پروتکل مدیریت کلید برای شبکه های ماهواره ای با استفاده از مشبکه ها

کیان کیقباد[*1]، رضا نخجوان شهرکی[2] سید داود منصوری[3]

۱- استادیار مجتمع دانشگاهی فناوری اطلاعات ، ارتباطات و امنیت دانشگاه صنعتی مالک اشتر

۲-دانشجوی کارشناسی ارشد مجتمع دانشگاهی فناوری اطلاعات ، ارتباطات و امنیت دانشگاه صنعتی مالک اشتر

3- محقق پژوهشکده امنیت مجتمع دانشگاهی فناوری اطلاعات ، ارتباطات و امنیت دانشگاه صنعتی مالک اشتر

چکیده

ارتباطات و رد و بدلِ اطلاعات مهم بین ماهواره و کاربران زمینی، باید از طریقِ کانال امنی صورت بگیرد. تا کنون برقراری کانالِ امن، با استفاده از ابزارهای رمزنگاری سنتی انجام می‌گرفت که امنیتشان بر مبنایِ سختیِ مسائلی مثلِ فاکتورگیری اعداد یا حل لگاریتم گسسته است. اما زمانی که کامپیوتر های کوانتومی با توان محاسباتی زیاد، ظهور کنند، امنیتی که بر مبنای مسائلِ سختی مثلِ فاکتورگیری اعداد یا حل لگاریتم گسسته بنا نهاده شده باشد، با تهدیدی جدّی مواجه خواهد شد و اکثر طرح ها از جمله RSA خواهند شکست. لذا باید به دنبالِ جایگزینی باشیم که بتوانیم امنیت را بر آن بنا نهیم. مشبکه ها به عنوانِ یکی از این جایگزین ها مورد تحقیق و پژوهش قرار گرفته اند. مشبکه ها، خود دارای مسائلِ گوناگونی هستند که با بررسی های انجام شده، این نتیجه بدست آمد که مسئله ی Ring-LWE[†] علاوه بر سختی که دارد، برای استفاده در شبکه های ماهواره ای موثرتر است، زیرا هم به حافظه ی کمتری نیاز دارد و هم طول کلید های کوچکتری دارد. از آنجا که جست و جو برای یافتنِ پروتکلی که بر مبنای مشبکه ها، برای استفاده در شبکه های ماهواره ای طراحی شده باشد، به نتیجه نرسید، لذا پس از پژوهش در این زمینه به این نتیجه رسیدیم که یکی از بهترین پروتکل های مبادله ی کلید که بر مبنای سختیِ Ring-LWE است، پروتکل مبادله ی کلید پایکرت می باشد. هرچند این پروتکل برای استفاده در محیط ماهواره ای طراحی نشده است، اما در این مقاله سعی شده است برای استفاده در چنین فضایی بهبود داده شود.

کلمات کلیدی: احراز اصالت، مدیریت کلید، شبکه های ماهواره ای، مشبکه ها،Ring-LWE

۱. مقدمه

[*] **Corresponding author**
Email: keyghobad.kiyan@gmail.com
[†] **Ring Learning With Error**

شبکه‌های ماهواره‌ای، صدا، ویدئو، عکس‌ها و دیتا را به‌وسیله‌ی سیگنال‌های الکترومغناطیسی از طریق هوا انتقال می‌دهند ازاین‌رو به قطع شدن و بهره برداری از دیتای منتقل شده در هوا حساس هستند. این شبکه‌ها، در برابر حملات اینترنتی که تهدیدی برای امنیت ملی و اقتصادی هستند نیز آسیب‌پذیرند. بنابراین برای مقابله با تهاجم سایبری گسترده آرایه‌ای از ابزارها و روش‌های امنیتی را به کار می‌گیرند تا اینکه از ادامه‌ی عملیات زیرساخت‌های حیاتی اطمینان پیدا کنند [1]. ارائه‌ی سرویس‌های امنیتی در سطح شبکه مستلزم وجود یک زیرساخت امنیتی بین اجزای شبکه است که به شکل مناسبی کلیدهای مشترکی را برای احراز اصالت و محرمانگی اطلاعات فراهم می‌نماید. به چارچوبی که در آن نیازمندی فوق برآورده می شود مدیریت کلید گفته می شود. برخی از روش‌های مدیریت کلید از امنیت بالایی برخوردارند اما به دلیل داشتن توان محاسباتی زیاد قابل استفاده در ماهواره نیستند. در سال های اخیر رمزنگاری مبتنی بر مشبکه ها توسط خواص خوبی نظیر تضمین امنیتی قابل اثبات قوی و مقاومت در برابر حملات کوانتومی، انعطاف پذیری برای داشتن ابزار قدرتمندی نظیر رمزنگاری تمام همریخت و کارایی تقریبی بالا، شناخته شده است. بعلاوه در چندین کار بیان شده است که برای وظایف پایه نظیر رمزنگاری و احراز اصالت، مشبکه ها می توانند عملکردی قابل مقایسه یا حتی بهتر از آن هایی داشته باشند که مبتنی بر روش های کلاسیک نظیر RSA یا دیفی هلمن هستند. با این وجود کارهای کمی بر روی توسعه ی مبادله ی کلید مبتنی بر مشبکه ها برای به کارگیری در سیستم های رمزنگاری و پروتکل ها شده است [2]. به این دلیل که بسیاری از رمزنگاری های کلید عمومی که استفاده می کنیم، نظیرِ RSA و ECC و DSA، مبتنی بر سختی مسائلی مانند فاکتورگیری اعداد و لگاریتم گسسته هستند، با ورود ماشینی که بتواند فاکتورگیری کوانتومی را پیاده سازی کند، امنیت ارتباطات به خطر خواهد افتاد [3]. بسیاری از الگوریتم های رمزنگاری مبتنی بر مشبکه ها مبتنی بر مسئله ی LWE* هستند که نوعی از مسائل مشبکه ای است که مانند چندین مسئله ی مشبکه ای بدترین حالت، برای حل کردن سخت است [2]. پروتکلی که در این مقاله قصد بهبودِ آن جهتِ استفاده برای شبکه های ماهواره ای تاکتیکی را داریم از محاسبات بسیار پیچیده از قبیل سیستم رمزنگاری کلید عمومی و سیستم رمزنگاری کلید خصوصی استفاده نمی کند و در بهبودِ آن نیز از چنین محاسباتی استفاده نشده است.

2. پیش نیاز ها

مسئله ی Ring-LWE:

یکی از ساختار هایی که در مشبکه های ایده آل می توان تعریف کرد، مسئله ی Ring-LWE است. فرض کنید $n \in N$ توانی از ۲ باشد، R_q حلقه ی $Z_q[x]/(x^n + 1)$ برای یک عدد صحیح q باشد و χ_σ توزیع خطای متناظر (n بعدی) روی R_q باشد. با داشتن $s,a \in R_q$ و $e \leftarrow \chi_\sigma$ ، عبارت A_{s,χ_σ} را به عنوان توزیعِ جفتِ

$$(a , as + e) \in R_q * R_q \qquad (1)$$

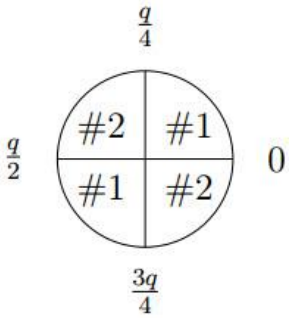
تعریف می کنیم [2]. a به صورت تصادفی یکنواخت انتخاب می شود [4]. هر المانی در R_q توسط یک چند جمله ای حداکثر از درجه ی n-1 نمایش داده می شود، که همچنین می تواند به عنوان یک بردار با درایه هایی که ضرایبِ چند

* Learning With Error

جمله ای هســـتند دیده شـــوند [5]. فرضِ $RLWE_{q,\sigma}$ بیان می دارد که برای هر الگوریتم زمان چند جمله ای با فقط تعدادی نمونه، تمایز قائل شدن بین A_{s,χ_σ} و توزیع یکنواخت بر روی $R_q * R_q$ مشکل است.

روش اصلاح سازی BCNS:

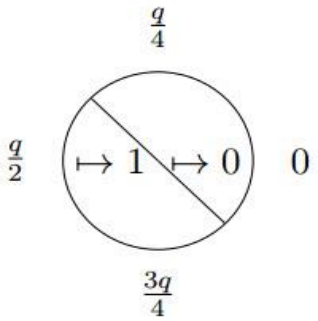
در سال ۲۰۱۴ پایکرت در [6] یک طرح انتقال کلید که ایده ی پایه ی یک سانی با پروتکل دینگ دا شت را ارائه کرد که ایده ی ارسال یک بیت سیگنال برای هر ضریب هم در آن استفاده شد. در این رویکرد بهبود یافته باب برای آلیس یک اطلاعات اصلاح سازی ارسال می کند، به طوری که $0\rightarrow$ region#1 , $1\rightarrow$ region#2، این مناطق در شکل1 نشان داده شده اند. به عبارت دیگر، یک طرف به عنوان اطلاعات اصلاح سازی، یک شماره ی منطقه برای طرف مقابل ارسال می کند که ضریبش در آن منطقه قرار گرفته ا ست. پس از آن بستگی به این منطقه، یک روش ا ستخراج کلید ویژه اعمال می شود. اگر u , v ضرایب متناظر از آلیس و باب باشند، و همچنین $|u - v| < \frac{q}{8}$ باشد، بنابراین، این روش همیشه کار می کند. با توجه به طراحی هوشمندانه ی مناطقِ گرد کردن* و روش های ا ستخراج کلید، افشای مناطق، هیچ اطلاعاتی در بر ندارد و امنیت را به خطر نمی اندازد. توجه کنید که پروتکل BCNS یک پیاده سازی از روشِ ا صلاح سازی پایکرت ا ست و یک روش اصلاح سازی جدید معرفی نمی کند.



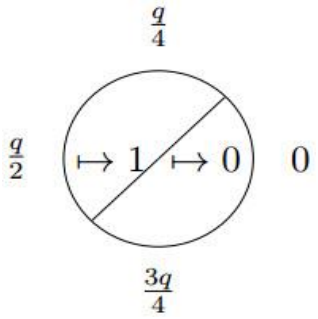
شکل ۱: چهار منطقه ی تابع گرد کردن

در دو شکل 2 و 3، روشِ rounding در پروتکل پایکرت بسته به مناطق نمایش داده شده است.

* Rounding



شکل ۲: وقتی که بیت اطلاعاتِ تابع گرد کردن شماره ی منطقه #1 باشد



شکل ۳: وقتی که بیت اطلاعاتِ تابع گرد کردن شماره ی منطقه #2 باشد

روش اصلاح سازی را طی مثالی توضیح می دهیم:

آلیس s و e کوچک در Z_q انتخاب می کند که هر دو از توزیع گوســـی نمونه برداری شـــده اند. همچنین باب s'و e'کوچک در Z_q را انتخاب می کند که هر دو از توزیع گو سی نمونه برداری شده اند. سپس مقادیری به صورتِ زیر برای هم ارسال می کنند:

Alice → Bob: b = as + e
Bob →Alice:b′= as′+ e′ , rounding region ∈ {0, 1}

آلیس راز مشترکِ $s * b' = s * (as' + e') = s * a * s' + s * e'$ را محاسبه می کند و باب نیز راز مشترکِ $s' * b = s' * (as + e) = s' * a * s + s' * e$ را محاســـبه می کند که این دو راز تقریبا با هم برابرند. در ادامه توصیف دقیق روشِ اصلاح سازی پایکرت می آید:

بازه های مجزای $I_0 = \{ 0 , 1 , \dots , \lceil \frac{q}{4} \rceil -1 \}$ و $I_1 = \{ \lceil \frac{3q}{4} \rceil , \dots , q-1 \}$ در mod q تعریف می کنیم.
حالا عملیات Cross-rounding را به صورت زیر تعریف می کنیم که $< . >_2 : Z_q \to Z_2$:

$$< v >_{q,2} = \begin{cases} 0 & v \in I_0 \cup (I_0 + \frac{q}{2}) \\ 1 & v \in I_1 \cup (I_1 + \frac{q}{2}) \end{cases} \qquad (2)$$

[2]. تابعی که آن را پیاده سازی می کند به صورت زیر تعریف می شود:

$$< v >_2 = \left\lfloor \frac{4}{q}\, v \right\rfloor \bmod 2 \qquad (3)$$

اگر v تصادفیِ یکنواخت باشد، پس $< v >_2$ نیز تصادفی یکنواخت است، اگر و فقط اگر $q/2$ زوج باشد، و در غیر اینصورت بایاسی به سمتِ صفر دارد[6] . تابعِ $\lceil . \rfloor_2 : Z_q \to Z_2$ را بصورت

$$\lceil v \rfloor_2 = \left\lceil \frac{2}{q} v \right\rfloor mod\ 2 \qquad (4)$$

در نظر بگیرید. هر دو تابع $< v >_2$ و $\lceil v \rfloor_2$، می توانند به صورت زیر به المان های R_q توسعه یابند [7]:

$$\lceil f \rfloor_{q,2} = (\lceil f_{n-1} \rfloor_{q,2}, \lceil f_{n-2} \rfloor_{q,2}, \dots, \lceil f_0 \rfloor_{q,2}) \qquad (5)$$

$$< f >_{q,2} = (< f_{n-1} >_{q,2}, < f_{n-2} >_{q,2}, \dots, < f_0 >_{q,2}) \qquad (6)$$

باب کلیدِ خود را توسط تابع $\lceil . \rfloor_2$ می سازد. اما آلیس با دریافتِ $< v >_2$ بازای هر ضریب و داشتن ضریبِ مربوط به این بیتِ ارسالی(u)، می تواند $\lceil v \rfloor_2$ را با استفاده از روش زیر بازیابی کند:

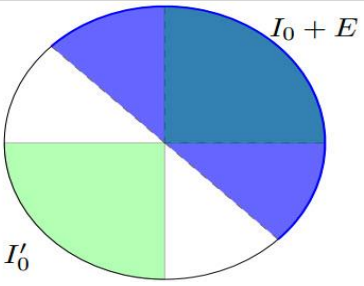
ابتدا به این نکته توجه کنید که ادعای 3.1 در [6] بیان می دارد که $< v >_2$ اطلاعاتی در مورد $\lceil v \rfloor_2$ فاش نمی کند [6]. این ادعا را در ادامه با ذکر مثالی توضیح خواهیم داد. برای محاسبه ی کلید مشترک تابع اصلاح سازیِ $rec : Z_q * Z_2 \to Z_2$ استفاده شده است [2]:

$$rec(v, b) = \begin{cases} 0 & \text{if } v \in I_b + \left(\left[-\frac{q}{8}, \frac{q}{8} \right) \cap Z \right) \bmod q \\ 1 & \text{otherwise} \end{cases} \qquad (7)$$

طبقِ ادعای 3.2 در [2]، اگر $E = \left[-\frac{q}{8}, \frac{q}{8} \right)$ ، برای q زوج و بازای $v \in Z_q$ و $e \in E$، اگر معادله ی $u = v + e \bmod q$ برقرار باشد، پس :

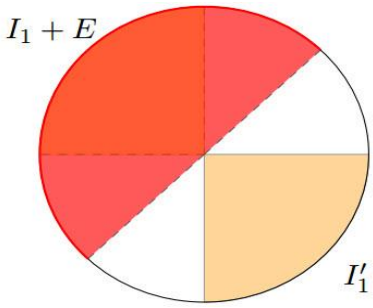
$$rec(u, < v >_2) = \lceil v \rfloor_2 \qquad (8)$$

اثباتِ آن به این صورت است که در نظر بگیرید، $b = < v >_2 \in \{0, 1\}$ ، به طوریکه $v \in I_b \cup (\frac{q}{2} + I_b)$ برقرار باشد. سپس $\lceil v \rfloor_2 = 0$ برقرار است اگر و فقط اگر $v \in I_b$ باشد، و این به نوبه ی خود یعنی $u \in I_b + E$، زیرا $((I_b + E) - E) \subseteq I_b + (-\frac{q}{4}, \frac{q}{4})$ و $(\frac{q}{2} + I_b)$ در modq مجزا هستند [6]. متغیر u، یک ضریب ازکلیدِ مشترکی است که در دستِ گیرنده است و $< v >_2$ اشاره ی دریافتی مبنی بر اینست که ضریبِ متناظر از کلیدِ مشترکی که دستِ فرستنده است در کدام ربع از دایره واقع شده است. فرض کنید $b = < v >_2 = 0$ باشد. پس $v \in I_b \cup I'_b$ و منطقه ی $I_0 + E$ به صورت تصویری ، منطقه ی هاشور زده شده در شکلِ زیر است:



شکل ۴:منطقه ی I_0 + E

حالا فرض کنید یک محدوده ی تضمین شده برای اینکه u و v چقدر می توانند از هم دور با شند دا شته با شیم که برابر است با $|u - v| < \frac{q}{8}$. سپس اگر $u \in I_0 + E$ باشد، v باید درونِ I_0 بوده باشد (($v \in (I_b \cup I'_b)$ را به یاد آورید) و یا اینکه $\frac{q}{8}$ دورتر از u بوده است. پس طبقِ تعریف داریم $\lceil v \rfloor_2 = 0$ و $rec(u, 0) = 0$.در غیر اینصورت اگر $u \notin I_0 + E$ باشدv باید در I'_0 بوده باشد، زیرا آن ها (I_0, I'_0)تنها مناطقِ ممکن برای مقدار v در فاصله ی $\frac{q}{8}$ از u هستند که در $I_0 + E$ نیستند و این به معنی اینست که $\lceil v \rfloor_2 = 1$. به طور مشابه فرض کنید که $b = \langle v \rangle_2 = 0$. پس $v \in (I_1 \cup I'_1)$ و منطقه ی $I_1 + E$ به صورت تصویری به شکلِ زیر است:



شکل ۵: منطقه ی I_1 + E

اگر $u \in I_1 + E$ پس برای اینکه $|u - v| < \frac{q}{8}$، v باید در I_1 بوده باشد، بنابراین $\lceil v \rfloor_2 = 0$ است و اگر $u \notin I_0 + E$ ، پس v باید در I'_1 بوده باشد، و بنابراین $\lceil v \rfloor_2 = 1$ است [4].

روش کپسوله کردنِ کلید:

روشِ KEM* در پروتکلِ پایکرت در ادامه معرفی می شود:

این KEM مبتنی بر Ring-LWE ا ست و رو شی کارآمد ا ست که در مقابل حملات غیر فعال (مثلِ شنود) امن ا ست. پایکرت این KEM را به عنوان قسمتی از پروتکل مبادله ی کلید احراز اصالت شده ی خود استفاده می کند. در روشِ کپسوله کردنِ کلید، کلید کپسوله شده به طور صریح توسط هیچ کدام از طرفین انتخاب نشده است. با استفاده از KEM نیازی نیست که فرستنده و گیرنده بر روی یک مقدار تقریبی توافق کنند و با استفاده از تابع اصلاح سازی کلید کوتاه مدت

* Key Encapsulation Mechanism

را از آن استخراج کنند. الگوریتم های KEM به صورت زیر هستند:

KEM.Setup ( ):

$a \leftarrow R_q$ را انتخاب می کند و pp = a را به عنوان خروجی بر می‌گرداند.

KEM.Gen (pp=a):

$s_0, s_1 \leftarrow \chi$ را انتخاب می کند، عبارتِ $b = a.\ s_1 + s_0 \in R_q$ را تشکیل می دهد، و در خروجیِ خود کلید عمومی pk = b وکلید خصوصی $sk = s_1$ را می دهد.

KEM.Encaps (pp = a, pk = b):

$e_0, e_1, e_2 \leftarrow \chi$ را به طور مستقل انتخاب کند. $u = e_0\ .\ a + e_1 \in R_q$ و $v = e_0.\ b + e_2 \in R_q$ را در نظر بگیرید. $\bar{v} \leftarrow dbl\ (v)$ و خروجی کپسوله سازیِ $c = (u, v' =< \bar{v} >_2) \in R_q * R_2$ و کلیدِ $\mu = \lceil \bar{v} \rfloor_2 \in R_2$ را در نظر بگیرید.

KEM.Decaps ($sk=s_1$, c = (u, v')):

عبارتِ $\omega = u.\ s_1 \in R_q$ را محاسبه می کند، و $\mu = rec\ (\omega\ ,\ v') \in R_2$ را در خروجی می دهد [6].

4. پروتکل پایکرت

همانطور که در [8] بیان شده است، پروتکل مبادله ی کلید احراز اصالت شده ی مبتنی بر مشبکه ی پایکرت، از بهترین پروتکل های مبادله ی کلید است. بنابراین، ما نیز پس از معرفی این پروتکل، بهبودی از آن را برای استفاده در شبکه های ماهواره ای دو به دو ارائه می کنیم. این پروتکل شامل یک طرح امضای دیجیتال SIG = (Sig.Gen, Sign, Ver) ، یک روش کپسوله کردنِ کلید KEM = (KeyGen, EnCap, DeCap) با فضای کلیدِ K ، یک تابعِ شبه تصادفی $F: K * \{0, 1\} \rightarrow K'$ و یک کد احراز اصالت پیام MAC* با پارامترهای Π = (Gen, O, Vrfy) با فضای کلید K' و فضای پیامِ $\{0,1\}^*$، می باشد. اجرای موفق این پروتکل یک کلید مخفی در K' را در نتیجه دارد [6]. فرض بر این ا ست که هر طرف یک کلید ام ضای طولانی مدت برای SIG دارد که کلید برر سی ام ضای متناظر آن، ثبت شده و به هویتِ فرد (ID) وابسته ا ست و برای همه ی طرف ها قابل د سترس ا ست. این امر، در یک روش استاندارد با استفاده از یک مرجع صدور گواهی†، امکان پذیر است. همچنین فرض می کنیم که پارامتر های عمومی مورد اعتماد (pp) برای KEM تو سط یک شخص مورد اعتماد‡با ا ستفاده از KEM.Setup تولید شده و برای همه در د سترس ا ست. اگر چنین شخ صی در د سترس نبود می توان KEM.Setup را در KEM.Gen جا کرد [8] و [6]. مراحلِ پروتکل به صورت زیر می باشد:

* Message Authentication Code
† Certificate Authority
‡ Trusted Party

۱) مقداردهی اولیه*

پروتکل توسط آغازگر (ID_I) با استفاده از یک شناساگر جلسه sid شروع می شود که این sid باید از همه ی آن هایی که در جلسات قبل استفاده شده بود، مجزا باشد. سپس آغازگر یک جفت کلید خصوصی-عمومیِ تازه تولید می کند $(S_s,P_s) \leftarrow Key.Gen\ (pp)$ و P_s را برای گیرنده ارسال می کند و(S_s,P_s) را به عنوان حالتِ جلسه ذخیره می کند [8] و [6].

$(I \rightarrow R): (sid,P_s)$

۲) پیامِ پاسخ

وقتی طرفِ گیرنده (ID_R) پیامِ (sid,P_s) را دریافت می کند، اگر sid هیچوقت در ID_R استفاده نشده باشد، جلسه را فعال می کند. مقدار $(c,k) \leftarrow EnCap(pp,P_s)$ را محاسبه می کند مقادیرِ $k_0 = F_k(0)$ و $k_1 = F_k(1)$ را بدست می آورد و مقادیرِ P_s و k را از حافظه اش پاک می کند. (k_0, k_1) را به عنوان حالتِ جلسه ذخیره می کند و k_0 را به عنوان کلید جلسه نزدِ خود دارد. او یک MAC به صورت $t_R \leftarrow O\ (k_1, (1, sid, ID_R))$ با استفاده از کلید k_1 برای پیامِ $(1, sid, ID_R)$ تولید می کند. همچنین یک امضا به صورتِ $\sigma_R = Sign\ (sk_R\ , (1, sid,P_s, c))$ با استفاده از کلید امضای طولانی مدتِ خود sk_R برای پیامِ $(1, sid,P_s, c)$ تولید می کند. او مقدارِ t_R و σ_R و c را برای آغازگر ارسال می کند.

$(R \rightarrow I): (sid, c,ID_R,\sigma_R,t_R)$

۳) پیامِ اتمام

هنگامی که آغازگر اولین پیام به فرمِ $(sid, c,ID_R,\sigma_R,t_R)$ دریافت کرد، با داشتنِ sid، به حالتِ (S_s,P_s) مرتبط با sid نگاه می کند و مقدار $k = Decap(c,S_s)$ را محاسبه می‌کند و مقادیرِ $k_0 = F_k(0)$ و $k_1 = F_k(1)$ را بدست می آورد. آغازگر، امضای σ_R از پیامِ $(1, sid,P_s, c)$ را با استفاده از کلید طولانی مدت بررسی امضای گیرنده (vk_R) ، با محاسبه ی $Ver\ (vk_R, (1, id,P_s, c), \sigma_R)$ بررسی می کند. او همچنین کد احراز اصالت پیام، از پیامِ $(1, sid, ID_R)$ را با استفاده از کلید k_1 با محاسبه ی $Vrfy\ (k_1, (1, sid, ID_R), t_R)$ بررسی می کند. اگر هر کدام از این بررسی ها شکست بخورد جلسه قطع و حالتِ جلسه، پاک می شود و خروجی جلسه $(abort, ID_I,sid)$ خواهد بود. اگر هر دو بررسی موفقیت آمیز باشد، او جلسه را به این صورت تکمیل می کند: کلید k_0 را به عنوان کلید جلسه می پذیرد، یک کدِ احراز اصالت پیام با استفاده از کلید k_1 برای پیامِ $(0,sid,I_s)$ به صورتِ $t_s \leftarrow O\ (k_1, (0, sid,I_s))$ تولید می کند بعلاوه یک امضای $\sigma_s = Sign\ (sk_s\ , (0, sid,P_s, c))$ با استفاده از کلید طولانی مدتِ امضای خود برای پیامِ $(0, sid,P_s, c)$ تولید می‌کند. سپس پیامی به صورت زیر به گیرنده ارسال می کند:

$(I \rightarrow R): (sid,ID_I,\sigma_s,t_s)$

۴) اتمامِ گیرنده

هنگامی که گیرنده پیامی به فرمِ (sid,ID_I,σ_s,t_s) دریافت کرد با داشتنِ sid، به حالتِ جلسه ی(k_0, k_1) مرتبط با sid نگاه می کند. سپس او کلید بررسی امضای آغازگر را بازیابی می کند، و از آن کلید برای بررسی امضای σ_s از پیامِ $(0, sid,P_s, c)$ استفاده می کند و امضاء را با محاسبه ی $Ver\ (vk_s, (0, sid,P_s, c),\sigma_s)$ بررسی می کند. بعلاوه او کدِ احراز اصالت پیامِ t_s برای پیامِ $(0, sid,I_s)$ را با استفاده از کلید k_1 با محاسبه ی $Vrfy(k_1, (0, sid,I_s), t_s)$، بررسی می کند. اگر هر کدام از بررسی ها شکست بخورد، جلسه قطع و حالتِ جلسه پاک می شود، و خروجی $(abort,ID_R,sid)$

* initiation

می باشد، در غیر اینصورت کلید k_0را به عنوان کلید جلسه می پذیرد [8] و [6].

این پروتکل را می توان در شکلِ زیر مشاهده کرد:

Sender		Receiver
Long term signing and verification key (vk_S,sk_S)		Long term signing and verification key (vk_R,sk_R)
$(S_s,P_s) \leftarrow$ Key.Gen (pp)	$\xrightarrow{(sid,P_S)}$	$(c,k) \leftarrow EnCap(pp,P_s)$ $k_0 = F_k(0)$ $k_1 = F_k(1)$ $t_R \leftarrow O\ (k_1, (1, sid, ID_R))$ σ_R = Sign $(sk_R , (1, sid,P_s, c))$
	$\xleftarrow{(sid,c,ID_R,\sigma_R,t_R)}$	
$k = Decap(c,S_s)$ $k_0 = F_k(0)$ $k_1 = F_k(1)$ Ver $(vk_R, (1, sid,P_s, c), \sigma_R)$ $Vrfy(k_1, (1, sid, ID_R), t_R)$ $t_s \leftarrow O\ (k_1, (0, sid,I_s))$ $\sigma_s \leftarrow$Sign $(sk_s , (0, sid,P_s, c))$		
	$\xrightarrow{(sid,ID_I,\sigma_s,t_s)}$	
Session key $k_0 = F_k(0)$		Ver $(vk_s, (0, sid,P_s, c),\sigma_s)$ $Vrfy(k_1, (0, sid,I_s), t_s)$ Session key k_0

شکل ۶: پروتکل مبادله ی کلید احراز اصالت شده ی پایکرت

5. پروتکلِ بهبود یافته ی پیشنهادی

این پروتکل دارای دو مرحله ی ثبت نام و احراز اصالت است.

۱) مرحله ی ثبت نام: کاربر به NCC* مراجعه کرده هویت دائمِ خود را به NCC می دهد و یک مقدار تصادفی به نام r به همراه آدرس حافظه ای که r در آنجا ذخیره شده است را دریافت می کند. NCC دارای لیست افرد مجاز می باشد. NCC

* **Network Control Center**

و کاربر در پایان هر نشست مقدارِ تصادفی جدید و یکسانی می سازند (r_{new}) که NCC این مقدار را در آدرسی دیگر ذخیره کرده و آن آدرس را به صورتی که در ادامه بیان می شود به اطلاع کاربر می رساند. همچنین کلید عمومی هر نفر با استفاده از کلید خصوصی NCC امضاء شده و در اختیار آن ها قرار میگیرد. همه ی کاربران کلید عمومی امضاء شده ی NCC را در اختیار دارند.

۲) مرحله ی احراز اصالت:

محاسبات کاربر به صورتِ زیر در زمانِ آفلاین انجام می شود:

$$s' \text{ و } e' \leftarrow X \quad (9)$$

این عبارت یعنی اینکه کاربر مقادیر s'وe' را از توزیع X که گوسی است نمونه برداری می کند.

$$b' \leftarrow a\, s' + e' \quad (10)$$

b' توسط کلید خصوصی NCC امضاء شده و در اختیار کاربر قرار گرفته است.

$$e'' \leftarrow X \quad (11)$$
$$v \leftarrow s'\, b + e'' \quad (12)$$
$$c \leftarrow <v>_{q,2} \quad (13)$$
$$k_B \leftarrow \lceil v \rfloor_{q,2} \quad (14)$$
$$k_B{}' \leftarrow k_B \oplus \text{date} \quad (15)$$
$$k_{common} \leftarrow H\,(k_B{}'||r) \quad (16)$$

در این مرحله مقدارِ هشِ $k_B{}'||r$ به عنوان کلید جلسه محاسبه می شود.

$$g' \leftarrow MAC_{k_{common}}(k_B) \quad (17)$$

کاربر پس از محاسبات بالا $f = certificate_{user} \oplus r, c, g', date$ و آدرس حافظه را برای ماهواره ارسال می کند. سپس ماهواره پس از دریافتِ این اطلاعات آن ها را به همراه هویت خود (LEO_{ID}) برای NCC ارسال می کند. NCC پس از دریافتِ f و آدرس حافظه، از آن آدرس حافظه مقدارِ r را بر می دارد و آن را با f، XOR می کند. سپس به $certificate_{user}$ می رسد. این گواهی را با کلید برر سی ام ضای خود برر سی می کند اگر شخصِ مورد نظر جزء افرادِ مجازِ لیست بود، پروتکل را ادامه می دهد، در غیر اینصورت پروتکل را ادامه نمی دهد و پیامِ قطعِ جلسه را به کاربر مخابره می کند. در صورتِ مجاز بودنِ فرد، در انتهای پروتکل، کلید مشترکی را با کاربر محاسبه می کند. NCC محاسبات زیر را برای احراز اصالت کاربر و رسیدنِ به کلید مشترک با او انجام می دهد:

$$k_A \leftarrow rec\,(sb', c) \quad (18)$$
$$k_A{}' \leftarrow k_A \oplus \text{date} \quad (19)$$
$$k_{common} \leftarrow H\,(k_A{}'||r) \quad (20)$$
$$g'' \leftarrow MAC_{k_{common}}(date) \quad (21)$$

پس از انجام محاسبات، مقدار$g'' \oplus (new_add)$ را برای کاربر می فرستد که new_add همان آدرس جدید حافظه می باشد، تا کاربر نیز ماهواره را احراز اصالت کند. کاربر با محاسبه ی $MAC_{k_{common}}(date)$ و XOR کردنِ آن با مقداری که از ماهواره دریافت کرده است، به new_add می رسد، اگر این آدرس معتبر بود، هم NCC را احراز اصالت می کند و هم تایید کلید را دارد و علاوه بر این ها به مقدارِ new_add برای ارتباط بعدی می رسد. مقدار r_{new} برای کاربر به صورتِ $h\,(k_B)$ و برای NCC به صورتِ $h\,(k_A)$ بدست می آید. کاربر می تواند محاسبه ی r_{new} را به بعد از

جلسه موکول کند و از اینرو زمانِ محاسبه ی یک هش از زمان محاسبات آنلاین کم می شود.

6. صحت طرح

در این طرح فقط کافیست نشان دهیم که k_A برابر با k_B می باشد. یعنی اگر u و v ضرایبِ متناظر از کلید هایی باشد که کاربر و ماهواره در دست دارند، باید برای هر ضریب، $|u - v| < \frac{q}{8}$ باشد، که اگر این موضوع رعایت شود، بنابراین این روش همیشه کار می کند. اگر $E = \left[-\frac{q}{8}, \frac{q}{8}\right)$ باشد، بازای $v \in Z_q$ و $e \in E$، اگر معادله ی $u = v + e \mod q$ برقرار باشد، پس :

$$\mathrm{rec}(u, \langle v \rangle_2) = \lceil v \rfloor_2 \qquad (22)$$

بر قرار است، که این عبارت یعنی $k_B = k_A$.

اثبات:

به این صورت است که در نظر بگیرید، $b = \langle v \rangle_2 \in \{0, 1\}$، به طوریکه $v \in I_b \cup (\frac{q}{2} + I_b)$ برقرار باشد. سپس $\lceil v \rfloor_2 = 0$ برقرار ا ست اگر و فقط اگر $v \in I_b$ با شد، و این به نوبه ی خود یعنی $u \in I_b + E$، زیرا $((I_b + E) - E)$ $\subseteq I_b + (-\frac{q}{4}, \frac{q}{4})$ و $(\frac{q}{2} + I_b)$ در modq مجزا هستند [6]. متغیر u، یک ضریب ازکلیدِ مشترکی ا ست که در د ستِ گیرنده است و $\langle v \rangle_2$ اشاره ی دریافتی مبنی بر اینست که ضریبِ متناظر از کلیدِ مشترکی که در دستِ فرستنده است در کدام ربع از دایره واقع شده است. فرض کنید $b = \langle v \rangle_2 = 0$ باشد. پس $v \in I_b \cup I'_b$. حالا فرض کنید یک محدوده ی تضمین شده برای اینکه u و v چقدر می توانند از هم دور باشند داشته باشیم که برابر است با $|u - v| < \frac{q}{8}$. سپس اگر $u \in I_0 + E$ باشد، v باید درونِ I_0 بوده باشد (($v \in (I_b \cup I'_b)$ را به یاد آورید) و یا اینکه $\frac{q}{8}$ دورتر از u بوده است. پس طبقِ تعریف داریم $\lceil v \rfloor_2 = 0$ و $\mathrm{rec}(u, 0) = 0$.در غیر اینصورت اگر $u \notin I_0 + E$ باشدv باید در I'_0 بوده با شد، زیرا آن ها (I_0, I'_0)تنها مناطقِ ممکن برای مقدار v در فا صله ی $\frac{q}{8}$ از u ه ستند که در $I_0 + E$ نیستند و این به معنی اینست که $\lceil v \rfloor_2 = 1$. به طور مشابه فرض کنید که $b = \langle v \rangle_2 = 0$. پس $v \in (I_1 \cup I'_1)$. اگر $u \in I_1 + E$ پس برای اینکه $|u - v| < \frac{q}{8}$، v باید در I_1 بوده باشد، بنابراین $\lceil v \rfloor_2 = 0$ است و اگر $u \notin I_0 + E$ ، پس v باید در I'_1 بوده باشد، و بنابراین $\lceil v \rfloor_2 = 1$ است. از اینرو هنگامی که برای تمامِ ضرایبِ k_A و k_B همین اثبات را در نظر بگیریم، اثباتِ صحتِ پروتکل تکمیل می شود و $k_B = k_A$ خواهد شد.

7. مقایسه ی طرح پیشنهادی با طرح پایکرت از نظر کارایی ویژگی ها و حملات

به این دلیل که پروتکل پایکرت برای اســتفاده در شــبکه های ماهواره ای طراحی نشــده اســت، لذا ملزومات آن با پروتکلی که برای استفاده در چنین شبکه هایی طراحی شده است متفاوت است.

بررسی پروتکل از نظر حملات:

۱) حمله ی تکرار

به این دلیل که مقدار کلید با date ترکیب شده است، مقادیری که قبلا توسط مهاجم شنود شده اند نمی توانند در زمانِ دیگری مورد استفاده قرار بگیرند.

۲) حمله ی ممانعت از سرویس

به این دلیل که NCC مقدارِ قبلیِ r را تا احراز اصالت بعدی در حافظه ی خود نگه می دارد، بنابراین اگر مقدارِ آدرس حافظه ی جدید به کاربر نرسیده باشد، کاربر می تواند با استفاده از همان آدرس قبلی برای ارتباط استفاده کند که در این صورت NCC متوجه می شود که این آدرس قدیمی است و آدرسِ جدید را برای کاربر ارسال می کند.

۳) حمله ی جعل هویت

چون کلید عمومی هر کاربر توسط کلید خصوصی NCC امضاء شده است و مهاجم کلید خصوصی متناظر با کلید عمومی کاربر را در اختیار ندارد نمی تواند کاربر را جعل کند. همچنین چون مهاجم به کلید خصوصی متناظر با کلید عمومی NCC دسترسی ندارد بنابراین NCC را نیز نمی تواند جعل کند.

۴) حمله ی الحاق

با توجه به اینکه مهاجم کلید خصوصی NCC را برای تولید یک گواهی معتبر ندارد، بنابراین هر اطلاعاتی از خود که وارد جدول بررسی NCC کند، بی فایده خواهد بود، چراکه NCC مقداری را که از کاربر دریافت می کند ابتدا با استفاده از کلید خصوصی خود اعتبار آن را بررسی می کند و اگر معتبر نباشد جلسه قطع می شود.

۵) حمله ی تصدیق کننده ی مسروقه

اگر جدول بررسی NCC بدست مهاجم بیفتد، اطلاعاتی را که بدست می آورد، به این دلیل که کلید خصوصی کاربر را برای محاسبه ی کلید نشست ندارد نمی تواند خود را به جای کاربر جعل کند.

بررسی پروتکل از نظر ویژگی های مورد نیاز:

۱) گمنامی

به دلیل اینکه هویت دائمی کاربر به صورتِ متن معلوم بر روی کانال عمومی ارسال نمی شود و مقدار r نیز برای هر جلسه به روزرسانی می شود، لذا ویژگی گمنامی بر قرار است.

۲) امنیت پیشرو

به دلیل اینکه مقدار r برای هر جلسه هم در سمتِ کاربر و هم در سمتِ NCC به روزرسانی می شود، مهاجم در صورتِ بدست آوردنِ کلیدهای خصوصی و بلند مدت طرفین، باز هم قادر به دستیابی به کلید های جلسه ی بعدی نخواهد بود.

۳) احراز اصالت متقابل

NCC با دریافت مقادیر f و g' از کاربر، اولا مطمئن می شود که فقط آن کاربر خاص می تواند کلید را محاسبه کند و ثانیا پس از اینکه کلید را بدست آورد و با بررسی مقدار g' و برابر بودنِ مقادیر، کاربر به NCC احراز اصالت می شود. همچنین کاربر نیز با در دست داشتنِ کلید عمومی امضاء شده ی NCC مطمئن است که فقط NCCمی تواند این کلید مشترک را حساب کند و ثانیا با دریافت $g'' \oplus (new\_add)$ از NCC و بررسی معتبر بودنِ آدرس جدید، NCC نیز به کاربر احراز اصالت می‌شود.

۴) محرمانگی

از آنجا که کاربر و NCC برای مبادلات خود کلید نشستی را محاسبه می کنند و پیام های خود را با استفاده از آن رمزنگاری می کنند، لذا ویژگی محرمانگی در نشست نیز بدست آمده است.

۵) نشست مستقل

برای هر جلسه مقدارِ جدیدی از r محاسبه می شود که مهاجم حتی با دستیابی به کلید های نشستِ قبلی نمی تواند آن را محاسبه کند و این مقدار در محاسبه ی کلید های نشست به کار می رود. از اینرو ویژگی نشست های مستقل در این پروتکل بدست آمده است.

۶) تازگی

در محاسبه ی کلید هر جلسه، مقدارِ date دخالت دارد، لذا کاربر و NCC مطمئن هستند که کلیدی را که از آن استفاده می کنند به همین جلسه اختصاص دارد و قبلا استفاده نشده است.

جدول ۱: مقایسه ی ویژگی های مورد نیاز در پروتکل پایکرت و پروتکل پیشنهادی

طرح پایکرت	طرح پیشنهادی	ویژگی های مورد نظر
-	+	گمنامی
+	+	امنیت پیشرو
+	+	احراز اصالت متقابل
+	+	محرمانگی
+	+	نشستِ مستقل
+	+	تازگی

جدول ۲ : مقایسه ی کارایی در پروتکل پایکرت و پروتکل پیشنهادی

طرح پایکرت	طرح پیشنهادی	
آنلاین	آفلاین	encapse
آنلاین	آنلاین	decaps
۴	۳	تعداد محاسبه ی MAC و هش
۲	-	محاسبه ی امضاء
۲	۱	محاسبه ی بررسی امضاء
۴	-	محاسبه ی تابع شبه تصادفیF
۳	۲	تعداد پیام های مبادله شده بین طرفین
-	۳	محاسبه ی XOR

جدول ۳:مقایسه ی امنیتی پروتکل پیشنهادی

پروتکل پایکرت	پروتکل پیشنهادی	حملات
به دلیل اینکه ویژگی گمنامی ندارد، حمله ی ممانعت از سرویس در این پروتکل موضوعیت ندارد	+	حمله ی ممانعت از سرویس

+	+	حمله ی تکرار
+	+	حمله ی الحاق
+	+	حمله ی تصدیق کننده ی مسروقه
+	+	حمله ی جعل هویت
بسته به اینکه کدام المان های sid در کارت ذخیره می شـــوند و کدام به عنوانِ اطلاعات شـــخصـــی نزد کاربر می ماند، می توان این حمله را انجام داد و یا خیر	+	حمله ی دزدیدنِ کارت هوشمند

8. نتیجه گیری

همانطور که در ابتدا نیز بیان شـــد، پروتکل پایکرت از بهترین پروتکل های احراز اصـــالت و مبادله ی کلید با اســـتفاده از مشبکه ها ست. لذا در این مقاله سعی شد این پروتکل را برای استفاده در شبکه های ماهواره ای تاکتیکی مناسب سازی کنیم. با توجه به اینکه سه گذره بودنِ پروتکل پایکرت باعث تاخیر اضافی در ارتباطات می شود، ابتدا با استفاده از تدابیری سه گذره بودن پروتکل، به دو گذره تبدیل شد. سپس از آنجا که گمنامی کاربر نقش بسیار مهمی در شبکه های ماهواره ای تاکتیکی دارد و در پروتکل پایکرت این موضوع فراهم نبود، در این مقاله سعی شد در ارتباطات چاره ای برای گمنامی کاربر اندیشه شود که با دخالت دادن پارامتر r در محاسبات این امر فراهم شده است. مهمتر اینکه برای افزایش سرعتِ ارتباطات، نیمی از محاســباتی که در پروتکل پایکرت انجام می شــد، شــامل تابع encapse و نمونه برداری از توزیع گوســی و انجام امضــاها، به بخشِ‌آفلاین منتقل شــدند. این انتقال به بخشِ آفلاین باعث افزایش کارایی پروتکل برای ارتباط در شــبکه های ماهواره ای می شود که این موضوع به صورت مقایسه ای در جدول2 نشان داده شده است.

9. مراجع

[1] A. Adam and H. Zare , "Satellite Network Hacking & Security Analysis," *International Journal of Computer Science and Security (IJCSS)*, vol. 10, no. 1, 2016.

[2] S. Hesamian, "Analysis of BCNS and Newhope Key-exchange Protocols," University of Wisconsin-Milwaukee, 2017.

[3] O. Maire , O. Elizabeth , M. Gavin , S. Markku-Juhani , M. Ciara , K. Ayesha , H. James , d. P. Rafael , A. Michel , R. Francesco , V. Felipe , G. Tim , O. Tobias and W. Adrian , "Secure Architectures of Future Emerging Cryptography," in *Proceedings of the ACM International Conference on Computing Frontiers*, Italy, 2016.

[4] V. Singh, Interviewee, "A Practical Key Exchange for the Internet using Lattice Cryptography", 2015.

[5] J. Ding, X. Xie and X. Lin, "A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem," Rutgers University, 2012.

[6] C. Peikert, "Lattice Cryptography for the Internet," Atlanta, 2014.

[7] J. W. Bos, C. Costello, M. Naehrig and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *supported by Australian Research Council (ARC) Discovery Project DP130104304*, Australian, October 26, 2016.

[8] S. Rieß, "An Analysis of Lattice-Based Key Exchange Protocols", Darmstadt, 2016.