Journal of Electronical & Cyber Defence
Vol. 9, No. 3, 2021, Serial No. 35

Design of a Secure Two-Party Protocol Based on the Expanded Cut-and-Choose Bilateral Oblivious Transfer

M. Azizi[1*], S. Ghorbanzadeh Havestin[2]

*Imam Hossein University

(Received: 12/09/2019, Accepted: 22/02/2021)

ABSTRACT

In the secure two-party computation, two parties wish to jointly compute a function with of their private inputs,while revealing only the output. Yao's garbled circuit protocol is a classic and solution to this problem. It is well-known that Yao's protocol is vulnerable to malicious behavior by its participants. The general approach as cut-and-choose techniques that proposes to solve this vulnerables, but cut-and-choose techniques creates new problems within itself including of selective failure attack and consisitecy inputs. In this paper we present a secure two party computation based on expanded cut-and-choose bilateral oblivious transfer protocol and show that proposed protocol solve selective failure attack and consisitecy inputs in addition our protocol is significantly more efficient and far simpler than previous works in computation complexity, symmetric encryption operations, bandwidth, and error probability by input recovery achive.

Keywords: Two-Party Ccomputation, Cut-and-Choose, Garbled Circuit, Oblivious Trasfer

*Corresponding Author Email: mahdiazizi@ihu.ac.ir

نشریه علمی "پدافند الکترونیکی و سایبری"
سال نهم، شماره ۳، پاییز ۱۴۰۰، ص ۳۷- ۲۱

علمی- پژوهشی

طراحی پروتکل محاسبات دوبخشی امن مبتنی بر انتقال کور دوطرفه

مهدی عزیزی[۱*]، سجاد قربانزاده هاوستین [۲]

۱- استادیار، ۲- دانشجوی کارشناسی ارشد دانشگاه امام حسین(ع)

(دریافت: ۱۳۹۹/۰۶/۲۲، پذیرش: ۱۳۹۹/۱۲/۰۴)

چکیده

پروتکل محاسبات امن دوبخشی، محاسبه مشترک تابع زمان چند جمله را برای دو عامل p_1 و p_2 با حفظ محرمانگی ورودی‌ها، میسر می‌کند. یائو[1] اولین پروتکل محاسبات امن دوبخشی، در الگوی عامل نیمه صادق را معرفی کرد. نشان داده شد که پروتکل یائو برابر مهاجم مخرب آسیب‌پذیر هست. برای برطرف شدن این آسیب‌پذیری روش برش- انتخاب در توسعه این پروتکل معرفی گردید. در پژوهش‌های بعدی نشان داده شد که استفاده از این روش از نظر پیچیدگی ارتباط و محاسبات، چالش‌هایی را ایجاد می‌کند. از مشکلات روش برش- انتخاب، تعداد مدارهای ساخته‌شده برای رسیدن به‌احتمال خطای موردنظر و آسیب‌پذیری در برابر حمله شکست انتخاب و سازگاری ورودی‌ها است. در این مقاله، پروتکل محاسبات دوبخشی امن مبتنی بر اولیه جدید انتقال‌کور برش- انتخاب دوطرفه بسط یافته، بر پایه مسئله سخت تصمیم دیفی هلمن طراحی‌شده است. نشان داده می‌شود پروتکل پیشنهادی نسبت به آسیب‌پذیری حمله شکست انتخاب و سازگاری ورودی‌ها مقاوم است، و همچنین نسبت به پروتکل‌های پیشین ازنظر مولفه‌های پیچیدگی محاسبات، تعداد عملیات رمزنگاری، پهنای باند نتایج بهبود یافته است. در طراحی پروتکل با استفاده از روش بازیابی ورودی بخش عامل سازنده مدار، احتمال خطای 2^{-8} برای پروتکل نیز ایجاد شده است که برای رسیدن به‌احتمال خطای 2^{-40} تعداد ۴۰ مدار کافی است.

کلید واژه‌ها: محاسبات دوبخشی امن، مدارهای مبهم، روش برش- انتخاب، انتقال کور

۱- مقدمه

محاسبات امن حجم زیادی از تحقیقات در حوزه رمزنگاری را به خود اختصاص می‌دهد. پتانسیل بالای پروتکل‌های محاسبات امن و فن‌های پیشرفته ارائه شده، جذابیت‌های فراوانی را برای تحقیق و توسعه پروتکل‌های محاسبات چند نهادی ایجاد کرده است. پروتکل‌های محاسبات چند نهادی در حوزه‌های مختلفی از جمله رأی‌گیری الکترونیکی، مزایده و مناقصه‌های الکترونیکی، دسترسی امن به پایگاه داده، حفظ حریم خصوصی و استخراج داده‌ها در ارتباط امن ماهواره‌ها و پهپادها کاربرد دارد [۱].

پروتکل‌ها محاسبات دوبخشی امن، پروتکلی برای دو بخش p_1 و p_2 با ورودی‌های خصوصی x و y برای محاسبه تابع f با حفظ مولفه‌های امنیتی مشخص، میسر می‌کند. مهم‌ترین این شاخص‌ها عبارت‌اند از:

حریم خصوصی[2]: هیچ کاربری نباید اطلاعاتی بیش از خروجی متناظر با خودش را دریافت کند، درعین‌حال محرمانگی ورودی‌ها حفظ گردد. برای مثال: در رأی‌گیری الکترونیکی رأی هر رأی‌دهنده باید مخفی بماند،

صحت[3]: پروتکل باید تمام کاربران را متقاعد کند، خروجی به‌دست آمده صحیح محاسبه شده است. برای مثال: در رأی‌گیری الکترونیکی باید نامزدی که رأی بیشتری دارد و یا در مزایده متقاضی که قیمت بالاتری پیشنهاد داده، پیروز شود.

استقلال ورودی‌ها[4]: به این معنی که هیچ یک از طرفین نتواند ورودی خودش را به‌عنوان تابعی از ورودی بخش دیگر انتخاب کند است.

محاسبه دوبخشی ابتدا در بحث مسئله میلیونر توسط یائو مطرح گردید [۲]، اولین پروتکل محاسبه دوبخشی در سال ۱۹۸۶ توسط خود یائو [۳ و ۴] ارائه گردید. پروتکل محاسبه دوبخشی یائو، در برابر مهاجم نیمه صادق امن، و از نظر کارآمدی، ساده و خالی از فرضیات سنگین ریاضیات است، اما در برابر مهاجم مخرب ناامن و آسیب‌پذیر است. اولین پروتکل دوبخشی امن در برابر مهاجم مخرب، با استفاده از پروتکل اثبات هیچ آگاهی[5]،

*رایانامه نویسنده مسئول: mahdiazizi@ihu.ac.ir

[1] Yao
[2] privacy
[3] Correctness
[4] Independence of Input
[5] Zero-Knowledge Proof

توسط گولدریچ در مقالات [۵ و ۶] ارائه شد، که از نظر کارآمدی غیرقابل اجرا و پیاده‌سازی‌اند و از نظر پیچیدگی محاسبات از فرض‌های سخت ریاضیاتی استفاده شده است. بنابراین بسیاری از پژوهشگران سعی در امن کردن پروتکل یائو در برابر مهاجم مخرب را دارند، که از نظر کارآمدی مناسب‌تر از پروتکل‌های ذکر شده است [۷].

برای مقاوم کردن پروتکل یائو در برابر رفتار مخرب و تضمین صحت مدارهای ساخته‌شده، روش برش- انتخاب[1] در مقاله‌ی [۸] معرفی شد. روش برش- انتخاب، یک روش احتمالی است، به عبارت بهتر درواقع با این روش احتمال عدم تشخیص فریب، به مقدار قابل قبولی کاهش پیدا می‌کند. روش برش- انتخاب به این شکل است که بخش p_1 که وظیفه ساختن مدار را دارد، باید تعداد s مدار مستقل از هم و متناظر با تابع f را بسازد و برای بخش p_2 بفرستد، سپس بخش p_2 به‌طور تصادفی تعدادی از s مدار دریافتی را انتخاب و از بخش p_1 تقاضا می‌کند تمام کلیدهای متناظر با مدارهای منتخب را برایش بفرستد. بخش p_2 با کلیدهای دریافتی مدارهای انتخاب شده را رمزگشایی، و درستی آن‌ها را با مقایسه خروجی مدار با خروجی تابع f بررسی می‌کند. در نتیجه مطابق با بحث احتمالات، می‌توان گفت مدارات باقی‌مانده نیز با احتمال بالایی صحیح و به‌درستی ساخته شده‌اند. اگرچه روش برش- انتخاب روشی بصری و ساده است اما با چالش‌ها و آسیب‌پذیری‌هایی که دارد روبرو هست. اولاً به‌دلیل مواجه با چندین مدار، طرفین باید روشی برای اطمینان از سازگاری[2] بین ورودی‌ها به‌کار گرفته شود (در غیر این صورت، همان‌طور که در ادامه نشان داده شده است، بخش مخرب می‌تواند اطلاعات بیشتر از خروجی کسب کند). دوماً، توصیف ارائه شده از برش- انتخاب بسیار مبهم است و در هنگام اجرای دانستن جزئیات حیاتی هست، سوماً بخش p_1 مخرب، ممکن است در پروتکل انتقال کور ورودی فاسدی[3] را ایجاد کند، به این شکل که، یکی از دو کلید متناظر با یک سیم ورودی بخش p_2 را نادرست بفرستد. اگر بخش p_2 متناظر با بیت ورودی خود، کلید فاسد را در پروتکل انتقال کور انتخاب کرده باشد، نمی‌تواند مدار مبهم را به‌درستی رمزگشایی کند بنابراین پروتکل را لغو[4] می‌کند؛ و اگر بخش p_2 کلید دستکاری شده را انتخاب نکرده باشد، متوجه فریب نخواهد شد، بنابراین بخش p_1 می‌تواند یکی از بیت‌های ورودی مخفی بخش p_2 را براساس ادامه پیدا کردن یا لغو پروتکل بفهمد، به این عمل حمله شکست انتخاب[5] گوییم.

[1] Cut-and-choose
[2] Consistency
[3] Corrupted
[4] Abort
[5] Selective Falure Attack

در ادامه مقاله در بخش دوم مفاهیم اولیه و مقدمات پروتکل و الگوی اثبات امنیت محاسبات چند نهادی را معرفی و در ادامه در فصل سوم با تعریف پروتکل انتقال کور برش- انتقال دوطرفه بسط یافته، پروتکل دوبخشی پیشنهادی را بیان می‌کنیم. در بخش بعدی اثبات امنیتی پروتکل و بررسی کارآمدی پروتکل را انجام می‌دهیم. در پایان نیز با مقایسه کارآمدی پروتکل پیشنهادی با پروتکل‌های پیشین، مزیت پروتکل پیشنهادی را اثبات می‌کنیم.

۲- مفاهیم اولیه

در این بخش به‌صورت خلاصه مفاهیم اولیه پروتکل‌های محاسبات چند نهادی از جمله پروتکل مدارهای مبهم[6]، انتقال کور[7] و نمونه‌های امنیتی را بیان می‌کنیم.

۲-۱- مدار مبهم

مدارهای مبهم روشی سنتی برای محاسبه امن تابع[8] (SFE) و برخی اهداف رمزنگاری است. در این نوع مدارها برای مخفی کردن مقدار صفر یا یک بودن بیت متناظر با هر سیم مدار بولی[9] دو رشته تصادفی در نظر می‌گیریم. در ادامه با فرض گیت‌هایی با دو سیم ورودی، یک جدول چهار ردیفی برای هر گیت تعریف می‌شود که کلیدهای هر ردیف برای رمزنگاری نامتقارنِ رشته سیم خروجی گیت استفاده می‌شود. مدار مبهم برای هر تابع دو ویژگی زیر را دارند:

- برای هر سیم مدار دو کلید مبهم، یکی متناظر با مقدار بیت صفر و دیگری متناظر با مقدار بیت یک، تعریف می‌شود.
- اگر برای هر سیم مدار، مقدار کلید مبهم متناظر با مقدار بیت سیم، داده شود می‌توان به‌طوری که هیچ اطلاعاتی به‌غیر از رشته خروجی آشکار نگردد مدار مبهم را محاسبه کرد.

۲-۲- انتقال کور

انتقال کور در محاسبات دوبخشی مورد توجه ویژه است؛ پروتکل انتقال کور یک از دو، را می‌توان به شکل $((x_1,x_2),\sigma) \to (x_\sigma)$ نمایش داد. به‌عبارت بهتر فرستنده زوج (x_1,x_2) و گیرنده مقدار بیت σ را به‌عنوان ورودی در اختیار دارند؛ و نتیجه پروتکل دستیابی گیرنده به مقدار x_σ (فقط x_σ)، بدون آشکار شدن مقدار بیت σ، است [۹] (در این مقاله برای اختصار از انتقال کور به‌جای

[6] Garbled Circuit
[7] Oblivious Transfer
[8] Secure Function Evaluation
[9] Boolean Circuit

انتقال کور یک از دو استفاده می‌شود). کلیت پروتکل انتقال کور یک از دو به‌صورت زیر است:

- ورودی‌ها
  - ورودی فرستنده دو رشته $x_1, x_2 \in \{0,1\}^n$ است.
  - ورودی گیرنده بیت σ است.
- خروجی
  - فرستنده خروجی ندارد.
  - گیرنده مقدار x_σ را به‌دست می‌آورد، بدون فهمیدن $x_{1-\sigma}$.

۲-۳- الگوهای امنیتی

در محاسبات دوبخشی مهاجمین با توجه به توانایی و اختیاراتی که دارند به دودسته مهاجم نیمه صادق و مخرب تقسیم می‌شوند.

مهاجم نیمه صادق [1] : بخش‌های فاسد شده توسط این نوع مهاجم به طور کامل از روند و دستورات پروتکل پیروی می‌کنند اما مهاجم با اطلاعاتی که از بخش فاسد شده به‌دست می‌آورد تلاش دارد تا به حریم خصوصی دیگر بخش‌ها تجاوز کند.

مهاجم مخرب [2]: بخش فاسد شده توسط مهاجم مخرب با روش‌های دلخواه سعی در منحرف کردن پروتکل و اخلال در تضمین ویژگی‌های امنیتی و تغییر خروجی به مقدار مدنظر خودش را دارد. در حالت کلی امنیت را در حضور مهاجم مخرب بررسی می‌کنیم تا اطمینان حاصل شود هیچ مهاجمی حمله موفقیت‌آمیزی نتواند انجام دهد [۹].

برای بررسی تعاریف ریاضی امنیت در حضور مهاجمین دانستن دو تعریف زیر حیاتی می‌باشد.

تعریف ۲-۱: با در نظر گرفتن n به‌عنوان طول ورودی و مولفه امنیتی، تابع $\mu(.)$ را در n ناچیز [3] گوییم اگر برای هر چند جمله‌ای $p(.)$ و مقادیر بزرگ n مقدار $\mu(n) < \frac{۱}{p(n)}$ برقرار باشد.

تعریف ۲-۲: فرض کنید برای مقدار نامتناهی s گروه‌های توزیع شده $X = \{X_s\}_{s \in S}$ و $Y = \{Y_s\}_{s \in S}$ را داریم. دو مجموعه

[1] Semi-Honest
[2] Malicious
[3] Negligible

X و Y را از نظر محاسباتی تمیزناپذیر [4] گوییم و با نماد $X \overset{c}{\equiv} Y$ نشان می‌دهیم اگر برای هر تابع تمایزگر زمان چندجمله‌ای احتمالی غیر یکنواختِ D و مقدار $s \in S$ به اندازه کافی بزرگ، رابطه $\left|Pr\left[D(X_s) = ۱\right] - Pr\left[D(Y_s) = ۱\right]\right|$ از نظر محاسباتی ناچیز باشد آنگاه X و Y را از نظر محاسباتی تمیزناپذیر گویند.

$$|\Pr[D(X_s) = 1] - \Pr[D(Y_s) = 1]| \le \mu(n)$$

روش استاندارد برای بررسی امنیت پروتکل چند نهادی، مقایسه خروجی بخش‌های صادق و مخرب [5]، در حالت اجرا واقعی [6] پروتکل با خروجی حالت اجرا ایده‌آل [7] است، که به دیاگرام واقعی/ ایده‌آل [8] شناخته می‌شود. به اجرا پروتکل توسط طرفین شرکت کننده نمونه واقعی و اجرا پروتکل با کمک بخش خارجی قابل اعتماد [9] و غیرقابل فاسد شدن، نمونه ایده‌آل گویند. در حالت ایده‌آل، تمام بخش‌ها ورودی خود را از طریق کانالِ امن به‌سادگی برای بخش قابل اعتماد می‌فرستند. سپس بخش قابل اعتماد طبق تابع انتخاب شده، محاسبه را انجام و خروجی متناظر با هر بخش شرکت‌کننده را به‌طور مستقل و مخفیانه برایش می‌فرستد. در حالت واقعی، هیچ بخش خارجی قابل اعتماد وجود ندارد و بخش‌ها پروتکل را به‌طور مشترک بدون کمک خارجی اجرا می‌کنند. اگر اجرا واقعی پروتکل به‌وسیله بخش‌ها بتواند تضمین کند که هیچ مهاجمی نیست که بتواند اطلاعاتی بیش از آنچه که در حالت ایده‌آل می‌تواند کسب کند، به‌دست آورد، در نتیجه می‌گویند پروتکل امن است [۱۰].

بخش‌های $p_۱$ و $p_۲$ به‌ترتیب ورودی x و y و مهاجم A ورودی کمکی z را انتخاب می‌کنند. بخش صادق p_j ورودی خود و بخش فاسد p_i که توسط مهاجم A کنترل می‌شود می‌تواند لغو [10] یا مقدار دلخواهی را به‌عنوان ورودی، به بخش قابل اعتماد بفرستد. بنابراین اگرورودی مهاجم A لغو باشد، بخش قابل اعتماد لغو را به‌عنوان خروجی را برای بخش صادق می‌فرستد، در غیر این صورت خروجی تابع را دریافت خواهد کرد . اجرا ایده‌آل تابع f با ورودی (x, y) و مقدار کمکی z و مولفه امنیتی n به‌صورت $IDEAL_{f,A(z),i}(x, y, n)$ نمایش داده می‌شود. در نمونه واقعی که طرفین، پروتکل π را اجرا می‌کنند. در این حالت، مهاجم A تمام پیام‌ها را به‌جای بخش فاسد ارسال می‌کند و می‌تواند یک راهبرد حمله زمان چند جمله‌ای دلخواه را دنبال کند. اجرا واقعی پروتکل را به صورت $REAL_{f,A(z),i}(x, y, n)$ نمایش می‌دهیم که

[4] Indistinguishable
[5] Malicious
[6] Real
[7] Ideal
[8] Ideal/Real Simulation Paradigm
[9] External Trusted Party
[10] Abort

به‌عنوان زوج خروجی بخش صادق و مخرب در نظر گرفته می‌شود [۹].

تعریف ۲-۳: فرض کنید f و π به‌ترتیب تابع و پروتکل باشند، می‌گوییم پروتکل π تابع f را بدون قطع و در حضور مهاجم مخرب، به‌طور امن محاسبه می‌کند اگر برای هر مهاجم زمان چندجمله‌ای احتمالی غیریکنواختِ A برای حالت واقعی، یک مهاجم زمان چند جمله‌ای احتمالی غیر یکنواخت s برای حالت ایده‌آل وجود داشته باشد به‌طوری که داشته باشیم:

$$\{IDEAL_{f,S(z),i}(x,y,n)\}_{x,y,z,n} \stackrel{c}{\equiv} \{REAL_{\pi,A(z),i}(x,y,n)\}_{x,y,z,n} \quad (۲-۱)$$

گویند پروتکل π تابع f را بدون لغو و به‌طور امن در حضور مهاجم مخرب، محاسبه می‌کند [۱۰].

۳- طرح پیشنهادی

در این بخش ابتدا اولیه انتقال کور برش- انتخاب دو طرفه بسط یافته را بیان و سپس پروتکل محاسبات دوبخشی پیشنهادی را ارائه می‌دهیم.

۳-۱- اولیه انتقال کور دوطرفه بسط یافته

انتقال کور برش- انتخاب دوطرفه، براساس طرح انتقال کور پیکرت[1] [۱۱] طراحی شده است. پروتکل انتقال کور پیکرت برپایه فرض مسئله سخت دیفی هلمن[2] و نمونه رشته مرجع مشترک[3] استوار است. رشته مرجع مشترک را با فرض g_0 به‌عنوان مولد گروه و $g_1 = (g_0)^y$ و $h_0 = (g_0)^a$ و $h_1 = (g_1)^b$، چهارتایی (g_0, g_1, h_0, h_1) در نظر گرفته می‌شود. اگر چهارتایی (g_0, g_1, h_0, h_1)، چهار تایی دیفی هلمن باشد آنگاه در پروتکل انتقال کور پیکرت [۱۱]، گیرنده هر دو مقدار ورودیِ فرستنده را دریافت، اما اگر چهارتایی دیفی هلمن نباشد تنها ورودی متناظر با بیت ورودی خود را به‌دست خواهد آورد. اگر رابطه $a \neq b$ یا برای سادگی، رابطه b = a + ۱ در چهارتایی (g_0, g_1, h_0, h_1) برقرار باشد، می‌توان اثبات کرد [۱۲]، چهارتایی $\left(g_0, g_1, h_0, \frac{h_1}{g_1}\right)$ چهارتایی دیفی هلمن است و گیرنده تنها توانایی بازیابی یکی از ورودی‌های فرستنده را خواهد داشت. حال اگر a = b باشد آن‌گاه چهارتایی (g_0, g_1, h_0, h_1) چهارتایی دیفی هلمن بوده و گیرنده امکان بازیابی هر دو مقدار ورودی فرستنده را دارد. یهودا[4] در مقاله [۱۲] با استفاده از این ایده، پروتکل انتقال کور برش- انتخاب را که در محاسبات دو بخشی استفاده می‌شود ارائه داد. در این مقاله با بسط دادن پروتکل انتقال کور برش- انتخاب دوطرفه [۱۳] به‌دنبال دو هدف زیر در پروتکل پیشنهادی هستیم.

۱- مقابله با حمله شکست انتخاب: نقطه ضعف اصلی پروتکل‌های محاسبات دوبخشی که سبب آسیب‌پذیری در مواجه با حمله شکست انتخاب[5] می‌شود، جدا بودن گام اجرا اولیه انتقال کور، از گام اجرا روش برش- انتخاب است [۱۲]. به بیان بهتر با فرض مخرب بودن بخش p_1، اگر چنان چه بخش p_1 در انتقال کور، کلیدهای متناظر با بیت صفر را در اولین سیم ورودی بخش p_2، در تمام مدارها را نادرست انتخاب و باقی‌مانده کلیدها را صحیح انتخاب کند. در ادامه اگر مقدار بیت اولِ ورودی بخش p_2 صفر باشد، کلیدهای نادرست و اگر یک باشد کلیدهای صحیح را دریافت خواهد کرد. با توجه به اینکه برای مدارهای کنترل[6] تمام زوج کلیدهای متناظر با هر سیم ورودی باید در اختیار بخش p_2 قرار گیرد، پس اگر بخش p_2 کلیدهای نادرست متناظر با اولین سیم ورودی‌اش در انتقال کور را دریافت کرده باشد با مقایسه کلیدهای در مدارات کنترل، رفتار مخرب بخش p_1 آشکار خواهد شد و بخش p_2 پروتکل را لغو خواهد کرد؛ اما اگر مقدار اولین بیت ورودی بخش p_2 یک باشد کلیدهای صحیح را دریافت و خروجی پروتکل نیز محاسبه خواهد شد. پس بخش مخرب p_1 با توجه به لغو یا پایان پروتکل می‌تواند مقدار اولین بیت ورودی بخش p_2 را بفهمد. با ارائه اولیه جدید انتقال کور برش- انتخاب، با اجرا برش- انتخاب و انتقال کور در یک مرحله، دیگر زمینه حمله شکست انتخاب از میان می‌رود. برای مثال اگر در یک سیم، کلید ورودی بخش p_2 نادرست باشد، از آنجا که حداقل یک مدار به‌عنوان مدار کنترل انتخاب خواهد شد و در مدارهای کنترل هر دو زوج کلید متناظر با هر سیم ورودی در اختیار بخش p_2 است پس رفتار فریبکارانه بخش p_1 برای بخش p_2 آشکار خواهد شد و بدون این‌که بخش p_1 از بیت ورودی بخش p_1 اطلاعاتی کسب کند، پروتکل را لغو خواهد کرد، بنابراین لغو یا ادامه پروتکل برای بخش p_1 سودی نخواهد داشت.

۲- سازگاری ورودی‌ها[7]: از چالش‌های اصلی روش برش- انتخاب، لزوم اثبات استفاده از ورودی یکسان در مدارات ارزیابی، است. به بیان دیگر، بخش p_1 می‌تواند کلیدهای متناظر با xهای متفاوتی را در مدارات ارزیابی استفاده کند. در پروتکل انتقال کور برش- انتخاب دوطرفه، با توجه به اینکه بخش p_1 قبل از فرستادن تمام کلیدهای مدار، از ارزیابی یا کنترلی بودن هیچ

[1] Peikert
[2] Decisional Diffie–Hellma
[3] Common Reference String
[4] Yehuda Lindell

[5] Selective Falure Attack
[6] Check Circuits
[7] Consistency

یک از مدارها اطلاع ندارد، پس با احتمال خیلی زیادی نمی‌داند کدام مدار برای ارزیابی یا کنترل انتخاب شده است در صورت استفاده از کلیدهای نادرست به احتمال زیاد افشاء خواهد شد.

در سال ۲۰۱۸ انتقال کور برش-انتخاب دوطرفه بر اساس فرض *DDH* معرفی گردید [۱۳]. هم کلیدهای متناظر با سیم‌های ورودی بخش p_1 و هم کلیدهای متناظر سیم‌های ورودی بخش p_2 را در یک پردازش، انتقال داده می‌شود. به بیان دیگر در یک دور، ورودی فرستنده دو زوج مرتب $\left(k_0^2,k_1^2\right)$ و $\left(k_0^1,k_1^1\right)$ و بیت τ، و ورودی گیرنده بیت σ و j است. در پایان پروتکل، گیرنده برای مدارهای کنترل که $j=0$ است تمام کلیدهای $k_0^1, k_1^1, k_0^2, k_1^2$ ، و برای مدارات ارزیابی که $j=1$ است مقادیر k_τ^1 و k_σ^2 را به‌دست خواهد آورد.

نمادهای پروتکل انتقال کور برش-انتخاب دو طرفه بسط یافته در جدول (۱) آمده است.

۳-۱-۱- پروتکل انتقال کور

ابتدا تابع *RAND* و توابع مشتق شده از آن را در جدول (۲) تعریف می‌کنیم. (با فرض گروه (G,q,g_0) و $g,h,\bar{g},\bar{h}\in G$ و انتخاب تصادفی مقادیر $s,t\leftarrow\mathbb{Z}_q$ تعریف می‌کنیم.)

جدول (۱): نمادهای پروتکل انتقال کور برش-انتخاب

نشانگر	نماد	نشانگر	نماد
بردار ورودی بخش p_2	$\vec{Z}$	شماره سیم	i
بردار ورودی بخش p_1	$\vec{Q}$	شماره مدار	j
مقدار ورودی مخفی بخش p_1	τ	تعهد جایگشت کلیدهای ورودی بخش p_1	m
عملیات تعهد	Com	مقدار ورودی مخفی بخش p_2	σ
بیت	b	مقدار مبهم سیم ورودی بخش p_1	$\mathbb{Z}_q$
بازگشایی تعهد	d	رشته بیت تصادفی	J
تعداد مدارهای مبهم	S	مقادیر تصادفی منتخب از گروه G	y,α

ورودی‌ها:

ورودی فرستنده: ورودی فرستنده شامل l بردار $\vec{z_1},\dots,\vec{z_l}$ به‌صورت $\vec{z_i}=\left((z_0^{i,1},z_1^{i,1}),(z_0^{i,2},z_1^{i,2}),\dots,(z_0^{i,s},z_1^{i,s})\right)$ برای کلیدهای متناظر با سیم‌های ورودی گیرنده، و l بردار $\vec{Q_1},\dots,\vec{Q_l}$ به‌صورت $\vec{Q_i}=\left((Q_0^{i,1},Q_1^{i,1},m^{i,1}),(Q_0^{i,2},Q_1^{i,2},m^{i,2}),\dots,(Q_0^{i,s},Q_1^{i,s},m^{i,s})\right)$ متناظر با سیم‌های ورودی فرستنده است. بیت‌های ورودی مخفی خودش $\tau_1,\dots,\tau_l$ و همچنین کلیدهای $key_1,\dots,key_s$ نیز ورودی فرستنده هستند.

ورودی گیرنده: رشته بیت ورودی مخفی خودش که با نماد $\sigma_1,\dots,\sigma_l$ و رشته $J\subset[s]$ که به‌صورت $b_1,\dots,b_s$ $b_j\in\{0,1\}$ نمایش می‌دهیم.

ورودی کمکی: (G,q,g_0)،گروه G با مرتبه q با طول n و مولد g_0 و تعهدات $com(Q_m^{1,1}),\dots,com(Q_m^{1,s}),\dots,com(Q_m^{l,s})$ و $com\left(Q_{1-m}^{1,1}\right),\dots,com\left(Q_{1-m}^{1,s}\right),\dots,com\left(Q_{1-m}^{l,s}\right)$ و $com\left(m^{1,1}\right),\dots,com\left(m^{1,s}\right),\dots,com\left(m^{1,s}\right)$ است.

➢ گام راه‌اندازی

۱- گیرنده به‌طور تصادفی $y\leftarrow\mathbb{Z}_q$ را انتخاب و مقدار $g_1=(g_0)^y$ را محاسبه می‌کند.

۲- گیرنده با اثبات هیچ آگاهی برای فرستنده اثبات می‌کند، لگاریتم گسسته g_1 که با g_0 رابطه دارد را می‌داند (اثبات هیچ‌آگاهی به شکل $\{((g_0,g_1),a)\mid g_1=(g_0)^a\}$).

۳- برای هر b_j که $j=1,\dots,s$ مقدار تصادفی $\alpha_j\leftarrow\mathbb{Z}_q$ را انتخاب و مقادیر $h_0^j=(g_0)^{\alpha_j}$ و $h_1^j=(g_1)^{\alpha_j+b_j}$ را محاسبه می‌کند. برای مدارات کنترل $b_j=0$ و برای مدارات ارزیابی $b_j=1$ است.

۴- گیرنده مقادیر $g_1, h_0^1, h_1^1, \dots, h_0^s, h_1^s$ را برای فرستنده می‌فرستد.

جدول (۲): مشتقات تابع RAND

۱	$RAND(g,h,\bar{g},\bar{h}) = (u,v)\ ;\ u = g^s * h^t\ ,v = \bar{g}^s * \bar{h}^t$
۲	$extenedRAND(g,h,g_1,h_1,\bar{g},\bar{h}) = ((u_0,v_0),(u_1,v_1))$ $RAND(g,h,\bar{g},\bar{h}) = (u_0,v_0)\ ;\ (u_1,v_1) = RAND(g_1,h_1,\bar{g},\bar{h})$
۳	$shrinkedRAND(g,h) = (u,v)$ $r \leftarrow z_q, \bar{g} = g^r, \bar{h} = h^r\ ; (u,v) = RAND(g,h,\bar{g},\bar{h})$
۴	$selfextendedRAND(g,h,g_1,h_1) = ((u_0,v_0),(u_1,v_1),(u_2,v_2))$ $(u_0,v_0) = shrinkedRAND(g,h)\ ;\ (u_1,v_1) = RAND(g,g_1,h,h_1)\ ; (u_2,v_2) = RAND(g,g_1,h,h_1)$
۵	$combinedRAND(g,h,g_1,h_1,\bar{g},\bar{h}) = ((u_0,v_0),(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_4,v_4))$ $((u_2,v_2),(u_3,v_3),(u_4,v_4)) = selfextendedRAND(g,h,g_1,h_1)$ $(u_3,v_3),(u_4,v_4) = extenedRAND(g,h,g_1,h_1,\bar{g},\bar{h})$

➢ گام انتقال

۱- گیرنده برای هر $i = 1, \dots, l$ به‌طور تصادفی $r_i \leftarrow \mathbb{Z}_q$ را انتخاب و $\bar{g}_\iota = (g_{\sigma_i})^{r_i}$ و $\overline{h_{\iota,1}}, \dots, \overline{h_{\iota,s}}$ را به‌شکل $\overline{h_{\iota,1}} = (h_{\sigma_i}^1)^{r_i}$، محاسبه می‌کند. و $(\bar{g}_\iota, \overline{h_{\iota,1}}, \dots, \overline{h_{\iota,s}})$ را برای فرستنده می‌فرستد.

۲- گیرنده با استفاده از اثبات هیچ‌آگاهی، اثبات می‌کند تمام چهارتایی‌های $\{(g_0, \bar{g}_i, h_0^j, \bar{h}_{i,j})\}_{j=1}^{s}$ یا $\{(g_1, \bar{g}_i, h_1^j, \bar{h}_{i,j})\}_{j=1}^{s}$ چهارتایی دیفی هلمن هستند.

۳- گیرنده برای هر $j = ۱, \dots, s$ به‌طور تصادفی $\rho_j \leftarrow \mathbb{Z}_q$ را انتخاب و $\tilde{h}_0^j = (h_0^j)^{\rho_j}$ را محاسبه و اگر $b_j = ۱$، $\tilde{h}_1^j = \left(\frac{h_1^j}{g_1}\right)^{\rho_j}$ در صورت $b_j = ۰$ نیز $\tilde{h}_1^j = (h_1^j)^{\rho_j}$خواهد بود. $(\tilde{h}_0^1, \tilde{h}_1^1, \dots, \tilde{h}_0^s, \tilde{h}_1^s)$ را برای فرستنده می‌فرستد.

۴- گیرنده با استفاده از اثبات هیچ‌آگاهی، اثبات می‌کند تمام از چهارتایی‌های $\{(g_0, g_1, \tilde{h}_0^j, \tilde{h}_1^j)\}_{j=1}^{s}$ چهارتایی دیفی هلمن هستند.

۵- فرستنده عملیات‌هایی که در ادامه آمده را انجام می‌دهد:

•

رای هر $i = ۱, \dots, l$ و $j = 1, \dots, s$ محاسبات زیر را انجام می‌دهد:

$$combinedRAND(g_0, h_0^j, g_1, h_1^j, \bar{g}_\iota, \overline{h_{\iota,J}}) = \left((u_0^{i,j}, v_0^{i,j}), (u_1^{i,j}, v_1^{i,j}), (u_2^{i,j}, v_2^{i,j}), (u_3^{i,j}, v_3^{i,j}), (u_4^{i,j}, v_4^{i,j})\right)$$

•

رای هر $j = 1, \dots, s$ مقادیر $w_0^{i,j} = v_0^{i,j} \times Q_\tau^{i,j}$ و $w_1^{i,j} = v_1^{i,j} \times Q_{1-\tau}^{i,j}$ و $w_2^{i,j} = v_2^{i,j} \times m_{i,j}$ و $w_3^{i,j} = v_3^{i,j} \times z_0^{i,j}$ و $w_4^{i,j} = v_4^{i,j} \times z_1^{i,j}$، را محاسبه کرده و به‌طور تصادفی زوج $(u_1^{i,j}, w_1^{i,j})$ ، $(u_0^{i,j}, w_0^{i,j})$ را جایگشت داده و به‌صورت $(u_1'^{i,j}, w_1'^{i,j})$ ، $(u_0'^{i,j}, w_0'^{i,j})$ نشان داده و $\left((u_2^{i,j}, w_2^{i,j}), (u_3^{i,j}, w_3^{i,j}), (u_4^{i,j}, w_4^{i,j})\right)$برای گیرنده می‌فرستد.

۶- برای هر $j = 1, \dots, s$ زوج $(u_5^j, v_5^j) = RAND(h_0^j, \frac{h_1^j}{g_1}, \tilde{h}_0^j, \tilde{h}_1^j)$ را به‌دست آورده و با محاسبه $w_5^j = v_5^j \times key_j$ مقدار زوج (u_5^j, w_5^j) را برای گیرنده می‌فرستد. ×

➢ گام خروجی

برای هر $i = ۱, \dots, l$ و $j = ۱, \dots, s$ ابتدا گیرنده مقادیر $d_0^{i,j} = \frac{w_0'^{i,j}}{(u_0'^{i,j})^{\alpha_j}}$ و $d_1^{i,j} = \frac{w_1'^{i,j}}{(u_1'^{i,j})^{\alpha_j}}$ را محاسبه می‌کند.

• اگر $b_j = ۰$

مقدار $m_{i,j} = \frac{w_2^{i,j}}{u_2^{i,j^{\alpha_j}}}$ را محاسبه کرده و تایید می‌کند که $\mathrm{com}(\mathrm{m_{i,j}})$ تعهد $m_{i,j}$ است و $\mathrm{com}\left(\mathrm{Q^{i,j}_{m_{i,j}}}\right)$ تعهد یکی از $\mathrm{d_0^{i,j}}$ و $d_1^{i,j}$، و $com\left(Q^{i,j}_{1-m_{i,j}}\right)$ تعهد مقدار دیگر است. و سپس با ترتیب $com\left(Q^{i,j}_{m_{i,j}}\right)$ و $com\left(Q^{i,j}_{1-m_{i,j}}\right)$ مقادیر $d_0^{i,j}$ و $d_1^{i,j}$ را نیز متناظر با صفر بودن ویک بودن مرتب می‌کند.

- اگر $\sigma_i = ۰$ ، $z_0^{i,j} = \frac{w_3^{i,j}}{\left(u_3^{i,j}\right)^{r_i}}$ و $z_1^{i,j} = \frac{w_4^{i,j}}{\left(u_4^{i,j}\right)^{r_i * y^{-1}}}$
- اگر $\sigma_i = ۱$ ، $z_0^{i,j} = \frac{w_3^{i,j}}{\left(u_3^{i,j}\right)^{r_i * y}}$ و $z_1^{i,j} = \frac{w_4^{i,j}}{\left(u_4^{i,j}\right)^{r_i}}$

• اگر $b_j = ۱$

- گیرنده تایید خواهد کرد که تنها یکی از $com\left(Q^{i,j}_{m_{i,j}}\right)$ و $com\left(Q^{i,j}_{1-m_{i,j}}\right)$ تعهد یکی از $d_1^{i,j}$ و $d_0^{i,j}$ است. و آن به‌عنوان $Q_\tau^{i,j}$ در نظر گرفته خواهد شد.
- اگر $\sigma_i = ۰$ ، $z_0^{i,j} = \frac{w_3^{i,j}}{\left(u_3^{i,j}\right)^{r_i}}$
- اگر $\sigma_i = ۱$ ، $z_1^{i,j} = \frac{w_4^{i,j}}{\left(u_4^{i,j}\right)^{r_i}}$
- $key_j = \frac{w_5^j}{\left(u_5^j\right)^{\rho_j}}$

۳-۱-۲- امنیت پروتکل انتقال کور

قضیه ۳-۱: با فرض برقراری مسئله سخت تصمیم دیفی هلمن (DDH) در گروه (G, g, q) و امن بودن پروتکل‌های اثبات هیچ‌آگاهی در حضور مهاجم مخرب، پروتکل ۳-۱-۱ تابع انتقال کور برش-انتخاب دوطرفه‌ی بسط یافته را به‌صورت امن در حضور مهاجم مخرب محاسبه خواهد کرد.

اثبات: امنیت پروتکل را طبق روش مقالات [۱۰, ۱۲] در نمونه ترکیبی[1] اثبات می‌کنیم برای این کار از فرض امن بودن پروتکل‌های هیچ آگاهی موجود در پروتکل ۳-۱-۱ بهره می‌بریم [۹]. برای اثبات امنیت پروتکل، شبیه‌ساز S را در نظر می‌گیریم به این‌صورت که شبیه‌ساز S مهاجم A را با ورودی‌ها و عمل‌های فریبکارانه‌اش، فراخوانی، و نمونه ایده‌آل را اجرا می‌کند. سپس نشان می‌دهیم توزیع خروجی پروتکل در حالت واقعی و ایده‌آل یکسان و غیر قابل تمیز است [۳]. برای اثبات امنیت باید امنیت پروتکل در دو حالت فاسد بودن گیرنده و فرستنده را بررسی کنیم.

گیرنده فاسد: فرض کنید مهاجم A گیرنده R را فاسد کرده و شبیه‌ساز S به‌صورت زیر عمل خواهد کرد:

۱- S ورودی $((g_0, g_1), y)$ که A برای تابع ایده‌آل اثبات هیچ‌آگاهی می‌فرستد، را دریافت و اگر رابطه $g_1 \neq (g_0)^y$ برقرار باشد، S لغو پروتکل را برای فرستنده شبیه‌سازی خواهد کرد در غیر این صورت، مقادیر $g_1, h_0^1, h_1^1, \dots, h_0^s, h_1^s$ را از A دریافت می‌کند.

۲- S برای هر $j = ۱, \dots, s$ با بررسی $h_1^j = \left(h_0^j\right)^y$ در صورت برابری، $b_j = 1$ و در غیر این‌صورت $b_j = 0$ قرار می‌دهیم.

۳- برای هر $i = ۱, \dots, l$ مقادیر $\left(\overline{g_\iota}, \overline{h_{\iota,1}}, \dots, \overline{h_{\iota,s}}\right)$ را از A دریافت می‌کند.

۴- S برای هر $i = 1, \dots, l$، گواهی r_i که A برای اثبات هیچ آگاهی دانش، استفاده کرده را دریافت می‌کند. اگر یکی از موارد $\left[\overline{g_\iota} = (g_0)^{r_i}, \overline{h_{\iota,1}} = \left(h_0{}^j\right)^{r_i}\right]$ یا $\left[\overline{g_\iota} = (g_1)^{r_i}, \overline{h_{\iota,1}} = \left(h_1{}^j\right)^{r_i}\right]$ برقرار نباشد، S لغو پروتکل را، شبیه‌سازی خواهد کرد و خروجی، مقدار خروجی A خواهد بود، در غیر این‌صورت برای هر i با فرض σ_i گام بعدی را اجراء خواهد کرد.

۵- S مقدار رشته J و $\sigma_1, \dots, \sigma_s$ را برای بخش مورد اعتماد می‌فرستد.

a. S برای هر $i = 1, \dots, l$ و $j = 1, \dots, s$ که در آن $b_j = 0$ است، زوج‌های $\left(z_0^{i,1}, z_1^{i,1}\right)$ و $\left(Q_0^{i,s}, Q_1^{i,s}, m^{i,s}\right)$ را دریافت می‌کند.

b. S برای هر $i = ۱, \dots, l$ و $j = ۱, \dots, s$ که در آن $b_j = 1$ است، مقادیر $Q_\tau^{i,j}$ و $z_{\sigma_i}^{i,j}$ را دریافت می‌کند.

c. S برای هر $j = ۱, \dots, s$ که در آن $b_j = ۱$، مقادیر $\mathrm{key_j}$ را به‌دست خواهد آورد.

۶- S گام انتقال را به شکل زیر شبیه‌سازی و برای A می‌فرستد.

a) برای هر $i = ۱, \dots, l$ و $j = ۱, \dots, s$ که در آن $b_j = 0$ زوج‌های $\left(u_0^{i,j}, w_0{}^{i,j}\right), \left(u_1{}^{i,j}, w_1{}^{i,j}\right), \left(u_2^{i,j}, w_2^{i,j}\right)$ $\left(u_3^{i,j}, w_3^{i,j}\right), \left(u_4^{i,j}, w_4^{i,j}\right)$ را همانند فرستنده صادق، محاسبه می‌کند و سپس دو مولفه $\left(u_0'^{i,j}, w_0'^{i,j}\right), \left(u_1'^{i,j}, w_1'^{i,j}\right)$ را جایگشت می‌دهد.

b) برای هر $\mathrm{i} = ۱, \dots, \mathrm{l}$ و $j = ۱, \dots, s$ که در آن $b_j = ۱$ است، $\left(u_{\sigma_i}^{i,j}, w_{\sigma_i}^{i,j}\right)$ و $\left(u_{\tau_i}^{i,j}, w_{\tau_i}^{i,j}\right)$ را محاسبه و مقادیر $\left(u_{1-\sigma_i}^{i,j}, w_{1-\sigma_i}^{i,j}\right)$ و $\left(u_{1-\tau_i}^{i,j}, w_{1-\tau_i}^{i,j}\right)$ و $\left(u_{2'}^{i,j}, w_{2'}^{i,j}\right)$ را به‌صورت تصادفی از G با طول n انتخاب می‌کند.

۷- S گام انتقال مربوط به key_j را به شکل زیر محاسبه، و برای A می‌فرستد.

a) S مقادیر $\left\{\tilde{h}_0^j, \tilde{h}_1^j\right\}_{j=1}^{s}$ و گواهی که A برای تابع هیچ آگاهی ایده‌آل می‌فرستد، را دریافت می‌کند. با بررسی

---
[1] Hybrid Model

گواهی در صورت صحت گواهی گام بعدی را اجرا در غیر این‌صورت لغو پروتکل را شبیه‌سازی می‌کند.

b) برای هر $j = ۱, \dots, s$ که در آن $b_j = ۱$ است، زوج (u_5^j, w_5^j) را محاسبه می‌کند.

c) برای هر $j = 1, \dots, s$ که در آن $b_j = 0$ است، زوج (u_5^j, w_5^j) را به‌صورت تصادفی انتخاب می‌کند.

۸- S هر آنچه مهاجم A به عنوان خروجی دریافت کند را خروجی شبیه‌ساز در نظر می‌گیرد.

ما ادعا می‌کنیم که توزیع خروجی ایده‌آل توسط شبیه‌ساز S، برابر با توزیع خروجی اجرا واقعی است. با نگاه دقیق‌تر به مراحل شبیه‌سازی، تنها تفاوت بین اجرا ایده‌آل و واقعی، در زوج‌های $(u_{1-\sigma_i}^{i,j}, w_{1-\sigma_i}^{i,j})$، $(u_{1-\tau_i}^{i,j}, w_{1-\tau_i}^{i,j})$، $(u_{2'}^{i,j}, w_{2'}^{i,j})$ و (u_5^j, w_5^j) است.

ابتدا از بررسی زوج (u_5^j, w_5^j) شروع می‌کنیم. واضح است شبیه‌ساز S برای زوج‌های (u_5^j, w_5^j) در $j = ۱, \dots, s$ که $b_j = ۰$ است، رابطه $h_1^j = (h_0^j)^y$ برقرار است. در نتیجه بردار (g_0, g_1, h_0^j, h_1^j) چهارتایی دیفی هلمن است، اما بردار $\left(g_0, g_1, h_0^j, \frac{h_1^j}{g_1}\right)$ دیفی هلمن نیست. حال در حالت واقعی، فرستنده مقدار (u_5^j, w_5^j) را به‌وسیله‌ی محاسبه تابع $RAND\left(g_0, g_1, h_0^j, \frac{h_1^j}{g_1}\right)$ تولید می‌کند. در نتیجه طبق پروتکل ۲-۱-۱ (u_5^j, w_5^j) به‌طور تصادفی بر روی گروه G محاسبه می‌شود بنابراین توزیع (u_5^j, w_5^j) برابر با توزیع مقداری که توسط S محاسبه می‌شود، است.

برای زوج‌های $(u_{1-\sigma_i}^{i,j}, w_{1-\sigma_i}^{i,j})$ و $(u_{1-\tau_i}^{i,j}, w_{1-\tau_i}^{i,j})$ و $(u_{2'}^{i,j}, w_{2'}^{i,j})$ که شرط $b_j = ۱$ برقرار است، طبق بالا عمل می‌کنیم و می‌دانیم که $h_1^j \neq (h_0^j)^y$ است بنابراین برای بردار $(\tilde{g}_i, \tilde{h}_{i,1}, \dots, \tilde{h}_{i,s})$ دو عبارت $\left[\tilde{g}_i = (g_0)^{r_i}, \tilde{h}_{i,j} = ({h_0}^j)^{r_i}\right]$ و $\left[\tilde{g}_i = (g_1)^{r_i}, \tilde{h}_{i,j} = ({h_1}^j)^{r_i}\right]$ نمی‌تواند همزمان برقرار باشد. شبیه‌ساز S با توجه به رابطه $\left[\tilde{g}_i = (g_{\sigma_i})^{r_i}, \tilde{h}_{i,1} = ({h_{\sigma_i}}^j)^{r_i}\right]$ مقدار σ_i را محاسبه می‌کند و در نتیجه رابطه $\left[\bar{g}_i = (g_{1-\sigma_i})^{r_i}, \overline{h_{i,1}} = ({h_{1-\sigma_i}}^j)^{r_i}\right]$ برقرار نخواهد بود و چهارتایی $(g_{1-\sigma_i}, \tilde{g}_i, h_{1-\sigma_i}, \tilde{h}_{i,j})$ دیفی هلمن نیست، پس توزیع خروجی شبیه‌ساز S برابر با خروجی چهارتایی دیفی هلمن خواهد بود.

فرستنده فاسد: فرض کنید مهاجم A فرستنده را فاسد، آن‌گاه شبیه‌ساز S به‌صورت زیر عمل خواهد کرد:

۱- شبیه‌ساز S مقدار $\mathbb{Z}_q \rightarrow y$ را انتخاب و $g_1 = (g_0)^y$ را محاسبه می‌کند و برای A می‌فرستد.

۲- شبیه‌ساز S برای A، اثبات می‌کند که لگاریتم گسسته g_1 را می‌داند.

۳- شبیه‌ساز S به‌عنوان گیرنده صادق، با انتخاب مقدار $b_j = ۱$ برای هر $j = ۱, \dots s$ و $\sigma_1 = \dots = \sigma_l = ۰$، عمل می‌کند:

a) برای هر $j = 1, \dots s$، S به‌طور تصادفی $\mathbb{Z}_q \rightarrow \alpha_j$ انتخاب و $h_0^j = (g_0)^{\alpha_j}$ و $h_1^j = (g_1)^{\alpha_j}$ را محاسبه و بردار $h_0^1, h_1^1, \dots, h_0^s, h_1^s$ را برای A می‌فرستد.

b) برای هر $i = ۱, \dots l$، S مقادیر $\bar{g}_i = (g_0)^{r_i}$ و $\overline{h_{i,j}} = (h_0^j)^{r_i}$ را محاسبه و بردار $(\bar{g}_i, \overline{h_{i,1}}, \dots, \overline{h_{i,s}})$ را برای A می‌فرستد.

۴- شبیه‌ساز S پس از دریافت $({u_0}^{i,j}, {w_0}^{i,j})$ $({u_1}^{i,j}, {w_1}^{i,j})$$(u_2^{i,j}, w_2^{i,j}), (u_3^{i,j}, w_3^{i,j}), (u_4^{i,j}, w_4^{i,j})$ از A، چون $b_j = ۱$ است، مقادیر $Q_0^{i,j}$ و $Q_1^{i,j}$ و $m_{i,j}$ و $z_0^{i,j}$ و $z_1^{i,j}$ را برای هر $j = ۱, \dots s$ و $\sigma_1 = \dots = \sigma_l = 0$ محاسبه می‌کند.

۵- شبیه‌ساز S برای شبیه‌سازی گام انتقال key_j، به‌صورت زیر عمل می‌کند:

a) برای هر $j = ۱, \dots s$ مقادیر $\tilde{h}_1^j = \left(\frac{h_1^j}{g_1}\right)^{\rho_j}$ و $\tilde{h}_0^j = (h_0^j)^{\rho_j}$ را محاسبه می‌کند.

b) شبیه‌ساز S اثبات هیچ‌آگاهی تمام $\{g_0, g_1, \tilde{h}_0^j, \tilde{h}_1^j\}_{j=1}^s$ در $j = 1, \dots s$ را به‌وسیله آنچه از تابع ایده‌آل هیچ آگاهی و از طریق A به‌دست آمده، شبیه‌سازی می‌کند.

c) شبیه‌ساز S زوج $(u_5^{i,j}, w_5^{i,j})$ را از A دریافت و گام انتقال key_j را انجام می‌دهد.

۶- تمام کلیدهای key_j را برای بخش قابل اعتماد می‌فرستد.

۷- شبیه‌ساز S خروجی که A به‌عنوان خروجی به‌دست آورده را مقدار خروجی در نظر می‌گیرد.

دو مطلب قابل ذکر در شبیه‌سازی گفته شده، وجود دارد. ابتدا به دلیل روش انتخاب g_0 و g_1، تمام بردارهای (g_0, g_1, h_0^j, h_1^j) و $(g_0, \bar{g}_i, h_0^j, \overline{h_{i,j}})$و $(g_1, \bar{g}_i, h_1^j, \overline{h_{i,j}})$، چهارتایی‌های دیفی هلمن خواهند بود. بنابراین S تمام کلیدهای متناسب و key_j را به‌دست‌خواهد آورد. ثانیا با فرض مسئله‌ی سخت DDH ، خروجی به‌دست آمده توسط S در حالت ایده‌آل با خروجی در اجرا واقعی بین A و گیرنده‌ی صادق، غیر قابل تمیز

است. دلیل این ادعا این است که تنها تفاوت بین حالت ایده‌آل و واقعی در موارد زیر است که بررسی می‌شوند:

۱- شبیه‌ساز S مقادیر $h_0^1, h_1^1, \dots, h_0^s, h_1^s$ را طوری انتخاب می‌کند که برای هر $j = 1, \dots, s$ رابطه $b_j = 0$ برقرار است و نیز مقادیر $\tilde{h}_0^1, \tilde{h}_1^1, \dots, \tilde{h}_0^s, \tilde{h}_1^s$ را طوری انتخاب می‌کند که برای هر $j = 1, \dots, s$، رابطه $b_j = 1$ برقرار باشد. از آنجا که مقادیر بالا بر پایه فرض مسئله سخت DDH هستند در نتیجه خروجی S با خروجی گیرنده صادق از نظر محاسباتی تمیز ناپذیر است.

۲- شبیه‌ساز S در عوض گیرنده، اثبات هیچ‌آگاهی دانش را اجرا می‌کند. با توجه به ویژگی اثبات هیچ‌آگاهی خروجی در دو حالت غیر قابل تمیز است [۱۲].

از آنجا که دو تفاوت ذکر شده ازنظر محاسباتی تمیزناپذیر هستند در نتیجه خروجی شبیه‌ساز S با خروجی گیرنده در حالت واقعی ازنظر محاسباتی غیر قابل تمیز هستند.

در پایان از آنجا که خروجی پروتکل در حالت ایده‌آل در هر دو مورد فاسدن شدن گیرنده و فرستنده از خروی پروتکل در حالت واقعی تمیز ناپذیر است پس پروتکل ۲-۱-۱ در برابر مهاجم مخرب امن است.

۳-۲- پروتکل محاسبه دوبخشی پیشنهادی

پروتکل محاسبه‌ی دوبخشی پیشنهادی، بر پایه‌ی اولیه‌ی انتقال کور برش- انتخاب دوطرفه که در بخش ۳-۱ ارائه شد و بازیابی ورودی بخش p_1 که اولین بار توسط یهودا در [۱۰] مطرح گردید، معرفی می‌شود.

طبق روش بازیابی ورودی‌ها در صورت اثبات فریب، بخش p_2 می‌تواند به صورت محلی ورودی مخفی x را بازیابی و سپس مقدار تابع $f(x,y)$ را محاسبه کند. جزئیات روش بازیابی به این شکل است که بخش p_1 تعداد s مدار را تولید و سپس با احتمال $\frac{1}{2}$ هر یک از مدارها را به عنوان مدار کنترل یا ارزیابی[1] توسط بخش p_2 انتخاب می‌شوند. در ادامه اگر بخش p_2 تمام مدارهای ارزیابی را محاسبه و خروجی یکسانی را به‌دست آورد، بخش p_2 خروجی صحیح را دریافت خواهد کرد و دیگر نخواهد توانست ورودی مخفی x را بازیابی کند. اما اگر پس از انجام محاسبه مدارهای ارزیابی، خروجی یکسانی در تمام مدارها ظاهر نشود، بخش p_2 خواهد توانست طبق روش بازیابی ورودی، مقدار مخفی x را بازیابی و تابع $f(x,y)$ را محاسبه کند.

طبق مطالب ذکر شده، خطای پروتکل زمانی رخ می‌دهد که پس از محاسبه‌ی تمام مدارات ارزیابی مقدار یکسان و نادرست در خروجی به‌دست آید، و همچنین تمام مدارات کنترل صحیح بوده باشند. در نتیجه با توجه به $\frac{1}{2}$ بودن احتمال انتخاب مدارات ارزیابی و کنترل، احتمال خطای پروتکل با توجه به روش بازیابی ورودی، مقدار 2^{-s} است. همان‌طور که مشخص است برای رسیدن به احتمال 2^{-40} وجود تعداد ۴۰ مدار کافی است. در ادامه کلیت پروتوکل را تشریح و سپس جزئیات پروتکل را بیان می‌کنیم.

در پروتکل پیشنهادی بخش p_1 وظیفه‌ی ساختن مدارهای مبهم را برعهده دارد. برای مبهم سازی مدار، باید برای هر سیم مدار دو رشته تصادفی متناظر با بیت صفر و یک، مشخص گردد. روش تولید رشته‌های تصادفی سیم‌های ورودی و خروجی اهمیت زیادی دارد. بخش p_1 برای تولید کلیدهای متناظر با ورودی خودش ابتدا تابع شبه تصادفی PRF (تابع شبه تصادفی PRF برای ورودی و بذر ثابت خروجی یکسانی تولید خواهد کرد) را تعیین می‌کند. سپس با استفاده از بذر $seed_j$ برای $j = 1, \dots, s$ برای هر یک از مدارها، کلیدهای متناظر با بیت $b \in \{0,1\}$ و سیم $i = 1, \dots, l$ در مدار $j = 1, \dots, s$ را به‌شکل $A_{i,j,b} = PRF_{seed_j}(i,b)$ محاسبه خواهد شد. دلیل استفاده از تابع شبه تصادفی در تولید کلیدهای متناظر با ورودی بخش p_1، بازیابی ورودی در صورت اثبات فریب است که در ادامه تشریح خواهد شد. کلیدهای متناظر با ورودی بخش p_2 نیز به‌صورت تصادفی انتخاب خواهند شد. مقادیر مبهم هر یک از سیم‌های خروجی مدار نیز با نماد $z_{i,j,b}$ نشان خواهیم داد.

پروتکل انتقال کور برش- انتخاب دو طرفه، مهمترین ایده ما در ارائه‌ی یک پروتکل محاسبات دو بخشی است. با استفاده از این ایده، چالش‌هایی که روش برش- انتخاب ایجاد می‌کند را بدون نیاز به روش‌های دیگر که موجب افزایش دور و پیچیدگی پروتکل می‌شود، بر طرف خواهیم کرد.

اولین چالش، مسئله حمله شکست انتخاب است؛ دلیل و زمینه‌ی اصلی حمله‌ی شکست انتخاب همان‌طور که [۱۲] بیان شده در جدا بودن گام انتقال کور و گام برش- انتخاب، از هم است، به شکلی که بررسی مدارهای کنترل نیز کمکی برای مقابله

[1] Evaluation Circuit

با این حمله نمی‌کند [۱۲]. برای حل این مشکل برای اولین بار در سال ۲۰۱۱، انتقال کور برش- انتخاب ارائه و سپس پروتکل انتخاب برش- انتخاب دوطرفه، نیز در برای اولین بار در مقاله [۱۳] مطرح گردید. پروتکل انتقال کور برش- انتخاب دوطرفه‌ی بسط یافته که در بخش ۳- ۱ بیان گردید بهبود یافته پروتکل‌های پیشین است که پروتکل پیشنهادی را در برابر حمله شکست انتخاب امن خواهد بود.

دومین چالش لزوم وجود روشی برای تضمین سازگاری ورودی‌ها است. با توجه به وجود s مدار مبهم، بخش p_1 باید سازگاری ورودی‌هایش را برای بخش p_2 اثبات کند. در پروتکل محاسبه دوبخشی پیشنهادی، با استفاده از پروتکل ۳-۱-۱ و روش تولید کلید و بازیابی ورودی، سازگاری ورودی بخش p_1 تضمین می‌گردد. عامل اصلی ضرورت تضمین سازگاری ورودی‌ها، جدا بودن گام برش- انتخاب و گام فرستادن کلیدهای بخش p_1 است. در پروتکل پیشنهادی با ادغام کردن این دو گام در پروتکل انتقال کور برش- انتخاب دوطرفه سبب می‌شویم، بخش p_1 قبل از فرستادن کلیدهای متناظر با ورودی خودش، از کنترل یا ارزیابی بودن مدار بی‌خبر باشد، و در نتیجه اگر کلیدهای متناظر با یک سیم ورودی را نادرست بفرستد با احتمال $\frac{1}{2}$ این رفتار مخرب آشکار خواهد شد، و اما اگر بخش p_2 رفتار مخرب را شناسایی نکرد، آنگاه حداقل در یک سیم خروج، در دو مدار دو مقدار خروجی ناسازگار به‌دست خواهد آورد که یا بخش p_2 خواهد توانست ورودی بخش p_1 را بازیابی یا مدار به عنوان مدار ناصحیح تلقی شده و بخش p_2 از پروتکل خارج خواهد شد. در نتیجه با استفاده از پروتکل انتقال کور برش- انتخاب دوطرفه‌ی احتمال موفقیت در ارائه‌ی ورودی‌های ناسازگاری، ناچیز و قابل چشم پوشی است. با استفاده از این روش، با حذف گام اثبات سازگاری، پیچیدگی دور و محاسبات پروتکل کاهش چشم‌گیری خواهد کرد و از طرفی احتمال موفیقت در ارائه‌ی ورودی ناسازگار هم ناچیز است.

در روش بازیابی ورودی بخش p_1 در صورت اثبات فریب، ابتدا بخش p_1 برای هر بیت از ورودی مخفی x که با $x[i]$ نمایش می‌دهیم، دو مقدار تصادفی $N_{i,1}$ و $N_{i,0}$ به‌طور تصادفی انتخاب و با استفاده از انتقال کور با بخش p_1 تبادل خواهد کرد. در انتقال کور ورودی بخش p_1 مقدار مخفی x و ورودی بخش p_2 نیز مقادیر $N_{i,1}$ و $N_{i,0}$ است. بخش p_1 متناظر با مقدار $x[i]$، مقدار $N_{i,x[i]}$ را دریافت خواهد کرد. سپس بخش p_1 با استفاده از $N_{i,x[i]}$ برای هر سیم i در مدار j با استفاده از تابع شبه تصادفی، مقدار $R_{j,i,[i]}$ را به‌صورت $R_{j,i,[i]} = PRF_{seed_j}(i,R) \oplus N_{i,x[i]}$ محاسبه و با کلید key_j رمز کرده و به شکل $E_{key_j}(R_{j,i,[i]})$ برای بخش p_2 می‌فرستد. بخش p_2 برای مدارهای ارزیابی، طبق گام دوم خواهد توانست $R_{j,i,[i]}$ را رمزگشایی کند. بخش p_1 همچنین برای هر سیم خروجی $i \in [n_3]$ مقدار Δ و $\Delta_{i,0}$ را انتخاب و $\Delta_{i,1} = \Delta_{i,0} \oplus \Delta$ را محاسبه خواهد کرد. سپس تمام مقادیر $H(\Delta_{i,b})$ را برای بخش p_2 می‌فرستد. مقدار $\Delta_{i,b}$ را برای هر سیم خروجی i را با $z_{j,i,b}$ رمز کرده

و به‌شکل $T_{j,i,b} = E_{z_{j,i,b}}(\Delta_{i,b})$ نشان می‌دهد. و مقدار $T_{j,i,b}$ را تعهد کرده و گواهی را با کلید key_j رمز کرده به‌صورت زوج $\left(c_j^T, E_{key_j}(d_j^T)\right)$ برای بخش p_2 می‌فرستد. بخش p_2 با محاسبه مدارهای ارزیابی و به‌دست آوردن مقدار $z_{j,i,b}$ در هر سیم خروجی $i \in [n_3]$ و مدار j، و مقایسه $H(Dec_{z_{j,i}}(T_{j,i,b}))$ با $H(\Delta_{i,b})$ برای $b \in \{0,1\}$ مقدار $z_j(i) = b$ را مشخص می‌کند. اگر برای حداقل یک سیم خروجی دو مقدار $z_{j,i}$ متفاوت به‌دست آید، بخش p_2 خواهد توانست Δ را محاسبه کند، و با مساوی قرار دادن $\Omega = \Delta$ با روشی که در ادامه تشریح داده می‌شود، خواهد توانست ورودی مخفی بخش مخرب p_1 را به‌دست آورد.

بخش p_2 بردار $(h, g_1, h_1) = (g^w, g^r, h^r\Omega)$ (به‌طوری که $w, r \in F_q$) را برای بخش p_1 می‌فرستد. سپس بخش p_1 برای هر مدار j مقادیر $s_j, t_j \in F_q$ را انتخاب و $C_j = g_1^{s_j} * h^{t_j}$ و $D_j = g_1^{s_j} * \left(\frac{h}{\Delta}\right)^{t_j}$ را محاسبه و C_j و $E_{D_j}(seed_j)$ را برای بخش p_2 می‌فرستد. اگر بخش p_2 مقدار Ω را برابر Δ قرار داده باشد، بخش p_2 مقدار x را با مقایسه رابطه $R_{j,i,x[i]} = PRF_{seed_j}(i,R) \oplus N_{i,0}$ و $R_{j,i,x[i]} = PRF_{seed_j}(i,R) \oplus N_{i,1}$ صفر یا یک بودن مقدار $x[i]$ مشخص و در غیر این‌صورت، $x[i] = \perp$ یا اگر دو مقدار x_j متفاوت به‌دست‌آید بخش p_2 از پروتکل خارج خواهد شد. در صورت به‌دست اوردن مقدار x، بخش p_2 خواهد توانست تابع $f(x,y)$ را محاسبه کند. در نتیجه مشخص است که تنها زمانی فریب بخش p_1 موفق است که تمام مدارهای کنترل صحیح، تمام مدارات ارزیابی نادرست و خروجی‌ها نیز یکسان باشند؛ با توجه با $\frac{۱}{۲}$ بودن احتمال انتخاب مدار، احتمال فریب موفیقت آمیز برابر است با ۲^{-s} است.

۳-۲-۱- جزئیات پروتکل پیشنهادی

نمادهای استفاده شده در پروتکل دوبخشی پیشنهادی در جدول (۳)، آمده است.

جدول (۳): نمادهای پروتکل دوبخشی پیشنهادی

نشانگر	نماد	نشانگر	نماد
بردار ورودی بخش p_2	$\vec{Z}$	شماره سیم	i
بردار ورودی بخش p_1	$\vec{Q}$	شماره مدار	j
مقدار جایگشت ورودی بخش p_1	m	مقدار بیت	b
عملیات تعهد	Com	تابع شبه تصادفی	PRF
مقدار تصادفی تولیدی بخش p_2	N	مقدار مبهم سیم ورودی بخش p_1	A
مولفه کمکی بازیابی ورودی	R	مقدار مبهم سیم خروجی	Z
بازگشایی تعهد	d	ورودی مخفی بخش p_1	x
چکیده ساز	H	رمزگشایی	Dec
مولفه بازیابی ورودی	$\boldsymbol{\Omega}$	بذر تابع شبه تصادفی	seed
تعهد	C	کلید رمزنگاری مولفه‌های هر مدار	Key
مولفه کمکی بازیابی ورودی	T	رمزنگاری	E

۱- بخش p_1 برای تولید کلیدهای متناظر با ورودی خودش، برای هر مدار $j\ (j=1,\dots,s)$ مقدار بذر $seed_j$ را به‌صورت تصادفی انتخاب، و سپس با استفاده از تابع شبه تصادفی PRF و بذر $seed_j$، برای هر سیم $i(i=0,\dots,l)$ و بیت $b\in\{0,1\}$، کلید مبهم را به شکل $A_{i,j,b}=PRF_{seed_j}(i,b)$ تولید، و کلیدهای متناظر با ورودی بخش p_2 را نیز به‌طور تصادفی انتخاب می‌کند، در ادامه نیز برای هر یک از سیم‌های خروجی، رشته بیت $z_{i,j,b}$ را به‌طور تصادفی در نظر می‌گیرد. در پایان بخش p_1 تعداد s مدار مستقل کپی، از مدار مبهم c را با کلیدهای تولید شده، تعریف می‌کند.

۲- طرفین، پروتکل انتقال کور برش- انتخاب بسط یافته که در بخش ۳-۱ تشریح شد را با ورودی‌های زیر اجرا می‌کنند. ورودی فرستنده: شامل l بردار $\overrightarrow{z_1},\dots,\overrightarrow{z_l}$ که $\overrightarrow{z_l}=((z_0^{i,1},z_1^{i,1}),(z_0^{i,2},z_1^{i,2}),\dots,(z_0^{i,s},z_1^{i,s}))$ برای کلیدهای متناظر با سیم‌های ورودی گیرنده و l بردار $\overrightarrow{Q_1},\dots,\overrightarrow{Q_l}$ که $\overrightarrow{z_l}=((Q_0^{i,1},Q_1^{i,1},m^{i,1}),\ (Q_0^{i,2},Q_1^{i,2},m^{i,2}),\dots,(Q_0^{i,s},Q_1^{i,s},m^{i,s}))$ و بیت‌های ورودی مخفی خودش $\tau_1,\dots,\tau_l$ و همچنین کلیدهای $key_1,\dots,key_s$ است. ورودی گیرنده: رشته بیت ورودی مخفی خودش که با نماد $\sigma_1,\dots,\sigma_l$ و رشته $J\subset[s]$ که به‌صورت $b_1,\dots,b_s$ $b_j\in\{0,1\}$ به‌طوری که احتمال یک شدن هر بیت برابر با $\frac{1}{2}$ است، نمایش می‌دهیم.

۳- بخش p_2 برای هر بیت از ورودی مخفی بخش p_1 دو مقدار تصادفی $N_{i,0}$ و $N_{i,1}$ $i=1,\dots,n$ انتخاب می‌کند. سپس طرفین برای هر بیت ورودی مخفی بخش p_1 یک پروتکل انتقال کور با ورودی‌های تعریف شده، اجرا می‌کنند. ورودی بخش p_1 مقدار بیت ورودی مخفی $x(i)$ و ورودی بخش p_2 $N_{i,0}$ و $N_{i,1}$ است. در پایان، بخش p_1 برای هر بیت ورودی $x(i)$ یک مقدار تصادفی $N_{i,x(i)}$ را به‌دست می‌آورد.

۴- بخش p_1 برای هر سیم ورودی خودش $i(i=0,\dots,n)$ در هر مدار $j\ (j=1,\dots,s)$، متناظر با مقدار بیت $x(i)$ مقدار $R_{j,i,x(i)}=PRF_{seed_j}(i,x(i))\oplus N_{i,x(i)}$ را محاسبه و با کلید key_j به شکل $E_{key_j}(R_{j,i,x(i)})$ را برای بخش p_2 می‌فرستد. در ادامه برای بخش p_1 مقدار رشته‌ی Δ و برای هر $i\in[n_3]$ مقدار $\Delta_{i,0}$ به‌طور تصادفی انتخاب و $\Delta_{i,1}=\Delta_{i,0}\oplus\Delta$ را محاسبه می‌کند. سپس برای هر سیم خروجی $i\in[n_3]$ و مدار $j\ (j=1,\dots,s)$ با استفاده از کلیدهای سیم خروجی $z_{j,i,b}$ مقدار $T_{j,i,b}=E_{z_{j,i,b}}(\Delta_{i,b})$ محاسبه و به شکل $(c_j^T,d_j^T)\leftarrow com(\{T_{j,i,b}\}_{i\in[n_3],b\in\{0,1\}}$ تعهد را ایجاد و گواهی تعهد را با کلید key_j رمز می‌کند. در پایان موارد $GC_j,c_j^T,En_{key_j}(d_j^T)$ برای $j\ (j=1,\dots,s)$ و برای هر سیم خروجی $H(\Delta_{i,b})$ $(i\in[n_3])$ را برای بخش p_2 می‌فرستد.

۵- محاسبه مدارات ارزیابی: ابتدا بخش p_2 برای تمام مدارهای ارزیابی که در پروتکل انتقال کور برش- انتخاب دوطرفه مقدار کلید key_j را به‌دست‌آورده است، با رمزگشایی d_j^T درستی تعهد $com(\{T_{j,i,b}\})$ را تایید می‌کند، در صورت عدم تایید از پروتکل خارج می‌شود. سپس با استفاده از کلیدهایی که در گام دوم به‌دست آورده مدار را محاسبه و در سیم‌های خروجی مقادیر مبهم $z_{j,i}$ را کسب خواهد کرد. اگر $H(Dec_{z_{j,i}}(T_{j,i,b}))$ با مقدار $H(\Delta_{i,b})$ برای $b \in \{0,1\}$ مطابقت داشته آنگاه $z'_j(i) = b$ در غیر این‌صورت $z'_j(i) = \perp$ است و سه حالت زیر رخ خواهد داد.

- مدار نامعتبر: اگر برای هر مدار ارزیابی، سیم خروجی i $z'_j(i) = \perp$ باشد، پس بخش p_2 مقادیر $z = \perp$ و $\Omega = 1$ را تایید می‌کند.

- خروجی ناسازگار: اگر برای برخی سیم خروجی i در دو مدار ارزیابی j_1, j_2 بخش p_2 مقادیر $z'_{j_1}(i) = 0$ و $z'_{j_2}(i) = 1$ باشد، سپس بخش p_2 مقدار $\Omega = Dec_{z_{j_1,i}}(T_{j,i,0}) \oplus Dec_{z_{j_2,i}}(T_{j,i,1})$ را تعیین می‌کند؛ اگر برای یک سیم خروجی Ω های متفاوت باشند آنگاه $z = \perp$ قرار داده می‌شود.

- خروجی سازگار: اگر برای تمام سیم خروجی i مقادیر $z'_j(i)$ یکسان باشد خواهیم داشت $z(i) = z'_j(i)$.

۶- بخش p_2 مقادیر $\omega, r \in f_q$ را انتخاب و بردار $(h, g_1, h_1) = (g^{\omega}, g^{r}, h^{r}\Omega)$ را محاسبه و برای بخش p_1 می‌فرستد. سپس بخش p_1 Δ و $\{\Delta_{i,b}\}_{i \in n_3, b \in \{0,1\}}$ را برای بخش p_2 می‌فرستد. بخش p_2 بررسی می‌کند که آیا $\{\Delta = \Delta_{i,1} \oplus \Delta_{i,0}\}_{i \in n_3}$ با مقدار $H(\Delta_{i,b})$ برابر است یا خیر؟ در صورت عدم برابری از پروتکل خارج می‌شود. بخش p_1 مقادیر $s_j, t_j \in f_q$ را انتخاب و دو مقدار C_j, D_j را با شکل $C_j = g^{s_j} h^{t_j}$ و $D_j = g_1^{s_j} \left(\frac{h_1}{\Delta}\right)^{t_j}$ محاسبه می‌کند و مقادیر $E_{D_j}(seed_j)$ و C_j را برای بخش p_2 می‌فرستد.

۷- اگر بخش p_2 مقدار Ω را برابر Δ قرار داده باشد، بخش p_2 خواهد توانست بذر $seed_j$ را به‌دست آورده و مقدار x را به این شکل که اگر رابطه $R_{j,i,x[i]} = PRF_{seed_j}(i,R) \oplus N_{i,0}$ برقرار بود، $x[i] = 0$ و اگر رابطه $R_{j,i,x[i]} = PRF_{seed_j}(i,R) \oplus N_{i,1}$ باشد، $x[i] = 1$ است. در غیر این‌صورت، $x[i] = \perp$ یا اگر دو مقدار x_j متفاوت به‌دست‌آید بخش p_2 از پروتکل خارج خواهد شد.

۸- بررسی درستی مدارات کنترل: بخش p_2 با استفاده از کلیدهایی که در گام دوم دریافت کرده درستی مدارهای کنترل را بررسی و رشته بیت‌های $z_{i,j,b}$ را به‌دست می‌آورد. در صورت عدم تایید درستی مدار، پروتکل را قطع می‌کند. بخش p_2 در مدارهای کنترل با استفاده $z_{i,j,b}$ ازمقدار $T_{j,i,b}$ را محاسبه و برابری $H(Dec_{z_{j,i}}(T_{j,i,b}))$ و $H(\Delta_{i,b})$ را بررسی می‌کند. برای مدارهای کنترل گام دوم، درستی تعهدهای $com(m_{i,j})$ و $com\left(Q^{i,j}_{m_{i,j}}\right)$ و $com\left(Q^{i,j}_{1-m_{i,j}}\right)$ باید تایید گردد. در صورت عدم تایید بخش p_2 از پروتکل خارج می‌شود، در غیر این‌صورت مقدار z را به‌عنوان خروجی در نظر می‌گیرد.

۴- امنیت و کارآمدی پروتکل پیشنهادی

بررسی امنیتی و کارآمدی پروتکل پیشنهادی به شرح زیراست:

۴-۱- اثبات امنیت پروتکل پیشنهادی

قضیه ۴-۱: با فرضیات، DDH بودن گروه (G, g, q)، پنهان‌سازی/ انقیاد[1] بودن طرح تعهد com، مقاومت تابع H در برابر تصادم، مبهم سازی مدارها طبق [۱۲] امن باشد و امن بودن پروتکل انتقال کور، پروتکل ۳-۲-۱ تابع f را در حضور مهاجم مخرب، با احتمال خطای $2^{-s} + \mu(.)$ ، به‌طور امن محاسبه می‌کند.

اثبات: قضیه ۴- ۱ در نمونه ترکیبی با در نظر گرفتن اینکه تابع انتقال کور و تابع اثبات هیچ‌آگاهی توسط بخش سوم مورد اعتماد، محاسبه می‌شود، اثبات می‌گردد. برای اثبات امنیت به‌طور جداگانه امنیت را در دو حالت فاسد شدن بخش p_1 و بخش p_2 بررسی می‌شود.

بخش p_1 فاسد: تنها رفتار فریبکارانه بخش p_1 تولید مدارهای نادرست است. برای موفقیت فریب باید تمام مدارهای کنترل معتبر، و تمام مدارهای ارزیابی نامعتبر و دارای خروجی یکسان، باشند. در غیر این‌صورت بخش p_2 با شناسایی فریب، پروتکل را لغو می‌کند و یا به‌وسیله‌ی بازیابی ورودی x خروجی صحیح را به‌دست می‌آورد.

فرض کنید مهاجم A، بخش p_1 را به کنترل خود در می‌آورد. طبق نمونه اثبات ترکیبی، انتقال کور و انتقال کور دوطرفه توسط بخش قابل اعتماد، اجرا می‌شود. ما یک شبیه‌ساز S که در حالت ایده‌آل با تعامل با بخش قابل اعتماد، تابع f را محاسبه می‌کند، را در نظر می‌گیریم. شبیه‌ساز S به‌صورت درونی[2] مهاجم A را اجرا،

[1] Hiding/Binding
[2] Internally

و نقش بخش $p_۲$ صادق را در برابر مهاجم A ایفا می‌کند. سپس به‌صورت خارجی با تعامل با بخش مورد اعتماد، تابع f را محاسبه می‌کند. شبیه‌ساز S به‌صورت زیرعمل می‌کند:

۱- شبیه‌ساز S به‌عنوان بخش صادق با ورودی $y = 0^l$، طبق نمونه ترکیبی، با مهاجم A برای اجرا پروتکل تعامل می‌کند.

۲- با فرض این‌که $x = \tau_1, \dots, \tau_l$ ورودی مهاجم A در انتقال کورِ برش- انتخاب گام دوم و انتقال کور گام سوم است، طبق نمونه ترکیبی، شبیه‌ساز S به‌طور مستقیم مقدار x را از مهاجم A دریافت می‌کند.

۳- اگر بخش $p_۲$ از پروتکل خارج شود، شبیه‌ساز S مولفه $\perp$ را برای تابع ایده‌آل محاسبه‌ی f می‌فرستد، در غیر این‌صورت مقدار x را خواهد فرستاد.

از آنجا که ورودی بخش $p_۲$ تنها در انتقال کورِ برش- انتخاب مورد استفاده قرار می‌گیرد، بنابراین منظره[1] A در حالت ایده‌آل برابر با منظره A در اجرا واقعی است، همچنین توزیع خروجی شبیه‌ساز S در حالت ایده‌آل برابر با توزیع خروجی مهاجم A در حالت اجرا واقعی پروتکل ۲-۲-۱ است. اما برای اثبات امنیت باید نشان دهیم توزیع خروجی بخش $p_۲$ در حالت ایده‌آل غیر قابل تمیز با توزیع خروجی بخش $p_۲$ در حالت واقعی است. حال اگر بخش $p_۲$ در اجرا واقعی پروتکل، پروتکل را لغو کند، شبیه‌ساز S نیز $\perp$ را برای بخش قابل اعتماد خواهد فرستاد. بنابراین توزیع برابری را خواهند داشت. حال باید نشان دهیم، وقتی بخش $p_۲$ پروتکل را لغو نمی‌کند، خروجی آن با احتمال خطای $2^{-s} + \mu(.)$ برابر با مقدار $f(x,y)$ است که در آن x متناسب با مقداری که توسط شبیه‌ساز S فرستاده شده، است.

ابتدا هر مدار مبهمی که کلیدهای متناظر با ورودی بخش‌های $p_۱$ و $p_۲$ آن را باز نکند، مدار نادرست می‌نامیم. طبق پروتکل ۳-۲-۱، واضح است بعد از گام چهارم هر یک از مدارها صحیح و یا ناصحیح هستند و دیگر تغییر نخواهند کرد، به‌دلیل اینکه کلیدهای سیم‌های ورودی و خروجی، تعهدات و مولفه‌های کمکی مشخص و غیر قابل تغییر خواهند بود. به‌طور واضح مشخص است حتی اگر یک مدار کنترل نادرست باشد، بخش $p_۲$ از پروتکل خارج می‌شود. در ضمن ما ادعا می‌کنیم که اگر یک مدار ارزیابیِ صحیح وجود داشته باشد و بخش $p_۲$ نیز از پروتکل خارج نشود، توزیع خروجی بخش $p_۲$ در حالت واقعی و ایده‌آل برابر خواهند بود (با احتمال خطای $2^{-s} + \mu(.)$). برای اثبات این ادعا، فرض کنیم تنها یک مدار صحیح در بین مدارهای ارزیابی قرار داشته باشد، که بخش $p_۲$ خروجی $f(x,y)$ را به‌دست خواهد آورد و دیگر مدارهای ارزیابی یکی از دو حالت را خواهند داشت. اول این‌که رشته‌های خروجی مدارهای ارزیابی باقی‌مانده، نامعتبر باشند یعنی این‌که رشته‌های خروجی هیچ یک از مقادیر $\Delta_{i,0}$و $\Delta_{i,1}$ را تولید نمی‌کنند. که در این حالت این مدارها را نادیده خواهیم گرفت. در حالت دوم، اگر مدار نادرستی باشد که رشته‌های خروجی معتبری ولی متفاوت از هم را در خروجی ظاهر کند، طبق پروتکل و روش بازیابی ورودی، بخش $p_۲$ خواهد توانست x را بازیابی و تابع $f(x,y)$را محاسبه کند. از آنجایی که در گام ششم، بردار $(\mathrm{g^w}, \mathrm{g^r}, \mathrm{g^r\Omega})$ و (g^w, g^r, g^r) بر اساس فرض مسئله سخت DDH تولید شده‌اند پس تمیزناپذیر خواهند بود، پس می‌توان امنیت بازیابی ورودی را تضمین کرد [۱۴]. بنابراین بخش $p_۲$ یا از پروتکل خارج یا اگر یک مدار ارزیابی صحیح وجود داشته باشد، خواهد توانست مقدار صحیح x را بازیابی، و خروجی را محاسبه کند. تنها حالت باقی‌مانده این است که هیچ رشته خروجی معتبری در تمام مدارهای ارزیابی وجود نداشته باشد. در این صورت اگر تنها یک مدار کنترل نادرست باشد بخش $p_۲$ از پروتکل خارج خواهد می‌شود، اما اگر تمام مدارهای کنترل صحیح باشند و مدارهای ارزیابی هم در سیم خروجی مقدار معتبری را به‌دست نیاورند، بخش $p_۲$ از پروتکل خارج نخواهد شد و این حالت تنها تفاوت با حالت ایده‌آل است به دلیل این‌که خارج شدن و نشدن از پروتکل به مقدار ورودی وابسته است. این نشان می‌دهد که شبیه‌ساز می‌تواند از توزیع خروجی واقعی منحرف شود اگر و فقط اگر تمام مدارهای کنترل صحیح و مدارهای ارزیابی نادرست باشند و از آنجا که بخش $p_۱$ از انتخاب مدارهای ارزیابی و کنترل آگاه نخواهد شد و همان‌طور که می‌دانیم احتمال انتخاب مدار به‌عنوان کنترل یا ارزیابی $\frac{۱}{۲}$ است. در نتیجه با فرض وجود s مدار، احتمال منحرف شدن شبیه‌ساز از توزیع خروجی حالت واقعی توزیع خروجی حالت واقعی ۲^{-s} است.

فاسد شدن بخش $p_۲$: در این بخش ما خروجی بخش $p_۲$ را شبیه‌سازی خواهیم کرد. شبیه‌ساز به‌عنوان بخش $p_۱$ صادق، با ورودی صفر است. با فرض این‌که بخش $p_۲$ مقادیر مبهم سیم‌های خروجی متناظر با $f(0,y)$ را می‌داند، جدول کدگذاری $T_{j,i,b}$ را

[1] View

برای اطمینان از درستی محاسبه خروجی $f(x,y)$ تنظیم می‌کنیم. با فرض اینکه مهاجم A بخش $p_۲$ را در اختیار می‌گیرد، شبیه‌ساز S را به‌شکل زیر می‌سازیم:

۱- شبیه‌ساز S مطابق بخش صادق $p_۱$ عمل کرده و ورودی J و مقدار y را که مهاجم A برای انتقال کور برش-انتخاب می‌فرستد، را به‌دست می‌آورد. سپس S زوج $(\mathrm{input}\,, y)$ را برای بخش قابل اعتماد می‌فرستد و $z = f(x,y)$ را دریافت می‌کند.

۲- شبیه‌ساز S مطابق بخش صادق $p_۱$ با ورودی $x = ۰$ عمل می‌کند و مقادیر $\{N_{i,b}\}$ را دریافت و $\{En_{key_j}(R_{j,i,0})\}$ را برای A می‌فرستد. اگر A لغو پروتکل را برای انتقال کور فرستاده باشد، S از پروتکل خارج خواهد شد.

۳- شبیه‌ساز S مطابق بخش صادق $p_۱$، عمل می‌کند، مگر در حالی که طبق محاسبه $z' = f(0,y)$ رابطه $z(i) \neq z'(i)$ برقرار باشد که در این‌صورت مقدار $T_{j,i,b}$ را به شکل $T_{j,i,b} = En_{Z_{j,i,b}}(\Delta_{i,1-b})$ تعیین می‌کنیم.

حال باید نشان دهیم توزیع خروجی بخش $p_۱$ و A در حالت واقعی، از توزیع حالت ایده‌آل غیر قابل تمیز است. برای اثبات از نمونه اثبات ترکیبی استفاده می‌کنیم.

H_1: مطابق با نمونه ترکیبی، شبیه‌ساز S نقش $p_۱$ صادق، را ایفا می‌کند.

H_2: شبیه‌ساز S مقدار ورودی y را در انتقال کور برش-انتخاب آورده و $(0,y)$ را برای بخش قابل اعتماد فرستاده و $T_{j,i,b}$ را با توجه به مقدار z محاسبه خواهد کرد.

توزیع H_1 و H_2 با هم برابرند مگر در دو مورد زیر:

۴- در H_1 بخش $p_۲$ مقادیر $\{R_{j,i,x(i)}\}$ محاسبه، در حالی که H_2 مقدار $\{R_{j,i,0}\}$ محاسبه می‌شود. از آنجا که $R_{j,i,b} = PRF_{seed_j}(i,R) \oplus N_{i,0}$ است و بخش $p_۲$ از $PRF_{seed_j}(i,R)$ آگاهی ندارد پس مقدار برای بخش $p_۲$ تصادفی خواهد بود. از طرفی با توجه به این‌که در هریک از H ها تنها یکی از $\{\mathrm{R_{j,i,b}}\}$ محاسبه می‌شود و این مقدار به طور یکنواخت تصادفی است. پس در نتیجه این تفاوت ازنظر محاسباتی تمیزناپذیر خواهد بود.

۵- در H_1، بخش $p_۲$ مقدار $z_{j,i,z(i)}$ و $T_{j,i,z(i)} = En_{Z_{j,i,z(i)}}(\Delta_{i,z(i)})$ را به‌دست خواهد آورد. اما در H_2 اگر $z(i) \neq z'(i)$ باشد بخش $p_۲$ در خروجی رشته $Z_{j,i1-,z(i)}$ و $T_{j,i1-,z(i)} = En_{Z_{j,i1-,z(i)}}(\Delta_{i,1-z(i)})$ را شاهد خواهد بود. در هر دو ترکیب به‌دلیل ویژگی مدارهای مبهم هیچ اطلاعاتی در خصوص مقدار مبهم متناظر با بیت دیگر را به‌دست آورد. و از آنجا که $Z_{j,i,0}$ و $Z_{j,i,1}$ غیر قابل تمیز از هم هستند در نتیجه دو مقدار $T_{j,i0}$ و $T_{j,i1}$ نیز غیر قابل تمیز از هم خواهند بود.

همان‌طور که مشاهده شد H_1 توزیع خروجی حالتِ اجرا واقعی و H_2 توزیع خروجی حالتِ اجرا ایده‌آل هستند، به دلایل ذکر شده و ویژگی‌های نمونه اثبات ترکیبی، H_1 و H_2 تمیز ناپذیر از هم هستند. در نتیجه پروتکل ۳-۲-۱ در حالتی که مهاجم مخرب، بخش $p_۲$ را در کنترل بگیرد امن است.

با اثبات امنیت در دو حالت فاسد شدن $p_۱$ و $p_۲$، امنیت پروتکل در حضور مهاجم مخرب نیز اثبات می‌شود.

۴-۲- کارآمدی پروتکل پیشنهادی

در این بخش به‌صورت دقیق تعداد مولفه‌های کارآمدی پروتکل پیشنهادی را بررسی می‌کنیم. کارآمدی پروتکل‌های محاسبات دوبخشی از نظر پیچیدگی محاسبات، شمارش تعداد عملیات ریاضی، عملیات رمزنگاری و پهنای باند بیان می‌شود. در مقایسه‌ی کارآمدی پروتکل‌های پیشنهادی از مقایسه‌ی این مولفه‌ها استفاده می‌شود.

مهمترین عملیات ریاضی پروتکل پیشنهادی عملیات نمایی است. عملیات نمایی ازنظر پیچیدگی محاسبات نسبت به عملیات رمزنگاری و تعهد، ساده‌تر است [۱۵]، بنابراین در سال‌های اخیر در عوض طرح‌های تعهد[1]، از عملیات‌های نمایی برپایه فرض DDH در پروتکل محاسبات چند نهادی استفاده می‌کنند. عملیات نمایی استاندارد، معمولاً از تکرار توان رسانی محاسبه می‌شود، برای گروهی از مرتبه q و طول بیت m به طور میانگین ۱/۵ m ضرب، برای توان رسانی کامل نیازمندیم. اگر چندین محاسبه نمایی دارای یک پایه ثابت با توان‌های متفاوت باشند، می‌توان با محاسبه کوچک‌ترین توان و استفاده از آن در دیگر توان‌رسانی‌ها، تعداد سربار عملیات را به مقدار ۰/۵ m ضرب، رساند. از طرفی در تابع $RAND$ از عملیات $x^a.y^b$ استفاده می‌شود که طبق [۱۵] ارزش ۱،۲۵ برابر توان رسانی استاندارد را دارد. در نتیجه برای شمارش تعداد عملیات نمایی با دو نوع نمایی با پایه ثابت[2] و نمایی منظم[3] مواجه خواهیم بود[۱۲].

در شمارش تعداد عملیات رمزنگاری متقارن با رمزنگاری کلیدهای متناظر با هر گیت در مدارهای مبهم، عملیات تابع

[1] Commitment
[2] Fixed-Base Exponentiations
[3] Regular Exponentiations

PRF و تعهدات مواجه هستیم. برای تولید هر مدار مبهم با فرض $|c|$ به‌عنوان تعداد گیت هر مدار، نیازمند $8|c|$ عملیات رمزنگاری هستیم. و با فرض انتخاب ۵۰ درصد مدارات به عنوان مدار کنترل، تعداد رمزگشایی‌ها برابر با $8|c|.\frac{s}{2}$ و برای مداهای ارزیابی،مقدار $2|c|\frac{s}{2}$ است. از طرفی تعداد محاسبه‌ی تابع PRF برابر با $2sl$، تعهدات نیز ... است. تعداد عملیات‌های پیچیدگی محاسباتی با s تعداد مدارها، l تعدادبیت ورودی هر یک از بخش‌ها و تعداد بیت خروجی، $|c|$ نشان‌دهنده تعدادگیت مدار، در جدول(۴) قابل مشاهده است.

جدول (۴): کارآمدی پروتکل پیشنهادی

گام	Fixed-base exponent	Regular exponent	symmetric encryptions	group elements	Symmetric comm
۱	––	–	$8\lvert c\rvert s+2sl$	–	–
۲	15.5sl+2s+l	4sl+6s+25	3sl	10sl+8s+19	$3sl^2$
۳	–	3l+2	4l	32(l+1)	-
۴	–	–	4Sl+3s+l	–	$sl^2+4sl\lvert c\rvert+2sl+l^2$
۵	–	–	$s\lvert c\rvert+\frac{sl}{2}+\frac{s}{2}$	-	-
۶	2.5s	3	s	3+s	$2l^2+sl+l$
۷	–	–	S+2sl	-	-
۸	-	-	$4\lvert c\rvert s+\frac{3sl}{2}$	-	sl
جمع	15.5sl+4.5s+l	4sl+6s+3l	$13\lvert c\rvert s+13sl+5.5s+5l$	10sl+9s	$(4s+3)l^2+4sl\lvert c\rvert+4sl+l$

۵- مقایسه کارآمدی

ازنظر پیچیدگی محاسبات، احتمال خطای پروتکل، تعداد مدارهای مبهم، تعداد عملیات رمزنگاری، عملیات نمایی طبق جدول (۵) با پروتکل‌های قبلی مقایسه شده اند.

از نظر پیچیدگی محاسبات تعداد مدار مبهم رابطه مستقیم با تعداد عملیات رمزنگاری و مولفه‌های دیگر دارد. از آنجا که کاهش احتمال خطا عامل اصلی کاهش تعداد مدار است، پس پروتکل‌هایی با احتمال خطای 2^{-s} ازنظر محاسبات کارآمد تر هستند. در نتیجه برای بررسی دقیق‌تر پروتکل پیشنهادی، در جدول (۵)، با پروتکل [۱۰] در جزئیات پروتکل مقایسه صورت می‌گیرد.

جدول (۵): مقایسه‌ی کارآمدی پروتکل پیشنهادی با پروتکل‌های پیشین

پروتکل	error probability	Number of circuits to 2^{-40}	symmetric encryptions	exponent	group elements	Symmetric communication
[۷]	$2^{-\frac{s}{4}}$	۱۶۰	$O(s\lvert c\rvert)$	$O(sl)$	-	$O(s\lvert c\rvert l)$
[۱۶]	$2^{-\frac{s}{17}}$	۶۸۰	$O(s\lvert c\rvert+s^2l)$	-	-	$O(s\lvert c\rvert l+s^2l)$
[۱۲]	$2^{-.311s}$	۱۲۸	$O(s\lvert c\rvert)$	$O(sl)$	$O(sl)$	-
[۱۰]	2^{-s}	۴۰	$O(s\lvert c\rvert)$	$O(sl)$	$O(sl)$	$O(s\lvert c\rvert l)$
[۱۴]	2^{-s}	۴۰	$O(s\lvert c\rvert+sl)$	-	$O(s)$	-
پروتکل پیشنهادی	2^{-s}	۴۰	$O(s\lvert c\rvert)$	$O(sl)$	$O(sl)$	$O(s\lvert c\rvert l)$

جدول (۶): مقایسه جزئیات کارآمدی پروتکل پیشنهادی

پروتکل	Fixed-base exponent	Regular exponent	symmetric encryptions	group elements	Symmetric communication
[۱۰]	$25sl+5040s$	$3.5sl+18l+480l$	$13s\|c\|+39sl$	$26sl$	$4ls\|c\| + \mathbf{14}sl^2$
پروتکل پیشنهادی	$\mathbf{15.5sl+4.5s+l}$	$\mathbf{4sl+6s+3l}$	$\mathbf{13\|c\|s + 13sl}$	$\mathbf{10sl+9s}$	$\mathbf{(4s+3)l^2+4sl\|c\|+4sl+l}$

برای درک بهتر مقایسه کارآمدی، جزئیات جدول(۶) را برای مثال، بر روی مدار الگوریتم AES با ۶۸۰۰ گیت و اندازه $l=$ ۱۲۸ بیت محاسبه می‌کنیم. برای به‌دست آوردن پهنای باند پروتکل، از محاسبه مجموع بیت‌هایی که در تبادل عناصر گروه و بیت‌های منتقل شده به عنوان رشته‌های رمز شده، با فرض ۱۲۸ بیتی بودن رشته‌های رمز شده و ۲۲۰ بیتی بودن هر یک از عناصر گروه به‌دست می‌آید [۱۰]. جزئیات محاسبه مقادیر در جدول(۷) مشاهده می‌شود.

جدول (۷): مقایسه مولفه‌های پیچیدگی محاسبات در مدارAES

پروتکل	Fixed-base exponent	Regular exponent	symmetric encryptions	bandwidth
[۱۰]	۳۰۹۱۲۰	۳۷۱۲۰	۳۷۴۹۶۰۰	۱۷۷۷۲۵۴۴۰
پروتکل پیشنهادی	۷۹۶۶۸	۲۱۱۰۴	۳۶۰۲۵۶۰	۱۵۳۲۹۸۴۰۰

۶- نتیجه‌گیری

در سال‌های اخیر با گسترش روز افزون شبکه‌های اینترنت، اینترنت اشیا و رایانش ابری، محاسبات توزیع شده کاربرد فراوانی یافته است. امروزه مهمترین چالش محاسبات توزیع شده حفظ محرمانگی و حریم خصوصی داده‌های هر یک از نودهای شبکه است. پروتکل‌های محاسبات دوبخشی راه‌حل مناسب برای حفظ حریم خصوصی داده‌ها به‌شمار می‌رود. امنیت و کارآمدی پروتکل‌های محاسبات دوبخشی از نکات مهم این پروتکل‌ها به‌شمار می‌روند.

پروتکل دوبخشی یائو در برابر مهاجم نیمه‌صادق امن بود اما در واقعیت با مهاجمین و بخش‌هایی مواجه هستیم که توانایی بالاتری دارند. مهاجم مخرب نمونه خوبی برای نمونه‌سازی این نوع مهاجمین است. برای امن کردن پروتکل یائو دربرابر مهاجم مخرب از روش برش-انتخاب در مقالات مختلفی بهره برده‌اند که دارای نقاط ضعف یا سبب افزایش پیچیدگی پروتکل شده اند. در پروتکل پیشنهادی با ادغام انتقال کور و روش برش-انتخاب پروتکل پیشنهادی در برابر مهاجم مخرب امن گردید.

کارآمدی پروتکل‌های محاسبه دو بخشی بر پایه‌ی پیچیدگی محاسبات و پهنای باند پروتکل سنجیده می‌شود. برای مقایسه، در محاسبه‌ی الگوریتم AES پروتکل پیشنهادی ازنظر پیچیدگی محاسبات و تعداد عملیات‌های رمزنگاری با کاهش تقریبا ۴٪ و در عملیات توان رسانی با کاهش ۷۰٪ و در پهنای باند پروتکل پیشنهادی ۱۳٪ کمتر از پروتکل یهودا است.

در این مقاله با ارائه‌ی پروتکل محاسبه دوبخشی پیشنهادی کارآمد با استفاده از روش‌های بازیابی ورودی و انتقال کور برش-انتخاب، پیچیدگی این نوع محاسبات را به‌طور چشم‌گیری کاهش دادیم.

۷- مراجع

[1] T. P. Jakobsen, “Practical aspects of secure multiparty computation, ” Department of Computer Science, Aarhus University, 2015.

[2] A. C. Yao, “Protocols for secure computations, ” in 23rd annual symposium on foundations of computer science, pp. 160-164, 1982.

[3] Y. Lindell and B. Pinkas, “A proof of security of Yao’s protocol for two-party computation, ” Journal of cryptology, vol. 22, pp. 161-188, 2009.

[4] A. C.-C. Yao, “How to generate and exchange secrets, ” in 27th Annual Symposium on Foundations of Computer Science (sfcs 1986), pp. 162-167, 1986.

[5] O. Goldreich, S. Micali, and A. Wigderson, “How to play any mental game," in Proceedings of the nineteenth annual ACM symposium on Theory of computing, pp. 218-229, 1987.

[6] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” SIAM Journal on computing, vol. 18, pp. 186-208, 1989.

[7] P. Mohassel and M. Franklin, “Efficiency tradeoffs for malicious two-party computation,” in International Workshop on Public Key Cryptography, pp. 458-473, 2006 .

[8] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, “Fairplay-Secure Two-Party Computation System,” in USENIX Security Symposium, p. 9, 2004 .

[9] C. Hazay and Y. Lindell, “Efficient secure two-party protocols: Techniques and constructions,” Springer Science & Business Media, 2010.

[10] Y. Lindell, “Fast cut-and-choose-based protocols for malicious and covert adversaries,” Journal of Cryptology, vol. 29, pp. 456-490, 2016.

[11] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in Annual international cryptology conference, pp. 554-571, 2008.

[12] Y. Lindell and B. Pinkas, "Secure two-party computation via cut-and-choose oblivious transfer," Journal of cryptology, vol. 25, pp. 680-722, 2012.

[13] H. Jiang, Q. Xu, C. Liu, Z. Zheng, Y. Tang, and M. Wang, "Cut-and-choose bilateral oblivious transfer protocol based on DDH assumption," Journal of Ambient Intelligence and Humanized Computing, pp. 1-11, 2018.

[14] X. Wang, A. J. Malozemoff, and J. Katz, "Faster secure two-party computation in the single-execution setting," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 399-424, 2017 .

[15] A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography: CRC press, 1996.

[16] Y. Lindell and B. Pinkas, "An efficient protocol for secure two-party computation in the presence of malicious adversaries," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 52-78, 2007.

Archive of SID.ir