

# مرور و بررسی روش‌های جمع چندسویه امن و چالش‌های موجود

شادیه عزیزی<sup>۱</sup>، مائده عاشوری تلوکی<sup>۲\*</sup> و حمید ملا<sup>۳</sup>

<sup>۱</sup> گروه مهندسی فناوری اطلاعات، دانشگاه اصفهان، اصفهان، ایران  
sh.azizi93@eng.ui.ac.ir

<sup>۲</sup> استادیار، مهندسی فناوری اطلاعات، دانشگاه اصفهان، اصفهان، ایران  
m.ashouri@eng.ui.ac.ir  
h.mala@eng.ui.ac.ir

## چکیده

در حوزه امنیت اطلاعات، انجام محاسبات ریاضی بر روی داده‌های خصوصی به صورت امن و گروهی (محاسبات چندسویه امن) بیش از پیش مورد توجه قرار گرفته است. نخستین بار، محاسبات چندسویه امن، در قالب مسئله میلیونرها مطرح شد که در آن دو میلیونر بدون افشای میزان سرمایه خود و بدون استفاده از طرف سوم مورد اعتماد، قصد داشتند بدانند کدامیک ثروتمندتر است. پس از آن مسائل دیگری در حوزه محاسبات چندسویه امن مطرح شد. در این پژوهش مسئله جمع چندسویه امن، در نظر گرفته شده است؛ در جمع چندسویه امن گروهی از کاربران قصد محاسبه مجموع داده محرمانه خود را دارند؛ به طوری که محرمانگی داده‌های آنها حفظ شود. در این مقاله پیشینه‌ای از راه‌حل‌های موجود در این حیثه بررسی و مقایسه شده‌اند. به علاوه چالش‌های موجود در این زمینه بررسی و پیشنهادهایی جهت راه‌کارهای آینده ارائه شده‌اند.

واژگان کلیدی: جمع چند سویی امن، حمله تبانی، کانال ناامن.

## ۱- مقدمه

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

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

در محاسبات چندسویه امن کاربران  $P_1$ ،  $P_2$ ، ... و  $P_n$  به ترتیب دارای مقادیر محرمانه  $d_1$ ،  $d_2$ ، ... و  $d_n$  بوده و قصد محاسبه امن تابع  $f(d_1, d_2, \dots, d_n)$  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 چندسویه امن هدف محاسبه مجموع داده محرمانه کاربران است به طوری که در پایان محاسبه، هر عضو تنها داده محرمانه خود و نتیجه حاصل جمع را می‌داند و از داده محرمانه دیگران مطلع نیست. پروتکل‌های جمع چندسویه امن را به دو دسته می‌توان تقسیم کرد:

دسته نخست، راه‌کارهایی هستند که فرض می‌کنند در بین اعضا کانال امن وجود دارد؛ در این راه‌کارها کانال

\* نویسنده عهده‌دار مکاتبات

عضو، مقدار تابع را به‌ازای شناسه عمومی هر عضو گروه  $\alpha_j$  محاسبه و برای او ارسال می‌کند ( $d_j = f_i(\alpha_j)$ ).

$$f_i = d_j + a_1x + \dots + a_t x^t \quad (1)$$

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. در واقع 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  $\alpha_j$  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 هزینه محاسباتی تولید تابع و هزینه ارتباطی ارسال مقدار تابع است. به‌علاوه با تبانی  $t + 1$  عضو، داده محرمانه کاربر افشا می‌شود.

کلیفتون<sup>۲</sup> و همکاران [2] در سال ۲۰۰۲ راه‌کاری را برای جمع چندسویه 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؛ داده 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<sup>۳</sup> و همکاران در مقاله [3]، پروتکلی تحت عنوان K-Secure Sum به‌منظور بهبود جمع چندسویه امن با فرض

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

در محاسبات چندسویه 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 داده محرمانه 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 چندسویه امن ارائه‌شده در بازه 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.

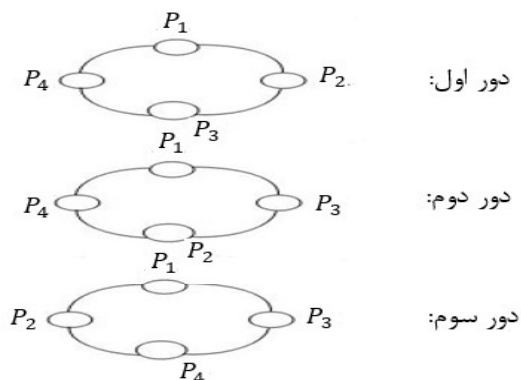
## ۲- راه‌کارهای با فرض کانال امن

در این بخش راه‌کارهایی با فرض وجود کانال ارتباطی امن و غیر قابل شنود در بین اعضا مرور می‌شوند. نخستین راه‌کار ارائه‌شده جهت جمع چندسویه امن توسط بن‌اور<sup>۱</sup> 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 محرمانه سایر اعضا نخواهند بود. طرف سوم راز را بین اعضا تقسیم و این روش از فرض کانال امن استفاده می‌کند. فرض کنیم  $n$  عضو  $P_1, P_2, \dots, P_n$  و به‌ترتیب با مقادیر محرمانه  $d_1, d_2, \dots, d_n$  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 مقادیر عددی  $\{\alpha_0, \alpha_1, \dots, \alpha_{n-1}\}$  را به‌عنوان شناسه 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

<sup>1</sup> Ben Or

<sup>2</sup> Clifton

<sup>3</sup> Sheikh



(شکل 1-): نمایی از جابه‌جایی اعضا در Ck-Secure Sum [4]

پروتکل بعدی شیخ و همکاران در [5] تحت عنوان Dk-Secure Sum<sup>1</sup> و با فرض کانال امن ارائه شد. اعضا داده محرمانه 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؛ به طوری که در نهایت 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 تسانی جزئی تا سطح  $n - 2$  امن است؛ اما با فرض وجود کانال امن هزینه ارتباطی آن از مرتبه  $O(n^2)$  است.

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

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

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 داده محرمانه 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  $P_i$  حاصل جمع جزئی را برای  $P_{(i+1) \bmod n}$  ارسال می‌کند و روال تا کامل‌شدن دور ادامه دارد. در پایان این دور حاصل جمع جزئی در اختیار  $P_1$  قرار می‌گیرد. دور دوم را برای بلوک‌های دوم آغاز و حاصل جمع 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 تسانی جزئی امن نیست و با تسانی دو کاربر در  $n$  دور، داده 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 داده 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  $m$  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 تسانی دو کاربر برای محاسبه داده محرمانه عضو میانی امن است؛ اما در برابر تسانی جزئی بیش از دو کاربر امن نیست؛ زیرا در هر دور فقط  $P_2$  جابه‌جا می‌شود.

<sup>1</sup> Distributed k-Secure Sum

<sup>2</sup> Youwen

<sup>3</sup> Collusion-Resisting Secure Sum Protocol

اضافه کردن بلوک دوم و مقدار تصادفی مربوط به آن تا رسیدن به عضو آخر ادامه می‌یابد.

پس از انجام روال ذکر شده توسط عضو آخر، مقدار  $(\sum_{i=1}^n (D_{i1} + D_{i2} + r_{i2}))$  محاسبه شده است. عضو آخر، مجموع حاصل شده در این مرحله را به عضو ما قبل خود می‌دهد؛ وی مقدار تصادفی بلوک دوم خود را حذف  $(r_{(n-1)2})$  و بلوک سوم و مقدار تصادفی آن را اضافه  $(D_{(n-1)3} + r_{(n-1)3})$  و نتیجه را برای عضو ماقبل خود ارسال می‌کند؛ این روال تا رسیدن به عضو نخست ادامه دارد. عضو نخست مجموع جزئی را به طرف سوم و طرف سوم به عضو آخر می‌دهد؛ عضو آخر مقدار تصادفی دوم خود را حذف  $(r_{n2})$  و بلوک سوم داده خویس  $(D_{n3})$  را اضافه و نتیجه را برای عضو ماقبل ارسال می‌کند. عضو آخر به بلوک سوم، مقدار تصادفی اضافه نمی‌کند. در این مرحله مجموع جزئی به دست آمده برابر  $(\sum_{i=1}^n (D_{i1} + D_{i2} + D_{i3})) + \sum_{i=1}^{n-1} r_{i3}$  است؛ سپس هر عضو  $P_i (i \neq n)$  تا رسیدن به عضو نخست، مقدار تصادفی سوم خود را  $(r_{i3})$  حذف و برای عضو ماقبل ارسال می‌کند. عضو نخست، پس از حذف مقدار تصادفی سوم خود  $(r_{13})$ ، مجموع نهایی  $(D_{i1} + D_{i2} + D_{i3})$  را محاسبه و برای طرف سوم می‌فرستد تا طرف سوم آن را پخش همگانی کند. در این روش داده محرمانه به بلوک‌هایی تقسیم و همراه هر بلوک عدد تصادفی نیز ارسال می‌شود. با انجام حمله تبانی، بلوک داده به همراه عدد تصادفی مربوط به آن آشکار می‌شود و داده محرمانه، مخفی باقی می‌ماند؛ بنابراین در برابر تبانی جزئی اعضا و طرف سوم امن است. این روش به طرف سوم معتمد نیاز دارد و هزینه ارتباطی و محاسباتی آن  $O(n)$  است.

راوتاری<sup>۳</sup> و همکاران در سال ۲۰۱۳ [8] پروتکل Distributed RK Secure Sum را با فرض کانال امن ارائه دادند. این پروتکل به منظور افزایش کارایی، اعضا را در چیدمان باس قرار می‌دهد. کاربران  $P_1, P_2, \dots, P_n$  هر یک داده خود را به  $n - 1$  بلوک تقسیم و نزد خود نگه می‌دارند. در دور نخست عضو نخست  $(P_1)$  بلوک نخست از داده خود را برای عضو دوم می‌فرستد؛ عضو دوم  $(P_2)$  نیز پس از جمع بلوک نخست داده خود با مقدار دریافتی از عضو نخست، حاصل را برای عضو سوم ارسال می‌کند روال تا رسیدن به عضو  $n$  ام ادامه دارد،  $P_n$  حاصل جمع جزئی دور نخست را نزد خود نگه می‌دارد؛ در دور دوم عضو دوم  $P_2$  با عضو سوم  $P_3$  جابه‌جا و روال دور نخست برای بلوک دوم تکرار می‌شود.

<sup>3</sup> Rautaray

یا به آن اضافه می‌کند و به‌طور محرمانه  $P_i$  را از عمل کم کردن و یا اضافه کردن مقدار ارسالی آگاه می‌سازد؛  $P_i$  عکس عمل انجام شده را بر روی  $m_i$  خود انجام می‌دهد.

در مرحله محاسبه مجموع، عضو نخست حلقه  $(P_1)$  مقدار  $m_i$  حاصل از مرحله نخست را با عدد تصادفی  $r$  جمع و برای عضو دوم حلقه ارسال می‌کند؛ عضو دوم مقدار دریافتی را با مقدار محرمانه نزد خود جمع می‌زند و حاصل را برای عضو سوم می‌فرستد. روال تا رسیدن به عضو  $n$  ام ادامه دارد. در پایان نتیجه برای عضو نخست ارسال می‌شود و او مقدار  $r$  را از مجموع به دست آمده کم کرده و نتیجه نهایی را پخش همگانی می‌کند. در این راه کار اگر در مرحله نخست هر کاربر  $n - 1$  عدد تصادفی تولید کند در برابر تبانی جزئی امن است در غیر این صورت امنیت در برابر تبانی جزئی کاهش می‌یابد. بنابراین برای امنیت بیشتر هزینه ارتباطی را افزایش می‌یابد؛ زیرا امنیت در برابر تبانی جزئی، به تعداد اعداد تصادفی تولید شده و به اشتراک گذاشته شده وابسته است.

در سال ۲۰۱۱ جانگدی<sup>۱</sup> و همکارانش در مقاله [7] راه‌کاری ترکیبی از مدل واقعی و ایده‌آل و با فرض کانال امن ارائه دادند. در مدل ایده‌آل یک طرف سوم معتمد<sup>۲</sup> وجود دارد که اعضا به‌طور محرمانه داده‌های خود را برای او می‌فرستند و طرف سوم همه محاسبات را انجام می‌دهد. در مدل واقعی از طرف سوم غیر معتمد استفاده می‌شود. در راه‌کار مقاله اعضای  $P_1, P_2, \dots, P_n$  باید داده محرمانه خود را به بلوک‌هایی تقسیم کنند. در ابتدای کار تعداد بلوک‌ها به‌طور آشکار تعیین می‌شود.

در مقاله [7] به منظور توضیح راه‌کار، اعضا داده محرمانه 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 کاربر  $P_i$  است. هر عضو، مجموع بلوک نخست به همراه مقدار تصادفی نظیر آن را برای طرف سوم می‌فرستد  $(D_{i1} + r_{i1})$ . طرف سوم پس از دریافت مقادیر کل اعضا، مجموع بلوک‌های نخست و مقادیر تصادفی  $(D_{i1} + r_{i1})$  را محاسبه و برای عضو نخست  $(P_1)$  می‌فرستد.  $P_1$  مقدار تصادفی جمع شده با بلوک نخست داده محرمانه خود را کم  $(r_{11})$  و بلوک دوم داده محرمانه و مقدار تصادفی مربوط به آن را اضافه  $(D_{i2} + r_{i2})$ ، و حاصل را برای عضو دوم ارسال می‌کند. روال حذف مقدار تصادفی نخست و

<sup>1</sup> Jangde

<sup>2</sup> Trusted Third Party (TTP)

$P_2$  ارسال می‌کند، 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 نگه می‌دارد.  $P_1$  دور دوم را برای جمع بلوک‌های دوم مشابه دور نخست آغاز می‌کند. روال برای  $n$  دور و جمع  $n$  بلوک ادامه می‌یابد و در نهایت  $P_n$  مجموع محاسبه‌شده را پخش همگانی می‌کند. این راه‌کار در برابر تبانی جزئی تا سطح  $n - 2$  نفر امن است؛ اما هزینه ارتباطی آن با فرض وجود کانال امن از مرتبه  $O(n^2)$  است.

جهان<sup>۱</sup> و همکاران پروتکل DRPM<sup>۲</sup> 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:

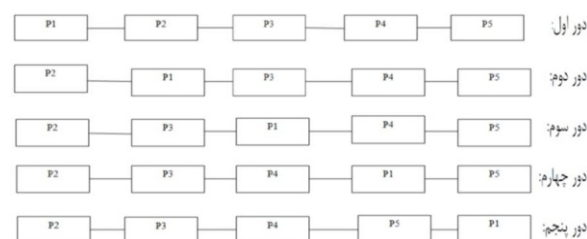
در بخش نخست، اعضا داده محرمانه خود را به تعداد مشخص بلوک  $m$  تقسیم می‌کنند. هر کاربر  $P_i$  یک آرایه از اعداد تصادفی با طول محرمانه و کمتر یا مساوی  $m$  را تولید می‌کند ( $R_i$ ).

در بخش دوم، برای هر کاربر  $P_i$  یک پرچم در نظر 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 در صورتی که از عدد تصادفی استفاده کرده باشند، مقدار پرچم مربوط به خود را  $true$  و در صورت عدم استفاده از مقدار تصادفی مقدار پرچم را  $false$  قرار می‌دهند؛ سپس طرف سوم مقدار  $value$  را به ترتیب 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  $value$  در صورتی که مقدار پرچم به‌ازای او  $true$  باشد، باید مقدار تصادفی را که در مرحله قبل اضافه کرده از  $value$  کم کند و سپس در صورت 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 عدد تصادفی، فقط بلوک دوم را به  $value$  اضافه کرده و  $Flag[i] = false$  قرار می‌دهد و  $value$  را برای

در دور سوم  $P_2$  با  $P_4$  جابه‌جا می‌شود و همین روال برای  $n - 1$  دور تکرار می‌شود. در دور  $m - 1$  کاربر  $P_2$  با  $P_n$  جابه‌جا و کاربر  $P_n$  حاصل جمع‌های دورهای قبلی را با بلوک  $m - 1$  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 حاصل جمع  $n - 1$  بلوک  $n$  کاربر محاسبه می‌شود. به دلیل 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 عضو 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. به طوری که در دور  $m$  ام 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 محرمانه 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

<sup>1</sup> Jahan

<sup>2</sup> Double Random Partitioned Model

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؛ به طوری که  $q | p - 1$ . گروه گردشی  $G_1$  از مرتبه  $q$  و با مولد  $g_1$  با شرط رابطه (۲) انتخاب می‌شود. در رابطه (۲) مقدار  $h \in \mathbb{R} Z_p$  است.

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

سپس گروه گردشی  $G_2$  از مرتبه  $q$  و با مولد  $g_2$  با شرط  $g_2 = g_1^p \bmod p^2$  انتخاب می‌شود. پروتکل ضرب و سپس پروتکل جمع در هر دو 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.

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

سپس هر عضو  $P_i$  داده محرمانه خود را به صورت رابطه (۴) رمز می‌کند.

$$C_i = d_i \cdot R_i = x_i \cdot (g_1^{r_{i+1}}/g_1^{r_{i-1}})^{r_i} \bmod p \quad (۴)$$

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 حاصل ضرب است ( $r_{n+1} = r_n$  و  $r_0 = r_1$ ).

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

در مدل یک تجمیع‌کننده، چون کانال نامن 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 داده محرمانه ندارد و  $C_i$  را محاسبه نمی‌کند؛ و

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 عدد تصادفی توسط یکدیگر اطلاع ندارند و قادر به حدس مقدار محرمانه یکدیگر نیستند. این راه‌کار به طرف سوم معتمد نیاز دارد. در صورتی که 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 نامن

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

جانگ<sup>۱</sup> و همکاران در سال [12] پروتکلی را بدون نیاز به کانال امن، بدون طرف سوم و در مدل شبه‌درستکار ارائه داده است. اساس آن مسئله سخت دیفی هلمن محاسباتی  $(CDH)$  است. اعضا  $P_1, P_2, \dots, P_n$  به ترتیب داده محرمانه  $d_1, d_2, \dots, d_n$  را دارند که هر  $d_i \in Z_p$  است. این مقاله روشی برای محاسبه مجموع مقادیر محرمانه  $(= f(d))$  و روشی دیگر برای محاسبه ضرب این مقادیر  $(= \prod_{i=1}^n d_i)$  راه‌کار ارائه داده است. هر روش در دو مدل ارائه شده است: مدل یک تجمیع‌کننده<sup>۳</sup>، مدل فقط اعضا<sup>۴</sup>.

در مدل یک تجمیع‌کننده، داده‌ها برای عضوی شبه‌درستکار مانند  $A$  ارسال می‌شود و فقط او  $f(d)$  را

<sup>1</sup> Jung

<sup>2</sup> Computational Diffie-Hellman problem

<sup>3</sup> One Aggregator Model

<sup>4</sup> Participants Only Model

مقدار  $C_i$  را برای  $A$  ارسال می‌کنند و  $A$  با ضرب کل مقادیر به صورت  $R_{n+1} \prod_{i=1}^n C_i = C \pmod{p^2}$  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  $P_i$  و  $P_{i+2}$  با هماهنگی در انتخاب  $r_i$  و  $r_{i+2}$  به صورت  $r_i = r_{i+2} - a$  قادر به محاسبه  $R_{i+1} = (g_1^a)^{r_{i+1}}$  و محاسبه داده محرمانه  $P_{i+1}$  است.  $P_i$  و  $P_{i+2}$  مقدار  $R_{i+1}$  را با انجام  $Y_i^a = (g_1^{r_i})^a$  محاسبه می‌کنند.

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

$$R_i = (g_1^{r_{i+k+1}} / g_1^{r_{i-1}})^{r_k r_{k-1} \dots r_{i+1} r_i} \in G_1 \quad (11)$$

این روش نیازی به کانال امن ندارد؛ ولی به دلیل انجام نمارسانی برای هر کاربر، هزینه محاسباتی بالایی دارد و از طرفی برای افزایش امنیت در برابر تیبانی جزئی هزینه محاسباتی کاربران بسیار افزایش می‌یابد؛ به طوری که به منظور داشتن امنیت در برابر تیبانی جزئی تا سطح  $n - 2$  نفر هزینه محاسباتی از مرتبه  $O(n^2)$  است.

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

گروه پیلیه  $G$  ( $Z_{N^*}^2$ ) با مولد تصادفی  $g \in G$  و مولد خاص  $g_s = 1 \pmod{N}$  در نظر گرفته می‌شود. تجزیه عوامل نخست  $N$  برای همگان مجهول است. به ازای  $g$  حل مسئله دیفی هلمن تصمیمی<sup>۱</sup> از لحاظ محاسباتی ناممکن و به ازای  $g_s$  مسئله لگاریتم گسسته<sup>۲</sup> قابل حل است.

اعضای  $P_1, P_2, \dots, P_n$  به ترتیب داده محرمانه  $d_1, d_2, \dots, d_n$  را داشته و بر روی  $(g_s, g, G)$  توافق دارند. پروتکل‌های ارائه شده به شرح زیر است:

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

سایر اعضا مقدار  $C_i$  را برای  $A$  ارسال می‌کنند و  $A$  با ضرب کل مقادیر به صورت  $R_{n+1} \prod_{i=1}^n C_i = \prod_{i=1}^n x_i \pmod{p}$  حاصل ضرب 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 به منظور محاسبه مجموع رابطه (۶) در نظر گرفته می‌شود.

$$(1+p)^m = \sum_{i=0}^m \binom{m}{i} p^i \quad (6)$$

$$= 1 + mp \pmod{p^2}$$

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

$$\prod_{i=1}^n (1+p)^{d_i} = \prod_{i=1}^n (1+d_i p) \quad (7)$$

$$= \left(1 + p \sum_{i=1}^n d_i\right) \pmod{p^2} = c$$

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

$$C_i = (1 + d_i p). R_i = (1 + d_i p). (g_1^{r_{i+1}} / g_1^{r_{i-1}})^{r_i} \pmod{p^2} \quad (8)$$

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.

$$C = \prod_{i=1}^n C_i = \prod_{i=1}^n (1 + d_i p). (g_1^{r_{i+1}} / g_1^{r_{i-1}})^{r_i} \quad (9)$$

$$= \left(1 + p \sum_{i=1}^n d_i\right). g_1^0$$

$$= \left(1 + p \sum_{i=1}^n d_i\right) \pmod{p^2}$$

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

$$\frac{c-1}{p} = \sum_{i=1}^n d_i \quad (10)$$

در مدل یک تجمیع‌کننده، عضو  $A$  به عنوان کاربر  $P_{n+1}$  محسوب می‌شود و مقدار  $R_{n+1}$  را محاسبه می‌کند؛ اما داده محرمانه ندارد و  $C_i$  را محاسبه نمی‌کند؛ و سایر اعضا

<sup>1</sup> Decisional Diffie-Hellman (DDH)  
<sup>2</sup> Discrete Logarithm (DL)

به طوری که از ضرب مقادیر شبکه و تنوی گمنام کلید کنفرانس حاصل می شود. در این پروتکل کلید مشترک گروه  $k = \prod_i g^{a_i b_i} = g^{\sum a_i b_i} = g^{2 \sum_{i=1}^n a_{i-1} a_i} = g^{2(a_1 a_2 + a_2 a_3 + \dots + a_n a_1)}$  این پروتکل شامل سه مرحله است:

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

$$g^{b_i} = g^{a_{i+1}} g^{a_{i-1}} \prod_{j=1, j \neq i, i+1, i-1}^n g^{(\text{sign}(i-j) a_j)} \quad (13)$$

هر کاربر  $P_i$  کلید را محاسبه می کند:  
 $k = k_i = (g^{a_{i-1}})^{2n} \cdot t_i^{2n-1} \cdot t_{i+1}^{2n-2} \cdot \dots \cdot t_{i-2}^2$   
 در مرحله سوم، هر کاربر  $P_i$  مقدار  $W_i = g^{a_i b_i} g_s^{d_i}$  را محاسبه و پخش همگانی می کند. با ضرب مقادیر ارسالی گروه ماسک شبکه تبدیل به کلید کنفرانس  $k$  شده و مجموع رمز شده مطابق با رابطه (۱۴) حاصل می شود.

$$\begin{aligned} \prod_i W_i &= \prod_i g^{a_i b_i} g_s^{d_i} \\ &= \prod_i g^{a_i b_i} \prod_i g_s^{d_i} \\ &= g^{\sum a_i b_i} g_s^{\sum d_i} \\ &= k g_s^{\sum d_i} \pmod{N^2} \end{aligned} \quad (14)$$

فقط اعضای گروه قادر به محاسبه  $k$  هستند و پس از حذف آن طبق رابطه (۱۲) قادر به محاسبه مجموع  $\sum_{i=1}^n d_i$  هستند. این پروتکل به دلیل آشکار بودن  $t_i$  در برابر تباری جزئی تا سطح  $4 - n$  نفر امن است و محرمانگی حاصل جمع را حفظ می کند. هزینه محاسباتی آن با نادیده گرفتن محاسبات کم هزینه به ازای هر کاربر 4 نمراسانی و به ازای کل کاربران  $4n$  نمراسانی است. در سه نسخه پروتکل از عملیات نمراسانی استفاده می شود و مدل مهاجم شبه در دستکار است و مجهول بودن تجزیه عوامل نخست  $N$  برای همگان فرض سنگینی است.

راه کار 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 می شوند؛ به طوری که  $q | p - 1$  گروه گردشگی  $G_1$  از مرتبه  $q$  و با مولد

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

در مرحله دوم، هر کاربر  $P_i$  مقدار  $W_i = g^{a_i b_i} g_s^{d_i}$  محاسبه و پخش همگانی می کند. با ضرب مقادیر ارسالی گروه ماسک شبکه و تنوی گمنام از بین می رود و  $\prod_i W_i = g_s^{\sum d_i} \pmod{N^2}$  حاصل می شود. اگر  $g_s = 1 + kN$  باشد، مجموع  $\sum_{i=1}^n d_i$  طبق رابطه (۱۲) قابل محاسبه است.

$$\sum_{i=1}^n d_i = \frac{g_s^{\sum d_i} - 1}{kN} \quad (12)$$

پروتکل SECURESUM V-1 به دلیل استفاده از شبکه و تنوی گمنام در برابر تباری جزئی تا سطح  $2 - n$  نفر امن است و هزینه محاسباتی آن با نادیده گرفتن محاسبات کم هزینه به ازای کل کاربران  $2n$  نمراسانی است.

SECURESUM V-2: پروتکل دوم، از شبکه و تنوی گمنام برای محرمانگی داده کاربران و از پروتکل اشتراک کلید کنفرانس [15] BD به منظور محرمانگی حاصل جمع در برابر مهاجم بیرونی استفاده می کند. این پروتکل نیز شامل سه مرحله است:

در مرحله 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 می کند. در مرحله دوم، هر کاربر  $P_i$  مقدار  $t_i = (g^{e_{i+1}}/g^{e_{i-1}})^{e_i}$  را محاسبه و پخش همگانی می کند. 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  $k = (g^{e_{i-1}})^{n e_i} \cdot t_i^{n-1} \cdot t_{i+1}^{n-2} \cdot \dots \cdot t_{i-2}^2 = g^{e_1 e_2 + e_2 e_3 + \dots + e_n e_1}$  را محاسبه و نزد خود نگاه می دارد.

در مرحله 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 می کند. با ضرب مقادیر ارسالی گروه ماسک شبکه از بین می رود و  $\prod_i W_i = k g_s^{\sum d_i} \pmod{N^2}$  حاصل می شود. فقط اعضای گروه مقدار کلید کنفرانس  $k$  را می دانند و با حذف آن طبق رابطه (۱۲) قادر به محاسبه مجموع  $\sum_{i=1}^n d_i$  هستند. پروتکل SECURESUM V-2 در برابر تباری جزئی تا سطح  $2 - n$  نفر امن است، محرمانگی حاصل جمع را حفظ می کند و هزینه محاسباتی آن با نادیده گرفتن محاسبات کم هزینه به ازای کل کاربران  $5n$  نمراسانی است.

SECURESUM V-3: این پروتکل محرمانگی داده کاربر و نتیجه حاصل جمع حفظ می کند و شبکه و تنوی گمنام را با پروتکل اشتراک کلید کنفرانس BD ترکیب می کند؛

برای  $\sum_{i=1}^n d_i = \frac{(G \times k^{-1}) - 1}{p}$  را به دست آورند. این راه‌کار در برابر تیبانی جزئی تا سطح  $n - 2$  نفر امن است. پیمانه محاسباتی یک عدد نخست بسیار بزرگ است و در مقایسه با راه‌کار [13] نیازی به انتخاب عدد مرکب  $N$  نیست. هزینه محاسباتی آن بدون در نظر گرفتن اثبات‌های صفردانش برای مقابله با مهاجم بدخواه  $5n$  نمارسانی و با در نظر گرفتن 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 تیبانی دو نفر امن نیست. در راه‌کارهای [4]، [6] و [10] از ایده راه‌کار کلیفتون [2] استفاده شده است و به‌منظور امنیت در برابر تیبانی 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.

و  $(h \in \mathbb{Z}_p)$   $g_1 = h^{(p-1)/q} \text{mod } p$ ،  $g_1 \neq 1 \text{mod } p$  سپس گروه گردشی  $G_2$  از مرتبه  $q$  و با مولد  $g_2 = g_1^p \text{mod } p^2$  انتخاب می‌شود. راه‌کار شامل دو مرحله است: در مرحله نخست اعضای گروه شبکه و توی 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  $a_i, e_i \in \mathbb{Z}_R$  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  $a_i, e_i \in \mathbb{Z}_R$  را نیز ارسال می‌کند؛ سپس کاربر  $P_i$  در صورت درستی مقادیر ارسالی سایر اعضای گروه  $g^{a_i b_i}$  و  $g^{b_i} = (\prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^n g^{a_j})$  کنفـرانس  $k = (g^{e_i-1})^{n e_i} \cdot t_i^{n-1} \cdot t_{i+1}^{n-2} \cdot \dots \cdot t_{i-2} = g^{\sum_{i=1}^n e_i - 1 e_i} = g^{e_1 e_2 + e_2 e_3 + \dots + e_n e_1}$  را محاسبه و نزد خود نگه می‌دارد.

در مرحله دوم، هر کاربر  $P_i$  مقدار  $w_i = (1 + d_i p) g^{e_i - 1 e_i} g^{a_i b_i} \text{mod } p^2$  را به همراه اثبات صفر دانش آن پخش همگانی می‌کند؛ سپس هر کاربر  $P_i$  در صورت 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،

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

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

اعضای گروه امن باشد که در عمل فرض سنگینی است؛ از این رو هزینه محاسباتی آن‌ها پایین است؛ اما هزینه پنهان

راه‌کارهای [11]-[2] با فرض وجود کانال امن طراحی شده‌اند و زمانی کاربرد دارند که کانال ارتباطی بین

توجه به بررسی‌های انجام‌شده چالش‌های موجود ذکر شده 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

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

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

با توجه به مقایسه‌ها و توضیحات ذکر شده در راه‌کارهای با فرض کانال امن، ارائه راه‌کاری که با هزینه ارتباطی کمتر از  $O(n^2)$  در برابر تسانی جزئی تا سطح  $n - 2$  نفر امن بوده و هزینه محاسباتی آن کمتر و یا مساوی مرتبه  $O(n^2)$  باشد، بهبود مناسبی است. در راه‌کارهای با فرض کانال ناامن، ارائه راه‌کاری که هزینه محاسباتی آن از مرتبه  $O(n)$  نامرسانی بوده و نسبت به راه‌کار عاشوری و همکاران [13] نیازی به فرض مجهول بودن تجزیه پیمان‌های محاسباتی  $N$  نداشته باشد و نسبت به راه‌کار عزیز و همکاران [16] نامرسانی‌های کمتری نیاز داشته باشد، بهبود مناسبی در این حیطه کاری محسوب می‌شود. به علاوه، مدل مهاجم تمامی راه‌کارها غیر از راه‌کار عزیز و همکاران شبه‌درستکار در نظر گرفته شده و ارائه راه‌کاری با مدل مهاجم بدخواه و بهینه‌تر از راه‌کار [16] حائز اهمیت است.

## ۵- جمع بندی

در این مقاله راه‌کارهای جمع‌چندسویه امن از سال ۱۹۸۸ تا ۲۰۱۷ مرور و بررسی شدند و تمامی راه‌کارها از لحاظ کارایی (هزینه ارتباطی و محاسباتی) و امنیت مقایسه شدند و با

دانشگاه اصفهان نیز می‌باشند. زمینه‌های پژوهشی مورد علاقه ایشان: امنیت شبکه‌های موبایل، گم‌نامی و حریم خصوصی کاربران، پروتکل‌های امنیتی، پروتکل‌های رمزنگاری توزیع‌شده و امنیت شبکه.

نشانی رایانامه ایشان عبارت است از:

[m.ashouri@eng.ui.ac.ir](mailto:m.ashouri@eng.ui.ac.ir)



**حمید ملا.** ایشان مدرک کارشناسی

مهندسی کامپیوتر را در سال ۱۳۸۲ و

مدرک کارشناسی ارشد را در سال 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 علاقه ایشان: طراحی و تحلیل رمزهای قالبی، امضای دیجیتال و پروتکل‌های امنیتی.

نشانی رایانامه ایشان عبارت است از:

[h.mala@eng.ui.ac](mailto:h.mala@eng.ui.ac)

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. عزیز، 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 عزیز.** 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 علاقه ایشان: استخراج قوانین انجمنی از پایگاه داده‌ها، حفظ حریم مکانی در خدمات مبتنی بر مکان، کنترل دسترسی و پروتکل‌های امنیتی.

نشانی رایانامه ایشان عبارت است از:

[sh.azizi93@eng.ui.ac.ir](mailto:sh.azizi93@eng.ui.ac.ir)



**مأده تلوقی عاشوری** مدرک کارشناسی

مهندسی کامپیوتر را در سال ۱۳۸۲ و

مدرک کارشناسی ارشد را در سال 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

