مجله علمی-پژوهشی «علوم و فناوری‌های پدافند غیرعامل»
سال دوم، شماره ۲، تابستان ۱۳۹۰: ص ۱۵۰- ۱۳۹

ارائه یک پروتکل کارآمد توافق کلید گروهی تصدیقی پویا مبتنی بر شناسه

مرتضی عریفی[۱*]، محمود گردشی[۲]

۱- کارشناس ارشد، ۲- مربی، مرکز تحقیقات فتح، دانشکده و پژوهشکده فناوری اطلاعات و ارتباطات، دانشگاه جامع امام حسین(ع)

(دریافت: ۱۳۸۹/۰۹/۱۳، پذیرش:۱۳۹۰/۰٤/۲۷)

چکیده:

پروتکل‌های تبادل کلید گروهی، نقش اساسی در برنامه‌های کاربردی گروهی از جمله سامانه‌های مخابراتی، رایانه‌ای و نظامی ایفا می‌کنند. بررسی این پروتکل‌ها با توجه به گستره وسیع استفاده، در ایجاد امنیت در ارتباطات بین‌المللی، تجاری و نظامی می‌تواند نقشی تعیین‌کننده در تأمین امنیت فضای ارتباطی کشور ایفا کند. اکثر پروتکل‌های تبادل کلید گروهی موجود، با تمرکز بر بسط پروتکل تبادل کلید دو عضوی مشهور دیفی- هلمن، سعی بر ایجاد پروتکل‌های تبادل کلید چند عضوی یا گروهی نموده‌اند. امنیت پروتکل‌های موجود مبتنی بر این فرض است که سختی حل مسئله دیفی- هلمن گروهی می‌تواند به سختی حل مسئله دیفی- هلمن دو عضوی کاهش یابد. برخلاف این فرض، تبادل کلید گروهی یک توسیع ساده از پروتکل‌های تبادل کلید دو عضوی نمی‌باشد. در حقیقت تفاوت‌ها و چالش‌های بسیاری برای طراحی و پیاده‌سازی پروتکل‌های تبادل کلید گروهی وجود دارند که باید بررسی شوند. این تفاوت ناشی از تفاوت در ماهیت ارتباطات نقطه به نقطه و ارتباطات گروهی است. در ارتباطات گروهی تنها به دست آوردن یک کلید نشست مشترک مد نظر نمی‌باشد، بلکه توجه به تغییرات گروه نیز دارای اهمیت است. در این مقاله، یک پروتکل توافق کلید گروهی پویا مبتنی بر شناسه با استفاده از زوج‌سازی ویل مطرح می‌شود. در پروتکل ارائه شده، از ساختار درختی سه‌تایی کامل استفاده می‌شود و همچنین تغییرات پویای گروه نیز در نظر گرفته شده است. در پایان نشان داده می‌شود که پروتکل ارائه شده ویژگی‌های امنیتی شناخته شده را برآورده می‌کند و نسبت به پروتکل‌های با ساختار مشابه کارآمدتر است.

کلیدواژه‌ها: توافق کلید گروهی، احراز اصالت مبتنی بر شناسه، زوج‌سازی، ساختار درخت

Efficient Authenticated ID-Based Group Key Agreement Protocol

M. Arifi[1*], M. Gardeshi[2]

Fath Research center, Faculty and Research Center of Communication and Information Technology,
Imam Hossein University

(Received: 12/04/2010, Accepted: 07/18/2011)

Abstract

Group key exchange protocols play main role in group applications such as computer networks, telecommunication and military systems. Given the wide range of usage of group applications, research and design of these protocols can play an important role in securing international exchanges, trade and military issues. Most existing group key exchange protocols establish multi-party or group key exchange protocols based on extending the two party well known Diffie-Hellman key exchanges. The security of these protocols is based on this assumption that difficulty of solving group Diffie-Hellman problem can be reduced to the difficulty of two party Diffie-Hellman problems. In spite of this assumption, Group key exchanges are not simply extension of two party protocols. In fact there are many differences and challenges that must be considered in design and implementation of these protocols. This difference is due to essential difference between peer to peer communication and group communication. In group communication in addition to considering the main goal e.g. obtaining group session key, consideration of group member changing is important too. In this paper, a Dynamic ID-based Group Key Agreement protocol based on Weil pairing is proposed. This complete ternary tree structure is used and dynamic changes in group are considered too. Finally we show that the proposed protocol is more efficient than other protocols with similar structure and also satisfies all known security requirements.

Keywords: Group Key Agreement, ID-Based Authentication, Bilinear Pairing, Tree Structure

* Corresponding author E-mail: Morteza.arifi@gmail.com

۱. مقدمه

با گسترش روزافزون استفاده از اینترنت، محبوبیت حوزه کاربردهای مبتنی بر ارتباطات گروهی به‌طور قابل توجهی افزایش یافته است. کنفرانس‌های دیجیتالی زنده با چندین عضو، بازی‌های ویدیوئی بر خط، سامانه‌های مشاوره و تشخیص بیماری از راه دور برای کاربردهای پزشکی، مذاکرات برای بستن قرارداد بین دولت‌ها و کاربردهای نظامی و ... از جمله برنامه‌های کاربردی گروهی می‌باشند. در ارتباطات گروهی یا فرستادن اطلاعات به‌طور هم‌زمان به چند شبکه، به‌جای آنکه n انتقال برای n عضو، مانند ارتباطات نقطه به نقطه صورت گیرد، تنها یک ارسال برای دست‌یابی تمام اعضا مورد نیاز است. استانداردهای موجود برای ارسال اطلاعات به چند شبکه به طور هم‌زمان نیازمند سازوکارهایی برای محافظت در برابر شنود، جعل هویت، سرقت نشست و غیره هستند. بنابراین استفاده از ارسال‌های چند پخشی[1] بدون در نظر گرفتن ویژگی‌های امنیتی این تهدید را ایجاد می‌کند که داده‌ها دست‌کاری و یا سرویس یا محتوا دزدیده شود.

یک پیشنهاد برای امن کردن تمام پیام‌های گروهی، استفاده از رمزنگاری کلید عمومی است. ولی الگوریتم‌های رمزنگاری کلید عمومی از نظر محاسباتی ۱۰۰۰ برابر کندتر از الگوریتم‌های رمزنگاری کلید خصوصی یا متقارن هستند. علاوه بر آن، برای امن کردن ارتباطات بین یک گروه با n کاربر با استفاده از این تکنیک، هر عضو باید n-1 کلید عمومی تایید شده را ذخیره کند. به‌علاوه اگر هر کاربر بخواهد یک پیام امن را به هر یک از کاربران گروه ارسال کند، باید پیام‌های زیادی را (به تعداد دریافت کنندگان) رمز و ارسال کند، این نتایج منجر به هزینه زیاد محاسبه‌ها و مخابرات گسترده می‌شود.

یک روش دیگر، استفاده از رمزنگاری کلید متقارن توسط یکی از دو روش زیر است :

۱) کلیدی که دو به دو به اشتراک گذاشته شده باشد

۲) استفاده از یک کلید مشترک گروهی

استفاده از کلیدهای دو به دو مشترک از نظر حافظه و تعداد فرایندهای رمزنگاری و ارسال مورد نیاز مانند روش رمزنگاری کلید عمومی، روش موثر و کارآمدی نمی‌باشد و استفاده از کلید گروهی مشترک به‌نظر راه حل مناسبی می‌باشد.

استفاده از رمزنگاری کلید متقارن مبتنی بر کلید گروهی مشترک نیازمند تولید و توزیع چنین کلیدی به تمام اعضای مجاز گروه می‌باشد. با در نظر گرفتن اینکه اگر تمام اعضای گروه بر یک کلید نشست مشترک گروهی به توافق برسند، تمام پیام‌های گروه با استفاده از این کلید به اشتراک گذاشته شده، می‌توانند رمز شوند.

اگر فرض شود که مهاجم از هیچ طریقی امکان دسترسی به کلید گروهی را نداشته باشد، آنگاه ارتباط بین اعضای گروه محافظت شده است. بنابراین یکی از چالش‌های اصلی طراحی در سامانه‌های ارتباطات گروهی امن و قابل اعتماد، ایجاد یک کلید گروهی مشترک قابل دسترس برای تمام اعضای گروه از طریق یک روش امن و کارآمد می‌باشد.

دو روش برای تولید کلید نشست وجود دارد: ۱- پروتکل‌های توزیع کلید ۲- پروتکل‌های توافق کلید. پروتکل‌های توزیع کلید به یک کنترلر گروه نیاز دارند تا اطلاعات تمام کاربران گروه را نگه دارد، اگر کنترلر گروه با مشکل مواجه شود و یا مورد حمله واقع شود، آنگاه گروه با شکست مواجه می‌شود و از آنجایی که در گروه‌های پویا اعضای گروه دارای تغییرات پویا هستند، کنترلر گروه ناکارآمد خواهد بود. بر خلاف پروتکل‌های توزیع کلید، پروتکل توافق کلید نیازی به کنترلر گروه نخواهد داشت؛ تمام کاربران گروه کلید نشست را با توافق بر کلید به دست می‌آورند. کلید نشست شامل اطلاعات تمام کاربران می‌باشد به طوری که هیچ کاربری نمی‌تواند کلید نشست را کنترل و یا پیش‌بینی نماید.

اولین پروتکل توافق کلید توسط دیفی-هلمن[2][۱] ارائه گردید. این پروتکل، امنیت ارتباط بین دو کاربر را تضمین می‌کند. ولی در این پروتکل کاربران تصدیق نمی‌شوند، بنابراین در مقابل حمله مردی در میانه[3] آسیب پذیر است. ژاکس[4] با استفاده از زوج سازی، یک پروتکل توافق کلید سه عضوی را مطرح نمود[۲].

زمانی که ۳ کاربر قصد توافق بر یک کلید نشست را داشته باشند، تنها یک پیام باید از طرف هر کاربر ارسال شود. اما پروتکل ژاکس نیز کاربران را تصدیق نمی‌کند و در برابر حمله مردی در میانه آسیب-پذیر است.

هر دو روش تاسیس کلید می‌توانند در هر دو حالت گروه‌های ثابت و پویا تجزیه و تحلیل شوند. به وضوح در گروه‌های پویا می‌توان کلید گروهی را در هر بار تغییر اعضای گروه با دوباره آغاز کردن پروتکل به دست آورد. ولی زمانی که گروه بزرگ باشد یا هزینه محاسبات گروه زیاد باشد این روش محاسبه کلید کارآمد نمی‌باشد. بنابراین بسیاری از پروتکل‌های تاسیس کلید پویا برای عملکرد بهتر در اضافه و کم شدن کاربران در گروه‌های پویا طراحی شدند.

درخت‌های تابع یک طرفه[5] می‌توانند برای محاسبه درخت کلیدها[6] استفاده شوند. کلیدها از برگ‌های درخت تا ریشه درخت محاسبه می‌شوند. کلیدهای سلسله مراتبی با توجه به آنکه کارآمدی پروتکل را در خلال تغییرات گروه بهبود می‌بخشند، در پروتکل‌های توزیع کلید پویا برای طرح‌های مشارکتی متداول می‌باشند.

استفاده از درخت‌های تابع یک طرفه برای کلیدهای گروهی اولین بار توسط شرمن[7] در مقاله[۳] مطرح شد. هر پروتکل توافق کلید دو عضوی که یک‌سری از ویژگی‌های مطرح شده در [۳] را برآورده کند، می‌تواند با استفاده از درخت‌های تابع یک طرفه به پروتکل توافق کلید n-عضوی توسیع داده شود. پروتکل دیفی-هلمن گروهی مبتنی

[1] Multi Cast

[2] Diffie-Hellman
[3] Maninthemiddle
[4] Joux
[5] Oneway Function Trees
[6] Key Trees
[7] Sherman

بر ساختار درختی[1] [۴] یکی از پروتکل‌هایی است که پروتکل دو عضوی دیفی- هلمن را با استفاده از درخت تابع یک طرفه به پروتکل توافق کلید گروهی توسیع داده است.

ردی و دیوا نالا[2][۵] نیز پروتکل توافق کلید تصدیقی دو عضوی را با استفاده از درخت تابع یک طرفه به پروتکل توافق کلید گروهی تصدیقی توسیع دادند و اولین پروتکل توافق کلید گروهی تصدیقی را ارائه کردند. در پروتکل آنها برگ‌های درخت معرف کاربران مستقل گروه هستند. پروتکل ارائه شده توسط شنگ[3] و همکارانش [۶] نیز از ساختار درختی استفاده می‌کند ولی ساختار درختی آنها باینری کامل است بدین معنی که در این ساختار درختی هر گره درخت معرف یک کاربر خواهد بود. پروتکل توافق کلید مبتنی بر ساختار درختی سه‌تایی نیز که توسیع پروتکل سه عضوی ژاکس[۲] می‌باشد، توسط باروآ[4] و همکارانش ارائه گردید[۷]. در این پروتکل کاربران گروه در برگ‌های درخت قرار می‌گیرند. ولی پروتکل آنها نیز حاوی احراز هویت نبود. دووتا[5] و همکارانش این پروتکل را با استفاده از امضاهای چندتایی[6] به پروتکل تصدیقی تبدیل کردند[۸].

در این مقاله، یک پروتکل توافق کلید گروهی پویا مبتنی بر زوج-سازی ویل ارائه می‌شود. در پروتکل ارائه شده از احراز اصالت مبتنی بر شناسه و ساختار درختی سه‌تایی کامل استفاده شده است. بدین معنی که هر گره درخت معرف یک کاربر گروه خواهد بود. اگر کاربری بخواهد به گروه اضافه و یا از گروه کم شود، برای به دست آوردن کلید نشست جدید نیازی نیست که تمام کاربران گروه در ساختار جدید، تمام محاسبه‌ها قبلی را بار دیگر محاسبه نمایند. از این جهت، پروتکل ارائه شده برای گروه‌های پویا بسیار مناسب می-باشد. سایر بخش‌های مقاله بدین صورت ارائه می‌گردند: در بخش ۲ مفاهیم مقدماتی و نیازهای امنیتی مطرح می‌شود، در بخش ۳ پروتکل ارائه شده معرفی می‌گردد و ویژگی‌های امنیتی مورد نظر پروتکل ارائه شده در بخش ۴ بررسی می‌شوند، در بخش ۵ حمله‌ای بر یکی از پروتکل‌های با ساختار درختی باینری کامل ارائه می‌گردد و در بخش ۶ نیز پروتکل ارائه شده را با پروتکل‌های دیگر مقایسه می-کنیم، و نهایتاً در بخش ۷ نتیجه‌گیری کار ارائه شده، مطرح می‌شود.

۲. مفاهیم اولیه

فرض کنید G_1 یک گروه جمعی از مرتبه اول q و G_2 یک گروه ضربی از مرتبه q باشد. P مولد گروه است. فرض می‌کنیم که مسئله لگاریتم گسسته در G_1 و G_2 غیر قابل حل باشد. e یک نگاشت دو خطی بین دو گروه است($e: G_1 \times G_1 \rightarrow G_2$). این نگاشت دوخطی باید شرایط زیر را داشته باشد:

دو خطی بودن: برای تمام نقاط $P, Q \in G_1$ و $a, b \in Z_q^*$، تساوی $e(aP, bQ) = e(P, Q)^{ab}$ برقرار باشد.

زوال ناپذیری: اگر p مولد G_1 باشد، آنگاه $e(P, P) \neq 1$ است.

محاسبه‌پذیری (قابل محاسبه بودن): یک الگوریتم کارآمد به منظور محاسبه $e(P, Q)$ برای تمام نقاط $P, Q \in G_1$ وجود دارد.

برای استفاده از نگاشت دو خطی به‌منظور پیاده‌سازی پروتکل، فرض-ها و مسائل زیر باید در نظر گرفته شوند:

- مسئله تصمیم دیفی-هلمن[۹] در گروه G_1 (DDH): با در اختیار داشتن (P, aP, bP, cP) برای $c \in Z_q^*$ و a, b تصمیم گرفته شود که آیا $cP = abP$ می‌باشد یا خیر. مسئله DDH[7] می‌تواند در زمان چندجمله‌ای توسط $e(aP, bP) = e(cP, P)$ حل شود.

  فرض DDH: هیچ الگوریتم زمان چندجمله‌ای برای حل مسئله DDH در G_2 وجود ندارد.
- مسئله تصمیم چکیده‌ساز دیفی - هلمن[۱۰](HDH): با داشتن (P, aP, bP, c) و تابع چکیده‌ساز $H_1: G_1 \rightarrow Z_q^*$ تصمیم گرفته شود که آیا $c = H_1(abP) \bmod q$ است.

  فرض HDH[8]: هیچ الگوریتم زمان چندجمله‌ای برای حل مسئله HDH در G_1 وجود ندارد.
- مسئله دیفی- هلمن دوخطی (BDH): با در اختیار داشتن چند تایی (P, aP, bP, cP) ، مقدار $e(P, P)^{abc}$ را محاسبه نماید.

  فرض BDH[9]: هیچ الگوریتم زمان چند جمله‌ای برای حل مسئله BDH وجود ندارد.
- مسئله تصمیم دو خطی چکیده ساز دیفی هلمن DHBDH: با در اختیار داشتن (P, aP, bP, cP, d) و چکیده‌ساز $H_2: G_2 \rightarrow Z_q^*$ تصمیم گرفته شود که آیا $d = H_2(e(P, P)^{abc}) \bmod q$ است.

  فرض DHBDH: هیچ الگوریتم زمان چندجمله‌ای برای حل مسئله DHBDH وجود ندارد.

۳. پروتکل ارائه شده

در این بخش، پروتکل جدید خود را ارائه می‌کنیم. به منظور اجرای احراز اصالت مبتنی بر شناسه، هر کاربر باید در فاز آماده سازی در مرکز تولید کلید ثبت نام کند. پروتکل دارای ۳ مرحله است: آماده-سازی، توافق کلید و تغیر در عضویت اعضا.

۳-۱. مرحله آماده‌سازی سیستم

در این زیر بخش نشان می‌دهیم که چگونه هر کاربر می‌تواند در مرکز تولید کلید ثبت نام کند. برای این منظور در ابتدا مرکز تولید کلید یک عدد تصادفی $s \in_R Z_q^*$ انتخاب می‌کند، آنگاه $P_{pub} = sP$ را محاسبه و آن‌را به‌عنوان کلید عمومی خود ارسال می‌کند. مرکز تولید کلید، s را به‌عنوان کلید اصلی به‌صورت محرمانه نگه می‌دارد.

شناسه هر کاربر U_i برابر با $ID_i \in \{0,1\}^*$ است کلید عمومی هر کاربر عبارت است از $Q_i = H(ID_i)$. هر کاربر برای ثبت نام در مرکز تولید

[1] Tree Based Group Diffie-Hellman (TGDH)
[2] Reddyand Divya Nalla
[3] Sheng-Hua Shiau
[4] Barua
[5] Dutta
[6] Multi- Signatures
[7] Decisional Diffie Hellman
[8] Hash Decisional Diffie Hellman
[9] Bilinear Diffie Hellman

کلید با استفاده از کلید عمومی خود از طریق یک کانال امن بدین صورت اقدام می‌کند:

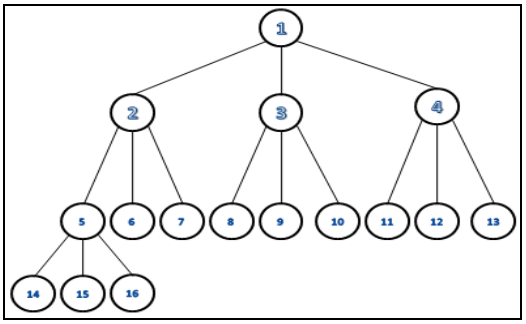
- کاربر U_i کلید عمومی Q_i را برای مرکز تولید کلید ارسال می‌کند.
- مرکز تولید کلید، کلید خصوصی بلند مدت کاربر را به صورت $S_i = sQ_i$ محاسبه و آن را برای کاربر U_i ارسال می‌کند.
- پارامترهای عمومی پروتکل عبارت اند از:

$(G_1, G_2, e, q, P, P_{pub}, H, H_1, H_2, H_3)$

که $H: \{0,1\}^* \rightarrow G_1$ ، $H_1: G_1 \rightarrow Z_q^*$ ، $H_2: G_2 \rightarrow Z_q^*$ و $H_3: G_1 \times G_1 \rightarrow Z_q^*$ توابع چکیده‌ساز رمزنگاری هستند.

۳-۲. مرحله توافق کلید

در این بخش، نشان می‌دهیم که چگونه کاربران مجاز برای محاسبه کلید نشست گروهی با یکدیگر همکاری می‌کنند. در این پروتکل فرایند توافق کلید مبتنی بر ساختار درختی سه‌تایی کامل (هر گره درخت معرف یک کاربر) است. شکل(۱) یک مثال از ساختار درختی سه‌تایی کامل با ۱۶ کاربر است.



شکل ۱. ساختار درختی سه تایی کامل با ۱۶ کاربر

فرض کنید n کاربر در گروه قرار دارد، هر کاربر U_i، $(i \in \{1,..,n\})$ زوج کلید خصوصی و عمومی $(S_i$ و $Q_i)$ را دارد. کاربران در هر بار اجرای پروتکل عدد تصادفی a_i را به‌عنوان کلید خصوصی کوتاه مدت خود انتخاب می‌کنند. در ساختار درختی سه‌تایی چهار نوع گره تعریف می‌شود: گره برگ، گره میانی با یک فرزند، گره میانی با دو فرزند و چهارمین حالت، گره میانی با سه فرزند است.

حالت اول- گره برگ، در این حالت $3i > n$ است.

- t_i را برابر a_i قرار می‌دهیم.
- کاربر U_i مقدار $t_i.P$ را برای گره والد و گره‌های مجاور خود ارسال می‌کند.

حالت دوم- گره میانی دارای یک فرزند باشد،

در این حالت $3i - 1 = n$ است.

کاربر U_i یک عدد تصادفی دیگر $\acute{a}_i$ را علاوه بر a_i انتخاب می‌کند.

۱. کاربر U_i پیام‌های $(P_i, \acute{P}_i, T_i)$ را برای کاربر U_{3i-1} ارسال می‌کند که در آن:

$P_i = a_i P$ ،$\acute{P}_i = \acute{a}_i P$ و $T_i = H_3(P_i, \acute{P}_i)\, S_i + a_i P_{pub}$ است.

۲. کاربر U_{3i-1} پیام‌های (P_{3i-1}, T_{3i-1}) را برای کاربر U_i ارسال می‌کند که در آن: $P_{3i-1} = a_{3i-1} P$

و $T_{3i-1} = H_1(P_{3i-1})\, S_{3i-1} + a_{3i-1} P_{pub}$ است.

۳. کاربر U_i برقراری رابطه: را بررسی می‌نماید.

۴. کاربر U_{3i-1} نیز برقراری رابطه:

$e(T_{3i-1}, P) = e(H_1(P_{3i-1})Q_{3i-1} + P_{3i-1}, P_{pub})$

را بررسی می‌کند.

اگر رابطه‌های بندهای ۳ و ۴ برقرار باشند، آنگاه:

۵. کاربر U_i کلید $K_i = e(P_{3i-1}, \acute{P}_i)^{a_i}$ و کاربر U_{3i-1} کلید $K_{3i-1} = e(P_i, \acute{P}_i)^{a_{3i-1}}$ را محاسبه می‌کنند. واضح است که هر دو کلید محاسبه شده توسط کاربرهای U_i و U_{3i-1} با یکدیگر برابر می‌باشد، $K_{3i-1} = e(P,P)^{a_{3i-1}\acute{a}_i a_i} = K_i$.

۶. اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر این‌صورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند.

حالت سوم- گره میانی دارای دو فرزند است، در این حالت $3i = n$ است.

در این حالت کاربرهای U_i، U_{3i-1} و U_{3i} بصورت خیلی ساده توافق کلید یک مرحله‌ای سه عضوی را انجام می‌دهند.

۱. کاربر U_i پیام $(P_i,\ T_i)$ را برای کاربرهای U_{3i-1} و U_{3i} و کاربر U_{3i-1} پیام‌های (P_{3i-1}, T_{3i-1}) رابرای کاربرهای U_i و U_{3i} و در نهایت کاربر U_{3i} پیام‌های (P_{3i}, T_{3i}) را برای کاربرهای U_i و U_{3i-1} ارسال می‌کند.

(در پیام‌های ارسال شده به‌طور کلی: $P_k = a_k P$

و $T_k = H_1(P_k) S_k + a_k P_{pub}$، $(k = i, 3i-1, 3i)$ است).

۲. در این مرحله به‌طور کلی هر کاربر U_k پیام‌های دریافتی $(P_A,\ T_A)$، $(P_B,\ T_B)$ از دو کاربر دیگر را به‌صورت هم‌زمان با استفاده از رابطه

$e(T_A + T_B, P) = e(H_1(P_A)Q_A + H_1(P_B)Q_B + P_A + P_B\ ,\ P_{pub})$

بررسی می‌کند.

اگر رابطه بند ۲ برقرار باشد،

۳. کاربر U_i کلید

$$K_i = e(P_{3i}, P_{3i-1})^{a_i} = e(P,P)^{a_i a_{3i} a_{3i-1}}$$

کاربر U_{3i-1} کلید

$$K_{3i-1} = e(P_{3i}, P_i)^{a_{3i-1}} = e(P,P)^{a_i a_{3i} a_{3i-1}}$$

کاربر U_{3i} کلید

$$K_{3i} = e(P_{3i-1}, P_i)^{a_{3i}} = e(P,P)^{a_i a_{3i} a_{3i-1}}$$

را محاسبه می‌کند.

مشاهده می‌شود که کلیدهای محاسبه شده با یکدیگر برابر می‌باشند یعنی $K_i = K_{3i} = K_{3i-1}$.

۴. اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد، بود در غیر این‌صورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند.

حالت چهارم-گره میانی دارای ۳ فرزند می‌باشد.

حالت‌هایی که توصیف شدند به‌نوعی شبیه ساختار درختی باینری توصیف شده توسط شنگ و همکارانش در مقاله [۶] بودند (شایان ذکر است در پروتکل ارائه شده، برای احراز اصالت از یک رابطه متفاوتی استفاده کردیم که در آن از دو زوج‌سازی استفاده می‌شود در حالی که احراز اصالت مقاله [۶] از سه زوج‌سازی استفاده می‌کند).

هر کاربر یک عدد تصادفی $a_k \in_R Z_q^*$ را انتخاب می‌کند و $P_k = a_k P$ و $T = H_1(P_k)\, S + aP_{pub}$ را محاسبه می‌کند.

۱. کاربر U_i پیام‌های (P_i, T_i) را برای کاربرهای U_{3i-1}، U_{3i+1} و U_{3i} ارسال می‌کند.

کاربر U_{3i-1} نیز پیام‌های (P_{3i-1}, T_{3i-1}) را برای کاربرهای U_i، U_{3i} و U_{3i+1} ارسال می‌کند.

کاربر U_{3i} نیز پیام‌های(P_{3i}, T_{3i}) را برای کاربرهای U_i، U_{3i-1} و U_{3i+1} ارسال می‌کند.

کاربر U_{3i+1} نیز پیام‌های (P_{3i+1}, T_{3i+1}) را برای کاربرهای U_i، U_{3i} و U_{3i-1} ارسال می‌کند.

۲. هر کاربر پیام‌های دریافتی از مرحله قبل را بررسی می‌کند. به طور کلی کاربر U_k پیام‌های دریافتی (P_A, T_A) ، (P_B, T_B) و (P_C, T_C) را با استفاده از رابطه زیر:

$$e(T_A + T_B + T_C, P) = e(H_1(P_A)Q_A + H_1(P_B)Q_B + H_1(P_C)Q_C + P_A + P_B + P_C\ ,\ P_{pub}) \quad (۱)$$

بررسی می‌کند. توجه شود که هر کاربر پیام‌های سه کاربر دیگر را به صورت هم‌زمان بررسی و تصدیق می‌کند.

اگر رابطه (۱) تصدیق شود،

۳. کاربر U_i پیام‌های $(P_{i,3i-1}, \acute{T}_{3i})$، $(P_{i,3i+1}, \acute{T}_{3i-1})$ و $(P_{i,3i-1}, \acute{T}_{3i+1})$ را محاسبه می‌کند و آنها را به ترتیب برای کاربرهای U_{3i-1}، U_{3i} و U_{3i+1} ارسال می‌کند.

کاربر U_{3i} نیز پیام $(P_{3i,3i-1}, \acute{T}_i)$ را محاسبه کرده و آن را برای کاربر U_i ارسال می‌کند.

به‌طور کلی $P_{i,j} = a_i P_j$ و $\acute{T}_k = H_1(P_{i,j})S_i + a_i P_{pub}$ است.

۴. به‌طور کلی، هر کاربر U_k پیام دریافتی $(P_{i,j}, \acute{T}_k)$ را با رابطه $e(\acute{T}_k, P) = e\big(H_1(P_{i,j})Q_i + P_i\ ,\ P_{pub}\big)$ بررسی می‌کند.

۵. در پایان اگر رابطه بند ۴ تصدیق شود هر کاربر کلید محرمانه را بدین صورت محاسبه می‌کند:

$$K_i = e(P_{3i,3i-1}, P_{3i+1})^{a_i} = e(P,P)^{a_i a_{3i} a_{3i-1} a_{3i+1}}$$
$$K_{3i-1} = e(P_{i,3i+1}, P_{3i})^{a_{3i-1}} = e(P,P)^{a_i a_{3i} a_{3i-1} a_{3i+1}}$$
$$K_{3i} = e\big(P_{i,3i-1}, P_{3i+1}\big)^{a_{3i}} = e(P,P)^{a_i a_{3i} a_{3i-1} a_{3i+1}}$$
$$K_{3i+1} = e(P_{i,3i-1}, P_{3i})^{a_{3i+1}} = e(P,P)^{a_i a_{3i} a_{3i-1} a_{3i+1}}$$

واضح است که تمام کلیدهای محاسبه شده با یکدیگر برابر هستند، یعنی $K_i = K_{3i} = K_{3i-1} = K_{3i+1}$..

۶. اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر این‌صورت، کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند.

هر کاربر مراحل فوق را دنبال می‌کند تا به ریشه درخت برسد، بنابراین تمام کاربران کلید نشست گروهی مشترک K_i را می‌توانند به‌دست آورند.

۳-۳. مرحله تغییر در عضویت کاربران

این احتمال وجود دارد تا یک‌سری از کاربران در هنگام اجرای پروتکل بخواهند به گروه اضافه شوند و یا اینکه گروه را ترک کنند. از نظر الزامات امنیتی، پروتکل باید به گونه‌ای باشد تا کاربرانی که گروه را ترک می‌کنند دیگر قادر به محاسبه پیام‌های جدید گروه نباشند و همچنین کاربرانی که جدیداً به گروه اضافه می‌شوند نیز نباید قادر به محاسبه پیام‌های قبلی گروه باشند. برای این منظور، زمانی که کاربری به گروه اضافه می‌شود و یا کاربری گروه را ترک می‌کند یکسری مراحل را برای برقراری الزامات امنیتی باید اجرا نماییم.

۳-۳-۱. پروتکل پیوستن به گروه

فرض می‌شود قبل از اینکه کاربری به گروه اضافه شود در ابتدا n کاربر در گروه وجود داشته باشد. مکان کاربر جدید در گروه در $(n+1)$-امین گره درخت کامل سه‌تایی است.

کاربر جدید مراحل زیر را طی می‌کند:

۱. کاربر U_1 اطلاعات گروه شامل تعداد کاربران گروه و کلیدهای عمومی تمام کاربران را برای کاربر جدید U_{n+1} ارسال می‌کند.

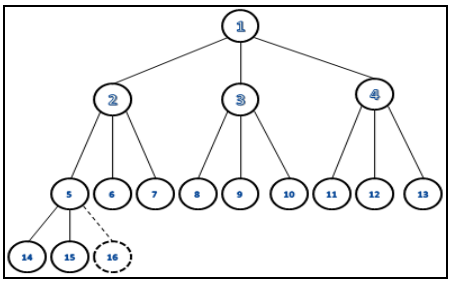
۲. کاربر U_{n+1} عدد تصادفی $a_{n+1} \in_R Z_q^*$ را به‌عنوان کلید خصوصی کوتاه مدت خود انتخاب و سپس $P_{n+1} = a_{n+1} P$ و امضای $T_{n+1} = H_1(P_{n+1})\, a_{n+1} + a_{n+1} P_{pub}$ را محاسبه و منتشر می‌کند.

۳. با توجه به حالت‌های زیر کلید نشست جدید محاسبه می‌شود و هر کلید میانی K_i-ای که در مسیر گره $(n+1)$-ام و گره ریشه قرار داشته باشد، تغیر خواهد کرد.

زمانی که کاربر U_{n+1} به گروه با n کاربر اضافه می‌شود، سه حالت مختلف ممکن است در ساختار اولیه گروه وجود داشته باشد که عبارتند از:

حالت اول- زمانی که $n = 0 \ mod\ 3$ یا $n = 3K$ باشد:

در این حالت پس از پیوستن کاربر جدید به گروه، آخرین گره والد (گره شماره ۵ در شکل (۲)) دارای سه گره فرزند خواهد بود. (شکل (۲)).



شکل۲. در ابتدا ۱۵ کاربر در گروه وجود دارد، گره ۱۶-ام مربوط به کاربر جدید است.

در این حالت i را برابر $n/3$ قرار می‌دهیم $(i = n/3)$ و U_i والد کاربر جدید U_{n+1} است. در این حالت U_i دارای سه گره فرزند است و مراحل محاسبه کلید را مانند حالت چهارم مرحله توافق کلید انجام می‌دهد که در این صورت U_{n+1} مانند U_{3i+1} در حالت چهارم عمل می‌کند. در پایان اجرای مراحل حالت چهارم از مرحله توافق کلید:

- کاربر U_i کلید $K_i = e(P_{3i,3i-1}, P_{n+1})^{a_i}$
- کاربر U_{3i-1} کلید $K_{3i-1} = e(P_{i,n+1}, P_{3i})^{a_{3i-1}}$
- کاربر U_{3i} کلید $K_{3i} = e(P_{i,3i-1}, P_{n+1})^{a_{3i}}$
- کاربر U_{n+1} کلید $K_{n+1} = e(P_{i,3i-1}, P_{3i})^{a_{n+1}}$

را محاسبه می‌کند. واضح است:

$$K_i = K_{3i} = K_{3i-1} = K_{n+1} = e(P,P)^{a_i a_{3i} a_{3i-1} a_{n+1}}$$

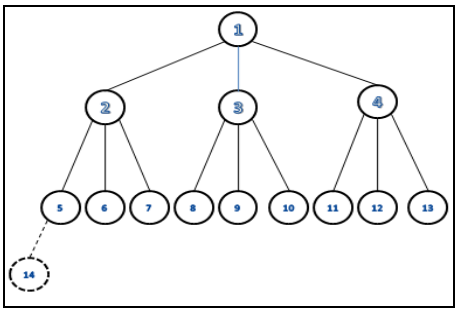
- اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر اینصورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند. آنگاه مرحله توافق کلید را ادامه می‌دهد تا به ریشه درخت برسند.

حالت دوم- در این حالت $n = 1 \ mod\ 3$ یا $n = 3k + 1$ است.

در این حالت پس از پیوستن کاربر جدید U_{n+1} به گروه، آخرین گره والد (گره شماره ۵ در شکل (۳)) دارای یک گره فرزند خواهد بود. (شکل (۳)).

در این حالت i را برابر با $(n+2)/3$ قرار می‌دهیم، U_i والد کاربر جدید U_{n+1} خواهد بود و U_{n+1} تنها فرزند آن است. مانند حالت دوم در مرحله توافق کلید:

- کاربر U_i یک عدد تصادفی دیگر $\acute{a}_i$ را علاوه بر a_i انتخاب کرده و $P_i = a_i P$، $\acute{P}_i = \acute{a}_i P$ و $T_i = H_3(P_i, \acute{P}_i)\, S_i + a_i P_{pub}$ را محاسبه می‌کند و سپس پیام $(P_i, \acute{P}_i, T_i)$ را برای کاربر U_{n+1} ارسال می‌کند.



شکل ۳. در ابتدا ۱۳ کاربر در گروه وجود دارد گره ۱۴-ام مربوط به کاربر جدید است.

- کاربر U_{n+1} نیز:

$P_{n+1} = a_{n+1} P$ و $T_{n+1} = H_1(P_{n+1})\, S_{n+1} + a_{n+1} P_{pub}$ را محاسبه و پیام (P_{n+1}, T_{n+1}) را برای کاربر U_i ارسال می‌کند.

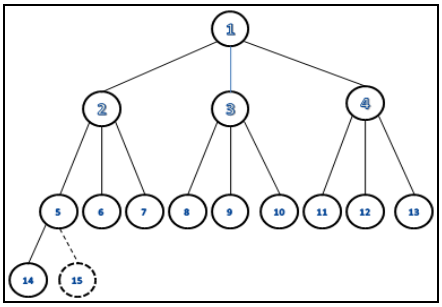
- کاربرهای U_i و U_{n+1} پیام‌های دریافتی خود را مانند بندهای ۳ و ۴ از حالت دوم مرحله توافق کلید بررسی می‌کنند. اگر رابطه تصدیق برقرار باشد آنگاه:

کاربر U_i کلید $K_i = e(P_{n+1}, \acute{P}_i)^{a_i}$ و کاربر U_{n+1} کلید $K_{n+1} = e(P_i, \acute{P}_i)^{a_{n+1}}$ را محاسبه می‌کند. توجه شود که $K_{n+1} = e(P,P)^{a_{n+1} \acute{a}_i a_i} = K_i$ است.

- اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر اینصورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند. آنگاه مرحله توافق کلید را ادامه می‌دهند تا به ریشه درخت برسند.

حالت سوم- در این حالت $n = 2 \ mod\ 3$ یا $n = 3K - 1$ است.

در این حالت پس از پیوستن کاربر جدید U_{n+1} به گروه، آخرین گره والد دارای دو گره فرزند خواهد بود. (شکل (۴)).



شکل ۴. در ابتدا ۱۴کاربر در گروه وجود دارد، گره ۱۵-ام مربوط به کاربر جدید است.

در این حالت i را برابر با $(n+1)/3$ قرار می‌دهیم، اکنون U_i دارای دو فرزند است و مراحل محاسبه کلید را مانند حالت سوم در مرحله توافق کلید انجام می‌دهد (U_{n+1} مانند U_{3i} در حالت ۳ عمل می‌کند).

پس از اجرای مرحله‌های بند ۱ و ۲ از حالت سوم مرحله توافق کلید:

- کاربر U_i کلید

$$K_i = e(P_{n+1}, P_{3i-1})^{a_i} = e(P,P)^{a_i a_{n+1} a_{3i-1}}$$

- کاربر U_{3i-1} کلید

$$K_{3i-1} = e(P_{n+1}, P_i)^{a_{3i-1}} = e(P,P)^{a_i a_{3i} a_{3i-1}}$$

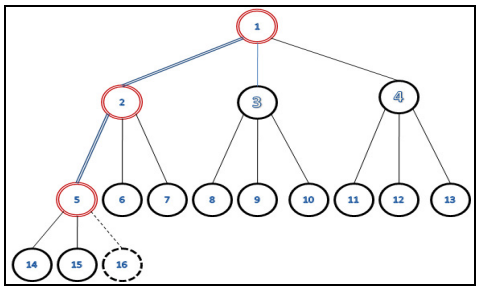
- کاربر U_{n+1} کلید

$$K_{n+1} = e(P_{3i-1}, P_i)^{a_{n+1}} = e(P,P)^{a_i a_{n+1} a_{3i-1}}$$

را محاسبه می‌کند، که $K_i = K_{n+1} = K_{3i-1}$ است.

- اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر این‌صورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند. آنگاه مرحله توافق کلید را ادامه می‌دهند تا به ریشه درخت برسند.

همان گونه که قبلاً نیز اشاره گردید برای بازسازی و بروز کردن کلید نشست گروهی، هر کلید میانی K_i مربوط به گره i که در مسیر گره $(n+1)$-ام و گره ریشه قرار گرفته است، تغییر خواهد کرد. بنابراین در هر سه حالتی که برای پیوستن کلید در این بخش توصیف شد کلیدهای K_5، K_2 و در نتیجه کلید K_1 که کلید نشست گروه است تغییر خواهند کرد. شکل (۵) مسیر این تغییرها را زمانی که کاربر U_{16} به یک گروه با ۱۵ کاربر اضافه می‌شود، نشان می‌دهد.



شکل ۵. زمانی که کاربر U_{16} به گروه اضافه می‌گردد کلیدهای K_5، K_2 و K_1 تغییر خواهند کرد.

۳-۳-۲. پروتکل ترک کردن گروه

فرض کنید که در ابتدا n کاربر در گروه باشد. کاربری که گروه را ترک می‌کند با U_l نشان می‌دهیم. برای اجرای پروتکل ترک کردن گروه، جای کاربر U_n و U_l را با یکدیگر عوض می‌کنیم و به سادگی کاربر U_l را حذف کرده و کلید نشست جدید را با توجه به ساختار جدید درخت محاسبه می‌کنیم. با توجه به موقعیت کاربر U_l در درخت سه‌تایی کامل سه حالت مختلف پیش می‌آید که بدین صورت است:

حالت اول- $\mathbf{l = n}$ است.

در این حالت گره ترک کننده گروه آخرین گره درخت می‌باشد و پروتکل می‌تواند مستقیماً کاربر U_n را حذف کند و کلید نشست جدید را تولید کند.

الف) اگر $n = 3k + 1$ باشد، i را برابر با $(n+2)/3$ قرار می‌دهیم. در این حالت پس از آنکه کاربر U_n گروه را ترک کرد، کاربر U_i دارای دو فرزند خواهد بود. U_i یک عدد تصادفی جدید $\acute{a}_i$ را بعنوان کلید خصوصی کوتاه مدت خود انتخاب می‌کند و مانند حالت سوم در مرحله توافق کلید عمل می‌کند:

- کاربر U_i مقادیر $\acute{P}_i = \acute{a}_i P$ و $\acute{T}_i = H_3(\acute{P}_i)\, S_i + \acute{a}_i P_{pub}$ را محاسبه می‌کند آنگاه پیام $(\acute{P}_i, \acute{T}_i)$ را برای کاربر U_{3i-1} و U_{3i} ارسال می‌کند و در نهایت کلید

$$K_i = e(P_{3i}, P_{3i-1})^{\acute{a}_i} = e(P,P)^{\acute{a}_i a_{3i} a_{3i-1}}$$

را محاسبه می‌کند.

- کاربر U_{3i-1} پس از بررسی پیام $(\acute{P}_i, \acute{T}_i)$ کلید

$$K_{3i-1} = e(P_{3i}, \acute{P}_i)^{a_{3i-1}} = e(P,P)^{\acute{a}_i a_{3i} a_{3i-1}}$$

را محاسبه می‌کند.

- کاربر U_{3i} نیز پس از بررسی پیام $(\acute{P}_i, \acute{T}_i)$ کلید

$$K_{3i} = e(P_{3i-1}, \acute{P}_i)^{a_{3i}} = e(P,P)^{\acute{a}_i a_{3i} a_{3i-1}}$$

را محاسبه می‌کند.

- اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر این‌صورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند. آنگاه مرحله توافق کلید را ادامه می‌دهند تا به ریشه درخت برسند.

ب) اگر $n = 3k$ باشد، i را برابر $n/3$ قرار می‌دهیم. در این حالت پس از آن که کاربر U_n گروه را ترک کرد، کاربر U_i تنها یک فرزند خواهد داشت و مانند حالت دوم در مرحله توافق کلید، یک عدد تصادفی دیگر $\acute{a}_i$ را علاوه بر a_i انتخاب می‌کند.

- کاربر U_i مقادیر

$$P_i = a_i P \text{ ،} \acute{P}_i = \acute{a}_i P \text{ و } T_i = H_3(P_i, \acute{P}_i)\, S_i + a_i P_{pub}$$

را محاسبه و پیام $(P_i, \acute{P}_i, T_i)$ را برای U_{3n-1} ارسال می‌کند:

کاربر U_{3n-1} نیز پیام (P_{3n-1}, T_{3n-1}) را برای کاربر U_i ارسال می‌کند که در آن:

$$P_{3n-1} = a_{3n-1} P \text{ و } T_{3n-1} = H_1(P_{3n-1})\, S_{3n-1} + a_{3n-1} P_{pub}$$

- کاربرهای U_i و U_{3n-1} پیام‌های دریافتی را مانند بند ۳ از حالت ۲ در مرحله توافق کلید بررسی می‌کنند. اگر رابطه تصدیق تایید شود U_i کلید $K_i = e(\acute{a}_i P_{3n-1}, \acute{P}_{3n-1})^{a_i}$ و کاربر U_{3n-1} کلید $K_{3n-1} = e(\acute{a}_{3n-1} P_i, \acute{P}_i)^{a_{3n-1}}$ را محاسبه می‌کند. می‌توان بررسی کرد که

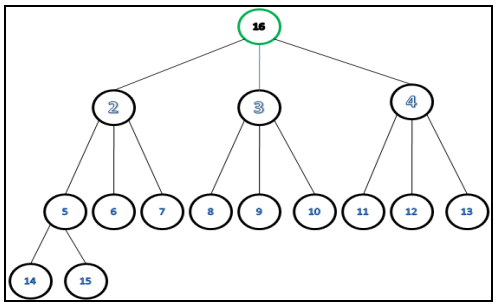
$$K_{3n-1} = e(P,P)^{\acute{a}_{3n-1} a_{3n-1} \acute{a}_i a_i} = K_i$$

- اگر $i = 1$ باشد آنگاه کلید نشست همان K_i خواهد بود در غیر این‌صورت کاربر U_i مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و $P_i = t_i P$ را برای گره والد و گره‌های مجاور خود و فرزندان گره‌های مجاور خود در گروه ارسال می‌کند. آنگاه مرحله توافق کلید را ادامه می‌دهند تا به ریشه درخت برسند.

اگر $n = 3k - 1$ باشد، i را برابر $(n+1)/3$ قرار می‌دهیم، در این حالت پس از آنکه کاربر U_n گروه را ترک کرد، U_i دیگر هیچگونه فرزندی نخواهد داشت بنابراین یک عدد تصادفی جدید $\acute{a}_i$ را بعنوان کلید خصوصی کوتاه مدت خود انتخاب می‌کند و t_i را با $\acute{a}_i$ جایگزین می‌کند و $\acute{P}_i = \acute{a}_i P$ و $\acute{T}_i = H_3(\acute{P}_i)\, S_i + \acute{a}_i \acute{P}_i$ را ارسال می‌کند. در نهایت U_i کلید K_i رابه‌روز می‌کند و فاز توافق کلید را تا زمانی که به ریشه درخت برسد اجرا می‌کند.

حالت دوم- $l = 1$ است.

در این حالت مکان کاربر ترک کننده گروه در درخت سه‌تایی کامل، ریشه درخت است. بنابراین پروتکل گره ریشه را حذف می‌کند و ریشه را با آخرین گره درخت U_n، جایگزین می‌کند و برای محاسبه کلید نشست جدید مانند حالت یک در پروتکل ترک کردن گروه عمل می‌کند. شکل (۶) یک مثال را نشان می‌دهد که در آن گروه دارای ۱۶ کاربر است و ریشه درخت یعنی کاربر U_1، گروه را ترک کرده است.



شکل ۶. گره ترک‌کننده گروه، گره ریشه است که با آخرین گره (گره ۱۶-ام) جایگزین شده است.

حالت سوم- $l \in \{2, \dots, n-1\}$ است.

در این حالت، پروتکل کاربر U_l را با U_n (آخرین گره در درخت سه‌تایی کامل) جایگزین می‌کند و مانند حالت اول توصیف شده در بخش پروتکل ترک کردن گروه ادامه می‌دهد و کلید نشست گروهی جدید را محاسبه می‌کند.

۴. تحلیل امنیتی پروتکل

در این بخش ویژگی‌های امنیتی را برای پروتکل ارائه شده مورد بررسی قرار می‌دهیم. این ویژگی‌های امنیتی به شرح زیر هستند:

- امنیت کلید شناخته شده:

این ویژگی امنیتی بیان می‌کند که اگر یک کلید نشست فاش شود، امنیت پروتکل‌های در حال اجرا نباید تحت تأثیر قرار گیرد. با فرض اینکه تنها چهار کاربر U_1، U_2، U_3 و U_4 در گروه وجود داشته باشند و کلید نشست قبلی برابر

$$K_{prev} = e(P_2, a_4 P_3)^{a_1} = e(a_4 P_1, P_3)^{a_2} = e(a_4 P_1, P_2)^{a_3} = e(a_2 P_1, P_3)^{a_4} = e(P,P)^{a_1 a_2 a_3 a_4}$$

باشد، اگر مهاجم بخواهد کلید خصوصی کوتاه مدت یکی از کاربران (برای مثال a_1)را بدست آورد باید مسئله BDH را در G_2 حل نماید، که طبق فرضی که در بخش ۲ معرفی گردید، معادل با حل یک مسئله سخت است. همچنین کلید نشست گروهی تولید شده وابسته به اعداد تصادفی انتخاب شده توسط کاربران در هر بار اجرای پروتکل می‌باشد، بنابراین در هر بار اجرای پروتکل کلید نشست متفاوت خواهد بود.

- تصدیق کلید:

ویژگی امنیتی تصدیق کلید (ضمنی) نیازمند آن است که هر کاربر مجاز پروتکل اطمینان یابد که هیچ کاربر دیگری به‌جز کاربرهای مجاز گروه نمی‌توانند کلید نشست گروهی را به دست آورند.

در پروتکل ارائه شده هر کاربر پیام تولید شده را با کلید خصوصی بلند مدت خود امضا می‌کند، بنابراین تمام کاربران با دریافت پیام‌های یکدیگر ابتدا پیام‌های دریافتی را بررسی می‌کنند و در صورت عدم تصدیق یکی از پیام‌ها پروتکل را ترک می‌کنند. از این جهت کاربران اطمینان می‌یابند که تنها کاربران مجاز پروتکل می‌توانند فرایند پروتکل را اجرا کرده و کلید نشست را محاسبه کنند.

- امنیت پیشرو:

این ویژگی امنیتی بیان می‌کند، اگر کلید خصوصی بلند مدت کاربری فاش شود امنیت کلیدهای نشست قبلی نباید تحت تأثیر قرار گیرد. از آنجایی که در پروتکل ارائه شده کلید خصوصی بلند مدت کاربران تنها برای تصدیق پیام‌ها استفاده شده است و برای تولید کلید نشست از کلید خصوصی بلند مدت کاربران استفاده نمی‌شود، با فاش شدن کلید خصوصی بلند مدت کاربران امنیت کلید نشست‌های قبلی به خطر نمی‌افتد. از این جهت پروتکل ارائه شده مورد هدف این حمله قرار نمی‌گیرد و پروتکل دارای امنیت پیشرو می‌باشد.

- امنیت در مقابل جعل هویت با کلید آشکار شده:

این ویژگی امنیتی پروتکل را در مقابل جعل هویت کاربران دیگر به جای کاربری که کلید خصوصی بلند مدت او توسط مهاجم فاش شده است مقاوم می‌کند. شایان ذکر است که کلیدهای خصوصی بلند مدت معمولاً برای تولید امضا و یا رمزگشایی استفاده می‌شوند؛ کلیدهای خصوصی بلند مدت در این پروتکل به منظور احراز اصالت استفاده می‌شوند و نه تولید کلید نشست گروهی، از این جهت پروتکل ارائه شده مورد هدف این حمله قرار نمی‌گیرد.

- کنترل کلید:

این ویژگی امنیتی بیان می‌کند که نباید هیچ کاربر مجازی در گروه وجود داشته باشد که قادر به پیش بینی و یا تحت تأثیر قرار دادن کلید نشست تولید شده باشد. در پروتکل ارائه شده کلید نشست گروهی با مشارکت تمام کاربران گروه به دست می‌آید لذا هیچ کاربری قادر به کنترل و یا پیش‌بینی کلید نشست نمی‌باشد.

- امنیت در مقابل مهاجم فعال و غیر فعال:

مهاجم غیر فعال به مهاجمی گفته می‌شود که تنها قادر به شنود کانال‌های ارتباطی می‌باشد و نمی‌تواند پیام‌ها را تغییر دهد و یا پیامی را وارد کانال کند. ولی مهاجم فعال قدرتمندتر بوده و فرض می‌شود که مهاجم فعال کنترل کاملی بر کانال ارتباطی دارد و قادر خواهد بود پیام‌های ارسال شده را تغییر دهد و یا پیام‌های خود را در هنگام اجرای پروتکل وارد کانال کند.

در پروتکل ارائه شده حالت چهارم از مرحله توافق کلید را که قسمت اصلی و نوآوری مقاله می‌باشد را در نظر می‌گیریم، در این حالت در مرحله ۱. کاربرها پیام‌های (P_i, T_i) را برای یکدیگر ارسال می‌نمایند و در مرحله ۴. کاربر U_i پیام‌های $(P_{i,3i+1}, \acute{T}_{3i-1})$، $(P_{i,3i-1}, \acute{T}_{3i})$ و $(P_{i,3i-1}, \acute{T}_{3i+1})$ را به ترتیب برای کاربرهای U_{3i-1}، U_{3i} و U_{3i+1} و کاربر U_{3i} نیز پیام $(P_{3i,3i-1}, \acute{T}_i)$ را برای کاربر U_i ارسال می‌کند.

حال فرض می‌شود که مهاجم غیر فعال پیام‌های تبادل شده را شنود کرده و در اختیار داشته باشد، برای به‌دست آوردن کلید نشست گروهی، مهاجم باید مانند مرحله ۵. یکی از کلیدهای K_i، K_{3i-1}، K_{3i} یا K_{3i+1} را محاسبه کند، برای این منظور، مهاجم باید به ترتیب کلید خصوصی کوتاه مدت کاربرهای a_i، a_{3i-1}، a_{3i} یا a_{3i+1} را در اختیار داشته باشد.

در صورتی که این کلیدها به صورت مخفی نزد کاربرها باقی می‌ماند و ارسال نمی‌گردد و برای محاسبه آنها، همان گونه که در امنیت کلید شناخته شده توضیح داده شد مهاجم با مسئله BDH در G_2 مواجه می‌شود.

مهاجم فعال علاوه بر شنود پیام‌های تبادل شده می‌تواند این پیام‌ها را تغییر دهد و یا پیام خود را جایگزین نماید، در پروتکل ارائه شده تمام کاربران با دریافت پیام‌های یکدیگر ابتدا پیام‌های دریافتی را (با استفاده از رابطه‌های مربوطه که در مرحله توافق کلید به آنها اشاره شد) بررسی می‌کنند و در صورت عدم تصدیق یکی از پیام‌ها پروتکل را ترک می‌نمایند.

از این جهت کاربران اطمینان می‌یابند که تنها کاربران مجاز پروتکل می‌توانند فرایند پروتکل را اجرا کرده و کلید نشست را محاسبه کنند. بنابراین پروتکل در مقابل مهاجم غیر فعال و مهاجم فعال امن می‌باشد.

۵. اجرای حمله ارائه‌شده توسط لان‌ژو روی پروتکل زنگ

در سال ۲۰۰۹ زنگ[1] و همکارانش در مقاله مطرح شده در [۱۱] پروتکل [۶] را با استفاده از پروتکل توافق کلید گروهی یک دوری شی[2] [۱۲] بهبود داده و حجم محاسبه‌ها را نسبت به پروتکل شنگ کاهش دادند. در سال ۲۰۰۶ لان ژو[3][۱۳] یک حمله کارآمد روی پروتکل شی ارائه داده بود که در پروتکل زنگ[۱۱] توجهی به این حمله نشده است. در این بخش حمله لان ژو بر پروتکل شی را بر پروتکل زنگ تعمیم داده و نشان می‌دهیم که این پروتکل نمی‌تواند امنیت مورد نظر را تأمین نماید. در این حمله ابتدا مهاجم $\mathcal{A}$ از KGC درخواست یک زوج کلید می‌کند.

بنابراین مهاجم $I_A = H_1(ID_A)$، کلید عمومی $Q_A = (I_A s_1 + s_2)P$ و کلید خصوصی $S_A = (I_A s_1 + s_2)^{-1}P$ که $(1 \le i \le n) I_A \ne I_i$ را دریافت می‌کند. آنگاه مهاجم $\mathcal{A}$ پیام‌های $a_i Q_{2i}$، $a_i Q_{2i+1}$ را که کاربر U_i برای کاربران U_{2i} و U_{2i+1} ارسال کرده است را انتخاب کرده و محاسبات زیر را انجام می‌دهد:

$$(I_{2i} - I_{2i+1})^{-1}(a_i Q_{2i} - a_i Q_{2i+1})$$
$$= (I_{2i} - I_{2i+1})^{-1}(a_i(I_{2i}s_1 + s_2)P - a_i(I_{2i+1}s_1 + s_2)P)$$
$$= (I_{2i} - I_{2i+1})^{-1}(a_i(I_{2i} - I_{2i+1})s_1 P) = a_i s_1 P$$
$$\left(I_{2i}^{-1} - I_{2i+1}^{-1}\right)^{-1}\left(I_{2i}^{-1} a_i Q_{2i} - I_{2i+1}^{-1} a_i Q_{2i+1}\right)$$
$$= \left(I_{2i}^{-1} - I_{2i+1}^{-1}\right)^{-1}\left(a_i(s_1 + I_{2i}^{-1}s_2)P - a_i(s_1 + I_{2i+1}^{-1}s_2)P\right)$$
$$= \left(I_{2i}^{-1} - I_{2i+1}^{-1}\right)^{-1}\left(a_i\left(I_{2i}^{-1} - I_{2i+1}^{-1}\right)s_2 P\right) = a_i s_2 P$$

و به همین صورت، مهاجم $\mathcal{A}$ می‌تواند مقادیر $a_{2i}s_1P$ ، $a_{2i}s_2P$ ، $a_{2i+1}s_1P$ و $a_{2i+1}s_2P$ را به‌دست آورده و مقادیر زیر را محاسبه کند:

$$a_i s_1 P I_A + a_i s_2 P = a_i(I_A s_1 + s_2)P = a_i Q_A$$
$$a_{2i} s_1 P I_A + a_{2i} s_2 P = a_{2i}(I_A s_1 + s_2)P = a_{2i} Q_A$$
$$a_{2i+1} s_1 P I_A + a_{2i+1} s_2 P = a_{2i+1}(I_A s_1 + s_2)P = a_{2i+1} Q_A$$

اکنون مهاجم $\mathcal{A}$ می‌تواند کلید K_i را بدین صورت محاسبه کند:

$$K_i = e(a_i Q_A + a_{2i} Q_A + a_{2i+1} Q_A, S_A)$$
$$= e(Q_A, S_A)^{(a_i + a_{2i} + a_{2i+1})}$$
$$= e(P, P)^{(a_i + a_{2i} + a_{2i+1})}$$

اگر $i = 1$، آنگاه کلید نشست K_i می‌باشد، در غیر این صورت مهاجم مقدار $t_i = H_2(K_i)$ را محاسبه می‌کند و این فرایند را ادامه می‌دهد تا به K_1 که کلید نشست گروه است، دست پیدا کند. همانطور که مشاهده شد، پروتکل مطرح شده در مقابل این حمله آسیب پذیر است.

در جدول (۱) پروتکل ارائه شده از نظر ویژگی‌های امنیتی با یک-سری از پروتکل‌های با ساختار درختی مورد مقایسه قرار گرفته است. همان طور که مشاهده می‌شود، پروتکل‌های [۶] و [۷] که مربوط به شنگ و باروآ می‌باشند نیز تمام ویژگی‌های امنیتی را برآورده می-سازند، ولی از نظر میزان محاسبات و سرعت، پروتکل ارائه شده بهتر از پروتکل‌های مذکور عمل می‌کند که در بخش بعدی این مقایسه صورت گرفته است.

[1] Zeng
[2] Yijuanshi
[3] LanZhou

جدول۱. مقایسه ویژگی‌های امنیتی پروتکل ارائه شده با برخی پروتکل‌ها

پروتکل	حمله کلید شناخته شده	تصدیق کلید	امنیت پیشرو	کنترل کلید
[۱۲]	✓	-	✓	✓
[۶]	✓	✓	✓	✓
[۷]	✓	✓	✓	✓
[۱۵]	✓	-	✓	-
[۱۶]	✓	-	✓	-
Our protocol	✓	✓	✓	✓

۶. بررسی عملکرد پروتکل ارائه شده

در این بخش پروتکل ارائه شده را از نظر میزان محاسبه‌ها و ارتباطات با پروتکل‌های شنگ [۶] و باروآ و همکارانش [۷] با توجه به شباهت در تولید کلید آنها مقایسه می‌کنیم. هر دو پروتکل شنگ و باروآ از ساختار درختی استفاده می‌کنند.

پروتکل ارائه شده توسط باروآ و همکارانش [۷]، از ساختار درختی سه‌تایی استفاده می‌کند و کاربران در برگ‌های درخت قرار می‌گیرند و باید کلید را با اجرای محاسبه‌ها از برگ‌های درخت به سمت ریشه درخت به دست آورند. پروتکل ارائه شده توسط شنگ [۶] از ساختار درختی باینری کامل استفاده می‌کند یعنی هر گره درخت معرف یک کاربر خواهد بود.

پروتکل ارائه شده در بخش (۳) از ساختار درختی سه‌تایی کامل استفاده می‌کند و هر گره درخت معرف یک کاربر است. برخلاف پروتکل باروآ و همکارانش پروتکل ارائه شده، از شناسه کاربران برای احراز اصالت استفاده می‌کند و به این ترتیب هزینه‌های PKI در سیستم ارائه شده کسر می‌شود.

عمل زوج‌سازی مهمترین و سنگین‌ترین عملیات در پروتکل‌های مبتنی بر زوج سازی می‌باشد، از این جهت تعداد زوج‌سازی‌های انجام شده در پروتکل می‌تواند معیار مناسبی برای میزان محاسبات استفاده شده در پروتکل و متعاقباً زمان مورد نیاز برای اجرای پروتکل باشد از این نظر که هرچه میزان محاسبات مورد نیاز برای اجرای پروتکل کمتر باشد زمان اجرای پروتکل نیز کمتر خواهد بود.

از این جهت در ادامه به محاسبه تعداد زوج‌سازی‌هایی که در اجرای پروتکل ارائه شده محاسبه می‌شوند می‌پردازیم و با استفاده از نرم‌افزار شبیه‌ساز MAPLE که مربوط به مدل‌کردن و شبیه‌سازی محاسبات ریاضی می‌باشد، نشان خواهیم داد که با افزایش تعداد کاربران میزان محاسبه‌ها چگونه تغییر خواهند کرد. با توجه به پویا بودن پروتکل ارائه شده بررسی کارایی آن با توجه به اضافه (کم) شدن کاربران به (از) گروه از اهمیت فراوانی برخوردار است.

برای محاسبه تعداد کامل زوج سازی‌های استفاده شده در پروتکل، حاصل جمع تعداد تمام زوج سازی‌هایی که کاربران برگ استفاده می‌کنند با تعداد کل زوج‌سازی‌های استفاده شده توسط گره‌های میانی را به‌دست می‌آوریم. برای به‌دست آوردن کلید نشست، گره‌های برگ باید فرایند محاسبه‌ها توصیف شده در بخش (۳-۲) را (با توجه به یکی از حالت‌های منطبق با خود) تا زمانی که به ریشه درخت برسند اجرا کنند.

بنابراین کاربری که در گره برگ قرار گرفته باشد باید به اندازه $R(n)$ مرتبه محاسبه‌ها را تکرار کند که $R(n)$ تعداد مرحله‌های پروتکل است و در پروتکلی که n کاربر دارد تعداد کاربرانی که در برگ‌های درخت قرار می‌گیرند برابر $\left(n-\left(\frac{3^{R(n)}-1}{2}\right)\right)$ است.

ولی باید توجه کرد که ممکن است یکسری از گره‌های برگ در آخرین سطح درخت یعنی $R(n)$-امین سطح قرار نداشته باشند و در $[R(n)-1]$-امین سطح درخت قرار داشته باشند، بنابراین این کاربران محاسبه‌ها را یک مرحله کمتر از کاربران گره برگ تکرار می‌کنند. در نتیجه باید تعداد آنها را از $\left(n-\left(\frac{3^{R(n)}-1}{2}\right)\right)R(n)$ کم کرد. تعداد گره‌های برگی که در سطح $[R(n)-1]$-ام قرار دارند برابر $3^{R(n)-1}-\left\lceil\frac{n-3^{R(n)}-1/2}{3}\right\rceil$ است. در مورد گره‌های میانی، در هر سطح l، تعداد گره‌ها 3^lاست.

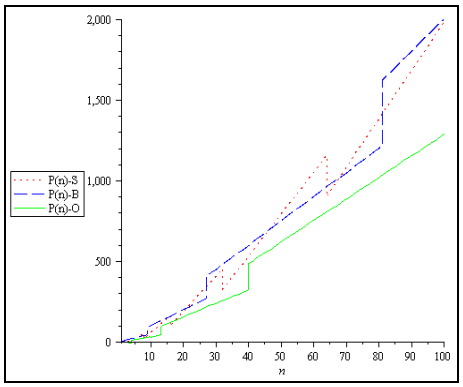
هر یک از گره‌های میانی، فرایند توصیف شده در بخش (۳-۲) را $l+1$ بار انجام می‌دهند. بنابراین تعداد کل آنها از سطح 0 تا $R(n)-1$ برابر با $\sum_{i=1}^{R(n)} i\,3^{i-1}$ است. در نهایت تعداد کل تکرارهای محاسبات توصیف شده در بخش (۳-۲) برابر است با:

$$\sum_{i=1}^{R(n)} i\,3^{i-1}+\left(n-\left(\frac{3^{R(n)}-1}{2}\right)\right)R(n)-3^{R(n)-1}+\left\lceil\frac{n-3^{R(n)}-1/2}{3}\right\rceil \quad (۲)$$

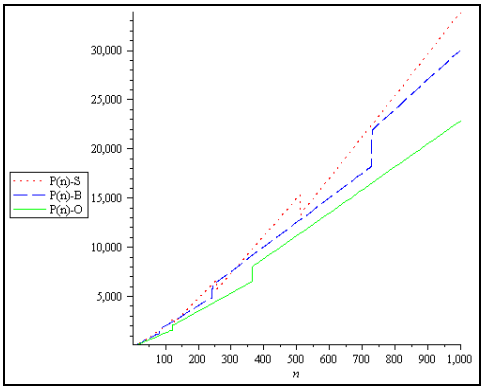
و برای محاسبه کلید نشست K_i با توجه به محاسبات بخش (۳-۲)، برای احراز اصالت پیام‌ها به ۴ زوج‌سازی و برای محاسبه K_i یک زوج‌سازی نیاز است. بنابراین برای به دست آوردن تعداد کل زوج-سازی استفاده شده در پروتکل ارائه شده کافی است تا رابطه (۲) را در عدد ۵ ضرب کنیم.

روند رشد میزان محاسبه‌ها پروتکل ارائه شده در بخش (۳) در مقایسه با دو پروتکل شنگ و باروآ در نمودار شکل‌های (۷) و (۸) آورده شده است. شکل(۷) زمانی که بیشینه تعداد کاربران ۱۰۰ است و شکل(۸) نمودار رشد محاسبه‌ها با توجه به افزایش تعداد کاربران زمانی که بیشینه کاربران ۱۰۰۰ است را نشان می‌دهد.

در نمودارها در محور عمودی $P(n)-S$، $P(n)-B$ و $P(n)-O$ به‌ترتیب معرف تعداد زوج‌سازی‌های پروتکل شنگ، پروتکل باروآ و پروتکل ارائه شده هستند. مشاهده می‌شود که پروتکل ارائه شده در مقایسه با دو پروتکل دیگر از نظر میزان محاسبه‌ها به‌خصوص زمانی که تعداد کاربران زیاد باشد بسیار بهینه است.



شکل ۷. مقایسه روند رشد میزان محاسبه‌ها زمانی که تعداد کاربران به عدد ۱۰۰ برسد.



شکل ۸. مقایسه روند رشد میزان محاسبه‌های زمانی که تعداد کاربران به عدد ۱۰۰۰ برسد.

برای محاسبه تعداد کل پیام‌های تبادل شده توسط کاربران نیز، هر گره میانی ۹ پیام را ارسال می‌کند و هر گره برگ ۳ پیام و گره ریشه نیز ۶ پیام را ارسال می‌کند، با ضرب کردن تعداد کل گره‌های میانی در عدد ۹ و تعداد کل گره های برگ در عدد ۳ و جمع کردن آنها با یکدیگر به‌علاوه ۶ پیام ارسالی از جانب گره ریشه، می‌توان بررسی نمود که تعداد کل پیام‌های ارسال شده حداکثر به اندازه $5(n-1)$ است.

$$\left[\left(\left(\frac{3^{R(n)}-1}{2}\right)\right)-3^{R(n)-1}+\left[\frac{n-3^{R(n)}-1/2}{3}\right]\right]\times 9+\left[\left(n-\left(\frac{3^{R(n)}-1}{2}\right)\right)+3^{R(n)-1}-\left[\frac{n-3^{R(n)}-1/2}{3}\right]\right]\times 3+6\cong 5(n-1)$$

مقایسه میزان محاسبه‌ها و تبادل اطلاعات پروتکل ارائه شده با دو پروتکل شنگ[۶] و باروآ [۷] در جدول (۲) آورده شده است. در جدول (۲) R(n) معرف تعداد مرحله‌های پروتکل، B(n) تعداد کل پیام‌های تبادل شده است و P(n) معرف تعداد کل زوج سازی‌هایی است که در پروتکل محاسبه می‌شود.

۷. نتیجه‌گیری

در این مقاله یک پروتکل توافق کلید گروهی تصدیقی پویا مبتنی بر شناسه ارائه گردید. در پروتکل ارائه شده از ساختار درختی سه‌تایی کامل استفاده شده است، بدین معنی که هر گره درخت معرف یک کاربر مجاز گروه است. هر کاربر گروه می‌تواند پیام‌های دریافت شده را با استفاده از ساختار مبتنی بر شناسه تصدیق نماید و نیازی به بررسی گواهینامه کلیدهای عمومی کاربران نمی‌باشد. از این جهت، هزینه‌های سنگین مربوط به PKI در پروتکل ارائه شده کاهش یافته و در نتیجه کارآمدی پروتکل افزایش یافته است. همچنین مکانیزم-هایی برای اضافه شدن کاربران به گروه و یا ترک کردن گروه ارائه گردید که نشان می‌دهد پروتکل ارائه شده برای گروه‌های پویا کارآمد و مناسب می‌باشد. همچنین، پروتکل ارائه شده، نیازهای امنیتی مهم از جمله امنیت کلید شناخته شده، تصدیق کلید، امنیت پیشرو، امنیت در مقابل جعل هویت با کلید آشکار شده و کنترل کلید را برآورده می‌سازد.

جدول۲. مقایسه میزان محاسبه‌ها و ارتباطات پروتکل‌ها با یکدیگر

	$R(n)$	$B(n)$	$P(n)$
Barua [2]	$\lceil log_3 n \rceil$	$\leq \frac{5}{2}(n-1)+n[R(n)-1]-\frac{3}{2}\left(3^{R(n)-1}-1\right)$	$\lceil log_3 n \rceil \times n \times 5$
Sheng [13]	$\lfloor log_2 n \rfloor$	$\leq 3(n-1)+1$	$\left\{\begin{matrix}\sum_{i=1}^{R(n)} i\,2^{i-1}+\left(n-\left(2^{R(n)}\right)\right)R(n) \\ -2^{R(n)}+\left\lfloor\frac{n+1}{2}\right\rfloor\end{matrix}\right\}\times 4$
Our protocol	$\lceil log_3(2n+1/3)\rceil$	$\leq 5(n-1)$	$\left\{\begin{matrix}\sum_{i=1}^{R(n)} i\,3^{i-1}+\left(n-\left(\frac{3^{R(n)}-1}{2}\right)\right)R(n) \\ -3^{R(n)-1}+\left\lfloor\frac{n-3^{R(n)}-1/2}{3}\right\rfloor\end{matrix}\right\}\times 5$

-R(n) معرف تعداد مرحله‌های پروتکل،

-B(n) معرف تعداد کل پیام‌های تبادل شده است، - P(n) معرف تعداد کل زوج سازی‌هایی است که در پروتکل محاسبه می‌شود.

۸. مراجع

[1]. Diffie, W.; Hellman, M. “New Directions in Cryptography.”; IEEE Transactions on Information Theory, 1976, 22, 644-654.

[2]. Joux, A. “A One-Round Protocol for Tripartite Diffie-Hellman.”; Proceedings of 4th Algorithmic Number Theory Symposium, LNCS 1838, 385-394, Springer-Verlag, 2000.

[3]. McGrew, D.; Sherman, A. "Key Establishment in Large Dynamic Groups Using One-Way Function Trees."; TIS Report No. 0755, TIS Lab at Network Associate, Inc, Glenwood, 1998.

[4]. Kim, Y.; Perrig, A.; Tsudik, G. "Simple and Fault Tolerant Key Agreement for Dynamic Collaborative Groups."; Proceedings of 7th ACM Conference on Computer and Communications Security, November 2000, 235-244.

[5]. Reddy, K. C.; Nalla, D. “Identity Based Authenticated Group Key Agreement Protocol.”; Proceedings of INDOCRYPT’02, LNCS 2551, 2002, 215–233.

[6]. Sheng, H.; Hwang, R.; Lin, M. "Key Agreement Protocol Based on Weil Pairing."; Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA 2005), 597-602.

[7]. Barua, R.; Dutta, R.; Sarkar, P. "Extending Joux's Protocol to Multi Party Key Agreement."; 3rd International Cryptology Conference in India -Indocrypt'2003, LNCS 2904, 205-217, Springer-Verlag, 2003.

[8]. Barua, R.; Dutta, R.; Sarkar, P. “Provably Secure Authenticated Tree Based Group Key Agreement.”; Proceedings of ICICS’04, LNCS 3269, 92-104, Springer-Verlag, 2004.

[9]. Boneh, D.; Franklin, M. "Identity-Based Encryption from the Weil Pairing."; In Advances in Cryptology – CRYPTO, LNCS 2139, 213-229, Springer-Verlag, 2001.

[10]. Abdalla, M.; Bllare, M.; Rogaway, P. "An encryption scheme based on the Diffie-Hellman problem."; CT-RSA 2001, 143-158.

[11]. Zeng, P. “An Efficient Identity-Based Group Key Agreement Protocol.”; IEEE Transactions on Computers, 2009, 597-602.

[12]. Shi, Y.; Chen, G.; Li, J. "ID-Based One Round Authenticated Group Key Agreement Protocol with Bilinear Pairings."; Preceding of IEEE the International Conference on Information Technology: Coding and Computing (ITCC), 2005.

[13]. Zhou, L. “Efficient ID-Based Authenticated Group Key Agreement from Bilinear Pairings.”; Proceedings of Mobile Ad-Hoc and Sensor Networks (MSN), LNCS 4325, 521-532, Springer-Verlag, 2006.

[14]. Kim, Y.; Perrig, A.; Tsudik, G. “Tree-Based Group Key Agreement,”; ACM Transactions on Information and System Security, February 2004, 7(1), 60–96.

[15]. Desmedt, Y.; Lange, T.; Burmester, M. “Scalable Authenticated Tree Based Group Key Exchange for Ad-Hoc Groups.”; In: Financial Cryptography and Data Security, LNCS 4886, 104–118, Springer, Heidelberg, 2007.

[16]. Desmedt, Y.; Lange, T. “Revisiting Pairing Based Group Key Exchange.”; In: Financial Cryptography and Data Security, FC 2008, LNCS 5143, 53–68, Springer-Verlag, 2008.