاه داده^{۴۷} خود ذخیره می‌کند.

$$U = h(p_{HA-MU_i} \parallel N_{MU_i}) \quad (7)$$

$$W_i = PW_{MU} \oplus N_{MU_i} \quad (8)$$

$$V_i = N_{MU_i} \oplus P_{HA-MU_i} \quad (9)$$

۳ ۴ ۳ - الگوریتم احراز هویت و توافق کلید نشست

در رمزنگاری منحنی بیضوی، کلید عمومی Q ، یک نقطه بر روی منحنی بیضوی و کلید خصوصی d ، یک عدد تصادفی محرمانه است. رابطه میان کلید عمومی و کلید خصوصی به‌صورت رابطه (۲) تعریف می‌شود.

$$Q = d.P \quad (2)$$

یکی از مزیت‌های اصلی رمزنگاری منحنی بیضوی نسبت به روش RSA ^{۴۲}، طول کوچک کلید آن است. یک کلید ۱۶۰ بیتی در رمزنگاری منحنی بیضوی امنیتی برابر با یک کلید ۱۰۲۴ بیتی در رمزنگاری کلید عمومی RSA تأمین می‌کند [۲۳]. همچنین، ضرب نردبانی روی گروه منحنی بیضوی افزایشی G_q ، با مرتبه d ، مطابق رابطه (۳) تعریف می‌شود.

$$d.P = P + P + \dots + P \quad (3)$$

۲ ۴ - استخراج‌کننده فازی

استخراج‌کننده فازی یک ابزار برای استخراج داده‌های کلیدی از بیومتریک^{۴۴} کاربر است و برای احراز هویت کاربر با استفاده از الگوی بیومتریک او مورد استفاده قرار می‌گیرد. از آنجایی که اگر ورودی تابع درهم‌ساز بیومتریک^{۴۵} به‌صورت جزئی تغییر کند، منجر به تغییر غیرقابل پیش‌بینی و گسترده در خروجی می‌گردد؛ بنابراین، مقالاتی که از تابع درهم‌ساز بیومتریک برای احراز هویت بیومتریک کاربر استفاده می‌کنند کارآمد نیستند [۲۴]. در این مقاله، مشابه [۲۵، ۲۶]، از استخراج‌کننده فازی به‌جای تابع درهم‌ساز بیومتریک استفاده می‌شود. استخراج‌کننده فازی فرآیندی کارآمد برای ذخیره‌سازی اطلاعات در کارت هوشمند و استفاده از آن‌ها در هنگام احراز هویت است. با استفاده از این فرآیند به‌راحتی می‌توان از بیومتریک‌هایی با پارازیت پایین استفاده کرد که شامل دو تابع زیر است:

(۱) $Gen(.)$: اگر بیومتریک کاربر را با نماد B_i نشان دهیم، تابع $Gen(.)$ مطابق رابطه (۴) تعریف می‌شود.

$$Gen(B_i) = (\sigma_i, \tau_i) \quad , \quad \sigma_i \in \{0,1\}^L \quad (4)$$

- MU ابتدا مقادیرهای S_6' و S_7' را، به ترتیب، مطابق رابطه (۲۲) و (۲۳) محاسبه می‌کند.

$$S_6' = h(ID_{FA} \parallel ID_{HA} \parallel h(PW_{MU} \parallel N_{MU_{i+1}})) \quad (22)$$

$$S_7' = h(aP.x \parallel h(PW_{MU} \parallel N_{MU_{i+1}})) \quad (23)$$

سپس، MU برابری تساوی‌های $S_6' = ?S_6$ و $S_7' = ?S_7$ را بررسی می‌کند. اگر برابر باشند FA و HA برای MU احراز هویت شده‌اند و یک عدد $Nonce$ و تصادفی b را انتخاب و مقدار bP را محاسبه می‌کند. سپس، مقادیرهای K_{MF} و C_{MF} را مطابق رابطه‌های (۲۴) و (۲۵) محاسبه می‌کند.

$$K_{MF} = h(abP.x) \quad (24)$$

$$C_{MF} = h(K_{MF} \parallel bP.x) \quad (25)$$

پس از آن، MU مقادیرهای W_{i+1} و V_{i+1} را، به ترتیب، مطابق رابطه‌های (۲۶) و (۲۷) محاسبه می‌کند. سپس، آن‌ها را، به ترتیب، جایگزین مقادیرهای W_i و V_i می‌کند و مقدار aP را ذخیره می‌کند. در نهایت، پیام $\langle C_{MF}, bP \rangle$ را برای FA ارسال می‌کند.

$$W_{i+1} = PW_{MU} \oplus N_{MU_{i+1}} \quad (26)$$

$$V_{i+1} = N_{MU_{i+1}} \oplus P_{HA-MU_i} \quad (27)$$

- FA مقدار K_{MF} و C_{MF}' را، به ترتیب، مطابق رابطه‌های (۲۴) و (۲۸) به دست می‌آورد.

$$C_{MF}' = h(K_{MF} \parallel bP.x) \quad (28)$$

سپس، FA برابری تساوی $C_{MF}' = ?C_{MF}$ را بررسی می‌کند. اگر برابر باشد، FA توانسته MU را احراز هویت کند و مقادیرهای C_{MF} و aP را در پایگاه داده خود ذخیره می‌کند.

۴-۳ - تحلیل امنیت طرح‌های گذشته

این بخش به تحلیل برخی از ضعف‌های امنیتی و آسیب‌پذیری‌های جدیدترین طرح‌های گذشته از جمله، $SARS$ [۱۱]، $SAMA$ [۱۴]، NUA [۱۵]، AWC [۱۶]، $MAKA$ [۱۷] و $PAKA$ [۱۸] که تاکنون در مقالات دیگر به آن‌ها اشاره‌ای نشده است، می‌پردازد.

۴-۴ - ناکارآمدی الگوریتم احراز هویت و کاربرپسند نبودن الگوریتم تغییر رمز عبور

طرح‌های NUA [۱۵] و $PAKA$ [۱۸] دارای الگوریتم احراز هویت ناکارآمد می‌باشند؛ زیرا درخواست احراز هویت یک کاربر غیرمجاز با اعمال هزینه محاسباتی قابل توجه (برابر با مجموع زمان اجرای یک ضرب نردبانی و دو رمزنگاری/ رمزگشایی نامتقارن برای طرح NUA و زمان اجرای یک ضرب نردبانی برای طرح $PAKA$) لغو می‌گردد و به راحتی موجب اجرای حمله منع سرویس توسط متخاصم می‌شود. الگوریتم تغییر رمز عبور در طرح $PAKA$ [۱۸] کاربرپسند نیست؛ زیرا کاربر برای تغییر رمز عبور خود نیازمند برقراری ارتباط با اپراتور مرجع خود است.

پس از ثبت نام در سیستم HA ، MU می‌تواند الگوریتم احراز هویت و توافق کلید نشست را، بر روی کانال عمومی، اجرا کند. مراحل اجرای این الگوریتم به صورت زیر است:

- MU کارت هوشمند را در دستگاه کارت خوان^{۴۸} قرار می‌دهد و مقادیرهای ID_{MU} و P_{MU} را وارد می‌کند. سپس، کارت هوشمند مقدار $N_{MU_{i+1}}$ را تولید می‌کند و به محاسبه مقادیرهای N_{MU_i} ، P_{HA-MU_i} ، S_1 ، S_2 ، S_3 و S_4 ، به ترتیب، مطابق رابطه‌های (۱۰)، (۱۱)، (۱۲)، (۱۳)، (۱۴) و (۱۵) می‌پردازد. در نهایت، MU مقدار $N_{MU_{i+1}}$ را ذخیره می‌کند و پیام $\langle ID_{HA}, S_1, S_2, S_3, S_4 \rangle$ را برای FA ارسال می‌کند.

$$N_{MU_i} = PW_{MU} \oplus W_i \quad (10)$$

$$P_{HA-MU_i} = N_{MU_i} \oplus V_i \quad (11)$$

$$S_1 = h(P_{HA-MU_i} \parallel N_{MU_i}) \quad (12)$$

$$S_2 = PW_{MU} \oplus N_{MU_{i+1}} \quad (13)$$

$$S_3 = h(N_{MU_{i+1}} \parallel ID_{FA}) \quad (14)$$

$$S_4 = h(PW_{MU} \parallel h(P_{HA-MU_i} \parallel N_{MU_{i+1}})) \quad (15)$$

- FA یک عدد $Nonce$ تصادفی جدید a را انتخاب می‌کند و مقدار aP را محاسبه می‌کند. سپس، مقادیرهای ID_{HA} ، a و aP را ذخیره می‌کند و در نهایت، پیام $\langle ID_{FA}, S_1, S_2, S_3, S_4, aP \rangle$ را برای HA ارسال می‌کند.

- HA مقادیرهای متناظر P_{HA-MU_i} و PW_{MU} را با استفاده از استخراج می‌کند. سپس، مقادیرهای $N_{MU_{i+1}}$ ، S_3' و S_4' را، به ترتیب، مطابق رابطه‌های (۱۶)، (۱۷) و (۱۸) محاسبه می‌کند.

$$N_{MU_{i+1}} = S_2 \oplus PW_{MU} \quad (16)$$

$$S_3' = h(N_{MU_{i+1}} \parallel ID_{FA}) \quad (17)$$

$$S_4' = h(PW_{MU} \parallel h(P_{HA-MU_i} \parallel N_{MU_{i+1}})) \quad (18)$$

پس از آن، HA برابری تساوی‌های $S_3' = ?S_3$ و $S_4' = ?S_4$ را بررسی می‌کند، اگر برابر باشند، HA توانسته MU و FA را احراز هویت کند. HA مقادیرهای S_5 ، S_6 و S_7 را، به ترتیب، مطابق رابطه‌های (۱۹)، (۲۰) و (۲۱) محاسبه می‌کند و مقدار S_1 را جایگزین مقدار $h(P_{HA-MU_i} \parallel N_{MU_{i+1}})$ در پایگاه داده خود می‌کند. سپس، پیام $\langle ID_{HA}, S_6, S_7 \rangle$ را برای FA ارسال می‌کند. وگرنه، HA را یک کاربر غیرقانونی می‌شناسد و ارتباط را خاتمه می‌دهد.

$$S_5 = h(PW_{MU} \parallel N_{MU_{i+1}}) \quad (19)$$

$$S_6 = h(ID_{FA} \parallel ID_{HA} \parallel S_5) \quad (20)$$

$$S_7 = h(aP.x \parallel S_5) \quad (21)$$

- FA پایگاه داده خود را چک می‌کند. اگر ID_{HA} را در پایگاه داده خود داشته باشد، FA توانسته HA را احراز هویت کند و پیام $\langle ID_{FA}, S_6, S_7, aP \rangle$ را برای MU ارسال می‌کند.

۴ ۴ ۳ - فراهم نکردن گمنامی کاربر

در طرح PAKA [۱۸]، متخاصم A با اجرای حمله داخلی از هویت کاربر (ID_{MU})، ذخیره شده در پایگاه داده HA، باخبر شده و گمنامی کاربر نقض می‌گردد.

۴ ۴ ۳ - قابل ردیابی بودن کاربر

در الگوریتم احراز هویت روش AWC [۱۶] و یا روش SARS [۱۱]، کاربر قابل ردیابی است؛ زیرا برخی از پارامترهای مهم موجود در پیام‌های باز، در هر ایجاد نشست، ثابت می‌مانند و این موجب قابل ردیابی بودن کاربر می‌شود.

۴ ۴ ۳ - فراهم نکردن احراز هویت متقابل

در الگوریتم احراز هویت روش‌های AWC [۱۶]، MAKA [۱۷] و PAKA [۱۸]، کاربر FA را احراز هویت نمی‌کند؛ زیرا کاربر به HA اعلام نمی‌کند که خواهان توافق کلید نشست با کدام اپراتور کمکی (FA) است؛ بنابراین، هر اپراتور کمکی معتبر دیگری (FA') می‌تواند با اجرای حمله فرد میانی، جلوی پیام احراز هویت کاربر را گرفته و آن را برای HA ارسال کند. سپس، HA اپراتور کمکی FA' را تنها به‌عنوان یک اپراتور کمکی معتبر و نه لزوماً همان اپراتور کمکی که MU خواهان ارتباط با آن است (همانند FA)، احراز هویت می‌کند.

۴ ۴ ۳ - امنیت ضعیف کلید نشست

در الگوریتم احراز هویت روش AWC [۱۶]، متخاصم A از مرحله (۷) به بعد هویت FA را جعل کرده و الگوریتم را اجرا می‌کند. به این صورت که ابتدا مقدار تصادفی r_A^1 را تولید می‌کند و به محاسبه مقدارهای n_{MA}^1 ، n_A^1 و SK_{MA} ، به ترتیب، مطابق رابطه‌های (۲۹)، (۳۰) و (۳۱) می‌پردازد.

$$n_A^1 = r_A^1 \cdot P \quad (29)$$

$$n_{MA} = r_A^1 n_{MU}^2 \quad (30)$$

$$SK_{MA} = h(n_{MA} \| t_{MU}) \quad (31)$$

که در آن، مقدار به‌دست‌آمده توسط MU و t_{MU} برچسب زمانی MU است. سپس، A پیام شماره (8) را، با اجرای حمله فرد میانی متوقف کرده و سپس، پیام $\langle ID_{FA}, C_{HA}^2, n_A^1 \rangle$ را برای MU ارسال می‌کند (C_{HA}^2 مقدار تولیدشده توسط HA است که بر روی کانال عمومی قرار دارد). MU پس از محاسبه C_{HA}^2 و مقایسه آن با مقدار دریافت شده در پیام، به‌اشتباه متخاصم را به‌جای FA احراز هویت می‌کند. سپس، MU به محاسبه مقدارهای n_{MA} و SK_{MA} ، به ترتیب، مطابق رابطه‌های (۳۲) و (۳۱) می‌پردازد.

$$n_{MA} = r_{MU}^2 n_A^1 \quad (32)$$

که در آن، مقدار تصادفی تولیدشده توسط MU است. درنهایت، MU کلید نشست SK_{MA} را با متخاصم A به اشتراک می‌گذارد و امنیت کلید نشست نقض می‌گردد.

در طرح PAKA [۱۸] امنیت کلید نشست فراهم نمی‌شود؛ زیرا همان‌طور که در زیر بخش ۴-۴ ذکر شد، هر اپراتور کمکی معتبری (همانند FA') می‌تواند هویت FA را جعل و پیام احراز هویت را با اجرای حمله فرد میانی برای HA ارسال کند. درنهایت، HA پارامترهای مربوط به FA' را برای MU ارسال می‌کند. درنتیجه، MU با اپراتور کمکی FA'، بدون آن‌که او را احراز هویت کند، کلید به اشتراک می‌گذارد و امنیت کلید نشست نقض می‌گردد.

۴ ۴ ۳ - حمله به پروتکل در صورت سرقت تصدیق‌کننده‌ها

طرح MAKA [۱۷] در مقابل حمله به پروتکل در صورت به سرقت رفتن تصدیق‌کننده‌ها آسیب‌پذیر است؛ زیرا HA نیازمند ذخیره‌سازی پارامترهای $\{Tr_{Seq}, ID_M, K_{uh}, h(\cdot)\}$ در پایگاه داده خود است؛ بنابراین، متخاصم A می‌تواند با به دست آوردن مقدار Tr_{Seq} که یک پارامتر منحصربه‌فرد و متعلق به کاربر است و همچنین، اطلاع از هویت کاربران (ID_M) و کلید مشترک کاربران با HA (K_{uh}) به‌راحتی کاربران را ردیابی کند.

۴ ۴ ۳ - حمله به پروتکل در صورت سرقت کارت هوشمند

در طرح SAMA [۱۴] اگر MU، پارامتر $N_{MU_{i+1}}$ را در کارت هوشمند ذخیره کرده و الگوریتم احراز هویت و توافق کلید نشست به پایان رسیده باشد، این حمله به‌صورت زیر اجرا می‌شود:

• متخاصم A با سرقت کارت هوشمند MU می‌تواند پارامترهای

$$P_{HA-MU_i} \text{ و } PW_{MU} \text{ را استخراج کند و } V_{i+1} \text{ و } W_{i+1} \text{، } N_{MU_{i+1}} \text{ را، به ترتیب، مطابق رابطه‌های (۳۳) و (۳۴) به دست آورد.}$$

$$PW_{MU} = W_{i+1} \oplus N_{MU_{i+1}} \quad (33)$$

$$P_{HA-MU_i} = N_{MU_{i+1}} \oplus V_{i+1} \quad (34)$$

سپس، A پارامتر تصادفی $N_{MU_{i+2}}$ را انتخاب و مقدارهای معتبر S_1^* ، S_2^* ، S_3^* و S_4^* را، به ترتیب، مطابق رابطه‌های (۳۵)، (۳۶)، (۳۷) و (۳۸) تولید می‌کند و پیام $\langle ID_{HA}, S_1^*, S_2^*, S_3^*, S_4^* \rangle$ را برای اپراتور کمکی دیگری (FA') ارسال می‌کند.

$$S_1^* = h(p_{HA-MU_i} \| N_{MU_{i+1}}) \quad (35)$$

$$S_2^* = PW_{MU} \oplus N_{MU_{i+2}} \quad (36)$$

$$S_3^* = h(N_{MU_{i+2}} \| ID_{FA}') \quad (37)$$

$$S_4^* = h(PW_{MU} \| h(p_{HA-MU_i} \| N_{MU_{i+2}})) \quad (38)$$

• A و HA مراحل طی شده در الگوریتم احراز هویت و توافق کلید نشست طرح SAMA [۱۴] را، با پارامترهای جدید تا انتها اجرا می‌کنند. A از سمت HA یک کاربر مجاز شناخته می‌شود. سپس، A با FA' به کلید نشست مشترک می‌رسد و از خدمات آن استفاده می‌کند.

۳ ۴ ۸ - حمله تکرار

در الگوریتم احراز هویت روش NUA [۱۵]، هیچیک از پیامها حاوی برچسب زمانی نیست؛ بنابراین، متخاصم A با اجرای حمله فرد میانی موجب حمله تکرار و حمله منع سرویس می‌شود. به این صورت که A پیام درخواست احراز هویت MU (m_1) را نگه می‌دارد و سپس، در زمان دیگری این پیام را برای FA ارسال می‌کند. مراحل اجرای این الگوریتم تا زمانی که پیام m_4 برای MU ارسال شود، ادامه می‌یابد. در این مرحله، HA برای MU به‌درستی احراز هویت نمی‌شود و الگوریتم ذکر شده لغو می‌گردد؛ بنابراین، متخاصم با اجرای این حمله موجب به هدر رفتن توان محاسباتی قابل توجه (تقریباً برابر با مجموع زمان اجرای یک ضرب نردبانی و دو رمزنگاری/ رمزگشایی نامتقارن) می‌شود.

در الگوریتم احراز هویت روش AWC [۱۶]، متخاصم با اجرای حمله تکرار، به‌راحتی پیام شماره (۲) را نگه می‌دارد. سپس، در زمانی دیگر برچسب زمانی فعلی خود را جایگزین برچسب زمانی موجود در پیام شماره (۲) کرده و پیام تغییر یافته را برای FA ارسال می‌کند. مراحل اجرای الگوریتم تا انتها پیش می‌رود. FA و MU به ترتیب، با استفاده از برچسب‌های زمانی متفاوت، کلیدهای نشست متفاوتی را به دست می‌آورند؛ بنابراین، FA و MU به کلید نشست مشترک نمی‌رسند و الگوریتم احراز هویت با به هدر رفتن توان محاسباتی FA ، لغو می‌گردد. متخاصم با اجرای این حمله، موجب اجرای حمله منع سرویس نیز می‌شود.

۳ ۴ ۹ - حمله جعل هویت کاربر/ اپراتور مرجع/ اپراتور کمکی و حمله فرد میانی

در الگوریتم احراز هویت و توافق کلید نشست طرح $SAMA$ [۱۴]، متخاصم A می‌تواند درخواست احراز هویت را برای FA ارسال و با اجرای حمله فرد میانی هویت HA را جعل کند و موجب احراز هویت خود برای FA شود. این حمله به‌صورت زیر اجرا می‌شود:

- پیام درخواست احراز هویت $\langle ID_{HA}, S_1', S_2', S_3', S_4' \rangle$ را با پارامترهای نامعتبر و دلخواه انتخاب و برای FA ارسال می‌کند.
- پارامتر تصادفی a را انتخاب کرده و اقدام به محاسبه aP می‌کند. سپس، ID_{HA} ، aP و a را در فهرست خود ذخیره و پیام $\langle ID_{FA}, aP, S_1', S_2', S_3', S_4' \rangle$ را برای HA ارسال می‌کند.
- میان ارتباطات FA و HA ، بر روی کانال عمومی قرار گرفته و جلوی پیام ارسال شده از سمت FA را می‌گیرد. سپس A هویت HA را جعل می‌کند و پارامترهای نامعتبر S_6' و S_7' را انتخاب و برای FA به‌صورت پیام $\langle ID_{HA}, S_6', S_7' \rangle$ ارسال می‌کند.
- در فهرست FA موجود است و HA احراز هویت می‌شود. سپس، پیام $\langle ID_{FA}, S_6', S_7', aP \rangle$ را برای A ارسال می‌کند.
- پارامتر c را انتخاب و مقدار cP را محاسبه می‌کند. سپس، مقدارهای K_{AF} و C_{AF}' را، به ترتیب، مطابق رابطه‌های (۳۹) و (۴۰) محاسبه و پیام $\langle cP, C_{AF}' \rangle$ را برای FA ارسال می‌کند.

$$K_{AF} = h(acP.x) \quad (۳۹)$$

$$C_{AF} = h(K_{AF} \| cP.x) \quad (۴۰)$$

• مقدارهای K_{AF} و C_{AF}' را، به ترتیب، مطابق رابطه‌های (۴۰) و (۴۱) به دست می‌آورد.

$$C_{AF}' = h(K_{AF} \| cP.x) \quad (۴۱)$$

سپس، FA برابری تساوی $C_{AF}' = ? C_{AF}$ را بررسی می‌کند و چون برابر است، FA متخاصم A را به‌اشتباه احراز هویت کرده و با او کلید نشست K_{AF} را به اشتراک می‌گذارد و اجازه استفاده از خدمات خود را به متخاصم A می‌دهد.

در الگوریتم تغییر کلید نشست روش AWC [۱۶]، متخاصم A به‌راحتی می‌تواند هویت کاربر مجاز را جعل کند؛ زیرا در این الگوریتم، FA کاربر را احراز هویت نمی‌کند؛ بنابراین، FA متخاصم A را مجاز می‌شناسد و متخاصم A به کلید نشست مشترک با FA می‌رسد.

در الگوریتم احراز هویت روش‌های $MAKA$ [۱۷] و $PAKA$ [۱۸]، همان‌طور که در زیر بخش ۴-۴ ذکر شده است، هر اپراتور کمکی معتبر دیگری (FA')، با اجرای حمله فرد میانی، به‌گونه‌ای می‌تواند هویت اپراتور کمکی FA را جعل کند و بدون آن که کاربر متوجه تغییر شود، برای HA احراز هویت شود. با اجرای این حمله، FA' می‌تواند با کاربر کلید نشست به اشتراک بگذارد و از پیام‌های محرمانه کاربر (که قصد ارسال به اپراتور کمکی FA داشته است) باخبر شود.

۳ ۴ ۱۰ - مشکل امنیتی در الگوریتم تغییر رمز عبور

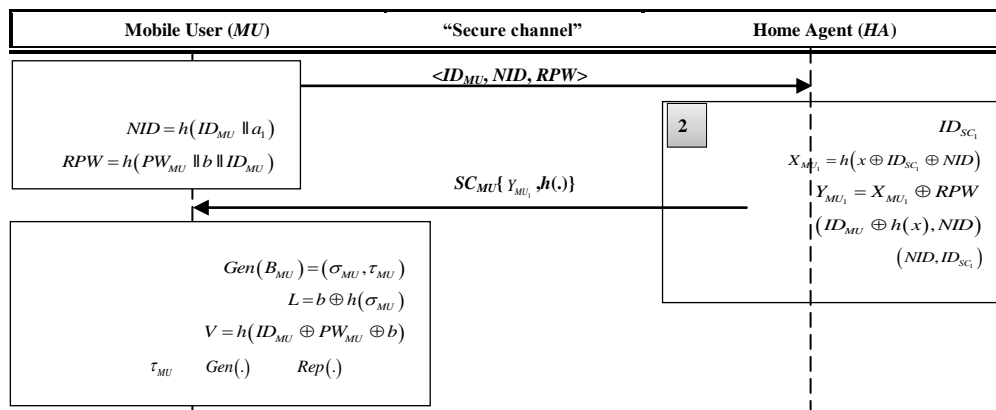
در طرح $SAMA$ [۱۴]، متخاصم A با دانستن رمز عبور قدیم کاربر (PW_{MU}) و با اجرای حمله فرد میانی و شنود پیام درخواست تغییر رمز عبور از روی کانال عمومی، می‌تواند رمز عبور جدید کاربر ($PW_{MU_{new}}$) را مطابق رابطه (۴۲) به‌دست آورد.

$$PW_{MU_{new}} = PW_{MU} \oplus h_1 \quad (۴۲)$$

h_1 مقدار تولید شده توسط MU بوده که بر روی کانال عمومی قابل شنود است؛ بنابراین، الگوریتم ذکر شده به‌صورت ناامن اجرا می‌شود.

۴ - طرح پیشنهادی

در این بخش، به‌منظور افزایش امنیت و کارایی مکانیزم احراز هویت مبتنی بر کارت هوشمند، یک طرح بهبود یافته احراز هویت از راه دور، مبتنی بر کارت هوشمند و همراه با حفظ گمنامی کاربر ارائه شده است که می‌تواند در شبکه‌های سیار سراسری و خدمات رومینگ بکار رود. لازم به‌ذکر است که طرح پیشنهادی، دارای سه الگوریتم، شامل الگوریتم ثبت‌نام، الگوریتم احراز هویت متقابل و توافق کلید نشست و الگوریتم تغییر رمز عبور است. علائم و نمادهای مورد استفاده برای ارائه طرح پیشنهادی در جدول (۲) نشان داده شده است. در شبکه‌های سیار سراسری، MU قبل از استفاده از خدمات رومینگ FA ، باید در سیستم HA ثبت‌نام کند. در ابتدا، یک تابع درهم‌ساز یک‌طرفه $\{0,1\}^k \rightarrow \{0,1\}^*$ ($h(\cdot)$) و کلید مخفی x (۱۰۲۴ بیت) را انتخاب می‌کند.



شکل ۱: مراحل اجرای الگوریتم ثبت نام طرح پیشنهادی

پس از آن، MU پیام درخواست ثبت نام $\langle ID_{MU}, NID, RPW \rangle$ را برای HA ارسال می کند.

- پس از دریافت پیام درخواست ثبت نام MU ، ابتدا مقدار $(ID_{MU} \oplus h(x))$ را به دست می آورد. سپس، HA در فهرست کاربران ثبت نام شده خود مقدار $(ID_{MU} \oplus h(x), NID)$ را جستجو می کند. اگر ID_{MU} موجود باشد، درخواست شناسه جدید را برای MU ارسال می کند. وگرنه، HA مقدار تصادفی ID_{SC1} را تولید و مقدارهای X_{MU1} (مقدار مخفی MU است که میان MU و HA محرمانه است) و Y_{MU1} را، به ترتیب، مطابق رابطه های (۴۵) و (۴۶) محاسبه می کند.

$$X_{MU1} = h(x \oplus ID_{SC1} \oplus NID) \quad (45)$$

$$Y_{MU1} = X_{MU1} \oplus RPW \quad (46)$$

- سپس، HA مقدارهای $\{Y_{MU1}, h(\cdot)\}$ را در کارت هوشمند کاربر (SC_{MU}) درج می کند و آن را به MU تحویل می دهد. علاوه بر این، HA مقدارهای $(ID_{MU} \oplus h(x), NID)$ و (NID, ID_{SC1}) را، به ترتیب، در فهرست کاربران ثبت نام شده و فهرست کاربران فعال خود ذخیره می کند. تأکید می شود که شناسه کاربر (ID_{MU}) به صورت $ID_{MU} \oplus h(x)$ ذخیره می شود؛ بنابراین، تنها HA می تواند آن آگاهی داشته باشد و نگرانی ها برای اجرای حمله داخلی به این پروتکل از بین خواهد رفت.
- پس از دریافت SC_{MU} بیومتریک خود (B_{MU}) را ثبت و فرآیند استخراج فازی را مطابق رابطه (۴۷) اعمال می کند.

$$Gen(B_{MU}) = (\sigma_{MU}, \tau_{MU}) \quad (47)$$

- سپس، MU مقدارهای V و L را، به ترتیب، مطابق رابطه های (۴۸) و (۴۹) محاسبه کرده و مقدارهای V, L, σ_{MU} ، $Gen(\cdot)$ و $Rep(\cdot)$ را در SC_{MU} ذخیره می کند.

$$V = h(ID_{MU} \oplus PW_{MU} \oplus b) \quad (48)$$

$$L = b \oplus H(\sigma_{MU}) \quad (49)$$

جدول ۲: علائم و نمادهای مورداستفاده در طرح پیشنهادی

نماد	تعریف
MU	کاربر مجاز تلفن همراه
FA	اپراتور کمکی
HA	اپراتور مرجع
X	کلید مخفی اپراتور مرجع
A	یک متخاصم
ID_A	هویت یک موجودیت A
SC_{MU}	کارت هوشمند شخصی MU
PW_{MU}	رمز عبور منحصربه فرد MU
$T_{MU1}, T_{MU2}, T_{MU3}$	برچسب های زمانی تولید شده توسط MU
T_{HA1}, T_{HA2}	برچسب های زمانی تولید شده توسط HA
$h(\cdot)$	تابع درهم ساز یک طرفه
B_{MU}	بیومتریک MU
σ_{MU}	پارامتر خصوصی تولید شده توسط استخراج کننده فازی
τ_{MU}	پارامتر عمومی تولید شده توسط استخراج کننده فازی
T	آستانه خطا
\oplus	عملیات XOR
I	عملیات الحاقی رشته ای
Δ	تأخیر زمانی معتبر در انتقال پیام

۴-۱ - الگوریتم ثبت نام

این الگوریتم توسط MU و HA ، از طریق کانال امن (ملاقات حضوری) اجرا می شود. مراحل اجرای این الگوریتم که در شکل (۱) نشان داده شده است، به صورت زیر است:

- MU ابتدا یک شناسه (ID_{MU}) و یک رمز عبور (PW_{MU}) را انتخاب و مقدارهای تصادفی b و a_1 را تولید می کند. سپس، MU یک شناسه مستعار^{۴۹} (NID) و مقدار RPW را، به ترتیب، مطابق رابطه های (۴۳) و (۴۴) به دست می آورد.

$$NID = h(ID_{MU} || a_1) \quad (43)$$

$$RPW = h(PW_{MU} || b || ID_{MU}) \quad (44)$$

بررسی برقراری رابطه (۵۶) به این معناست که از زمانی که پیام توسط MU ارسال شده (T_{MU_1}) تا زمانی که پیام به دست HA رسیده (T_{MU_1}'')، مدت زمانی کمتر از دو تأخیر زمانی معتبر در انتقال پیام ($2\Delta t$) سپری شده است. مقدار مجاز برای $2\Delta t$ از قبل میان MU و HA هماهنگ شده است. اگر رابطه برقرار نباشد نشست خاتمه می‌یابد و گرنه مراحل اجرای الگوریتم ادامه می‌یابد. علاوه بر این، HA با بررسی کردن هر دو مقدار (برچسب زمانی کاربر که روی کانال قرار گرفته و برچسب زمانی کاربر که داخل پارامتر رمز شده M_{MU} است) و برابر بودن آن‌ها مطمئن می‌شود که حمله تکرار رخ نداده است. اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه، HA مقدار (NID, ID_{SC_1}) را از فهرست کاربران فعال خود بازیابی می‌کند. اگر شناسه مستعار MU (NID)، در فهرست کاربران فعال HA وجود نداشته باشد، تلاش برای احراز هویت MU خاتمه می‌یابد. وگرنه، HA مقدار M_{MU}' را مطابق رابطه (۵۷) به دست می‌آورد.

$$M_{MU}' = h(h(x \oplus ID_{SC_1} \oplus NID) \| T_{MU_1} \| ID_{FA}) \quad (57)$$

سپس، HA برابری تساوی $M_{MU}' = ? M_{MU}$ را بررسی می‌کند. اگر برابر نباشد، نشست خاتمه می‌یابد (این الگوریتم کارآمد است؛ زیرا درخواست احراز هویت کاربر غیرمجاز با هزینه محاسباتی پایینی لغو می‌گردد و از حمله منع سرویس جلوگیری می‌شود). وگرنه، MU و FA برای HA احراز هویت شده‌اند. سپس، HA و FA از طریق روش CL - AKA [۲۷] کلید K_{HA-FA} را به اشتراک می‌گذارند. HA پارامتر تصادفی K را انتخاب و مقدار M_{HA} و کلید نشست مشترک میان MU و FA (SK) را، به ترتیب، مطابق رابطه‌های (۵۸) و (۵۹) تولید می‌کند (با مقداری تصادفی K در هر نشست، محرمانگی کلید شناخته شده^{۵۱} فراهم می‌شود. همچنین، در محاسبه کلید نشست، پارامترهای بلندمدت x و PW_{MU} به کار نمی‌رود و محرمانگی کامل روبه‌جلو، محرمانگی کامل رو به عقب در این طرح تضمین می‌شود).

$$M_{HA} = h(h(x \oplus ID_{SC_1} \oplus NID) \| T_{MU_1} \| T_{HA_1} \| ID_{FA}) \quad (58)$$

$$SK = h(k \| NID \| ID_{FA}) \quad (59)$$

سپس، HA مقدار SK را، به صورت جداگانه، برای FA و MU ، به ترتیب، در قالب E_1 مطابق رابطه (۶۰) و در قالب E_2 مطابق رابطه (۶۱) رمز می‌کند.

$$E_1 = E_{K_{HA-FA}}(SK) \quad (60)$$

$$E_2 = E_{X_{MU}}(SK) \quad (61)$$

لازم به ذکر است که HA کلید مشترک میان FA و MU (SK) را در فهرست خود ذخیره نمی‌کند و نگرانی‌ها برای اجرای حمله داخلی برطرف خواهد شد. پس‌از آن، HA مقدار E_3 را مطابق رابطه (۶۲) محاسبه می‌کند.

$$E_3 = E_{K_{HA-FA}}(NID \| M_{HA} \| T_{HA_1}) \quad (62)$$

همان‌طور که در شکل (۱) مشاهده می‌شود، MU پیام ثبت‌نام را ایجاد و آن را برای HA ، از طریق کانال امن، ارسال می‌کند. در صورتی که MU قبلاً در سیستم HA ثبت‌نام نکرده باشد، HA کارت هوشمند را تولید و ارسال می‌کند. در پایان این الگوریتم، پارامترهای $Y_{MU_1}, h(\cdot), V, L, \tau_{MU}, t, Gen(\cdot)$ و $Rep(\cdot)$ در SC_{MU} ذخیره شده است.

۴-۴ الگوریتم احراز هویت متقابل و توافق کلید نشست

این الگوریتم توسط MU ، HA و FA ، از طریق کانال عمومی، اجرا می‌شود. در این الگوریتم، HA و FA با استفاده از روش CL - AKA [۲۷]، کلید به اشتراک می‌گذارند. مراحل اجرای این الگوریتم که در شکل (۲) نشان داده شده است، به صورت زیر است:

- ابتدا، MU کارت هوشمند را در دستگاه کارت‌خوان قرار می‌دهد و ID_{MU} ، PW_{MU} و B_{MU} را وارد می‌کند. سپس، SC_{MU} برای بررسی صحت ورودی‌ها، مقدارهای σ_{MU}' ، b' و V' را، به ترتیب، مطابق رابطه‌های (۵۰)، (۵۱) و (۵۲) به دست می‌آورد.

$$\sigma_{MU}' = Rep(B_{MU}, \tau_{MU}) \quad (50)$$

$$b' = L \oplus H(\sigma_{MU}') \quad (51)$$

$$V' = h(ID_{MU} \oplus PW_{MU} \oplus b') \quad (52)$$

SC_{MU} درستی تساوی $V' = ? V$ را بررسی می‌کند. اگر برابر نباشد، نشست خاتمه می‌یابد. وگرنه، کاربر احراز هویت شده است و SC_{MU} مقدارهای RPW ، X_{MU_1} و M_{MU} را، به ترتیب، مطابق رابطه‌های (۴۴)، (۵۳) و (۵۴) محاسبه و پیام احراز هویت $\langle ID_{HA}, NID, M_{MU}, T_{MU_1} \rangle$ را برای FA ارسال می‌کند.

$$X_{MU_1} = Y_{MU_1} \oplus RPW \quad (53)$$

$$M_{MU} = h(X_{MU_1} \| T_{MU_1} \| ID_{FA}) \quad (54)$$

- FA ، پس از دریافت پیام در زمان T_{MU_1}' ، ابتدا درستی رابطه (۵۵) را بررسی می‌کند.

$$|T_{MU_1}' - T_{MU_1}| \leq ? \Delta t \quad (55)$$

بررسی برقراری رابطه (۵۵) به این معناست که از زمانی که پیام توسط MU ارسال شده (T_{MU_1}) تا زمانی که پیام به دست FA رسیده (T_{MU_1}')، مدت زمانی کمتر از یک تأخیر زمانی معتبر در انتقال پیام (Δt) سپری شده است. مقدار مجاز برای Δt از قبل میان MU و FA هماهنگ شده است. اگر رابطه برقرار نباشد، نشست خاتمه می‌یابد. وگرنه، FA مقدار (NID, ID_{HA}) را در فهرست کاربران خود ذخیره و پیام $\langle ID_{FA}, NID, M_{MU}, T_{MU_1} \rangle$ را برای HA ارسال می‌کند.

- HA ، پس از دریافت پیام در زمان T_{MU_1}'' ، ابتدا درستی رابطه (۵۶) را بررسی می‌کند.

$$|T_{MU_1}'' - T_{MU_1}| \leq ? 2\Delta t \quad (56)$$

$$|T_{MU_2}'' - T_{MU_2}| \leq 2\Delta t \quad (۶۹)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه، مقدار E_4 را، با استفاده از X_{MU_1} رمزگشایی می‌کند. سپس، HA برابری NID دریافت شده از پیام را با NID' حاصل از رمزگشایی E_4 مقایسه و درستی مقدار T_{MU_2} را بررسی می‌کند. اگر برابر نباشند، نشست خاتمه می‌یابد. وگرنه، HA به مقدارهای صحیح NID_{NEW} و RPW می‌رسد. سپس، HA مقدار ID_{SC_2} را تولید و مقدارهای X_{MU_2} ، Y_{MU_2} و E_5 را، به ترتیب، مطابق رابطه‌های (۷۰)، (۷۱) و (۷۲) محاسبه می‌کند.

$$X_{MU_2} = h(x \oplus ID_{SC_2} \oplus NID_{NEW}) \quad (۷۰)$$

$$Y_{MU_2} = X_{MU_2} \oplus RPW \quad (۷۱)$$

$$E_5 = E_{X_{MU_1}}(NID_{NEW} \parallel Y_{MU_2} \parallel T_{HA_2}) \quad (۷۲)$$

پس از آن، HA پیام $\langle NID, E_5, T_{HA_2} \rangle$ را برای FA ارسال می‌کند تا مقدار Y_{MU_2} را به اطلاع MU برساند و به MU اعلام کند که مقدار صحیح NID_{NEW} را به دست آورده است.

• FA ، پس از دریافت پیام در زمان T_{HA_2}' ، ابتدا درستی رابطه (۷۳) را بررسی می‌کند.

$$|T_{HA_2}' - T_{HA_2}| \leq \Delta t \quad (۷۳)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه FA پیام $\langle NID, E_5, T_{HA_2} \rangle$ را برای MU ارسال می‌کند.

• MU ، پس از دریافت پیام در زمان T_{HA_2}'' ، ابتدا درستی رابطه (۷۴) را بررسی می‌کند.

$$|T_{HA_2}'' - T_{HA_2}| \leq 2\Delta t \quad (۷۴)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه MU مقدار E_5 را با استفاده از X_{MU_1} رمزگشایی و برابری $NID_{NEW}' = NID_{NEW}$ و درستی مقدار T_{HA_2} را بررسی می‌کند. اگر برابر نباشند، نشست خاتمه می‌یابد. وگرنه، MU به مقدار صحیح Y_{MU_2} می‌رسد. MU به محاسبه مقدارهای X_{MU_2} و E_6 ، به ترتیب، مطابق رابطه‌های (۷۵) و (۷۶) می‌پردازد و Y_{MU_2} را جایگزین Y_{MU_1} در کارت هوشمند خود می‌کند. سپس، پیام $\langle NID, E_6, ID_{HA}, T_{MU_3} \rangle$ را برای FA ارسال می‌کند تا به اطلاع HA برساند که مقدار را به درستی دریافت کرده است.

$$X_{MU_2} = Y_{MU_2} \oplus RPW \quad (۷۵)$$

$$E_6 = E_{X_{MU_2}}(NID \parallel NID_{NEW} \parallel T_{MU_3}) \quad (۷۶)$$

• FA ، پس از دریافت پیام در زمان T_{MU_3} ، ابتدا درستی رابطه (۷۷) را بررسی می‌کند.

$$|T_{MU_3}' - T_{MU_3}| \leq \Delta t \quad (۷۷)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه FA پیام $\langle NID, E_6 \rangle$ را برای HA ارسال می‌کند.

• HA ، پس از دریافت پیام در زمان T_{MU_3}'' ، ابتدا درستی رابطه (۷۸) را بررسی می‌کند.

مطابق رابطه (۶۲)، HA برای جلوگیری از حمله‌های جعل هویت MU ، جعل هویت HA و تکرار، پارامترهای NID ، M_{HA} و T_{HA_1} را بر روی کانال عمومی هم به صورت باز و هم به صورت رمز شده برای FA ارسال می‌کند؛ زیرا اگر نویزی به صورت عمدی و یا غیر عمد ایجاد شد، FA متوجه آن شده و نشست خاتمه می‌یابد. سپس، HA پیام $\langle ID_{HA}, T_{HA_1}, E_1, E_2, E_3, M_{HA}, NID, E_1, E_2, E_3, M_{HA}, T_{HA_1} \rangle$ را برای FA ارسال می‌کند.

• FA ، پس از دریافت پیام در زمان T_{HA_1}' ، ابتدا درستی رابطه (۶۳) را بررسی می‌کند.

$$|T_{HA_1}' - T_{HA_1}| \leq \Delta t \quad (۶۳)$$

اگر برقرار نباشد نشست خاتمه می‌یابد. وگرنه، مقدار E_3 را، با کلید K_{HA-FA} رمزگشایی می‌کند. سپس، برابری تساوی‌های $NID' = NID$ و $M_{HA}' = M_{HA}$ و درستی مقدار T_{HA_1} را بررسی می‌کند. اگر برابر نباشند، نشست خاتمه می‌یابد و FA مقدار (NID, ID_{HA}) را از فهرست کاربران خود حذف می‌کند. وگرنه، MU و HA برای FA احراز هویت شده‌اند. FA مقدار K_{FA} را با کلید K_{HA-FA} رمزگشایی می‌کند و به کلید نشست SK می‌رسد و پیام $\langle M_{HA}, E_2, T_{HA_1} \rangle$ را برای MU ارسال می‌کند.

• MU ، پس از دریافت پیام در زمان T_{HA_1}'' ، ابتدا درستی رابطه (۶۴) را بررسی می‌کند.

$$|T_{HA_1}'' - T_{HA_1}| \leq 2\Delta t \quad (۶۴)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه، مقدار M_{HA}' را مطابق رابطه (۶۵) به دست می‌آورد.

$$M_{HA}' = h(X_{MU_1} \parallel T_{MU_1} \parallel T_{HA_1} \parallel ID_{FA}) \quad (۶۵)$$

MU برابری تساوی $M_{HA}' = M_{HA}$ را بررسی می‌کند، اگر برابر نباشد، نشست خاتمه می‌یابد. وگرنه، HA و FA برای MU احراز هویت شده‌اند. MU ، E_2 را با X_{MU_1} رمزگشایی می‌کند و به کلید نشست SK می‌رسد. سپس، MU مقدار تصادفی a_2 را تولید کرده و به محاسبه شناسه مستعار جدید خود (NID_{NEW}) و مقدار E_4 ، به ترتیب، مطابق رابطه‌های (۶۶) و (۶۷) می‌پردازد.

$$NID_{NEW} = h(ID_{MU} \parallel a_2) \quad (۶۶)$$

$$E_4 = E_{X_{MU_1}}(NID \parallel NID_{NEW} \parallel RPW \parallel T_{MU_2}) \quad (۶۷)$$

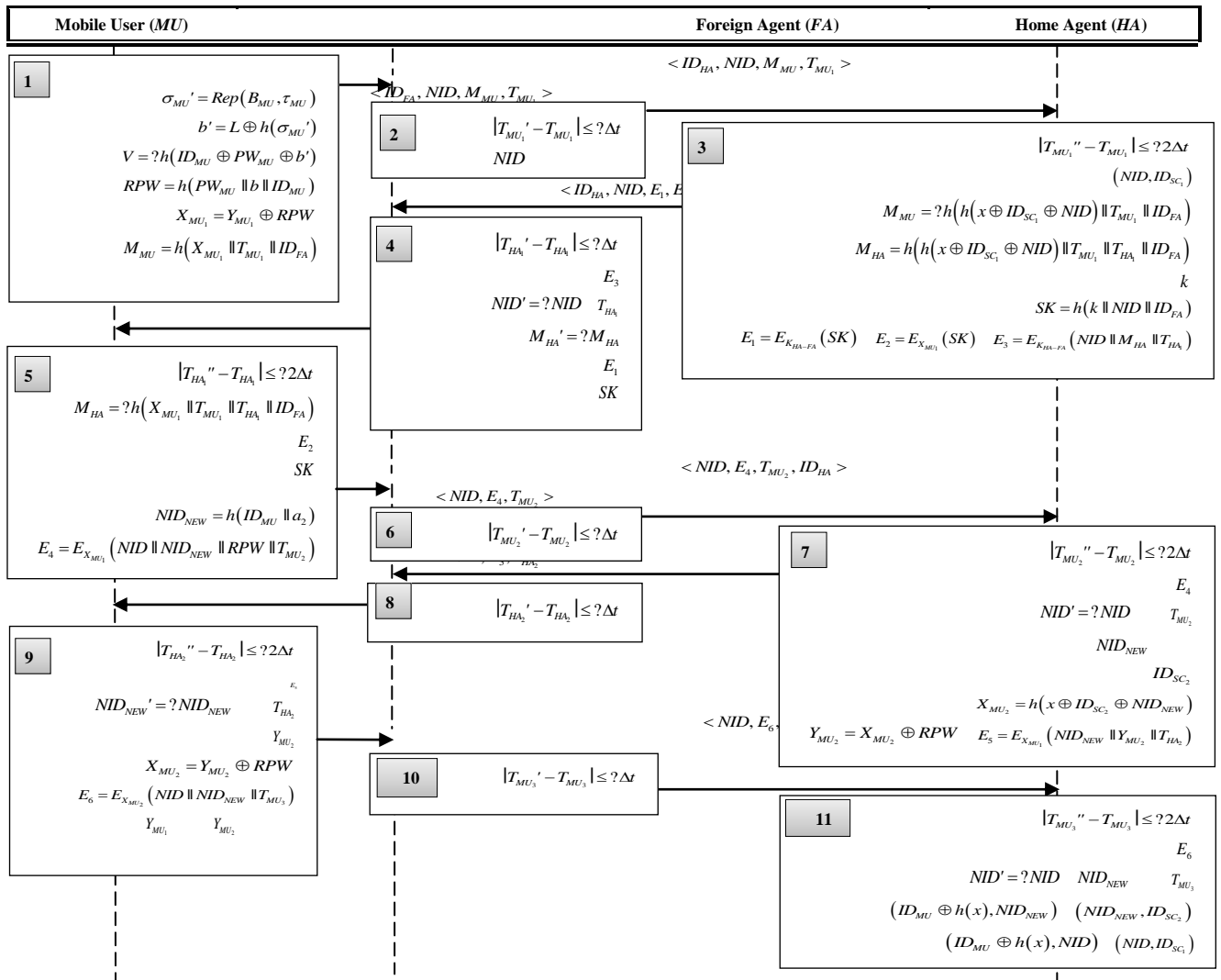
حال، MU پیام $\langle NID, E_4, T_{MU_2}, ID_{HA} \rangle$ را برای FA ارسال می‌کند تا شناسه مستعار جدید خود را به اطلاع HA برساند.

• FA ، پس از دریافت پیام در زمان T_{MU_2}' ، ابتدا درستی رابطه (۶۸) را بررسی می‌کند.

$$|T_{MU_2}' - T_{MU_2}| \leq \Delta t \quad (۶۸)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه FA پیام $\langle NID, E_4, T_{MU_2} \rangle$ را برای HA ارسال می‌کند.

• HA ، پس از دریافت پیام در زمان T_{MU_2}'' ، ابتدا درستی رابطه (۶۹) را بررسی می‌کند.



شکل ۲: مراحل اجرای الگوریتم احراز هویت متقابل و توافق کلید نشست طرح پیشنهادی

خدمات FA استفاده می‌کند. شناسه مستعار MU در انتهای هر بار اجرای این الگوریتم، تغییر می‌کند و گمنامی کاربر به صورت کامل حفظ می‌شود. تأکید می‌شود که از برچسب زمانی به عنوان سنجش شمارنده برای مقابله با حمله تکرار استفاده می‌شود.

۴-۴ الگوریتم تغییر رمز عبور کاربر

این الگوریتم در یک ارتباط امن میان کاربر و کارت هوشمند اجرا می‌شود. زمانی که کاربری رمز عبور خود را فراموش کرده و یا رمز عبورش لو رفته است، می‌تواند با اجرای این الگوریتم آن را تغییر دهد. این الگوریتم در صورت لزوم، بلافاصله بعد از الگوریتم ثبت نام قابل اجراست و به صورت کاربرپسند طراحی شده است. به این منظور که کاربر بدون دخالت و اتلاف وقت اپراتور مرجع و تنها به کمک کارت هوشمند خود، می‌تواند رمز عبور خود را تغییر دهد. مراحل اجرای این الگوریتم که در شکل (۳) نشان داده شده است، به صورت زیر است:

$$|T_{MU_3}'' - T_{MU_3}| \leq ? 2\Delta t \quad (78)$$

اگر برقرار نباشد، نشست خاتمه می‌یابد. وگرنه، مقدار E_6 را با X_{MU_2} رمزگشایی می‌کند. سپس، درستی تساوی $NID' = ? NID$ و مقدارهای NID_{NEW} و T_{MU_3} را بررسی می‌کند. اگر برابر نباشند، نشست خاتمه می‌یابد. وگرنه، مقدارهای (NID_{NEW}, ID_{SC_2}) و $(ID_{MU} \oplus h(x), NID_{NEW})$ را، به ترتیب، در فهرست کاربران فعال و در فهرست کاربران ثبت نام شده خود ذخیره و مقدارهای (NID, ID_{SC_1}) و $(ID_{MU} \oplus h(x), NID)$ را حذف می‌کند.

همان طور که در شکل (۲) مشاهده می‌شود، در صورتی که MU و FA برای HA، به درستی احراز هویت شوند، مجاز بودن MU را به اطلاع FA می‌رساند. علاوه بر این، HA، کلید نشست مشترک میان MU و FA را تولید و به صورت جداگانه برای آن‌ها ارسال می‌کند. در انتهای این الگوریتم، هر سه شرکت کننده مجاز، به صورت متقابل و به درستی برای یکدیگر احراز هویت شده‌اند و MU به صورت گمنام از

سپس، SC_{MU} برای تساوی $V' = ?V$ را بررسی می‌کند. اگر تساوی برابر نباشد (کاربر احراز هویت نشود)، نشست خاتمه می‌یابد. وگرنه، SC_{MU} مقادارهای RPW و X_{MU_2} قدیم را، به ترتیب، مطابق رابطه‌های (۸۲) و (۸۳) به دست می‌آورد و رمز عبور جدید را از MU درخواست می‌کند.

$$RPW = h(PW_{MU}^{OLD} \parallel b \parallel ID_{MU}) \quad (82)$$

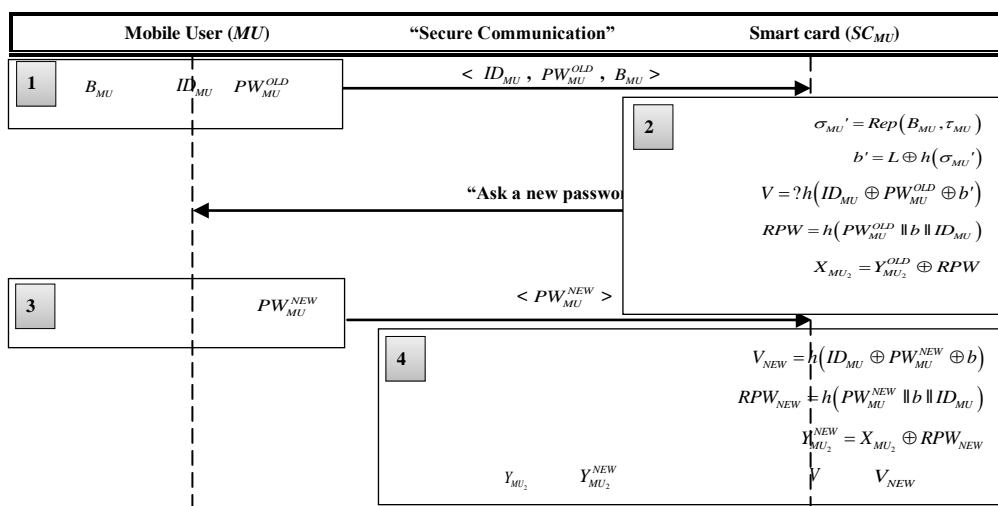
$$X_{MU_2} = Y_{MU_2}^{OLD} \oplus RPW \quad (83)$$

• MU ابتدا کارت هوشمند خود (SC_{MU}) را در دستگاه کارت‌خوان قرار می‌دهد و مقادارهای ID_{MU} ، PW_{MU}^{OLD} و B_{MU} را وارد می‌کند.
• پس از دریافت ورودی‌ها، مقادارهای b' ، σ_{MU}' و V' را، به ترتیب، مطابق رابطه‌های (۷۹)، (۸۰) و (۸۱) به دست می‌آورد.

$$\sigma_{MU}' = Rep(B_{MU}, \tau_{MU}) \quad (79)$$

$$b' = L \oplus h(\sigma_{MU}') \quad (80)$$

$$V' = h(ID_{MU} \oplus PW_{MU}^{OLD} \oplus b') \quad (81)$$



شکل ۳: مراحل اجرای الگوریتم تغییر رمز عبور طرح پیشنهادی

در طرح پیشنهادی، هیچ‌یک از پیام‌های مبادله شده بر روی کانال عمومی، دارای ID_{MU} نیست و هیچ متخاصمی نمی‌تواند با دانستن NID به ID_{MU} برسد. همچنین، ID_{MU} به صورت $ID_{MU} \oplus h(x)$ در فهرست کاربران فعال HA ذخیره می‌شود و تنها HA از آن آگاهی دارد.

۴ + ۵ - عدم ردیابی کاربر

در طرح پیشنهادی تمام پارامترهای پیام‌های مربوط به MU ، در هر الگوریتم احراز هویت و توافق کلید نشست، با تغییر مقادارهای تصادفی انتخاب‌شده توسط MU و HA (k_1 و a_2 ، ID_{SC_2})، تغییر می‌کنند.

۴ + ۵ - حفظ امنیت در صورت سرقت کارت هوشمند

متخاصم A با سرقت SC_{MU} و ثبت پیام‌های مبادله شده روی کانال عمومی، نمی‌تواند پیام درخواست احراز هویت معتبر $\langle ID_{HA}, NID, M_{MU}^*, T_A \rangle$ را در برچسب زمانی فعلی (T_A) تولید کند؛ زیرا باید مقدار M_{MU}^* را مطابق رابطه (۸۷) به دست آورد.

$$M_{MU}^* = h(X_{MU_1} \parallel T_A \parallel ID_{FA}) \quad (87)$$

برای دستیابی به X_{MU_1} معتبر از Y_{MU_1} ، دانستن پارامترهای امنیتی MU (PW_{MU} ، ID_{MU} و B_{MU}) ضروری است.

۴ + ۵ - مقاومت در مقابل حمله حدس رمز عبور به صورت برون خط

• MU رمز عبور جدید خود (PW_{MU}^{NEW}) را انتخاب و وارد دستگاه کارت‌خوان می‌کند.
• SC_{MU} مقادارهای V_{NEW} ، RPW_{NEW} و $Y_{MU_2}^{NEW}$ را، به ترتیب، مطابق رابطه‌های (۸۴)، (۸۵) و (۸۶) محاسبه می‌کند.

$$V_{NEW} = h(ID_{MU} \oplus PW_{MU}^{NEW} \oplus b) \quad (84)$$

$$RPW_{NEW} = h(PW_{MU}^{NEW} \parallel b \parallel ID_{MU}) \quad (85)$$

$$Y_{MU_2}^{NEW} = X_{MU_2} \oplus RPW_{NEW} \quad (86)$$

در نهایت، SC_{MU} مقادارهای V_{NEW} و $Y_{MU_2}^{NEW}$ را در کارت هوشمند ذخیره و مقادارهای V و Y_{MU_2} را از کارت هوشمند حذف می‌کند. همان‌طور که در شکل (۳) مشاهده می‌شود، در انتهای این الگوریتم، رمز عبور کاربر مجاز بدون دخالت HA ، به درستی تغییر می‌یابد. همچنین، در طول اجرای این الگوریتم مقدار X_{MU_2} تغییر نمی‌کند؛ بنابراین، HA هر زمان می‌تواند آن را به درستی بازیابی کند.

۵ - تحلیل امنیت طرح پیشنهادی

۴ + ۵ - تحلیل امنیت طرح پیشنهادی به روش غیررسمی

این بخش، به تحلیل امنیت غیررسمی طرح پیشنهادی می‌پردازد و در جدول (۳) امنیت قوی آن را با طرح‌های گذشته مقایسه می‌کند.

۴ + ۵ - حفظ گمنامی کاربر

در طرح پیشنهادی شناسه ID_{MU} مخفی است و از روی پیام‌های مبادله شده روی کانال عمومی و یا NID ، به دست نمی‌آید.

۵ + ۴ - مقاومت در مقابل حمله جعل هویت کاربر/ اپراتور مرجع/ اپراتور کمکی و حمله فرد میانی

فرضیه‌های متفاوت برای اجرای این حملات توسط متخاصم A :

- A با جعل هویت MU ، برای احراز هویت خود تلاش می‌کند. او برای تولید پیام معتبر $\langle ID_{HA}, NID, M_A, T_A \rangle$ با برچسب زمانی فعلی T_A ، مقدار X_{MU_i} را نیاز دارد و با شکست مواجه می‌شود.
 - A با جعل هویت MU پیام نامعتبر $\langle ID_{HA}, NID, M_A, T_A \rangle$ را با برچسب زمانی فعلی (T_A) تولید و ارسال می‌کند. سپس، A با جعل هویت HA میان ارتباطات HA و FA قرار می‌گیرد. حال A باید کلید مشترک میان HA و FA را داشته باشد تا بتواند پارامترهای E_3 و E_I را به درستی محاسبه کرده و برای FA احراز هویت شود؛ بنابراین، با شکست مواجه می‌شود.
- همان‌طور که در جدول (۳) مشاهده می‌شود، طرح پیشنهادی علاوه بر آن‌که برخی از ویژگی‌های امنیتی، از جمله گمنامی و عدم ردیابی کاربر، احراز هویت متقابل، امنیت کلید نشست، محرمانگی

متخاصم A برای اجرای این حمله، حتی با داشتن SC_{MU} و پیام‌های مبادله شده روی کانال عمومی میان هر سه موجودیت، با شکست مواجه می‌شود؛ زیرا برای بررسی درستی رمز عبور حدس زده شده توسط پارامتر V و یا پارامترهای M_{MU} و M_{HA} ، به ترتیب، نیاز به دانستن ID_{MU} و b و یا X_{MU_i} است.

۵ + ۵ - حفظ امنیت در صورت سرقت تصدیق کننده‌ها

در طرح پیشنهادی، HA مقدار PW_{MU} را ذخیره نمی‌کند و تنها NID و $ID_{MU} \oplus h(x)$ را ذخیره می‌کند. برای بازیابی ID_{MU} از $ID_{MU} \oplus h(x)$ و یا NID ، به ترتیب، پارامترهای x و a_1 مورد نیاز است. همچنین، A با دسترسی به پارامترهای ذخیره شده در فهرست FA و یا SC_{MU} ، نمی‌تواند مدارک یا گواهی‌نامه‌های کاربران را بازیابی کند.

۵ + ۶ - مقاومت در مقابل حمله جعل

در الگوریتم احراز هویت و توافق کلید نشست طرح پیشنهادی، HA کلیدهای مخفی مختلفی را برای درخواست‌های متفاوت کاربر تولید می‌کند؛ بنابراین، اگر متخاصم A خواهان استفاده از NID برای دستیابی به X_{MU_i} باشد، موفق نخواهد بود؛ زیرا در محاسبه X_{MU_i} ، مقدار ID_{SC_i} ، حتی برای NID های مشابه، به صورت تصادفی تولید می‌شود و X_{MU_i} برای هر درخواست، متفاوت خواهد بود. علاوه بر این،

جدول ۳: مقایسه ویژگی‌های امنیتی طرح پیشنهادی با طرح‌های پیشین

طرح پیشنهادی	PAKA [۱۸]	MAKA [۱۷]	AWC [۱۶]	NUA [۱۵]	SAMA [۱۴]	SARS [۱۱]	ویژگی‌های امنیتی مورد نیاز
✓	✗	✓	✓	✗	✗	✓	الگوریتم احراز هویت کارآمد
✓	✗	✓	✓	✓	✗	-	الگوریتم تغییر رمز عبور کاربر پسند
✓	✓	✓	✓	✓	✗	-	الگوریتم تغییر رمز عبور ایمن
✓	✗	✗	✓	✗	✗	✗	احراز هویت اولیه با استفاده از کارت هوشمند
✓	✗	✗	✓	✓	✓	✗	گمنامی کاربر
✓	✓	✓	✗	✓	✓	✗	عدم ردیابی کاربر
✓	✗	✗	✗	✓	✗	✗	احراز هویت متقابل
✓	✓	✗	✓	✓	✓	✓	محرمانگی کامل روبه جلو
✓	✓	✓	✓	✓	✓	✗	محرمانگی کامل رو به عقب
✓	✓	✓	✓	✓	✓	✓	محرمانگی کلید شناخته شده
✓	✗	✗	✗	✓	✗	✓	امنیت کلید نشست
✓	✓	✗	✓	✓	✗	-	امن ماندن در صورت سرقت کارت هوشمند
✓	✓	✗	✓	✓	✓	✓	مقاومت در مقابل حمله حدس رمز عبور به صورت برون خط
✓	✓	✗	✓	✓	✗	✓	امن ماندن در صورت سرقت تصدیق کننده‌ها
✓	✗	✗	✓	✓	✗	✗	مقاومت در مقابل حمله داخلی
✓	✓	✗	✗	✗	✓	✗	مقاومت در مقابل حمله تکرار
✓	✓	✗	✓	✓	✓	✓	مقاومت در مقابل حمله جعل
✓	✓	✓	✗	✓	✗	✗	مقاومت در مقابل حمله فرد میانی و جعل هویت کاربر
✓	✓	✓	✓	✓	✗	✗	مقاومت در مقابل حمله فرد میانی و جعل هویت اپراتور مرجع
✓	✗	✗	✗	✓	✗	✗	مقاومت در مقابل حمله فرد میانی و جعل هویت اپراتور کمکی
✓	✗	✗	✗	✗	✗	✓	مقاومت در مقابل حمله منع سرویس

```

role alice (MU, HA, FA, KGC : agent,
Kmuha : symmetric_key,
Pfa1,Pfa2,Rfa1,Rfa2,Pha1,Pha2,Rha1,Rha2,Ppub : public_key,
% Hash is hash function
Hash, Add, Sub, Mul, Rep, Gen : hash_func,
SND_MH, RCV_MH, SND_MF, RCV_MF : channel(dy))
played_by MU
def=
local State := nat,

IDmu, IDha, IDfa, PWmu, A1, A2, B, K2, Bmu, Tmu1, Tmu2, Tmu3, Tha1, Tha2 : text,
SK : symmetric_key,
NID, NIDnew, RPW, V, L, Xmu1, Xmu2, Ymu1, Ymu2, Mmu, Mha : message,
Inc : hash_func
const server1_alice_sk, alice_server1_nidnew, server1_alice_y_mu2, server1_server2_sk,
subs1, subs2, subs3, subs4, subs5, subs6, subs7, subs8, subs9, subs10, subs11, subs12,
subs13 : protocol_id
init State := 0
transition
1. State = 0 ^ RCV_MH(start) =>
State := 2 ^ B := new() ^ A1 := new()
^ NID := Hash(IDmu, A1)
^ RPW := Hash(PWmu, B, IDmu)
^ SND_MH((IDmu, NID, RPW) _Kmuha)
^ secret((IDmu), subs1, {MU, HA})
^ secret({PWmu, B, A1'}, subs2, MU)
2. State = 2 ^ RCV_MH(Ymu1) _Kmuha =>
State := 4 ^ Tmu1 := new()
^ V := Hash(xor(xor(IDmu, PWmu), B))
^ L := xor(B, Hash(Bmu))
^ Xmu1 := xor(Ymu1, RPW)
^ Mmu := Hash(Xmu1, Tmu1, IDfa)
^ SND_MF(IDha, NID, Mmu, Tmu1)
3. State = 4 ^ RCV_MF(Mha) (SK) _Xmu1, Tha1 =>
State := 6 ^ Tmu2 := new() ^ A2 := new()
^ NIDnew := Hash(IDmu, A2)
^ SND_MF(NID, {NID, NIDnew, RPW, Tmu2} _Xmu1, Tmu2, IDha)
^ request(MU, HA, server1_alice_sk, SK)
^ witness(MU, HA, alice_server1_nidnew, NIDnew)
^ secret({A2'}, subs3, MU)
^ secret({NIDnew'}, subs4, {MU, HA})
4. State = 6 ^ RCV_MF(NID, {NIDnew, Ymu2, Tha2} _Xmu1, Tha2) =>
State := 8 ^ Xmu2 := xor(Ymu2, RPW) ^ Tmu3 := new()
^ SND_MF(NID, {NID, NIDnew, Tmu3} _Xmu2, IDha, Tmu3)
^ request(MU, HA, server1_alice_y_mu2, Ymu2)
end role

```

شکل ۴: مشخصات زبان HLPSSL برای نقش آغازکننده MU

اعلامیه^{۵۷} $secret(\{ID_{MU}\}, \{subs1\}, \{MU, HA\})$ نشان می‌دهد که شناسه کاربر (ID_{MU}) را تنها MU و HA می‌دانند و اعلامیه $secret(\{PW_{MU, B}, A1'\}, \{subs2\}, MU)$ نشان می‌دهد که رمز عبور PW_{MU} و مقادیرهای جدید b و a_1 را تنها MU می‌داند. سپس، یک کارت هوشمند، شامل اطلاعات $\{Y_{MU}, h(\cdot)\}$ را که با کلید متقارن K_{muha} رمز شده، با کمک عملیات RCV_MH ، از HA دریافت می‌کند. در طول الگوریتم احراز هویت متقابل و توافق کلید نشست، MU پیام درخواست احراز هویت $\langle ID_{HA}, NID, M_{MU}, T_{MU1} \rangle$ را، با کمک عملیات SND_MF ، برای FA ارسال کرده و پیام $\langle M_{HA}, E_2, T_{HA} \rangle$ را، با کمک عملیات RCV_MF ، از FA دریافت می‌کند. در نتیجه، به کلید مشترک میان خود و FA (SK) می‌رسد و به راحتی HA و FA را احراز هویت می‌کند. پس از آن، MU پیام $\langle NID, E_4, T_{MU2}, ID_{HA} \rangle$ را، با کمک عملیات SND_MF ، برای FA ارسال می‌کند. اعلامیه $request(MU, HA, server1_alice_sk, SK')$ نشان می‌دهد که MU مقدار SK را که توسط HA تولید می‌شود، دریافت کرده است. اعلامیه $witness(MU, HA, alice_server1_nidnew, NIDnew')$ نشان می‌دهد که MU مقدار جدید NID_{NEW} را محاسبه می‌کند. اعلامیه $secret(\{A2'\}, \{subs3\}, MU)$ نشان می‌دهد که مقدار جدید a_2' را تنها MU می‌داند. $secret(\{NIDnew'\}, \{subs4\}, \{MU, HA\})$ نشان می‌دهند که مقدار جدید NID_{NEW} را تنها MU و HA می‌دانند. سپس، MU پیام

کامل روبه‌جلو، محرمانگی کامل رو به عقب و محرمانگی کلید شناخته‌شده را فراهم می‌آورد، در مقابل حملات فعال و غیرفعال موجود در شبکه، از جمله حمله تکرار، حمله داخلی، حمله جعل هویت کاربر/ اپراتور مرجع/ اپراتور کمکی، حمله به پروتکل در صورت سرقت کارت هوشمند و یا تصدیق‌کننده‌ها، حمله حدس رمز عبور به صورت برون‌خط، حمله فرد میانی و حمله منع سرویس نیز مقاوم است؛ بنابراین، طرح پیشنهادی نسبت به طرح‌های پیشین کارآمدتر بوده و از امنیت بالاتری برخوردار است.

۴-۵ - تحلیل امنیت طرح پیشنهادی به روش رسمی^{۵۴}

در این بخش برای تحلیل امنیت طرح پیشنهادی به روش رسمی از نرم‌افزار شبیه‌ساز AVISPA استفاده می‌شود [۲۸]. نتایج شبیه‌سازی با این ابزار این اطمینان را می‌دهد که طرح پیشنهادی امن و یا ناامن است. در این ابزار برای توصیف پروتکل پیشنهادی و مشخص کردن ویژگی‌های امنیتی مورد نظر، از زبان HLPSSL استفاده می‌شود. HLPSSL یک زبان نقش‌گرا است که هر شرکت‌کننده نقش مستقل از دیگران را اجرا می‌کند و ارتباط با دیگر نقش‌ها از طریق کانال‌ها امکان‌پذیر است. نقش‌های HLPSSL شامل نقش‌های پایه و مرکب است و نقش‌های پایه توسط نقش‌های مرکب به صورت موازی اجرا می‌شود. مشخصات^{۵۴} طرح پیشنهادی در الگوریتم ثبت‌نام و الگوریتم احراز هویت و توافق کلید نشست توسط زبان HLPSSL به اجرا گذاشته می‌شود. در این اجراء نقش‌های alice، server اول، server دوم، session و environment شرح داده شده‌اند. سه نقش پایه server، alice، server اول و server دوم، به ترتیب، نشان‌دهنده شرکت‌کننده‌های کاربر معتبر تلفن همراه (MU)، اپراتور مرجع (HA) و اپراتور کمکی (FA) می‌باشند. برخی از اصطلاحات در زیر تعریف شده‌اند:

- Agent: نشان‌دهنده هویت اصلی موجودیت‌ها.
 - Symmetric_key: نشان‌دهنده کلید مورد استفاده برای رمزنگاری متقارن.
 - Public_key: نشان‌دهنده کلید عمومی موجودیت‌ها.
 - Hash_func: نشان‌دهنده تابع درهم‌ساز رمزنگاری.
 - Channel(dy): نشان‌دهنده کانال‌های ارتباطی از نوع مدل متخاصم Dolev-Yao میان موجودیت‌ها.
 - Played_by: نشان‌دهنده نام موجودیت بازی‌کننده نقش.
 - New(): تابع تولیدکننده اعداد تصادفی.
- مشخصات زبان HLPSSL برای نقش آغازکننده MU، در شکل (۴) نشان داده شده است.

همان‌طور که در شکل (۴) مشاهده می‌شود، در طول الگوریتم ثبت‌نام، نقش آغازکننده MU برای اولین بار پیام "start" را دریافت می‌کند و پیام $\langle ID_{MU}, NID, RPW \rangle$ را، با کلید متقارن K_{muha} رمز کرده و با کمک عملیات SND_MH ، آن را برای HA ارسال می‌کند.

```

role server1 (MU, HA, FA, KGC: agent,
  Kmuha : symmetric_key,
  Pfa1,Pfa2,Rfa1,Rfa2,Pha1,Pha2,Rha1,Rha2,Ppub : public_key,
  % Hash is hash function
  Hash, Add, Sub, Mul, Rep, Gen : hash_func,
  SND_HM, RCV_HM, SND_HF, RCV_HF: channel(dy))
played_by HA
def=
local State : nat,
IDmu, IDha, IDfa, IDsc1, IDsc2, Lha1, Lha2, P, K1, Sha1, Sha2, X,
  Xha1, Xha2, Tmu1, Tmu2, Tmu3, Tha1, Tha2, Tfa : text,
  SK, Khafa : symmetric_key,
NID, NIDnew, RPW, Xmu1, Xmu2, Ymu1, Ymu2, Mmu, Wfa1, Wfa2,
  Efa1, Efa2, Eha1, Eha2, D1, D2, D3, D4, Khf1, Khf2, Khf3, Khf4, Mha : message,
  Inc : hash_func
const server1_alice_sk, alice_server1_nidnew, server1_alice_ymu2, server1_server2_sk,
  subs1, subs2, subs3, subs4, subs5, subs6, subs7, subs8, subs9, subs10, subs11, subs12,
  subs13 : protocol_id
init State := 1
transition
1. State = 1  $\wedge$  RCV_HM({IDmu.NID.RPW}, Kmuha) =>
  State' := 3  $\wedge$  IDsc1' := new()  $\wedge$  Xmu1' := Hash(xor(xor(X,IDsc1),NID))
   $\wedge$  Ymu1' := xor(Xmu1'.RPW)  $\wedge$  SND_HM({Ymu1'}, Kmuha)
   $\wedge$  secret({Xmu1'}, subs5, {HA, MU})
   $\wedge$  secret({X, IDsc1'}, subs6, HA)
2. State = 3  $\wedge$  RCV_HF(IDfa.NID.Mmu'.Tmu1') =>
  State' := 5  $\wedge$  Lha1' := new()  $\wedge$  Lha2' := new()
   $\wedge$  Eha1' := Mul(Lha1'.P)  $\wedge$  Eha2' := Mul(Lha2'.P)
   $\wedge$  SND_HF(HA.Eha1'.Eha2'.Rha1.Rha2)
   $\wedge$  secret({Lha1', Lha2'}, subs7, HA)
3. State = 5  $\wedge$  RCV_HF(IDfa.Efa1'.Efa2'.Rfa1.Rfa2) =>
  State' := 7  $\wedge$  Wfa1' := Add(Rfa1.Mul(Hash(IDfa.Rfa1).Ppub))
   $\wedge$  Wfa2' := Add(Rfa2.Mul(Hash(IDfa.Rfa2).Ppub))
   $\wedge$  D1' := Hash(IDfa.IDha.Efa1'.Efa2'.Eha1.Eha2)
   $\wedge$  D2' := Hash(IDha.IDfa.Eha1.Eha2.Efa1'.Efa2')
   $\wedge$  D3' := Hash(IDfa.IDha.Rfa1.Rfa2.Rha1.Rha2.Pfa1
    .Pfa2.Pha1.Pha2.Efa1'.Efa2'.Eha1.Eha2)
   $\wedge$  D4' := Hash(IDha.IDfa.Rha1.Rha2.Rfa1.Rfa2.Pha1
    .Pha2.Pfa1.Pfa2.Eha1.Eha2.Efa1'.Efa2')
   $\wedge$  Khf1' := Mul(Add(Mul(D2.Lha1), Sha1.Mul(D4.Xha2)),
    Add(Mul(D1.Efa2), Wfa2.Mul(D3.Pfa1)))
   $\wedge$  Khf2' := Mul(Add(Mul(D2.Lha2), Sha2.Mul(D4.Xha1)),
    Add(Mul(D1.Efa2), Wfa2.Mul(D3.Pfa1)))
   $\wedge$  Khf3' := Mul(Add(Mul(D2.Lha1), Sha1.Mul(D4.Xha2)),
    Add(Mul(D1.Efa1), Wfa1.Mul(D3.Pfa2)))
   $\wedge$  Khf4' := Mul(Add(Mul(D2.Lha2), Sha2.Mul(D4.Xha1)),
    Add(Mul(D1.Efa1), Wfa1.Mul(D3.Pfa2)))
   $\wedge$  Khafa' := Hash(IDha.IDfa.Eha1.Eha2.Efa1'
    .Efa2'.Khf1'.Khf2'.Khf3'.Khf4)
   $\wedge$  Tha1' := new()  $\wedge$  K' := new()
   $\wedge$  Mha' := Hash(Hash(xor(xor(X,IDsc1),NID)).Tmu1.Thal'.IDfa)
   $\wedge$  SK' := Hash(K'.IDmu.IDfa)
  SND_HF(IDha.NID.{SK'}_Khafa'.{SK'}_Xmu1.Mha'.Thal'.{NID.Mha'.Thal'}_Khafa')
   $\wedge$  witness(HA,FA,server1_server2_sk,SK')
   $\wedge$  witness(HA,MU,server1_alice_sk,SK')
   $\wedge$  secret({Khafa'}, subs8, {HA,FA})
   $\wedge$  secret({SK'}, subs9, {FA,MU})
   $\wedge$  secret({K',Sha1,Sha2,Xha1,Xha2}, subs10, HA)
4. State = 7  $\wedge$  RCV_HF(NID.{NID.NIDnew'.RPW.Tmu2'}_Xmu1.Tmu2') =>
  State' := 9  $\wedge$  IDsc2' := new()  $\wedge$  Tha2' := new()
   $\wedge$  Xmu2' := Hash(xor(xor(X,IDsc2),NIDnew'))
   $\wedge$  Ymu2' := xor(Xmu2'.RPW)
   $\wedge$  SND_HF(NID.{NIDnew'.Ymu2'.Tha2'}_Xmu1.Tha2')
   $\wedge$  secret({IDsc2'}, subs11, HA)
   $\wedge$  secret({Xmu2'}, subs12, {HA, MU})
   $\wedge$  secret({IDsc2'}, subs13, HA)
   $\wedge$  witness(HA,MU,server1_alice_ymu2,Ymu2')
   $\wedge$  request(HA,MU,alice_server1_nidnew,NIDnew')
5. State = 9  $\wedge$  RCV_HF(NID.{NID.NIDnew.Tmu3'}_Xmu2.Tmu3') =>
  State' := 11
end role

```

شکل ۵: مشخصات زبان HLPSSL برای نقش پاسخ‌دهنده HA

ID_{SC_2} مقدار جدید $secret(\{IDsc2'\}, subs11, HA)$ را تنها HA می‌داند. اعلامیه $secret(\{Xmu2'\}, subs12, \{HA, MU\})$ نشان می‌دهد که مقدار مخفی جدید کاربر (X_{MU_2}) را تنها HA و MU می‌دانند. اعلامیه $witness(HA, MU, server1_alice_ymu2, Ymu2')$ نشان می‌دهد که HA مقدار جدید Y_{MU_2} را محاسبه می‌کند. اعلامیه $request(HA, MU, alice_server1_nidnew, NIDnew')$ که HA مقدار جدید NID_{NEW} را که توسط MU تولید می‌شود، دریافت کرده است. درنهایت، پیام $\langle NID, E_6, T_{MU_3} \rangle$ را با کمک عملیات RCV_HF از FA دریافت می‌کند. مشخصات زبان HLPSSL برای نقش پاسخ‌دهنده FA، در شکل (۶) نشان داده شده است.

$\langle NID, E_5, T_{HA_2} \rangle$ را با کمک عملیات RCV_MF از FA دریافت می‌کند و درنهایت، پیام $\langle NID, E_6, ID_{HA}, T_{MU_3} \rangle$ را با کمک عملیات SND_MF برای FA ارسال می‌کند. اعلامیه $request(MU, HA, server1_alice_ymu2, Ymu2')$ نشان می‌دهد که مقدار جدید Y_{MU_2} را که توسط HA تولید می‌شود، دریافت کرده است. نوع کانال اعلام شده dy است؛ بنابراین، متخاصم توانایی ره‌گیری، شنود، تحلیل و تغییر پیام ارسالی روی کانال ناامن را دارد. مشخصات زبان HLPSSL برای نقش پاسخ‌دهنده HA، در شکل (۵) نشان داده شده است.

همان‌طور که در شکل (۵) مشاهده می‌شود، در طول الگوریتم ثبت‌نام، HA پیام درخواست ثبت‌نام $\langle ID_{MU}, NID, RPW \rangle$ را که با کلید متقارن K_{muha} رمز شده و با کمک عملیات RCV_HM از MU دریافت کرده و یک کارت هوشمند شامل اطلاعات $\{Y_{MU}, h(\cdot)\}$ را، به‌صورت رمز شده با کلید متقارن K_{muha} و با کمک عملیات SND_HM برای MU ارسال می‌کند. اعلامیه $secret(\{X, IDsc1'\}, subs6, HA)$ نشان می‌دهد که مقدار مخفی جدید X_{MU_1} را تنها HA و MU می‌دانند. در طول الگوریتم احراز هویت متقابل و توافق کلید نشست، HA پیام $\langle ID_{FA}, NID, M_{MU}, T_{MU_1} \rangle$ را، با کمک عملیات RCV_HF از FA دریافت کرده و سپس، با کمک عملیات SND_HF پیام $\langle ID_{HA}, E_{HA_1}, E_{HA_2}, R_{HA_1}, R_{HA_2} \rangle$ را برای FA ارسال می‌کند. اعلامیه $secret(\{Lha1', Lha2'\}, subs7, HA)$ و e_{HA_1} و e_{HA_2} را تنها HA می‌داند. سپس، HA پیام $\langle ID_{FA}, E_{FA_1}, E_{FA_2}, R_{FA_1}, R_{FA_2} \rangle$ را، با کمک عملیات RCV_HF از FA دریافت کرده و به محاسبه کلید مشترک میان خود و FA، به روش $CL\text{-}AKA$ [۲۷] می‌پردازد. علاوه بر این، HA به‌راحتی می‌تواند MU و FA را احراز هویت کند. سپس، HA، با کمک عملیات SND_HF پیام $\langle ID_{HA}, NID, E_1, E_2, E_3, M_{HA}, T_{HA_1} \rangle$ را برای FA ارسال می‌کند. $witness(HA, MU, server1_alice_sk, SK')$ و هر دو نشان می‌دهند که HA مقدار جدید SK را محاسبه می‌کند. اعلامیه $secret(\{Khafa'\}, subs8, \{HA, FA\})$ و $secret(\{SK'\}, subs9, \{FA, MU\})$ نشان می‌دهند که مقدار جدید K_{HA-FA} را تنها HA و FA می‌دانند. نشان می‌دهند مقدار جدید SK را تنها FA و MU می‌دانند. مقدارهای $K, s_{HA_1}, s_{HA_2}, x_{HA_1}, x_{HA_2}$ را تنها HA می‌داند و سپس، نشان می‌دهند مقدار جدید SK را تنها FA و MU می‌دانند. $secret(\{K', Sha1, Sha2, Xha1, Xha2\}, subs10, HA)$ مقدارهای $K, s_{HA_1}, s_{HA_2}, x_{HA_1}, x_{HA_2}$ را تنها HA می‌داند و سپس، $secret(\{K', Sha1, Sha2, Xha1, Xha2\}, subs10, HA)$ نشان می‌دهد که مقدارهای $K, s_{HA_1}, s_{HA_2}, x_{HA_1}, x_{HA_2}$ را تنها HA می‌داند و سپس، پیام $\langle NID, E_4, T_{MU_2} \rangle$ را با کمک عملیات RCV_HF از FA دریافت می‌کند و پیام $\langle NID, E_5, T_{HA_2} \rangle$ را با کمک عملیات SND_HF برای FA ارسال می‌کند.

request(FA, HA , اعلامیه می‌کند. ارسال MU برای SND_FM server1_server2_sk, SK) را که توسط HA تولید می‌شود، دریافت کرده است. FA پیام $<NID, E_4, T_{MU_2}, ID_{HA}>$ را، با کمک عملیات RCV_FM از MU دریافت می‌کند و پیام $<NID, E_4, T_{MU_2}>$ را با کمک عملیات SND_FH برای HA ارسال می‌کند. پس از آن، FA پیام $<NID, E_5, T_{HA_2}>$ را با کمک عملیات RCV_FH از HA دریافت می‌کند و پیام $<NID, E_5, T_{HA_2}>$ را با کمک عملیات SND_FM برای MU ارسال می‌کند. در نهایت، FA پیام $<NID, E_6, ID_{HA}, T_{MU_3}>$ را با کمک عملیات RCV_FM از MU دریافت می‌کند و پیام $<NID, E_6, T_{MU_3}>$ را با کمک عملیات SND_FH برای HA ارسال می‌کند. مشخصات زبان $HLPSL$ برای نقش نشست در شکل (۷) و برای نقش محیط و هدف در شکل (۸) نشان داده شده است.

role session (MU, HA, FA, KGC: agent, Kmuha : symmetric_key, Pfa1, Pfa2, Rfa1, Rfa2, Pha1, Pha2, Rha1, Rha2, Ppub : public_key, Hash, Add, Sub, Mul, Rep, Gen : hash_func)	def=local
SMH, RMH, SMF, RMF, SHM, RHM, SHF, RHF, SFM, RFM, SFH, RFH : channel (dy)	composition
alice (MU, HA, FA, KGC, Kmuha, Pfa1, Pfa2, Rfa1, Rfa2, Pha1, Pha2, Rha1, Rha2, Ppub, Hash, Add, Sub, Mul, Rep, Gen, SMH, RMH, SMF, RMF)	
server1 (MU, HA, FA, KGC, Kmuha, Pfa1, Pfa2, Rfa1, Rfa2, Pha1, Pha2, Rha1, Rha2, Ppub, Hash, Add, Sub, Mul, Rep, Gen, SHM, RHM, SHF, RHF)	
server2 (MU, HA, FA, KGC, Pfa1, Pfa2, Rfa1, Rfa2, Pha1, Pha2, Rha1, Rha2, Ppub, Hash, Add, Sub, Mul, Rep, Gen, SFM, RFM, SFH, RFH)	
	end role

شکل ۷: مشخصات زبان $HLPSL$ برای نقش نشست

همان‌طور که در شکل (۷) مشاهده می‌شود، در نقش نشست، تمام نقش‌های پایه، با آرگومان‌های خود، فراخوانی شده‌اند. در شکل (۸) نقش محیط شامل ثابت‌های سراسری و ترکیبی از یک یا چند نشست است که در آن متخاصم ممکن است برخی از نقش‌ها را به‌عنوان یک کاربر قانونی بازی کند. ۱۳ هدف امنیتی و چهار احراز هویت در طرح پیشنهادی تصدیق شده است، موارد تصدیق شده به‌صورت زیر تعریف می‌شوند:

- Secrecy_of subs1: شناسه کاربر (ID_{MU})، میان کاربر مجاز و اپراتور مرجع مخفی نگه‌داشته می‌شود.
- Secrecy_of subs2: رمز عبور (PW_{MU}) و پارامتر تصادفی a_1 و b تنها برای کاربر مجاز مخفی نگه‌داشته می‌شوند.
- Secrecy_of subs3: پارامتر تصادفی a_2 تنها برای کاربر مجاز مخفی نگه‌داشته می‌شود.
- Secrecy_of subs4: شناسه مستعار جدید کاربر (NID_{new})، میان کاربر مجاز و اپراتور مرجع مخفی نگه‌داشته می‌شود.
- Secrecy_of subs5: مقدار مخفی کاربر (X_{MU})، میان کاربر مجاز و اپراتور مرجع مخفی نگه‌داشته می‌شود.
- Secrecy_of subs6: پارامترهای x و ID_{SC_i} تنها برای اپراتور مرجع مخفی نگه‌داشته می‌شوند.

role	server2 (MU, HA, FA, KGC: agent, Pfa1, Pfa2, Rfa1, Rfa2, Pha1, Pha2, Rha1, Rha2, Ppub : public_key, Hash, Add, Sub, Mul, Rep, Gen : hash_func, SND_FM, RCV_FM, SND_FH, RCV_FH: channel(dy))	played_by FA
		def=local State : nat,
	IDha, IDfa, Lfa1, Lfa2, P, Sfa1, Sfa2, Xfa1, Xfa2, Tmu1, Tmu2, Tmu3, Tha1, Tha2, Tfa: text,	
	SK, Khafa : symmetric_key,	
	NID, Mmu, Mha, Wha1, Wha2, Efa1, Efa2, Eha1, Eha2, D1, D2, D3, D4, Kfh1, Kfh2, Kfh3, Kfh4 : message,	
	M : {symmetric_key}_symmetric_key,	
	N : {message.message.message.text}_symmetric_key,	
	O : {message.message.text}_symmetric_key,	
	Q : {message.message.text}_symmetric_key,	
	Inc: hash_func	
	const server1_alice_sk, alice_server1_nidnew, server1_alice_ymu2, server1_server2_sk, subs1, subs2, subs3, subs4, subs5, subs6, subs7, subs8, subs9, subs10, subs11, subs12, subs13 : protocol_id	
	init State := 13	
	transition	
	1. State = 13 \wedge RCV_FM(IDha.NID.Mmu'.Tmu1') => State := 15 \wedge SND_FH(IDfa.NID.Mmu'.Tmu1')	
	2. State = 15 \wedge RCV_FH(IDha.Eha1'.Eha2'.Rha1.Rha2) => State := 17 \wedge Lfa1' := new() \wedge Lfa2' := new() \wedge Efa1' := Mul(Lfa1'.P) \wedge Efa2' := Mul(Lfa2'.P) \wedge Wha1' := Add(Rha1.Mul(Hash(HA.Rha1).Ppub)) \wedge Wha2' := Add(Rha2.Mul(Hash(HA.Rha2).Ppub)) \wedge D1' := Hash(IDfa.IDha.Efa1'.Efa2'.Eha1'.Eha2) \wedge D2' := Hash(IDha.IDfa.Eha1'.Eha2'.Efa1'.Efa2) \wedge D3' := Hash(IDfa.IDha.Rfa1.Rfa2.Rha1.Rha2.Pfa1.Pfa2.Pha1.Pha2.Efa1'.Efa2'.Eha1'.Eha2) \wedge D4' := Hash(IDha.IDfa.Rha1.Rha2.Rfa1.Rfa2.Pha1.Pha2.Pfa1.Pfa2.Eha1'.Eha2'.Efa1'.Efa2) \wedge Kfh1' := Mul(Add(Mul(D1.Lfa2'), Sfa2, Mul(D3.Xfa1)) .Add(Mul(D2.Eha1'), Wha1, Mul(D4.Pha2))) \wedge Kfh2' := Mul(Add(Mul(D1.Lfa2'), Sfa2, Mul(D3.Xfa1)) .Add(Mul(D2.Eha2'), Wha2, Mul(D4.Pha1))) \wedge Kfh3' := Mul(Add(Mul(D1.Lfa1'), Sfa1, Mul(D3.Xfa2)) .Add(Mul(D2.Eha1'), Wha1, Mul(D4.Pha2))) \wedge Kfh4' := Mul(Add(Mul(D1.Lfa2'), Sfa1, Mul(D3.Xfa2)) .Add(Mul(D2.Eha2'), Wha2, Mul(D4.Pha1))) \wedge Khafa' := Hash(IDha.IDfa.Eha1'.Eha2'.Efa1'.Efa2'.Kfh1'.Kfh2'.Kfh3'.Kfh4) \wedge SND_FH(IDfa.Efa1'.Efa2'.Rfa1.Rfa2) \wedge secret({Lfa1', Lfa2', Sfa1, Sfa2, Xfa1, Xfa2}, subs13, FA) \wedge secret(Khafa', subs8, {HA, FA})	
	3. State = 17 \wedge RCV_FH(IDha.NID. {SK}'_Khafa.M'. {NID.Mha}'_Khafa.Mha'.Tha1') => State := 19 \wedge SND_FM(Mha'.M'.Tha1') \wedge request(FA, HA, server1_server2_sk, SK)	
	4. State = 19 \wedge RCV_FM(NID.N'.Tmu2'.IDha) => State := 21 \wedge SND_FH(NID.N'.Tmu2')	
	5. State = 21 \wedge RCV_FH(NID.O'.Tha2') => State := 23 \wedge SND_FM(NID.O'.Tha2')	
	6. State = 23 \wedge RCV_FM(NID.Q'.IDha.Tmu3') => State := 25 \wedge SND_FH(NID.Q'.Tmu3')	
	end role	

شکل ۸: مشخصات زبان $HLPSL$ برای نقش پاسخ‌دهنده FA

در شکل ۸، اپراتور کمکی FA پیام درخواست احراز هویت $<ID_{HA}, NID, M_{MU}, T_{MU_1}>$ را، در طول الگوریتم احراز هویت و با کمک عملیات RCV_FM ، از MU دریافت کرده و پیام $<ID_{FA}, NID, M_{MU}, T_{MU_1}>$ را، با کمک عملیات SND_FH ، برای HA ارسال می‌کند. سپس، FA ، با کمک عملیات RCV_FH ، پیام $<ID_{HA}, E_{HA_1}, E_{HA_2}, R_{HA_1}, R_{HA_2}>$ را از HA دریافت می‌کند و به محاسبه کلید مشترک میان خود و HA به روش $CL-ACA [27]$ می‌پردازد. پس از آن، FA ، با کمک عملیات SND_FH ، پیام $<ID_{FA}, E_{FA_1}, E_{FA_2}, R_{FA_1}, R_{FA_2}>$ را برای HA ارسال می‌کند. اعلامیه $(secret(K_{HA-FA}, subs8, \{HA, FA\}))$ و اعلامیه $(secret(\{Lfa1', Lfa2', Sfa1, Sfa2, Xfa1', Xfa2'\}, subs13, FA))$ ، به ترتیب، نشان می‌دهند که مقدار تازه K_{HA-FA} را تنها HA و FA می‌دانند و مقدارهای $L_{FA_1}, L_{FA_2}, S_{FA_1}, S_{FA_2}, X_{FA_1}, X_{FA_2}$ را تنها FA می‌داند. سپس، FA ، پیام $<ID_{HA}, NID, E_1, E_2, E_3, M_{HA}, T_{HA}>$ را، با کمک عملیات RCV_FH ، از HA دریافت کرده و به‌راحتی HA و MU را احراز هویت می‌کند و پیام $<M_{HA}, E_2, T_{HA}>$ را، با کمک عملیات

- اپراتور مرجع NID_{new} را از پیام‌های کاربر مجاز دریافت کند، پس اپراتور مرجع، کاربر مجاز را احراز هویت کرده است.
 - $Authentication_on\ server1_alice_ymu2$: اپراتور مرجع مقدار Y_{MU_2} را برای کاربر مجاز تولید کرده است. به طوری که Y_{MU_2} را تنها کاربر مجاز می‌داند. اگر کاربر مجاز Y_{MU_2} را از پیام‌های اپراتور مرجع دریافت کند، پس کاربر مجاز، اپراتور مرجع را احراز هویت کرده است.
 - $Authentication_on\ server1_server2_SK$: اپراتور مرجع یک کلید نشست SK برای کاربر مجاز و اپراتور کمکی تولید کرده است. به طوری که SK را تنها اپراتور مرجع می‌داند. اگر اپراتور کمکی، SK را از پیام‌های اپراتور مرجع دریافت کند، اپراتور کمکی، اپراتور مرجع را احراز هویت کرده است.
- نتایج شبیه‌سازی برای تحلیل تصدیق امنیت طرح پیشنهادی با استفاده از $OFMC^{SM}$ در شکل (۹) نشان داده شده است.



شکل ۹: خروجی گزارش شده توسط نرم‌افزار AVISPA برای طرح پیشنهادی

شکل (۹) نتایج شبیه‌سازی برای تحلیل تصدیق امنیت طرح پیشنهادی با استفاده از ابزار AVISPA و جستجوگر مدل $OFMC$ را نشان می‌دهد. جستجوگر مدل $OFMC$ برای تشخیص سریع حملات و اثبات درستی پروتکل استفاده می‌شود. همچنین، مشخصات $OFMC$ $HLPSSL$ طرح پیشنهادی را بررسی می‌کند و تشخیص می‌دهد که آیا حمله‌ای غیرفعال نی‌ز رخ داده است یا خیر. نتایج این اطمینان را به ما می‌دهند که طرح پیشنهادی در مقابل انواع حملات فعال و غیرفعال، از جمله حمله تکرار و حمله فرد میانی، امن است. خلاصه‌ای از نتایج تحت $OFMC$ ، گزارش امن بودن طرح پیشنهادی را نشان می‌دهد.

۶ - تحلیل کارایی طرح پیشنهادی

این بخش، همان‌طور که در جدول (۴) نشان داده شده است، به محاسبه و مقایسه هزینه محاسباتی برحسب پیچیدگی زمانی عملگرهای مورداستفاده، زمان اجرا (در الگوریتم ثبت‌نام و الگوریتم احراز هویت)، هزینه ارتباطی (تنها در الگوریتم احراز هویت) و سربار



شکل ۸: مشخصات زبان HLPSSL برای نقش محیط و هدف

- $Secrecy_of\ subs7$: پارامترهای تصادفی e_{HA_2} و e_{HA_1} تنها برای اپراتور مرجع مخفی نگه‌داشته می‌شوند.
- $Secrecy_of\ subs8$: کلید مشترک K_{HA-FA} میان اپراتور مرجع و اپراتور کمکی مخفی نگه‌داشته می‌شود.
- $Secrecy_of\ subs9$: کلید نشست KS ، میان کاربر مجاز و اپراتور کمکی مخفی نگه‌داشته می‌شود.
- $Secrecy_of\ subs10$: پارامترهای $K_I, s_{HA_2}, s_{HA_1}, x_{HA_2}$ و x_{HA_1} تنها برای اپراتور مرجع مخفی نگه‌داشته می‌شوند.
- $Secrecy_of\ subs11$: پارامتر تصادفی ID_{SC_2} تنها برای اپراتور مرجع مخفی نگه‌داشته می‌شود.
- $Secrecy_of\ subs12$: مقدار مخفی جدید کاربر (X_{MU_2}) ، میان کاربر مجاز و اپراتور مرجع مخفی نگه‌داشته می‌شود.
- $Secrecy_of\ subs13$: پارامترهای $e_{FA}, x_{FA_2}, x_{FA_1}, s_{FA_2}, s_{FA_1}$ و e_{FA} تنها برای اپراتور مرجع مخفی نگه‌داشته می‌شوند.
- $Authentication_on\ server1_alice_SK$: اپراتور مرجع یک کلید نشست (SK) برای کاربر مجاز و اپراتور کمکی تولید کرده است. به طوری که کلید نشست ذکر شده را تنها اپراتور مرجع می‌داند (در طرح پیشنهادی اپراتور مرجع کلید نشست را ذخیره نمی‌کند). اگر کاربر مجاز، کلید نشست SK را از پیام‌های اپراتور مرجع دریافت کند، پس کاربر مجاز، اپراتور مرجع را احراز هویت کرده است.
- $Authentication_on\ alice_server1_nidnew$: کاربر مجاز یک شناسه مستعار جدید (NID_{new}) را برای خود و اپراتور مرجع تولید کرده است. به طوری که NID_{new} را تنها کاربر مجاز می‌داند. اگر

در این مقاله برای ارائه هزینه محاسباتی و ارتباطی دقیق، از داده‌های آزمایشی گزارش شده در [۲۹] استفاده می‌شود که در آن نویسندگان زمان اجرای عملیات رمزنگاری را با استفاده از نرم‌افزار Visual C++ نسخه چاپ شده در سال ۲۰۰۸، کتابخانه MIRACL و در محیطی با مشخصات واحد پردازش مرکزی Intel(R) Core(TM)2T6570 همراه با فرکانس ۲/۱ گیگاهرتز، حافظه چهار گیگابایت، سیستم عامل ویندوز هفت (۳۲ بیتی)، سطح امنیتی ۱۶۰ بیت در میدان اول F_p ، گروه منحنی ۱۰۲۴ بیتی، AES به عنوان روش رمزنگاری / رمزگشایی متقارن و $SHA-1$ به عنوان تابع درهم ساز یک طرفه، اندازه گیری کرده اند؛ بنابراین، در این مقاله زمان اجرای عملیات رمزنگاری به صورت $T_m \approx 7.3529 \text{ ms}$ ، $T_s \approx 0.1303 \text{ ms}$ و $T_{fe} \approx 0.0004 \text{ ms}$ مطابق [۳۰]، [۱۴] و [۲۶]، به ترتیب، $T_{MAC} \approx T_m$ ، $T_A \approx T_m$ و $T_{fe} \approx T_m$ است؛ بنابراین، مطابق [۲۹]، $T_{MAC} \approx 0.0004 \text{ ms}$ ، $T_A \approx 7.3529 \text{ ms}$ و $T_{fe} \approx 7.3529 \text{ ms}$ است. همچنین، از هزینه محاسباتی زمان اجرای یک عملیات XOR و زمان اجرای

ذخیره سازی کارت هوشمند طرح پیشنهادی با طرح های پیشین، از جمله SARS [۱۱]، SAMA [۱۴]، NUA [۱۵]، AWC [۱۶]، MAKA [۱۷] و PAKA [۱۸]، می پردازد. در شکل (۱۰) کارایی طرح پیشنهادی، از نظر زمان اجرا، با طرح های مشابه مقایسه شده است. برای ارزیابی هزینه محاسباتی نهایی، زمان اجرای عملگرهای مورد نیاز که در بدنه طرح های احراز هویت مورد استفاده قرار می گیرند، به صورت زیر تعریف شده اند:

- T_m (زمان اجرای یک ضرب نردبانی)
- T_{SSL} (زمان ایجاد یک کانال امن با استفاده از پروتکل SSL)
- T_{fe} (زمان اجرای عملیات استخراج کننده فازی)
- T_A (زمان اجرای عملیات رمزنگاری / رمزگشایی نامتقارن)
- T_s (زمان اجرای یک عملیات رمزنگاری / رمزگشایی متقارن)
- T_h (زمان اجرای یک تابع درهم ساز یک طرفه)
- T_{MAC} (زمان اجرای یک کد اصالت سنجی پیام^{۵۹})

جدول ۴: مقایسه هزینه محاسباتی طرح پیشنهادی با طرح های پیشین

مقایسه کارایی	SARS [۱۱]	SAMA [۱۴]	NUA [۱۵]	AWC [۱۶]	MAKA [۱۷]	PAKA [۱۸]	طرح پیشنهادی
کاربر تلفن همراه (MU)	$T_h 4T_m + T_s + 2$	$T_h 10T_m + 2$	$T_h 9T_m + 2$	$T_h 6T_m + T_s + 2$	$T_h 7T_s +$	$T_h 12T_m + 2$	$T_h 8T_{fe} + 2T_s + 4$
اپراتور مرجع (HA)	$T_h 5$	$T_h 7$	$T_h 12T_A + T_{SSL} +$	$T_h 8T_s + 4T_m + 5$	$T_h 17T_s + 3T_m + 4$	$T_h 21T_m + T_s + 4$	$T_h 14T_s + 6T_m + 4$
اپراتور کمکی (FA)	$T_h 3T_m + T_s + 2$	$T_h 2T_m + 2$	$T_h 4T_m + T_A + T_{SSL} + 2$	$T_h 8T_m + T_s + 6$	$T_h 8T_s + 2T_m + 4$	$T_h 12T_m + T_s + 6$	$T_h 7T_s + 2T_m + 4$
مجموع	$T_h 12T_m + 4$	$T_h 19T_m + 4$	$T_h 25T_{SSL} + 2T_A + 2T_m + 4$	$T_h 22T_s + 6T_m + 13$	$T_h 32T_s + 6T_m + 8$	$T_h 45T_s + 2T_m + 12$	$T_h 29T_{fe} + 2T_s + 12T_m + 8$
زمان اجرا (میلی ثانیه)	۲۹/۶۷۷	۲۹/۴۱۹۲	۴۴/۶۵۰۲	۹۶/۳۷۸۳	۵۹/۶۱۷۸	۸۸/۵۱۳۴	۷۵/۱۰۴۲
تعداد پیام ها	۵	۵	۵	۵	۴	۴	۱۲
هزینه ارتباطی (بیت)	۲۳۶۸	۳۶۸۰	۵۸۸۸	۵۳۷۶	۵۴۷۲	۷۳۶۰	۷۳۶۰
حافظه SC_{MU} (بیت)	-	۹۶۰	۶۴۰	۹۲۸	۴۸۰	۸۰۰	۷۲۰

گرفته شود؛ بنابراین، هزینه ایجاد یک کانال امن با استفاده از پروتکل SSL ، تقریباً برابر با مجموع دو هزینه رمزنگاری متقارن و دو هزینه کد اصالت سنجی پیام است. در نتیجه، زمان مورد نیاز جهت ایجاد یک کانال امن با استفاده از پروتکل SSL ، تقریباً برابر با $T_{SSL} \approx 2T_s + 2T_{MAC} \approx 0.2614 \text{ ms}$ است.

در این مقاله فرض می‌شود که طول برجسته زمانی ۳۲ بیت، خروجی تابع درهم ساز ($SHA-1$)، طول پارامتر در نظر گرفته شده برای شناسه کاربر و عدد / Nonce تصادفی ۱۶۰ بیت، خروجی رمزنگاری / رمزگشایی متقارن ($AES-128$) ۱۲۸ بیت، نقطه منحنی بیضوی ۳۲۰ بیت و خروجی رمزنگاری / رمزگشایی نامتقارن (RSA) ۱۰۲۴ بیت

یک عملیات الحاق صرف نظر می‌شود. همچنین، فرض شده است که پروتکل مورد استفاده برای ایجاد کانال امن، SSL است. تأکید می‌شود که پروتکل SSL یک پروتکل رمزنگاری است و برای تأمین امنیت ارتباطات از طریق اینترنت، بنا شده است. پروتکل SSL سه نیازمندی امنیتی شامل محرمانگی، صحت داده و احراز هویت را فراهم می‌آورد که به ترتیب، با استفاده از رمزنگاری متقارن بر روی محتویات بسته های ارسالی در شبکه، کد اصالت سنجی پیام و استاندارد $x-509$ تضمین می‌شود. همچنین، در این پروتکل برای تبادل کلید متقارن از رمزنگاری نامتقارن استفاده می‌شود. برای ایجاد یک کانال امن توسط پروتکل SSL ، باید مجموع هزینه های هر سه نیازمندی امنیتی در نظر

۷ - نتیجه‌گیری

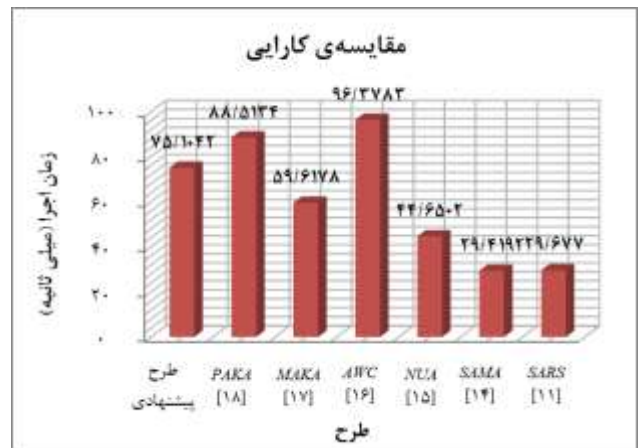
در این مقاله، یک طرح سبک‌وزن، گمنام و کارآمد، جهت احراز هویت متقابل مبتنی بر کارت هوشمند در شبکه‌های سیار سراسری ارائه شده است. طرح پیشنهادی در کاربردهایی همانند خدمات رومینگ دستگاه‌های تلفن همراه و هر کاربرد دیگری که نیاز به یک اپراتور مرجع برای ثبت‌نام و احراز هویت و یک اپراتور کمکی برای دریافت سرویس، همراه با حفظ گمنامی کاربر دارد، می‌تواند مورد استفاده قرار گیرد. با مقایسه امنیت و کارایی طرح پیشنهادی با طرح‌های پیشین، مشخص می‌شود که طرح پیشنهادی، علاوه بر مقاومت در مقابل اکثر حملات شناخته‌شده فعال و غیرفعال موجود در شبکه، از جمله حمله تکرار، حمله داخلی، حمله جعل هویت کاربر/ اپراتور مرجع/ اپراتور کمکی، حمله حدس رمز عبور به صورت برون خط، حمله فرد میانی و حمله منع سرویس، دارای ویژگی‌های امنیتی مهمی، از جمله، حفظ گمنامی و عدم ردیابی کاربر، احراز هویت متقابل، محرمانگی کامل روبه‌جلو، محرمانگی کامل رو به عقب، محرمانگی کلید شناخته‌شده و امنیت کلید نشست نیز است؛ بنابراین، طرح پیشنهادی امنیت بالاتری را نسبت به طرح‌های پیشین فراهم می‌آورد. همچنین، همان‌طور که مشاهده می‌شود، طرح پیشنهادی زمان محاسباتی کمتری از طرح‌های AWC و PAKA دارد و دارای هزینه ارتباطی برابر با طرح PAKA است. علاوه بر این، طرح پیشنهادی سربار ذخیره‌سازی کمتری نسبت به طرح‌های SAMA، AWC و PAKA دارد. با توجه به اولویت امنیت نسبت به زمان محاسباتی در پروتکل‌های امنیتی و امنیت بالای طرح پیشنهادی، این طرح در مقایسه با روش‌های گذشته، دارای کارایی قابل قبولی نیز است.

مراجع

- [۱] محمد لاری، «تخصیص منابع جهت کمینه‌سازی تأخیر ارسال در سامانه‌های مخابراتی تغذیه‌شونده به صورت بی‌سیم»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۳، صفحه ۱۲۰۵-۱۲۱۲، پاییز ۱۳۹۶.
- [۲] سعید سیدطاهری، علی رضا عندلیب، «طراحی و انافتگرهای مبتنی بر بلورهای فوتونی با قابلیت تواناسازی مناسب برای سامانه‌های مخابرات نوری»، مجله مهندسی برق دانشگاه تبریز، جلد ۴۷، شماره ۲، صفحه ۵۶۳-۵۷۰، تابستان ۱۳۹۶.
- [3] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 1, pp. 231-235, June 2004.
- [4] C. C. Lee, M. S. Hwang and I. E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683-1687, Oct 2006.
- [5] C. C. Wu, W. B. Lee and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722-723, Oct 2008.

است. همچنین، با در نظر گرفتن سطح امنیتی یکسان با پارامترهای تصادفی در این مقاله، فرض می‌شود که τ_{MU} دارای ۱۶۰ بیت است و حد آستانه t مطابق [۳۰] برابر با 2^{-80} و نیازمند ۸۰ بیت است.

طرح پیشنهادی دارای ۱۰ پیام در الگوریتم احراز هویت و توافق کلید نشست است. در طرح پیشنهادی، پیام درخواست احراز هویت کاربر $\langle ID_{FA}, NID, M_{MU}, T_{MU_1} \rangle$ و پیام $\langle ID_{HA}, NID, M_{MU}, T_{MU_1} \rangle$ نیازمند ۱۰۲۴ بیت، پیام $\langle ID_{HA}, NID, E_1, E_2, E_3, M_{HA}, T_{HA_1} \rangle$ نیازمند ۸۹۶ بیت، پیام‌های $\langle M_{HA}, E_2, T_{HA_1} \rangle$ ، $\langle NID, E_4, T_{MU_2} \rangle$ و $\langle NID, E_5, T_{HA_2} \rangle$ نیازمند ۹۶۰ بیت، دو پیام $\langle NID, E_6, T_{MU_3} \rangle$ نیازمند ۶۴۰ بیت، پیام $\langle NID, E_4, ID_{HA}, T_{MU_2} \rangle$ نیازمند ۹۶۰ بیت و پیام $\langle NID, E_6, ID_{HA}, T_{MU_3} \rangle$ نیازمند ۹۶۰ بیت و برای برقراری کلید مشترک K_{HA-FA} (توسط روش $CL-ACA$ [۲۷])، نیازمند ۲۸۸۰ بیت است.



شکل ۱۰: نمودار مقایسه کارایی طرح پیشنهادی با طرح‌های گذشته

همان‌طور که در جدول (۴) و شکل (۱۰) مشاهده می‌شود، زمان اجرای طرح پیشنهادی از طرح‌های AWC [۱۶] و PAKA [۱۸] کمتر اما از طرح‌های SARS [۱۱]، SAMA [۱۴]، NUA [۱۵] و MAKA [۱۷] بیشتر است. همچنین، طرح پیشنهادی هزینه ارتباطی برابر با طرح PAKA [۱۸] و بالاتر از طرح‌های پیشین دارد. گرچه زمان اجرا و هزینه ارتباطات در پروتکل پیشنهادی از برخی از پروتکل‌های پیشین بالاتر است، اما این موضوع توجیه‌پذیر است؛ زیرا همان‌طور که در جدول (۳) نشان داده شد، طرح‌های SARS [۱۱]، SAMA [۱۴]، NUA [۱۵]، AWC [۱۶]، MAKA [۱۷] و PAKA [۱۸] برخی از ویژگی‌های امنیتی مهم را فراهم نمی‌آورند و در مقابل برخی از حملات فعال و غیرفعال موجود در شبکه آسیب‌پذیرند. علاوه بر این، طرح پیشنهادی دارای کارت هوشمند کم‌هزینه‌تر نسبت به طرح‌های SAMA [۱۴]، AWC [۱۶] و PAKA [۱۸] است، اما سربار ذخیره‌سازی بیشتر نسبت به طرح‌های NUA [۱۵] و MAKA [۱۷] دارد.

- Communication Networks*, vol. 9, no. 16, pp. 3527-3542, Nov 2016.
- [19] C. C. Lee, Y. M. Lai, C. T. Chen and S. D. Chen, "Advanced Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1281-1296, June 2017.
- [20] X. Li, J. Niu, S. Kumari, F. Wu and K. K. R. Choo, "A robust biometrics based three-factor authentication scheme for Global Mobility Networks in smart city," *Future Generation Computer Systems*, in press, doi.org/10.1016/j.future.2017.04.012, Apr 2017.
- [21] R. Madhusudhan and K. S. Suvidha, "An Efficient and Secure User Authentication Scheme with Anonymity in Global Mobility Networks," *International Conference on Advanced Information Networking and Applications Workshops*, pp. 19-24, May 2017.
- [22] M. Nikooghadam, E. Malekian and A. Zakerolhosseini, "A Versatile Reconfigurable Bit-Serial Multiplier Architecture in Finite Fields GF(2^m)," *Communications in Computer and Information Science*, vol. 6, no. 1, pp. 227-234, Feb 2008.
- [23] M. Nikooghadam, A. Zakerolhosseini and M. Ebrahimi Moghaddam, "Efficient utilization of elliptic curve cryptosystem for hierarchical access control," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1917-1929, Oct 2010.
- [24] D. Mishra, A. Chaturvedi, S. Mukhopadhyay, "Design of a lightweight two-factor authentication scheme with smart card revocation," *journal of information security and applications*, vol. 23, pp. 44-53, Aug 2015.
- [25] Y. Dodis, R. Ostrovsky, L. Reyzin and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *Society for Industrial and Applied Mathematics journal on computing*, vol. 38, no. 1, pp. 97-139, Sep 2008.
- [26] D. He, N. Kumar, J. H. Lee and R. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 30-37, Feb 2014.
- [27] H. Sun, Q. Wen and W. Li, "A strongly secure pairing-free certificateless authenticated key agreement protocol under the CDH assumption," *Science China Information Sciences*, vol. 59, no. 3, pp. 1-16, March 2016.
- [28] T. Team, "AVISPA v1. 1 User manual," *Information Society Technologies Programme*, <http://avispa-project.org>, March 2006.
- [29] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of medical systems*, vol. 39, no. 2, pp. 1-9, Jan 2015.
- [30] A. G. Reddy, E. J. Yoon, A. K. Das and K. Y. Yoo, "Lightweight authentication with key-agreement protocol for mobile network environment using smart cards," *The Institution of Engineering and Technology Information Security*, vol. 10, no. 5, pp. 272-282, Feb 2016.
- [6] C. C. Chang, C. Y. Lee and W. B. Lee, "Cryptanalysis and improvement of a secure authentication scheme with anonymity for wireless communications," *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 902-904, Nov 2009.
- [7] J. S. Lee, J. H. Chang and D. H. Lee, "Security flaw of authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 13, no. 5, pp. 292-29, May 2009.
- [8] J. Xu and D. Feng, "Security flaws in authentication protocols with anonymity for wireless environments," *Electronics and Telecommunications Research Institute journal*, vol. 31, no. 4, pp. 460-462, Aug 2009.
- [9] P. Zeng, Z. Cao, K. K. Choo and S. Wang, "On the anonymity of some authentication schemes for wireless communications," *IEEE Communications Letters*, vol. 13, no. 3, pp. 170-171, March 2009.
- [10] D. He, M. Ma, Y. Zhang, C. Chen and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Communications*, vol. 34, no. 3, pp. 367-374, March 2011.
- [11] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1, pp. 214-222, Apr 2012.
- [12] JS. Kim and J. Kwak, "Secure and efficient anonymous authentication scheme in global mobility networks," *Journal of Applied Mathematics*, vol. 2013, pp. 1-12, Sep 2013.
- [13] D. Zhao, H. Peng, L. Li and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 247-269, Apr 2014.
- [14] W. C. Kuo, H. J. Wei and J. C. Cheng, "An efficient and secure anonymous mobility network authentication scheme," *journal of information security and applications*, vol. 19, no. 1, pp. 18-24, Feb 2014.
- [15] H. D. Le, C. C. Chang and Y. C. Chou, "A Novel Untraceable Authentication Scheme for Mobile Roaming in GLOMONET," *International Journal of Network Security*, vol. 17, no. 4, pp. 395-404, July 2015.
- [16] Y. Lu, X. Wu and X. Yang, "A Secure Anonymous Authentication Scheme for Wireless Communications Using Smart Cards," *International Journal of Network Security*, vol. 17, no. 3, pp. 237-245, May 2015.
- [17] P. Gope and T. Hwang, "An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks," *Journal of Network and Computer Applications*, vol. 62, pp. 1-8, Feb 2016.
- [18] F. Wu, L. Xu, S. Kumari, X. Li, A. k. Das, M. K. Khan, M. Karuppiah and R. Baliyan, "A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks," *Security and*

⁴ Privacy⁵ Users's Anonymity⁶ Authentication⁷ Mobility network⁸ Home Agent⁹ Foreign Agent¹⁰ Session key backward secrecy¹¹ Mutual authentication¹² Forgery attack¹ wireless² The mobile user³ Security

-
- ¹³ Off-line guessing attack
 - ¹⁴ Perfect Forward Secrecy
 - ¹⁵ Replay attack
 - ¹⁶ Roaming service
 - ¹⁷ Secure Authentication for Roaming Service
 - ¹⁸ Man-in-the middle attack
 - ¹⁹ Mobile User/Foreign Agent/Home Agent impersonation attack
 - ²⁰ Insider attack
 - ²¹ Smart card
 - ²² Secure Anonymous Mobility Authentication
 - ²³ Update
 - ²⁴ Verifier attack
 - ²⁵ Denial of service attack
 - ²⁶ Untraceability
 - ²⁷ Novel Untraceable Authentication
 - ²⁸ Local authentication
 - ²⁹ User-friendly password change phase
 - ³⁰ Authentication for Wireless Communications
 - ³¹ Mutual Authentication and Key Agreement
 - ³² Session key security
 - ³³ Provably Authentication and Key Agreement
 - ³⁴ Stolen smart card attack
 - ³⁵ Elliptic curve cryptography
 - ³⁶ Fuzzy extractor
 - ³⁷ Public key
 - ³⁸ Elliptic curve discrete logarithm problem
 - ³⁹ Galois field
 - ⁴⁰ Abelian group
 - ⁴¹ Secret key
 - ⁴² Private key
 - ⁴³ Rivest Shamir Adleman
 - ⁴⁴ Biometric
 - ⁴⁵ Bio-hash function
 - ⁴⁶ Secure channel
 - ⁴⁷ Database
 - ⁴⁸ Card reader
 - ⁴⁹ Pseudo-identity
 - ⁵⁰ Certificate-less Authenticated Key Agreement
 - ⁵¹ Known-key secrecy
 - ⁵² Long-term
 - ⁵³ Informal
 - ⁵⁴ Formal
 - ⁵⁵ High Level Protocol Specification Language
 - ⁵⁶ Specification
 - ⁵⁷ Declaration
 - ⁵⁸ On-the-fly Model-Checker
 - ⁵⁹ Message authentication code