مرور و بررسی روش‌های جمع چندسویهٔ امن و چالش‌های موجود

شادیه عزیزی[1]، مائده عاشوری تلوکی[2*] و حمید ملا[3]

[1] گروه مهندسی فناوری اطلاعات، دانشگاه اصفهان، اصفهان، ایران
sh.azizi93@eng.ui.ac.ir

[2 و 3] استادیار، مهندسی فناوری اطلاعات، دانشگاه اصفهان، اصفهان، ایران
m.ashouri@eng.ui.ac.ir
h.mala@eng.ui.ac.ir

چکیده

در حوزهٔ امنیت اطلاعات، انجام محاسبات ریاضی بر روی داده‌های خصوصی به‌صورت امن و گروهی (محاسبات چندسویهٔ امن) بیش از پیش مورد توجه قرار گرفته است. نخستین‌بار، محاسبات چندسویهٔ امن، در قالب مسئلهٔ میلیونرها مطرح شد که در آن دو میلیونر بدون افشای میزان سرمایهٔ خود و بدون استفاده از طرف سوم مورد اعتماد، قصد داشتند بدانند کدامیک ثروتمندتر است. پس از آن مسائل دیگری در حوزه محاسبات چندسویهٔ امن مطرح شد. در این پژوهش مسئلهٔ جمع چندسویهٔ امن، در نظر گرفته شده است؛ در جمع چندسویهٔ امن گروهی از کاربران قصد محاسبه مجموع دادهٔ محرمانه خود را دارند؛ به‌طوری‌که محرمانگی داده‌های آنها حفظ شود. در این مقاله پیشینه‌ای از راه‌حل‌های موجود در این حیطه بررسی و مقایسه شده‌اند. به‌علاوه چالش‌های موجود در این زمینه بررسی و پیشنهادهایی جهت راه‌کارهای آینده ارائه شده‌اند.

واژگان کلیدی: جمع چند سویهٔ امن، حملهٔ تبانی، کانال ناامن.

۱- مقدمه

در محاسبات چندسویه امن کاربران P_1 ، P_2 ، ... و P_n به‌ترتیب دارای مقادیر محرمانه d_1 ، d_2 ، ... و d_n بوده و قصد محاسبه امن تابع $f(d_1,d_2,...,d_n)$ را دارند؛ اما چون اعتماد کامل در بین اعضا وجود ندارد؛ باید علاوه بر درستی نتیجه تابع f مقدار d_i را فقط P_i بداند و از دید سایرین پنهان بماند. در پروتکل‌های جمع چندسویهٔ امن هدف محاسبه مجموع دادهٔ محرمانه کاربران است به‌طوری‌که در پایان محاسبه، هر عضو تنها داده محرمانهٔ خود و نتیجه حاصل‌جمع را می‌داند و از داده محرمانهٔ دیگران مطلع نیست. پروتکل‌های جمع چندسویهٔ امن را به دو دسته می‌توان تقسیم کرد:

دستهٔ نخست، راه‌کارهایی هستند که فرض می‌کنند در بین اعضا کانال امن وجود دارد؛ در این راه‌کارها کانال ارتباطی بین اعضا غیر قابل شنود و امن است؛ بنابراین تلاش می‌شود اطلاعاتی که کاربران در جهت محاسبهٔ مجموع در اختیار یکدیگر قرار می‌دهند، باعث افشای مقدار محرمانه آنها نشود. هر داده‌ای که در کانال ارسال می‌شود، فقط توسط گیرنده مشاهده می‌شود. به‌عنوان نمونه در محاسبه مجموع مقادیر در چند پایگاه داده فرض وجود کانال امن کاربرد دارد.

دستهٔ دوم راه‌کارهای با فرض کانال ناامن هستند؛ در این راه‌کارها فرض می‌شود، اطلاعاتی که در کانال ارتباطی ارسال می‌شود، قابل شنود و علاوه‌بر گیرندهٔ اطلاعات، سایرین نیز قادر به شنود و دریافت این اطلاعات هستند؛ بنابراین لازم است تمهیداتی جهت امنیت داده‌های ارسالی

* نویسندهٔ عهده‌دار مکاتبات

[Downloaded from monadi.isc.org.ir on 2025-06-10]

اندیشیده شود؛ مانند استفاده از رمزنگاری هم‌ریخت در محاسبات ابری تلفن همراه و خدمات مبتنی بر مکان.

در محاسبات چندسویهٔ امن دو مدل مهاجم وجود دارد: مدل مهاجم شبه‌درستکار و مدل مهاجم بدخواه. در مدل مهاجم شبه‌درستکار، اعضای گروه از روال پروتکل تبعیت، اما جهت به‌دست‌آوردن اطلاعاتی راجع به دادهٔ محرمانهٔ سایر اعضای گروه کنجکاوی می‌کنند. در مدل مهاجم بدخواه، اعضای گروه به‌دلخواه داده ارسال می‌کنند و از روال پروتکل تبعیت نمی‌کنند.

در این مقاله پروتکل‌های جمع چندسویهٔ امن ارائه‌شده در بازهٔ سال‌های ۱۹۸۸ تا ۲۰۱۷ بررسی و مقایسه شده‌اند. ساختار مقاله بدین‌صورت است که در بخش دوم راه‌کارهای ارائه‌شده با فرض کانال امن و در بخش سوم راه‌کارهای با فرض عدم وجود کانال امن، مرور و بررسی می‌شوند و در بخش چهارم راه‌کارهای ارائه‌شده مقایسه و چالش‌های موجود بیان و پیشنهادهایی جهت کارهای آینده ارائه می‌شود؛ درنهایت در بخش پنجم مطالب ذکرشده جمع‌بندی می‌شوند.

۲- راه‌کارهای با فرض کانال امن

در این بخش راه‌کارهایی با فرض وجود کانال ارتباطی امن و غیر قابل شنود در بین اعضا مرور می‌شوند. نخستین راه‌کار ارائه‌شده جهت جمع چندسویهٔ امن توسط بن‌اور[1] و همکارانش [1] در سال ۱۹۸۸ ارائه شد که از تسهیم راز شمیر برای محاسبه جمع چندسویه امن استفاده می‌کند و به‌صورت t-private است؛ زیرا با تبانی t بازیکن و یا کمتر از آن قادر به محاسبه داده محرمانهٔ سایر اعضا نخواهند بود. طرف سوم راز را بین اعضا تقسیم و این روش از فرض کانال امن استفاده می‌کند. فرض کنیم n عضو P_1، P_2، ... و P_n به‌ترتیب با مقادیر محرمانه d_1، d_2، ... و d_n داریم. اعضا ورودی‌های محرمانه خود را به تابع F داده و به‌صورت t-private نتیجه را محاسبه می‌کنند. این روش شامل سه مرحله است: مرحله ورودی، مرحله محاسبات، مرحله پایانی. در مرحله ورودی اعضا به‌ترتیب مقدارهای عددی $\{\alpha_0، \alpha_1، \dots، \alpha_{n-1}\}$ را به‌عنوان شناسهٔ عمومی دریافت می‌کنند. هر عضو P_i یک تابع چندجمله‌ای f_i از درجه t با ضرایب تصادفی $a_i، (i = 1، \dots، t)$ می‌سازد، به‌طوری‌که، مقدار ثابت تابع را برابر با ورودی محرمانه خود d_i قرار می‌دهد. رابطه (۱) چند جمله‌ای f_i را نشان می‌دهد. پس هر عضو، مقدار تابع را به‌ازای شناسه عمومی هر عضو گروه α_j محاسبه و برای او ارسال می‌کند ($d_j = f_i(\alpha_j)$).

$$f_i = d_j + a_1 x + \cdots + a_t x^t \qquad (۱)$$

در مرحله محاسبات، مقدار ثابت تابع حاصل از جمع توابع دریافتی f کل اعضا که در مرحله ورودی ایجاد کرده‌اند، برابر جمع ورودی محرمانه اعضا است. درواقع اگر دو تابع $f(x)$ و $g(x)$ به‌ترتیب دارای مقادیر ثابت a و b باشند، اگر تابع $k(x) = f(x) + g(x)$ باشد، مقدار ثابت تابع $k(x)$ برابر $a + b$ است. بنابراین هر عضو P_j مقادیر دریافتی کل اعضا $f(\alpha_j)$ را جمع کرده تا تابع k به‌ازای α_j محاسبه شود ($k(\alpha_j)$). در ادامه باید اعضا با تسهیم راز مقدار ثابت تابع k را به‌دست آورند. با اشتراک هر $t + 1$ عضو، مقدار مجموع محاسبه می‌شود.

در مرحله پایانی که در آن مقدار نهایی تابع F سهم‌های مشترک برای یک عضو و یا همه آشکار می‌شود. اگر در تابع نهایی مقدار متغیر ورودی برابر صفر قرار داده شود، مجموع محاسبه می‌شود. از معایب این راه‌کار هزینهٔ محاسباتی تولید تابع و هزینه ارتباطی ارسال مقدار تابع است. به‌علاوه با تبانی $t + 1$ عضو، دادهٔ محرمانه کاربر افشا می‌شود.

کلیفتون[2] و همکاران [2] در سال ۲۰۰۲ راه‌کاری را برای جمع چندسویهٔ امن با فرض کانال امن به‌عنوان ابزاری در راستای داده‌کاوی ارائه دادند. در این راه‌کار اعضا در یک چیدمان گردشی قرار می‌گیرند؛ یکی از آن‌ها به‌عنوان آغازگر انتخاب می‌شود؛ دادهٔ محرمانه خود را با مقدار تصادفی r جمع می‌زند و حاصل را برای عضو بعدی در حلقه می‌فرستد. عضو دوم، مقدار دریافتی را صرفاً با مقدار محرمانه خود جمع و برای عضو بعدی می‌فرستد؛ روال تا کامل‌شدن دور ادامه می‌یابد و نتیجه نهایی در اختیار آغازگر قرار می‌گیرد؛ وی مقدار تصادفی r را از نتیجه نهایی کم می‌کند و مجموع مقادیر محرمانه به‌دست می‌آید. در این راه‌کار هر دو نفر با تبانی مقدار محرمانه عضو میانی را می‌توانند افشا کنند. بنابراین اگرچه هزینه ارتباطی آن از مرتبه $O(n)$ است و کاربر فقط عملیات جمع انجام می‌دهد، اما در برابر تبانی جزئی حتی دو نفر امن نیست.

شیخ[3] و همکاران در مقاله [3]، پروتکلی تحت عنوان K-Secure Sum به‌منظور بهبود جمع چندسویه امن با فرض

[1] Ben Or

[2] Clifton

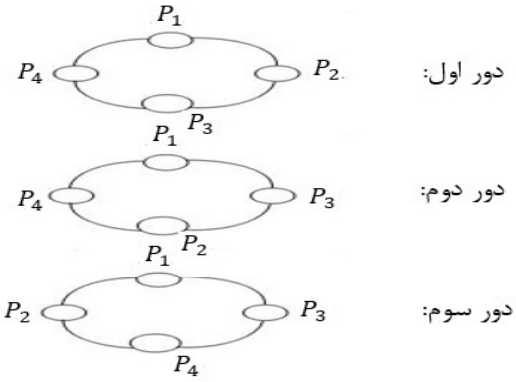
[3] Sheikh

[Downloaded from monadi.isc.org.ir on 2025-06-10]

کانال امن، ارائه دادند. اعضا داده محرمانه خود را به بلوک‌هایی تقسیم می‌کنند. تعداد قطعات کاربران باید با هم برابر باشد(k قطعه). در بهترین حالت امنیتی اعضا دادهٔ محرمانهٔ خود را به تعداد کل اعضا یعنی n قطعه تقسیم می‌کنند؛ سپس روال زیر طی می‌شود:

اعضا در یک چیدمان حلقه قرار می‌گیرند و عضو آغازگر (P_1) یکی از بلوک‌های خود را انتخاب و برای عضو دوم (P_2) ارسال می‌کند؛ عضو دوم آن را با یکی از بلوک‌های خود جمع می‌زند و برای عضو سوم ارسال و به همین ترتیب هر P_i حاصل‌جمع جزئی را برای $P_{(i+1) mod\, n}$ ارسال می‌کند و روال تا کامل‌شدن دور ادامه دارد. در پایان این دور حاصل‌جمع جزئی در اختیار P_1 قرار می‌گیرد. P_1 دور دوم را برای بلوک‌های دوم آغاز و حاصل‌جمع جزئی دور نخست را با بلوک دوم خود جمع و برای عضو دوم ارسال می‌کند و روال مشابه دور نخست تکرار می‌شود. پس از n دور، مجموع نهایی نزد P_1 حاصل می‌شود. به‌منظور ارتقای امنیت روش در هر دور عضو آغازگر، بلوک اولیه را با یک مقدار تصادفی نیز جمع می‌زند؛ پس از محاسبه مجموع حاصل شده از n دور، P_1 باید مجموع اعداد تصادفی اضافه‌شده را از آن کم کند تا جمع داده‌های محرمانه اعضا حاصل شود. این روش به‌علت ثابت‌بودن چیدمان در برابر تبانی جزئی امن نیست و با تبانی دو کاربر در n دور، دادهٔ محرمانه عضو میانی محاسبه می‌شود.

شیخ و همکاران به‌منظور تأمین امنیت بیشتر، نسخه‌های بهبودیافته‌ای از پروتکل قبلی را با نام پروتکل Ck-Secure Sum ارائه کرده‌اند[4]. در پروتکل تحت عنوان Ck-Secure Sum اعضا دادهٔ محرمانه خود را به قطعاتی تقسیم می‌کنند. تعداد قطعات کاربران با هم برابر و در این پروتکل برابر $n-1$ قطعه است. روال مشابه پروتکل [3] است. با این تفاوت که در هر دور عضو دوم P_2 جایگاه خود را با عضو بعدی جابه‌جا می‌کند تا در مکان عضو nام قرار گیرد، بدین ترتیب که پیش از آغاز دور دوم P_2 با P_3 و پیش از دور سوم P_2 با P_4 و به همین ترتیب جابه‌جا می‌شود. در شکل (۱) این روال برای چهار کاربر نشان داده شده است. در پایان، مشابه [3] عضو نخست P_1 نتیجه نهایی را محاسبه و پخش می‌کند. به‌دلیل جابه‌جایی P_2 این راه‌کار در برابر تبانی دو کاربر برای محاسبه داده محرمانه عضو میانی امن است؛ اما در برابر تبانی جزئی بیش از دو کاربر امن نیست؛ زیرا در هر دور فقط P_2 جابه‌جا می‌شود.



(شکل-۱): نمایی از جابه‌جایی اعضا در Ck-Secure Sum [4]

پروتکل بعدی شیخ و همکاران در [5] تحت عنوان Dk-Secure Sum[1] و با فرض کانال امن ارائه شد. اعضا داده محرمانهٔ خود را به قطعاتی تقسیم می‌کنند. تعداد قطعات هر کاربر برابر تعداد کل اعضا، یعنی n قطعه است. در ادامه هر عضو P_i هر قطعه را برای یکی از اعضا ارسال می‌کند؛ به‌طوری‌که درنهایت هر عضو n بلوک داده دارد و یکی از آنها متعلق به خود اوست. روال کار مشابه پروتکل [3] است در دور نخست با شروع از P_1 اعضا مجموع بلوک‌های نخست خود را محاسبه می‌کنند و سپس P_1 دور بعدی را آغاز می‌کند. پس از n دور مجموع محاسبه می‌شود و P_1 آن را پخش می‌کند. در این پروتکل نیازی به افزودن مقادیر تصادفی و جابه‌جایی نیست. این راه‌کار در برابر تبانی جزئی تا سطح $n-2$ امن است؛ اما با فرض وجود کانال امن هزینهٔ ارتباطی آن از مرتبهٔ $O(n^2)$ است.

یوون[2] و همکاران در مقاله [6] راه‌کار $CR-SSP$[3] را در مدل شبه‌درستکار و با فرض وجود کانال امن ارائه داده‌اند این پروتکل با شرط $n \geq 4$ به‌طور قطع در برابر تبانی دو کاربر برای به‌دست‌آوردن داده کاربر میانی امن است. n کاربر در چیدمان حلقه قرار می‌گیرند. روال کار در دو مرحله انجام می‌شود: مرحلهٔ ۱) پوشش دادهٔ محرمانه، مرحلهٔ ۲) محاسبه مجموع.

مرحلهٔ پوشش داده محرمانه شامل دو مرحله است: در مرحله نخست، هرکاربر P_i، $n-1$ عدد تصادفی r_{ij} (j = 1،2، ... ،$n-1$) را تولید کرده و عدد r_{ij} را به‌طور محرمانه برای کاربر P_j ارسال می‌کند. کاربر P_i مقدار m_i را برابر d_i (دادهٔ محرمانه P_i) قرار می‌دهد ($m_i = d_i$). در مرحلهٔ دوم، کاربر P_j پس از دریافت r_{ij} به‌طور تصادفی آن را از m_j کم و

[1] Distributed k-Secure Sum
[2] Youwen
[3] Collusion-Resisting Secure Sum Protocol

[Downloaded from monadi.isc.org.ir on 2025-06-10]

یـا بـه آن اضـافه مـی‌کنـد و بـه‌طور محرمانـه P_i را از عمـل کم‌کردن و یا اضافه‌کردن مقـدار ارسـالی آگـاه مـی‌سـازد؛ P_i عکس عمل انجام‌شده را برروی m_i خود انجام می‌دهد.

در مرحلهٔ محاسبه مجموع، عضو نخسـت حلقـه (P_1) مقدار m_i حاصل از مرحلهٔ نخست را با عدد تصادفی r جمـع و برای عضـو دوم حلقـه ارسـال مـی‌کنـد؛ عضـو دوم مقـدار دریافتی را با مقدار محرمانه نزد خود جمع می‌زند و حاصـل را برای عضو سوم می‌فرستد. روال تـا رسـیدن بـه عضـو nام ادامه دارد. در پایان نتیجه برای عضو نخست ارسال می‌شود و او مقدار r را از مجموع به‌دست‌آمده کم کرده و نتیجه نهـایی را پخش همگانی می‌کند. در این راه‌کار اگر در مرحلهٔ نخست هر کاربر $n-1$ عدد تصادفی تولید کند در برابر تبانی جزئی امن است در غیر ایـن‌صـورت امنیـت در برابـر تبـانی جزئـی کاهش می‌یابد. بنابراین برای امنیت بیشتر هزینه ارتبـاطی را افزایش می‌یابد؛ زیرا امنیت در برابر تبـانی جزئـی، بـه تعـداد اعداد تصادفی تولیدشده و به اشـتراک گذاشته‌شـده وابسـته است.

در سال ۲۰۱۱ جانگدی[1] و همکـارانش در مقالـه [7] راه‌کاری ترکیبی از مدل واقعی و ایده‌آل و با فرض کانال امن ارائه دادند. در مدل ایده‌آل یک طـرف سـوم معتمـد[2] وجـود دارد کـه اعضـا بـه‌طور محرمانـه داده‌هـای خـود را بـرای او می‌فرستند و طرف سوم همه محاسبات را انجام می‌دهـد. در مدل واقعی از طرف سوم غیر معتمـد اسـتفاده مـی‌شـود. در راه‌کار مقاله اعضای P_1، P_2، ... و P_n باید داده محرمانه خـود را به بلوک‌هایی تقسیم کنند. در ابتدای کار تعـداد بلـوک‌هـا به‌طور آشکار تعیین می‌شود.

در مقالـه [7] بـه‌منظور توضـیح راه‌کـار، اعضـا دادهٔ محرمانهٔ خود را به ۳ بلوک تقسیم می‌کنند؛ و برای هر بلوک یک مقدار تصادفی در نظر می‌گیرند؛ D_{ij}، بلوک j ام کاربر P_i و r_{ij}، عدد تصادفی بلوک j ام از کـاربر P_i اسـت. هـر عضـو، مجموع بلوک نخست به همـراه مقـدار تصـادفی نظیـر آن را برای طرف سوم می‌فرستد ($D_{i1}+r_{i1}$). طـرف سـوم پـس از دریافت مقادیر کل اعضا، مجموع بلوک‌های نخست و مقـادیر تصادفی ($D_{i1}+r_{i1}$) را محاسبه و برای عضـو نخسـت (P_1) می‌فرستد. P_1 مقدار تصادفی جمع‌شده با بلوک نخسـت دادهٔ محرمانهٔ خود را کم (r_{11}) و بلوک دوم دادهٔ محرمانه و مقـدار تصادفی مربوط به آن را اضافه ($D_{i2}+r_{i2}$)، و حاصل را برای عضو دوم ارسال می‌کند. روال حذف مقدار تصادفی نخست و اضافه‌کردن بلـوک دوم و مقـدار تصـادفی مربـوط بـه آن تـا رسیدن به عضو آخر ادامه می‌یابد.

پس از انجام روال ذکرشده توسط عضـو آخـر، مقـدار ($\sum_{i=1}^{n}(D_{i1}+D_{i2}+r_{i2})$ محاسبه شده اسـت. عضـو آخـر، مجموع حاصل‌شده در این مرحله را به عضـو مـا قبـل خـود مـی‌دهـد؛ وی مقـدار تصـادفی بلـوک دوم خـود را حـذف ($r_{(n-1)2}$) و بلـوک سـوم و مقـدار تصـادفی آن را اضـافه ($D_{(n-1)3}+r_{(n-1)3}$) و نتیجـه را بـرای عضـو ماقبـل خـود ارسال می‌کند؛ این روال تا رسـیدن بـه عضـو نخسـت ادامـه دارد. عضو نخست مجموع جزئی را بـه طـرف سـوم و طـرف سوم به عضو آخر می‌دهد؛ عضو آخر مقدار تصادفی دوم خود را حذف ($\boldsymbol{r_{n2}}$) و بلوک سوم دادهٔ خـویش (D_{n3}) را اضـافه و نتیجه را برای عضو ماقبل ارسال می‌کند. عضو آخر به بلـوک سوم، مقدار تصادفی اضافه نمی‌کند. در ایـن مرحلـه مجمـوع جزئـی به‌دسـت‌آمده برابـر $\sum_{i=1}^{n}(D_{i1}+D_{i2}+D_{i3})+$ $\sum_{i=1}^{n-1}r_{i3}$ است؛ سپس هر عضـو P_i ($i\neq n$)، تـا رسـیدن بـه عضو نخست، مقدار تصادفی سوم خود را (r_{i3})، حذف و برای عضو ماقبل ارسال می‌کند. عضو نخست، پس از حذف مقـدار تصـادفی سـوم خـود ($\boldsymbol{r_{13}}$)، مجمـوع نهـایی ($\sum_{i=1}^{n}(D_{i1}+$ $D_{i2}+D_{i3})$) را محاسبه و برای طـرف سـوم مـی‌فرسـتد تـا طـرف سـوم آن را پخـش همگـانی کنـد. در ایـن روش دادهٔ محرمانه به بلوک‌هایی تقسیم و همراه هر بلوک عدد تصادفی نیز ارسال می‌شود. با انجام حملهٔ تبانی، بلوک داده به همـراه عدد تصادفی مربوط به آن آشکار مـی‌شـود و دادهٔ محرمانـه، مخفی باقی می‌ماند؛ بنـابراین در برابـر تبـانی جزئـی اعضـا و طرف سوم امن است. این روش به طـرف سـوم معتمـد نیـاز دارد و هزینه ارتباطی و محاسباتی آن $O(n)$ است.

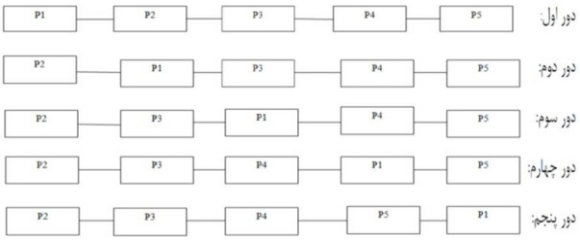
راوتـاری[3] و همکـاران در سـال ۲۰۱۳ [8] پروتکـل Distributed RK Secure Sum را با فرض کانـال امـن ارائـه دادنـد. ایـن پروتکـل بـه‌منظور افـزایش کـارایی، اعضـا را در چیدمان باس قرار می‌دهد. کاربران P_1، P_2، ... و P_n هر یـک دادهٔ خود را به $n-1$ بلوک تقسیم و نزد خود نگه‌مـی‌دارنـد. در دور نخست عضو نخست (P_1) بلوک نخسـت از دادهٔ خـود را برای عضو دوم می‌فرستد؛ عضو دوم (P_2) نیز پس از جمـع بلوک نخست داده خود بـا مقـدار دریـافتی از عضـو نخسـت، حاصل را برای عضو سوم ارسال می‌کند روال تـا رسـیدن بـه عضو n ام ادامه دارد، P_n حاصل‌جمـع جزئـی دور نخسـت را نزد خود نگه‌می‌دارد؛ در دور دوم عضو دوم P_2 با عضـو سـوم P_3 جابه‌جا و روال دور نخست برای بلوک دوم تکرار می‌شود.

[1] Jangde
[2] Trusted Third Party (TTP)
[3] Rautaray

[Downloaded from monadi.isc.org.ir on 2025-06-10]

در دور سوم P_2 با P_4 جابه‌جا می‌شود و همین روال برای $n-1$ دور تکرار می‌شود. در دور $n-1$ام کاربر P_2 با P_n جابه‌جا و کاربر P_n حاصل‌جمع‌های دورهای قبلی را با بلوک $n-1$ام خود و مقدار دریافتی از کاربر P_{n-1} جمع زده و برای P_2 ارسال می‌کند و P_2 مقدار دریافتی از P_n را با بلوک $n-1$ ام خود جمع کرده و بدین طریق حاصل‌جمع $n-1$ بلوک n کاربر محاسبه می‌شود. به‌دلیل جابه‌جایی کاربران در برابر تبانی دو کاربر علیه کاربر میانی امن است؛ اما در برابر تبانی جزئی امن نیست؛ زیرا در هر دور فقط کاربر P_2 با عضوی دیگر جابه‌جا می‌شود.

راوتاری و همکاران در سال ۲۰۱۳ پروتکل قبلی خود را بهبود داد و پروتکل Modified Distributed RK Secure Sum [9] را با فرض کانال امن ارائه دادند. بدین طریق که اعضا داده خود را به n بلوک تقسیم می‌کنند و پیش از شروع هر دور غیر از دور نخست، عضو نخست (P_1) با سایر اعضا جابه‌جا می‌شود. به‌طوری‌که در دور nام در جایگاه P_n قرار می‌گیرد و پس از n دور مجموع در نزد P_1 محاسبه می‌شود و P_1 آن را پخش همگانی می‌کند. نمایی از جابه‌جایی کاربر P_1 در شکل (۲) نشان داده شده است. مابقی روال پروتکل مشابه مقاله [8] است. در این پروتکل احتمال نشت اطلاعات کاهش یافته اما در برابر تبانی جزئی امن نیست.



(شکل-۲): نمایی از جابه‌جایی کاربران در روش Modified [9] Distributed RK Secure Sum

سپس راوتاری و همکاران [10]، در سال ۲۰۱۳ راه‌کاری را تحت عنوان Distributed Database RK secure sum با فرض کانال امن ارائه دادند. در این روش اعضا داده محرمانهٔ خود را به n بلوک تقسیم می‌کنند؛ سپس، هر عضو بلوک‌های خود را برای سایر اعضا ارسال می‌کند بدین طریق که برای هر عضو یک بلوک ارسال می‌کند؛ بنابراین پس از توزیع بلوک توسط کل اعضا، هر عضو n بلوک دارد و یکی از آن‌ها متعلق به خودش است.

در ادامه پس از قرارگرفتن اعضا در چیدمان باس، نخستین عضو در توپولوژی (P_1)، یک بلوک را برای عضو دوم P_2 ارسال می‌کند، P_2 آن را با بلوک نخست خود جمع زده و برای عضو سوم می‌فرستد، روال تا رسیدن به عضو n ام (P_n) ادامه می‌یابد و P_n مجموع جزئی حاصل از بلوک‌های نخست را نزد خود نگه‌می‌دارد. P_1 دور دوم را برای جمع بلوک‌های دوم مشابه دور نخست آغاز می‌کند. روال برای n دور و جمع n بلوک ادامه می‌یابد و درنهایت P_n مجموع محاسبه‌شده را پخش همگانی می‌کند. این راه‌کار در برابر تبانی جزئی تا سطح $n-2$ نفر امن است؛ اما هزینهٔ ارتباطی آن با فرض وجود کانال امن از مرتبهٔ $O(n^2)$ است.

جهان[1] و همکارن پروتکل DRPM[2] را در سال ۲۰۱۵ در مقاله [11] در مدل مهاجم شبه‌درستکار ارائه دادند که از طرف سوم قابل اعتماد و بلوک‌بندی تصادفی داده استفاده می‌کند. این راه‌کار شامل دو بخش است:

در بخش نخست، اعضا دادهٔ محرمانهٔ خود را به تعداد مشخص بلوک m تقسیم می‌کنند. هر کاربر P_i یک آرایه از اعداد تصادفی با طول محرمانه و کمتر یا مساوی m را تولید می‌کند (R_i).

در بخش دوم، برای هر کاربر P_i یک پرچم درنظر گرفته می‌شود ($Flag[i]$). اعضا به‌طور موازی، مجموع بلوک نخست خود و مقدار تصادفی مربوط به آن و یا بدون استفاده از مقدار تصادفی را برای طرف سوم معتمد می‌فرستند و طرف سوم مجموع بلوک‌های نخست و اعداد تصادفی را محاسبه می‌کند که آن را با $value$ نشان می‌دهیم.

اعضا درصورتی‌که از عددتصادفی استفاده کرده باشند، مقدار پرچم مربوط به خود را $true$ و درصورت عدم استفاده از مقدار تصادفی مقدار پرچم را $false$ قرار می‌دهند؛ سپس طرف سوم مقدار $value$ را به‌ترتیب برای اعضا ارسال می‌کند. طرف سوم به همراه ارسال $value$ برای کاربر P_i، پرچم نظیر کاربر ($Flag[i]$) را نیز برای او ارسال می‌کند. هر عضو P_i پس از دریافت $value$ درصورتی‌که مقدار پرچم به‌ازای او $true$ باشد، باید مقدار تصادفی را که در مرحله قبل اضافه کرده از $value$ کم کند و سپس درصوت تمایل به استفاده از عدد تصادفی، مجموع بلوک دوم و عدد تصادفی دیگری را که انتخاب می‌کند به $value$ اضافه و دوباره $Flag[i]=true$ قرار می‌دهد؛ در صورت عدم تمایل به استفاده از عددتصادفی، فقط بلوک دوم را به $value$ اضافه کرده و $Flag[i]=false$ قرار می‌دهد و $value$ را برای

[1] Jahan
[2] Double Random Partitioned Model

[Downloaded from monadi.isc.org.ir on 2025-06-10]

طرف سوم ارسال می‌کند؛ تا طرف سوم آن را برای سایر اعضا ارسال کند. همین روال برای m بلوک تکرار می‌شود.

به منظور حذف مقدار تصادفی ارسالی در مجموع نهایی، هر عضو هنگام ارسال بلوک بعدی، مقدار تصادفی بلوک ماقبل را از آن کم می‌کند و از طرفی مقدار $Flag[i]$ برای طرف سوم آشکار است و اگر عضوی مقدار تصادفی داشته که هنوز حذف نشده است، باید مقدار $value$ برای او ارسال شود تا مقدار تصادفی را حذف کند و بدین طریق مجموع نهایی محاسبه می‌شود.

در این پروتکل اعضا با هم ارتباط ندارند و طرف سوم درستکار است؛ و طرف سوم فقط مقدار $value$ را در بین اعضا ارسال می‌کند و بدین طریق اعضا از تغییرات $value$ مطلع می‌شوند؛ اما از استفاده و یا عدم استفاده از عددتصادفی توسط یکدیگر اطلاع ندارند و قادر به حدس مقدار محرمانه یکدیگر نیستند. این راه‌کار به طرف سوم معتمد نیاز دارد. درصورتی‌که طرف سوم بدخواه باشد و با اعضا در تبانی شرکت کند، چون اختیارات او زیاد است، امنیت کل اعضا به خطر می‌افتد. هزینه ارتباطی و محاسباتی راه‌کار پایین و از مرتبه $O(n)$ است.

۳- راه‌کارهای با فرض کانال ناامن

در این بخش راه‌کارهایی مرور می‌شوند که فرض می‌کنند کانال ارتباطی بین اعضای گروه قابل شنود است؛ در این راه‌کارها باید امنیت دادهٔ کاربران در حین ارسال مقادیر در کانال حفظ شود.

جانگ[1] و همکاران در سال [12] پروتکلی را بدون نیاز به کانال امن، بدون طرف سوم و در مدل شبه‌درستکار ارائه داده است. اساس آن مسئلهٔ سخت دیفی هلمن محاسباتی (CDH[2]) است. اعضا P_1، P_2، ... و P_n به‌ترتیب دادهٔ محرمانهٔ d_1، d_2، ... و d_n را دارند که هر $d_i \in Z_p$ است. این مقاله روشی برای محاسبهٔ مجموع مقادیر محرمانه ($f(d) = \sum_{i=1}^{n} d_i$) و روشی دیگر برای محاسبهٔ ضرب این مقادیر ($f(d) = \prod_{i=1}^{n} d_i$) راه‌کار ارائه داده است. هر روش در دو مدل ارائه شده است: مدل یک تجمیع‌کننده[3]، مدل فقط اعضا[4].

در مدل یک تجمیع‌کننده، داده‌ها برای عضوی شبه‌درستکار مانند A ارسال می‌شود و فقط او $f(d)$ را محاسبه می‌کند. در مدل فقط اعضا، n کاربر حضور دارند و عضوی مانند A وجود ندارد.

دو عدد نخست بسیار بزرگ p و q انتخاب می‌شوند؛ به‌طوری‌که $q \mid p-1$. گروه گردشی G_1 از مرتبه q و با مولد g_1 با شرط رابطه (۲) انتخاب می‌شود. در رابطه (۲) مقدار $h \in_R Z_p$ است.

$$g_1 = h^{(p-1)/q} mod\ p\ , g_1 \neq 1\ mod\ p \qquad (۲)$$

سپس گروه گردشی G_2 از مرتبه q و با مولد g_2 با شرط $g_2 = {g_1}^p mod\ p^2$ انتخاب می‌شود. پروتکل ضرب و سپس پروتکل جمع در هردو مدل توضیح داده می‌شوند.

پروتکل ضرب: اعضا در چیدمان گردشی قرار می‌گیرند. در مدل یک تجمیع‌کننده و مدل فقط اعضا هر عضو P_i عدد تصادفی $r_i \in Z_q$ را انتخاب و مقدار $Y_i = {g_1}^{r_i} \in G_1$ را برای دو عضو کناری خود در حلقه (P_{i+1} و P_{i-1}) ارسال می‌کند؛ سپس مقدار R_i را طبق رابطه (۳) محاسبه می‌کند.

$$R_i = (Y_{i+1}/Y_{i-1})^{r_i} = ({g_1}^{r_{i+1}}/{g_1}^{r_{i-1}})^{r_i} \in G_1 \qquad (۳)$$

سپس هر عضو P_i دادهٔ محرمانه خود را به‌صورت رابطه (۴) رمز می‌کند.

$$C_i = d_i . R_i = x_i . ({g_1}^{r_{i+1}}/{g_1}^{r_{i-1}})^{r_i}\ mod\ p \qquad (۴)$$

در مدل اعضا، هر عضو P_i پس از محاسبه C_i آن را پخش همگانی می‌کند؛ سپس هر کاربر با انجام محاسبات رابطه (۵) قادر به محاسبه حاصل‌ضرب است($r_{n+1} = r_n$ و $r_0 = r_1$).

$$\begin{aligned}\prod_{i=1}^{n} C_i &= \prod_{i=1}^{n} d_i . ({g_1}^{r_{i+1}}/{g_1}^{r_{i-1}})^{r_i}\ mod\ p \\ &= \prod_{i=1}^{n} d_i \prod_{i=1}^{n} ({g_1}^{r_{i+1}}/{g_1}^{r_{i-1}})^{r_i}\ mod\ p \\ &= \prod_{i=1}^{n} d_i \, . \, {g_1}^{\sum_{i=1}^{n}(r_{i+1}r_i - r_{i-1}r_i)}\ mod\ p \\ &= \prod_{i=1}^{n} d_i \, . \, {g_1}^{0}\ mod\ p = \prod_{i=1}^{n} d_i\ mod\ p\end{aligned} \qquad (۵)$$

در مدل یک تجمیع‌کننده، چون کانال ناامن است عضو A به‌عنوان کاربر P_{n+1} محسوب می‌شود و مقدار Y_{n+1} را برای عضو P_1 و P_n ارسال و مقدار R_{n+1} را محاسبه می‌کند؛ اما دادهٔ محرمانه ندارد و C_i را محاسبه نمی‌کند؛ و

[1] Jung
[2] Computational Diffie-Hellman problem
[3] One Aggregator Model
[4] Participants Only Model

[Downloaded from monadi.isc.org.ir on 2025-06-10]

سایر اعضا مقدار C_i را برای A ارسال می‌کنند و A با ضرب کل مقادیر به‌صورت $R_{n+1}\prod_{i=1}^{n}C_i=\prod_{i=1}^{n}x_i \ mod\ p$ حاصل‌ضرب مقادیر را محاسبه و پخش می‌کند.

پروتکل جمع: اعضا در چیدمان حلقه قرار گرفته و هر عضو P_i مقدار Y_i و سپس R_i را محاسبه می‌کند. با این تفاوت که اعداد عضو گروه G_2 هستند؛ اما به‌منظور محاسبه مجموع رابطه (۶) درنظرگرفته می‌شود.

$$(1+p)^m=\sum_{i=0}^{m}\binom{m}{i}p^i \\ =1+mp\ mod\ p^2 \qquad (۶)$$

بنابراین اگر d_i جایگزین m شود، $f(d)=\sum_{i=1}^{n}d_i$ را طبق رابطه (۷) می‌توان محاسبه کرد.

$$\prod_{i=1}^{n}(1+p)^{d_i}=\prod_{i=1}^{n}(1+d_ip) \\ =\left(1+p\sum_{i=1}^{n}d_i\right)\ mod\ p^2=c \qquad (۷)$$

چون کانال ناامن است، کاربر P_i مقدار C_i را با انجام عملیات (۸) محاسبه می‌کند.

$$C_i=(1+d_ip).R_i= \\ (1+d_ip).(g_1{}^{r_{i+1}}/g_1{}^{r_{i-1}})^{r_i}\ mod\ p^2 \qquad (۸)$$

در مدل فقط اعضا، کاربر P_i مقدار C_i را پخش همگانی می‌کند؛ سپس هر کاربر P_i با ضرب مقادیر ارسالی کل اعضا مقدار C که در رابطه (۹) نشان داده شده است، محاسبه می‌کند.

$$C=\prod_{i=1}^{n}C_i=\prod_{i=1}^{n}(1+d_ip).(g_1{}^{r_{i+1}}/g_1{}^{r_{i-1}})^{r_i} \\ =\left(1+p\sum_{i=1}^{n}d_i\right).g_1{}^{0} \\ =\left(1+p\sum_{i=1}^{n}d_i\right)\ mod\ p^2 \qquad (۹)$$

پس از محاسبه C، مجموع مقادیر محرمانه با انجام رابطه (۱۰) محاسبه می‌شود.

$$\frac{c-1}{p}=\sum_{i=1}^{n}d_i \qquad (۱۰)$$

در مدل یک تجمیع‌کننده، عضو A به‌عنوان کاربر P_{n+1} محسوب می‌شود و مقدار R_{n+1} را محاسبه می‌کند؛ اما دادهٔ محرمانه ندارد و C_i را محاسبه نمی‌کند؛ و سایر اعضا مقدار C_i را برای A ارسال می‌کنند و A با ضرب کل مقادیر به‌صورت $R_{n+1}\prod_{i=1}^{n}C_i=C\ mod\ p^2$ مقدار C را محاسبه و با انجام رابطه (۱۰) مقدار $\sum_{i=1}^{n}d_i$ را محاسبه پخش می‌کند.

در این روش دو کاربر P_{i+2} و P_i با هماهنگی در انتخاب r_{i+2} و r_i به‌صورت $r_i=r_{i+2}-a$ قادر به محاسبه $R_{i+1}=(g_1{}^{a})^{r_{i+1}}$ و محاسبهٔ دادهٔ محرمانه P_{i+1} است. P_{i+2} و P_i مقدار R_{i+1} را با انجام $Y_i{}^{a}=(g_1{}^{r_i})^{a}$ محاسبه می‌کنند.

برای مقابله با تبانی دو کاربر، مقدار R_i با دخیل‌کردن مقادیر بیش از دو نفر محاسبه می‌شود. برای مقاومت در برابر تبانی k نفر R_i با انجام رابطه (۱۱) محاسبه می‌شود.

$$R_i \\ =(g_1{}^{r_{i+k+1}}/g_1{}^{r_{i-1}})^{r_kr_{k-1}\dots r_{i+1}r_i} \\ \in G_1 \qquad (۱۱)$$

این روش نیازی به کانال امن ندارد؛ ولی به‌دلیل انجام نمارسانی برای هر کاربر، هزینه محاسباتی بالایی دارد و از طرفی برای افزایش امنیت در برابر تبانی جزئی هزینه محاسباتی کاربران بسیار افزایش می‌یابد؛ به‌طوری‌که به‌منظور داشتن امنیت در برابر تبانی جزئی تا سطح $n-2$ نفر هزینهٔ محاسباتی از مرتبهٔ $O(n^2)$ است.

عاشوری و همکاران درسال ۲۰۱۶ [13] با فرض عدم وجود کانال امن سه پروتکل را با نیازمندی‌های متفاوت برای جمع چندسویهٔ امن در مدل شبه درستکار ارائه دادند.

گروه پیلیه G $(Z_{N^2}^{*})$ با مولد تصادفی $g\in G$ و مولد خاص $g_s=1\ mod\ N$ درنظرگرفته می‌شود. تجزیهٔ عوامل نخست N برای همگان مجهول است. به‌ازای g حل مسئله دیفی هلمن تصمیمی[1] از لحاظ محاسباتی ناممکن و به‌ازای g_s مسئلهٔ لگاریتم گسسته[2] قابل حل است.

اعضای P_1 ، P_2 ، ... و P_n به‌ترتیب دادهٔ محرمانه d_1 ، d_2 ، ... و d_n را داشته و برروی (g_s,g,G) توافق دارند. پروتکل‌های ارائه‌شده به شرح زیر است:

SECURESUM V-1: در این پروتکل به‌منظور حفظ محرمانگی دادهٔ کاربران، اعضای گروه شبکهٔ وتوی گم‌نام [14] راه‌اندازی می‌کنند. پروتکل شامل دو مرحله است: در مرحلهٔ نخست، هر کاربر P_i عدد تصادفی $a_i\in_R G$ را انتخاب و g^{a_i} را پخش همگانی می‌کند؛ سپس $=g^{b_i}$

[1] Decisional Diffie-Hellman (DDH)
[2] Discrete Logarithm (DL)

[Downloaded from monadi.isc.org.ir on 2025-06-10]

$(\prod_{j=1}^{i-1} g^{a_j}/\prod_{j=i+1}^{n} g^{a_j})$ و $g^{a_i b_i}$ را محاسبه و نزد خود نگه‌می‌دارد.

در مرحلهٔ دوم، هر کاربر P_i مقدار $W_i = g^{a_i b_i} g_s^{d_i}$ را محاسبه و پخش همگانی می‌کند. با ضرب مقادیر ارسالی گروه ماسک شبکهٔ وتوی گمنام از بین می‌رود و $\prod_i W_i =$ $g_s^{\sum_i d_i} \bmod N^2$ حاصل می‌شود. اگر $g_s = 1 + kN$ باشد، مجموع $\sum_{i=1}^{n} d_i$ طبق رابطهٔ (۱۲) قابل محاسبه است.

$$\sum_{i=1}^{n} d_i = \frac{g_s^{\sum_i d_i} - 1}{kN} \qquad (۱۲)$$

پروتکل SECURESUM V-1 به‌دلیل استفاده از شبکهٔ وتوی گم‌نام در برابر تبانی جزئی تا سطح $n-2$ نفر امن است و هزینه محاسباتی آن با نادیده‌گرفتن محاسبات کم‌هزینه به‌ازای کل کاربران $2n$ نمارسانی است.

SECURESUM V-2: پروتکل دوم، از شبکهٔ وتوی گم‌نام برای محرمانگی دادهٔ کاربران و از پروتکل اشتراک کلید کنفرانس BD [15] به‌منظور محرمانگی حاصل‌جمع در برابر مهاجم بیرونی استفاده می‌کند. این پروتکل نیز شامل سه مرحله است:

در مرحلهٔ نخست، هر کاربر P_i دو عدد تصادفی $a_i، e_i \in_R G$ را انتخاب و ($g^{a_i}، g^{e_i}$) را محاسبه و پخش همگانی می‌کند. در مرحلهٔ دوم، هر کاربر P_i مقدار $t_i = (g^{e_{i+1}}/g^{e_{i-1}})^{e_i}$ را محاسبه و پخش همگانی می‌کند. سپس $g^{b_i} = (\prod_{j=1}^{i-1} g^{a_j}/\prod_{j=i+1}^{n} g^{a_j})$ و $g^{a_i b_i}$ و کلید کنفرانس $k =$ $(g^{e_{i-1}})^{n e_i}. t_i^{n-1}. t_{i+1}^{n-2}. \dots . t_{i-2} = g^{\sum_{i=1}^{n} e_{i-1} e_i} =$ $g^{e_1 e_2 + e_2 e_3 + \cdots + e_n e_1}$ را محاسبه و نزد خود نگه‌می‌دارد.

در مرحلهٔ سوم، هر کاربر P_i مقدار $W_i =$ $g^{a_i b_i} g^{e_{i-1} e_i} g_s^{d_i}$ را پخش همگانی می‌کند. با ضرب مقادیر ارسالی گروه ماسک شبکه از بین می‌رود و $\prod_i W_i =$ $k g_s^{\sum_i d_i} \bmod N^2$ حاصل می‌شود. فقط اعضای گروه مقدار کلید کنفرانس k را می‌دانند و با حذف آن طبق رابطهٔ (۱۲) قادر به محاسبهٔ مجموع $\sum_{i=1}^{n} d_i$ هستند. پروتکل SECURESUM V-2 در برابر تبانی جزئی تا سطح $n-2$ نفر امن است، محرمانگی حاصل‌جمع را حفظ می‌کند و هزینه محاسباتی آن با نادیده‌گرفتن محاسبات کم‌هزینه به‌ازای کل کاربران $5n$ نمارسانی است.

SECURESUM V-3: این پروتکل محرمانگی دادهٔ کاربر و نتیجهٔ حاصل‌جمع حفظ می‌کند و شبکهٔ وتوی گمنام را با پروتکل اشتراک کلید کنفرانس BD ترکیب می‌کند؛ به‌طوری‌که از ضرب مقادیر شبکهٔ وتوی گمنام کلید کنفرانس حاصل می‌شود. در این پروتکل کلید مشترک گروه برابر $k = \prod_i g^{a_i b_i} = g^{\sum a_i b_i} = g^{2\sum_{i=1}^{n} a_{i-1} a_i} =$ $g^{2(a_1 a_2 + a_2 a_3 + \cdots + a_n a_1)}$ است. این پروتکل شامل سه مرحله است:

در مرحلهٔ نخست، هر کاربر P_i عدد تصادفی $a_i \in_R G$ را انتخاب و g^{a_i} را پخش همگانی می‌کند. در ادامه و در مرحلهٔ دوم هر کاربر P_i مقدار $t_i = (g^{a_{i+1}}/g^{a_{i-1}})^{a_i}$ را محاسبه و پخش همگانی می‌کند؛ سپس $\boldsymbol{g^{b_i}}$ را طبق رابطهٔ (۱۳) و $g^{a_i b_i}$ را محاسبه و نزد خود نگه‌می‌دارد.

$$g^{b_i} = g^{a_{i+1}} g^{a_{i-1}} \prod_{j=1, \neq i, i+1, i-1}^{n} g^{(sign(i-j) a_j)} \qquad (۱۳)$$

هر کاربر P_i کلید را محاسبه می‌کند:

$$k = k_i = (g^{a_{i-1}})^{2n} \ . t_i^{2n-1}. t_{i+1}^{2n-2}. \dots . t_{i-2}^{2}.$$

در مرحلهٔ سوم، هر کاربر P_i مقدار $W_i = g^{a_i b_i} g_s^{d_i}$ را محاسبه و پخش همگانی می‌کند. با ضرب مقادیر ارسالی گروه ماسک شبکه تبدیل به کلید کنفرانس k شده و مجموع رمزشده مطابق با رابطهٔ (۱۴) حاصل می‌شود.

$$\begin{aligned} \prod_i W_i &= \prod_i g^{a_i b_i} g_s^{d_i} \\ &= \prod_i g^{a_i b_i} \prod_i g_s^{d_i} \\ &= g^{\sum a_i b_i} g_s^{\sum_i d_i} \\ &= k g_s^{\sum_i d_i} \bmod N^2 \end{aligned} \qquad (۱۴)$$

فقط اعضای گروه قادر به محاسبهٔ k هستند و پس از حذف آن طبق رابطهٔ (۱۲) قادر به محاسبهٔ مجموع $\sum_{i=1}^{n} d_i$ هستند. این پروتکل به‌دلیل آشکاربودن t_i در برابر تبانی جزئی تا سطح $n-4$ نفر امن است و محرمانگی حاصل‌جمع را حفظ می‌کند. هزینه محاسباتی آن با نادیده‌گرفتن محاسبات کم‌هزینه به‌ازای هر کاربر 4 نمارسانی و به‌ازای کل کاربران $4n$ نمارسانی است. در سه نسخه پروتکل از عملیات نمارسانی استفاده می‌شود و مدل مهاجم شبه‌درستکار است و مجهول‌بودن تجزیهٔ عوامل نخست N برای همگان فرض سنگینی است.

راه‌کار بعدی توسط عزیزی و همکاران در سال ۲۰۱۷ در مدل بدخواه ارائه شده است [16]. در این پروتکل مشابه روش جانگ و همکاران [12] دو عدد نخست بسیار بزرگ p و q انتخاب می‌شوند؛ به‌طوری‌که $q | p-1$. گروه گردشی G_1 از مرتبه q و با مولد

[Downloaded from monadi.isc.org.ir on 2025-06-10]

$g_1 \neq 1 \bmod p$ ، $g_1 = h^{(p-1)/q} \bmod p$ ($h \in_R Z_p$) و سپس گروه گردشی G_2 از مرتبه q و با مولد $g_2 = g_1^p \bmod p^2$ انتخاب می‌شود. راه‌کار شامل دو مرحله است: در مرحلهٔ نخست اعضای گروه شبکهٔ وتوی گم‌نام [14] راه‌اندازی می‌کنند و با استفاده از پروتکل توافق کلید کنفرانس BD [15] کلید جلسه به اشتراک می‌گذارند. بنابراین، هر کاربر P_i دو عدد تصادفی $a_i, e_i \in_R G$ را انتخاب و (g^{a_i}, g^{e_i}) را محاسبه و پخش همگانی می‌کند؛ سپس هر کاربر P_i مقدار $t_i = (g^{e_{i+1}}/g^{e_{i-1}})^{e_i}$ را محاسبه و پخش همگانی می‌کند. به‌منظور مقابله با مهاجم بدخواه، کاربر P_i اثبات صفردانش $a_i, e_i \in_R G$ را نیز ارسال می‌کند؛ سپس کاربر P_i در صورت درستی مقادیر ارسالی سایر اعضای گروه $g^{b_i} = (\prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^{n} g^{a_j})$ و $g^{a_i b_i}$ و کلید کنفرانس $k = (g^{e_{i-1}})^{ne_i} . t_i^{n-1} . t_{i+1}^{n-2} . \dots . t_{i-2}$ $g^{\sum_{i=1}^{n} e_{i-1} e_i} = g^{e_1 e_2 + e_2 e_3 + \cdots + e_n e_1}$ را محاسبه و نزد خود نگه‌می‌دارد.

در مرحلهٔ دوم، هر کاربر P_i مقدار $w_i = (1 + d_i p) g^{e_{i-1} e_i} g^{a_i b_i} \bmod p^2$ را به همراه اثبات صفر دانش آن پخش همگانی می‌کند؛ سپس هر کاربر P_i درصورت صحت w_i ها آن‌ها را درهم ضرب می‌کند و ماسک شبکه از بین می‌رود و $c = (1 + p \sum_{i=1}^{n} d_i) \times k$ حاصل می‌شود؛ سپس اعضای گروه قادرند، $\sum_{i=1}^{n} d_i = \frac{(C \times k^{-1}) - 1}{p}$ را به‌دست آورند. این راه‌کار در برابر تبانی جزئی تا سطح $n-2$ نفر امن است. پیمانهٔ محاسباتی یک عدد نخست بسیار بزرگ است و در مقایسه با راه‌کار [13] نیازی به انتخاب عدد مرکب N نیست. هزینه محاسباتی آن بدون درنظرگرفتن اثبات‌های صفردانش برای مقابله با مهاجم بدخواه $5n$ نمارسانی و با درنظرگرفتن هزینه اثبات صفردانش شامل اثبات و ارزیابی مقادیر $13n$ نمارسانی است.

۴- مقایسه

ویژگی‌های روش‌های مختلف جمع چندسویه امن به‌طور خلاصه در جدول (۱) نشان داده شده‌اند. همان‌گونه که در جدول (۱) مشخص شده است، راه‌کار کلیفتون ساده و کم هزینه است؛ اما در برابر تبانی دو نفر امن نیست. در راه‌کارهای [4] ، [6] و [10] از ایدهٔ راه‌کار کلیفتون [2] استفاده شده است و به‌منظور امنیت در برابر تبانی جزئی تا سطح $n-2$ نفر هزینه ارتباطی افزایش می‌یابد و از مرتبه $O(n^2)$ است. در راه‌کار [7] و [11] هزینه محاسباتی و ارتباطی بسیار مناسب و کمتر و یا مساوی $O(n)$ است؛ اما به طرف سوم معتمد نیاز دارند.

(جدول-۱): مقایسه روش‌های جمع چندسویه امن، n تعداد کاربران و m تعداد قطعات داده

	نیاز به طرف سوم	نیاز به کانال امن	امنیت در برابر تبانی	هزینه محاسبات	هزینه ارتباطات	مدل مهاجم
کلیفتون و همکاران [2]	×	✓	×	$O(n)$ جمع	$O(n)$	شبه‌درستکار
شیخ و همکاران[4]	×	✓	$n-2$	$O(n^2)$ جمع	$O(n^2)$	شبه‌درستکار
یوون و همکاران[6]	×	✓	$n-2$	$O(n^2)$ جمع	$O(n^2)$	شبه‌درستکار
جانگدی و همکاران[7]	✓	✓	$n-2$	$O(n)$ جمع	$O(n)$	شبه‌درستکار
راوتاری و همکاران[10]	×	✓	$n-2$	$O(n)$ جمع	$O(n^2)$	شبه‌درستکار
جهان و همکاران[11]	✓	✓	$n-2$	$O(logm)$ جمع	$O(m)$	شبه‌درستکار
جانگ و همکاران [12]	×	×	$n-2$	$O(n^2)$ نمارسانی	$(n^2-2n)\lceil log_2 p^2 \rceil$ بیت	شبه‌درستکار
عاشوری و همکاران Securesum-2[13]	×	×	$n-2$	$O(n)$ نمارسانی	$\lceil \log_2 N^2 \rceil 4n$ بیت	شبه‌درستکار
عزیزی و همکاران [16]	×	×	$n-2$	$O(n)$ نمارسانی	$4n \lceil \log_2 p^2 \rceil$ بیت	بدخواه

راه‌کارهای [11]-[2] با فرض وجود کانال امن طراحی شده‌اند و زمانی کاربرد دارند که کانال ارتباطی بین اعضای گروه امن باشد که درعمل فرض سنگینی است؛ ازاین‌رو هزینهٔ محاسباتی آن‌ها پایین است؛ اما هزینهٔ پنهان

[Downloaded from monadi.isc.org.ir on 2025-06-10]

پیاده‌سازی کانال امن در این راه‌کارها وجود دارد؛ اما در راه‌کارهای با فرض کانال ناامن هزینهٔ تبادل امن اطلاعات نیز محاسبه شده است؛ درواقع می‌توان گفت راه‌کارهای با فرض کانال ناامن درعمل بهینه‌تر از راه‌کارهای با فرض کانال امن هستند.

راه‌کارهای [12]، [13] و [16] به کانال امن نیاز ندارند ازاین رو هزینه محاسباتی آن‌ها، هزینهٔ نمارسانی پیمانه‌ای است. راه‌کار جانگ و همکاران [12] به‌منظور تأمین امنیت در برابر تبانی جزئی، دارای هزینهٔ محاسباتی از مرتبه $O(n^2)$ نمارسانی و هزینهٔ ارتباطی $O(n^2)$ است. راه‌کار عاشوری و همکاران [13] بدون نیاز به کانال امن راه‌کار کارایی با هزینه محاسباتی $O(n)$ نمارسانی است؛ اما نیاز به عدد مرکب N دارد که تجزیهٔ آن برای همگان مجهول باشد. راه‌کار عزیزی و همکاران [16] در مدل مهاجم بدخواه و بدون نیاز به کانال امن، دارای هزینه محاسباتی $O(n)$ نمارسانی است؛ بنابراین ارائه راه‌کاری که با هزینه محاسباتی کمتر در مدل مهاجم بدخواه قادر به محاسبه مجموع به‌صورت امن باشد، همچنان به‌عنوان یک مسأله پژوهشی مطرح است.

با توجه به مقایسه‌ها و توضیحات ذکرشده در راه‌کارهای با فرض کانال امن، ارائهٔ راه‌کاری که با هزینهٔ ارتباطی کم‌تر از $O(n^2)$ در برابر تبانی جزئی تا سطح $n-2$ نفر امن بوده و هزینهٔ محاسباتی آن کم‌تر و یا مساوی مرتبهٔ $O(n^2)$ باشد، بهبود مناسبی است. در راه‌کارهای با فرض کانال ناامن، ارائهٔ راه‌کاری که هزینه محاسباتی آن از مرتبهٔ $O(n)$ نمارسانی بوده و نسبت به راه‌کار عاشوری و همکاران [13] نیازی به فرض مجهول‌بودن تجزیهٔ پیمانهٔ محاسباتی N نداشته باشد و نسبت به راه‌کار عزیزی و همکاران [16] نمارسانی‌های کم‌تری نیاز داشته باشد، بهبود مناسبی در این حیطهٔ کاری محسوب می‌شود. به‌علاوه، مدل مهاجم تمامی راه‌کارها غیر از راه‌کار عزیزی و همکاران شبه‌درستکار درنظر گرفته شده و ارائهٔ راه‌کاری با مدل مهاجم بدخواه و بهینه‌تر از راه‌کار [16] حائز اهمیت است.

۵- جمع‌بندی

در این مقاله راه‌کارهای جمع‌چندسویهٔ امن از سال ۱۹۸۸ تا ۲۰۱۷ مرور و بررسی شدند و تمامی راه‌کارها از لحاظ کارایی (هزینهٔ ارتباطی و محاسباتی) و امنیت مقایسه شدند و با توجه به بررسی‌های انجام‌شده چالش‌های موجود ذکرشده و پیشنهادهایی ارائه شد.

۶- مراجع

[1] Or, M. B., Goldwasser, S., and Wigderson , A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. ACM Symposium on Theory of Computing. ACM. 1988. pp. 1-10.

[2] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., and Zhu, M. Y. Tools for Privacy Preserving Distributed Data Mining. ACM SIGKDD Explorations Newsletter. 2002. volume 4, 28-34.

[3] Sheikh, R., Kumar, B., and Mishra, D. K. Privacy-Preserving k-Secure Sum Protocol. International Journal of Computer Science and Information Security (IJCSIS), 2009. vol. 6, no. 2, 184-188.

[4] Sheikh, R., Kumar, B., and Mishra, D. K. A Distributed k-Secure Sum Protocol for Secure Multi-Party Computations. Journal of Computing, 2010. vol. 2, no. 3.

[5] Sheikh, R., Kumar, B., and Mishra, D. K. Changing Neighbors k-Secure Sum Protocol for Secure Multi-Party Computation. International Journal of Computer Science and Information Security (IJCSIS), 2010. vol. 7, no. 1, 239-243.

[6] Youwen, Z., Liusheng, H., Wei, Y., and Xing, Y. Efficient Collusion-Resisting Secure Sum Protocol. Chinese Journal of Electronics, 2011. 407-413.

[7] Jangde, M., Chandel, M. S., and Mishra, M. K. Hybrid Technique For Secure Sum Protocol. World of Computer Science and Information Technology Journal (WCSIT), 2011. vol. 1, no. 5, 198-201.

[8] Rautaray, J., and Kumar, R. DISTRIBUTED DATABASE RK-SECURE SUM PROTOCOL. International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), March 2013. vol. 2, no. 3, 559-562.

[9] Rautaray, J., and Kumar, R. Distributed RK-Secure Sum Protocol for Privacy Preserving. IOSR Journal of Computer Engineering (IOSR-JCE), Feb. 2013.vol. 9, no. 1, 49-52.

[10] Rautaray, J., Kumar, R., and Bajpai, G. Modified Distributed Rk Secure Sum Protocol. International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), March 2013. vol. 2, no. 3, 734-736.

[11] Jahan, I., Sharmy, N. N., Jahan, S., Ebha, F. A., and Lisa, N. J. Design of a Secure Sum Protocol using Trusted Third Party System for Secure Multi-Party Computations. 6th International

[Downloaded from monadi.isc.org.ir on 2025-06-10]

Conference on Information and Communication Systems (ICICS) IEEE. 2015. pp. 136-141.

[12] Jung, T., and Yang Li, X. Collusion-Tolerable Privacy-Preserving Sum and Product Calculation without Secure Channel. IEEE Transactions on Dependable and secure computing, 2015. 45-57.

[13] Talouki, M. A., and Dastjerdi, A. B. Cryptographic collusion-resistant protocols for secure sum. Electronic Security and Digital Forensics, Vol 9, 2016.

[14] Hao, F., and Zielinski, P. A 2-Round Anonymous Veto Protocol. In Security Protocols . Springer Berlin Heidelberg. 2009. pp. 202-211.

[15] Burmester, M., and Desmedt, Y. A secure and efficient conference key distribution system. In Advances in Cryptology . Springer-Verlag. 2006. pp. 275-286.

[۱۶] ش. عزیزی, م. عاشوری و ح. ملا, پروتکل کارا برای جمع چند سویه امن در مدل بدخواه با فرض کانال ناامن, در بیست و دومین کنفرانس ملی سالانه کامپیوتر ایران, تهران, ۱۳۹۵.

شادیه عزیزی. ایشان مدرک کارشناسی مهندسی فناوری اطلاعات را در سال ۱۳۹۱ از دانشگاه کردستان اخذ کرد و از سال ۱۳۹۳ دانشجوی کارشناسی ارشد دانشگاه اصفهان در رشته مهندسی فناوری اطلاعات گرایش امنیت اطلاعات است. زمینه‌های پژوهشی مورد علاقهٔ ایشان: استخراج قوانین انجمنی از پایگاه داده‌ها، حفظ حریم مکانی در خدمات مبتنی بر مکان، کنترل دسترسی و پروتکل‌های امنیتی.

نشانی رایانامهٔ ایشان عبارت است از:

sh.azizi93@eng.ui.ac.ir

مائده تلوکی عاشوری مدرک کارشناسی مهندسی کامپیوتر را در سال ۱۳۸۲ و مدرک کارشناسی ارشد را درسال ۱۳۸۵ و مدرک دکترا را نیز در سال ۱۳۹۱ از دانشگاه اصفهان اخذ کرده است و در حال حاضر عضو هیئت علمی و استادیار دانشکده کامپیوتر دانشگاه اصفهان نیز می‌باشند. زمینه‌های پژوهشی مورد علاقهٔ ایشان: امنیت شبکه‌های موبایل، گم‌نامی و حریم خصوصی کاربران، پروتکل‌های امنیتی، پروتکل‌های رمزنگاری توزیع‌شده و امنیت شبکه.

نشانی رایانامهٔ ایشان عبارت است از:

m.ashouri@eng.ui.ac.ir

حمید ملا. ایشان مدرک کارشناسی مهندسی کامپیوتر را در سال ۱۳۸۲ و مدرک کارشناسی ارشد را درسال ۱۳۸۴ و مدرک دکترا را نیز در سال ۱۳۸۹ از دانشگاه صنعتی اصفهان اخذ کرده است و در حال حاضر عضو هیئت علمی و استادیار دانشکده کامپیوتر دانشگاه اصفهان می‌باشند. زمینه‌های پژوهشی مورد علاقهٔ ایشان: طراحی و تحلیل رمزهای قالبی، امضای دیجیتال و پروتکل‌های امنیتی.

نشانی رایانامهٔ ایشان عبارت است از:

h.mala@eng.ui.ac

[Downloaded from monadi.isc.org.ir on 2025-06-10]

[Downloaded from monadi.isc.org.ir on 2025-06-10]