دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۲، جلد ۲۱، شماره ۱ (۴۹-۶۵)
https://monadi.isc.org.ir

مروری بر مدل‌های امنیتی با تاکید بر پروتکل‌های تبادل کلید نشت‌تاب

ناصر زربی[۱]، علی زعیم‌باشی*[۱،] و نصور باقری[۲]

[۱] دانشکده علوم پایه، دانشگاه تربیت دبیر شهید رجائی، تهران، ایران
[۲] دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجائی، تهران، ایران

اطلاعات مقاله

تاریخچه مقاله:
تاریخ دریافت: ۲۵ آذر ۱۴۰۲
تاریخ پذیرش: ۱ اسفند ۱۴۰۲
انتشار آنلاین: ۲۳ اسفند ۱۴۰۲

کلمات کلیدی:
مدل امنیتی
نشت‌تاب
پروتکل‌های تبادل کلید
مهاجم

نوع مقاله: مروری

چکیده

در رمزنگاری نشت‌تاب، هدف طراحی پروتکل‌های تبادل کلیدی است که بتواند در برابر حملات نشت مقاومت کند. این پروتکل‌ها با استفاده از یک مدل امنیتی نشت‌تاب مورد بررسی قرار می‌گیرند تا مشخص شود که آیا ویژگی‌های امنیتی ادعا شده را دارا هستند یا خیر. بررسی ویژگی‌های امنیتی بر این تمرکز دارد که چگونه یک مدل امنیتی نشت‌تاب طی سال‌ها تکامل‌یافته است تا نیازهای امنیتی فزاینده را برآورده کند و طیف وسیع‌تری از حملات را پوشش دهد. با مطالعه و بررسی ویژگی‌های امنیتی ارائه‌شده توسط این مدل‌ها، آسیب‌پذیری‌های احتمالی در طراحی پروتکل‌ها می‌تواند به طور مؤثری برطرف شود. این مقاله به بررسی انواع مدل‌های امنیتی نشت‌تاب مبتنی بر دو مدل CK و eCK می‌پردازد و نمونه‌هایی از پروتکل‌های امن تبادل کلیدی را که در این مدل‌ها تعریف‌شده‌اند، ارائه می‌کند. علاوه بر این، ارتباط بین قابلیت‌های مهاجمان در این مدل‌ها و انواع طرح‌های حمله در دنیای واقعی را مورد بررسی قرار می‌دهد و با ارائه بینشی در مورد مدل‌های مختلف امنیتی نشت‌تاب، حملات نشتی و توسعه پروتکل‌های امن، به پیشرفت دانش در این زمینه کمک می‌کند.

۱ مقدمه

پروتکل‌های توافق کلید احراز اصالت شده(AKA [۱])، ابزارهای رمزنگاری مشهوری هستند که در آن‌ها تبادل کلید بین دو طرف انجام می‌شود تا با ایجاد یک کلید مخفی مشترک، ارتباط امنی را برقرار کنند. این پروتکل‌ها در حضور مهاجم [۲] هم از ارتباط بین دو طرف محافظت می‌کنند. برای تجزیه و تحلیل امنیت پروتکل‌های تبادل کلید [۳] ، سه مدل متداول وجود دارد: مدل Dolev-Yao [۱]، مدل Canetti-Krawczyk (CK) [۲] و مدل Canetti-Krawczyk توسعه‌یافته (eCK [۴]) [۳]. این مدل‌ها توانایی‌های مختلف مهاجمان را با هدف تضمین امنیت ارتباطات در نظر می‌گیرند. پژوهشگران در زمینه رمزنگاری به پیشنهاد پروتکل‌های تبادل کلید کارآمدتر برای مقابله با مهاجمانی با توانایی بیشتر پرداخته‌اند. به همین دلیل، مدل‌های امنیتی قوی‌تری برای اثبات امنیت این پروتکل‌ها درحال ایجاد است. رمزنگاری نشت‌تاب روش جدیدی است که برای محافظت در برابر حملات کانال جانبی طراحی شده است و هدف آن ایجاد طرح‌هایی است که در صورت بروز حملات نشت مانند نشت کلیدهای مخفی بلندمدت و موقت، نشت اطلاعات محاسباتی، حافظه و نشت اطلاعات از راه کانال‌های جانبی، همچنان امن باقی می‌مانند. با افزایش قدرت مهاجم در دنیای واقعی، مدل CK کارایی خود را در مقابله با حملات احتمالی از دست می‌دهد. ازاین‌رو با توسعه این مدل، مدل eCK

*نویسنده مسئول
آدرس‌های رایانامه: nasser_zarbi@yahoo.com (ناصر زربی)، azaeembashi@sru.ac.ir (علی زعیم‌باشی)، Nbagheri@sru.ac.ir (نصور باقری)

[1] Authenticated Key Agreement [2] Adversary [3] Key Exchange Protocols

[4] Extended Canetti-Krawczyk

حاصل می‌شود که طیف وسیع‌تری از حملات را پوشش می‌دهد. با این حال، حملات کانال جانبی چالش‌های دیگری را ایجاد می‌کند که نیازمند طرح‌های پیشرفته در مدل امنیتی eCK است. این حملات می‌توانند از آسیب‌پذیری‌های موجود در پروتکل‌های رمزنگاری، از جمله پروتکل‌های تبادل کلید در مدل‌های امنیتی مستعد نشتی، سوء استفاده کنند. در این مقاله، علاوه بر مدل‌های امنیتی Dolev- Yao، CK و eCK، شش مدل امنیتی نشت‌تاب موجود نیز بررسی می‌شوند: نشتی تاب در مدل CK (LR-CK) [۴]، نشتی تاب در مدل eCK (LR-eCK) [۵]، نشت پس از وقوع محدود eCK (BAFL-eCK) [۶]، نشت پس از وقوع مداوم eCK (CAFL-eCK) [۷]، نشت مداوم eCK (GCL-eCK) [۸] و نشت‌تاب وابسته به چالش eCK (CLR-eCK) [۹]. هر مدل امنیتی پیشنهادی، مبانی رمزنگاری و ویژگی‌های امنیتی خاص خود را دارد که در طراحی پروتکل‌های تبادل کلید به کار می‌رود. این بررسی می‌تواند به پیشنهاد مفاهیم امنیتی قوی‌تر برای پروتکل‌های تبادل کلید کمک کند. این مقاله مروری بر شش پروتکل [۴، ۶–۱۰] دارد که به ویژگی‌های امنیتی مهم در مدل‌های امنیتی پایه مربوطه می‌پردازد. در این پروتکل‌ها ویژگی‌های امنیتی در رابطه با تنظیمات نشتی تعریف‌شده و همچنین قابلیت‌های مهاجم بررسی و مشخص می‌شود که چگونه مدل‌های امنیتی در طول سال‌ها برای برآوردن نیازهای امنیتی پروتکل‌های تبادل کلید، تکامل یافته‌اند. در ادامه رابطه بین مدل‌های امنیتی و طرح‌های دنیای واقعی را بررسی می‌کند. مقاله‌ی حاضر به صورت زیر سازماندهی شده است: بخش ۲ مروری کوتاه بر مدل‌های امنیتی Dolev-Yao، CK و eCK ارائه می‌کند. بخش ۳ پروتکل‌های مختلف تبادل کلید ارائه شده بر اساس مدل‌های امنیتی پایه مربوطه و مبانی رمزنگاری را مورد بحث قرار می‌دهد. در بخش ۴، جدولی برای مقایسه تنظیمات نشت پروتکل‌های تبادل کلید موجود و بررسی تفاوت‌ها از نظر قابلیت‌های مهاجم ارائه شده است.

۲ مدل‌های امنیتی برای پروتکل‌های تبادل کلید

۱.۲ مدل Dolev-Yao

مدل Dolev-Yao یک چارچوب ساده و مفید برای تحلیل پروتکل‌های امنیتی است اما، مهاجم را محدود فرض می‌کند.

حال به بررسی قدرت مهاجم در این مدل می‌پردازیم.

فرض کنیم پیام‌ها اعضای یک جبر A باشند. دو نوع پیام داریم.

- متن‌ها (T) که خود دو نوع هستند: شناساگرها [۱] (عمومی و قابل پیش‌بینی که با M نمایش داده می‌شوند) و تک‌شمارهای تصادفی [۲] (خصوصی و غیرقابل پیش‌بینی که با R نمایش داده می‌شوند) و
- کلیدها (K) که خود به عمومی (K_{Pub}) و خصوصی (K_{Priv}) افراز می‌شوند.

پیام‌های ترکیبی نیز با دو عمل جبری زیر ایجاد می‌شوند:

- رمزگذاری: $K_{Pub} \times A \to A$
- جفت کردن: $A \times A \to A$

علاوه بر این، جبر را آزاد در نظر می‌گیریم؛ یعنی هر مقدار یک نمایش منحصربه‌فرد دارد.

در این مدل دو نوع بخش فعال وجود دارد: شرکت کنندگان صادق و مهاجم. شرکت‌کنندگان صادق گام‌های پروتکل را بدون انحراف دنبال می‌کنند. آن‌ها می‌توانند چندین اجرای پروتکل را هم‌زمان و با طرف‌های مختلف درگیر کنند. این مدل فقط شامل پیام‌هایی است که آن‌ها ارسال و دریافت می‌کنند؛ حالت‌های داخلی به صراحت مدل‌سازی نمی‌شوند.

فرض می‌شود که کل شبکه تحت کنترل مهاجمی است که می‌تواند پیام‌ها را ضبط، حذف، بازپخش، تغییر مسیر و یا تغییر ترتیب دهد. این کار با فرض این‌که مهاجم یک شبکه است، عملی می‌شود.

شرکت‌کنندگان صادق پیام‌هایشان را فقط به مهاجم ارسال و فقط از مهاجم دریافت می‌کنند. چون شرکت‌کنندگان صادق مورد علاقه ما نیستند، آن‌ها را در یک فرآیند «محیطی» خلاصه می‌کنیم. بنابراین می‌توانیم هر اجرای پروتکل را دنباله‌ای متناوب از پیام‌های مهاجم $(q_i \in A)$ و پاسخ‌های محیط $(r_i \subseteq A)$ در نظر بگیریم:

$$r_o q_1 r_1 q_2 r_2 \dots q_{n-1} r_{n-1} q_n r_n$$

در ادامه به بررسی این‌که مهاجم چگونه دستورات را تولید می‌کند، می‌پردازیم. مجموعه دانش اولیه مهاجم شامل موارد زیر است:

- کلیدهای عمومی (K_{Pub})،
- کلیدهای خصوصی شرکت‌کنندگان آسیب‌دیده $(K_{Adv} \subseteq K_{Priv})$،
- اسامی اصلی (M) و
- تک‌شمارهایی که خود (R_{Adv}) را تولید می‌کنند، مجزا از همه‌ی تک‌شمارهای تولیدشده توسط شرکت‌کنندگان صادق فرض می‌شوند.

برای این که پیام داده‌شده M، از مجموعه پیام‌های S قابل استخراج باشد، باید با چند بار به کارگیری اعمال زیر روی S بتوان آن را تولید کرد:

- رمزگشایی با کلیدهای خصوصی یاد گرفته‌شده یا شناخته‌شده،
- رمزگذاری با کلیدهای عمومی،
- جفت کردن دو عضو شناخته‌شده و
- جداسازی یک عضو «اتصال» به عناصر تشکیل‌دهنده آن.

تعریف ۱ (بستار [۳]). بستار S که با $C[S]$ نمایش داده می‌شود برابر است با کوچک‌ترین زیرمجموعه A به طوری که:

(۱) $S \subseteq C[S]$،
(۲) $M \cup K_{Pub} \cup K_{Subv} \cup R_{Adv} \subseteq C[S]$،
(۳) اگر $\{|M|\}_K \in C[S]$ و $K^{-1} \in C[S]$، آنگاه $M \in C[S]$،
(۴) اگر $M \in C[S]$ و $K \in C[S]$، آنگاه $\{|M|\}_K \in C[S]$،
(۵) اگر $MN \in C[S]$، آنگاه $M \in C[S]$ و $N \in C[S]$ و
(۶) اگر $M \in C[S]$ و $N \in C[S]$، آنگاه $MN \in C[S]$.

[1] Identifiers [2] Random nonces

[3] Closure

این فرض اصلی مدل Dolev-Yao است که عمل بستار، محدودیت توانایی مهاجم را برای ایجاد پیام‌های جدید نشان می‌دهد.

تعریف ۲ (مهاجم Dolev-Yao). اگر مهاجم Dolev-Yao، مجموعه پیام‌های S را تشخیص دهد، آنگاه فقط می‌تواند پیام‌هایی را که در $C[S]$ هستند، تولید کند.

بنابراین در مدل Dolev-Yao باید

$$q_i \in C[S_o \cup r_o \cup \ldots \cup r_{i-1}]$$

برقرار باشد طوری که S_o مجموعه پیام‌هایی است که مهاجم در شروع اجرا می‌داند.

۲.۲ مدل Canetti-Krawczyk (CK)

در مدل CK شناسه نشست برای شرکت کننده‌ها به منظور فعال‌سازی نشست الزامی است [۲]. شکل کلی یک پروتکل تبادل کلید به صورت $(P_A, P_B, s, role)$ هست که P_A و P_B طرف‌های پروتکل، s شناسه و $role$ آغازگر یا پاسخ‌دهنده است. در ادامه به سه پرسمان قابل‌توجه می‌پردازیم:

(۱) آشکارکننده کلید که هدفش فاش کردن کلید نشست‌های انجام‌شده است.

(۲) آشکارکننده وضعیت نشست که هدفش آشکارسازی اطلاعات داخل نشست است. به طور عموم مدل CK نمی‌تواند به اطلاعاتی از وضعیت داخل نشست دست یابد اما با به کارگیری پروتکل تبادل کلید به این مهم دست پیدا می‌کند به شرطی که اطلاعات داخل نشست شامل اطلاعات مخفی بلندمدت شرکت‌کنندگان پروتکل نباشد.

(۳) پرسمان آسیب که هدفش افشای اطلاعات وضعیت داخلی نشست است.

این مدل دنباله‌ای از پرسمان‌ها را به‌منظور حفظ تازگی نشست، محدود می‌کند. برای این‌که در مدل CK یک نشست در معرض افشا باشد، مهاجم باید یکی از پرسمان‌های زیر را اجرا کند:

- پرسمان آشکارکننده وضعیت در یک نشست،
- پرسمان آشکارکننده کلید نشست،
- تخریب شرکت‌کننده پروتکل قبل از منقضی‌شدن نشست.

برای تعیین امنیت یک پروتکل تبادل کلید در مدل CK باید موارد زیر را در نظر گرفت:

(۱) دو بخش آسیب ندیده با موفقیت یک نشست تطابق CK را تکمیل و کلید نشست یکسانی به دست آورند.

(۲) هیچ مهاجم زمانی چندجمله‌ای احتمالاتی[1] وجود نداشته باشد که بتواند با احتمال قابل توجه در بازی تشخیص‌ناپذیری موفق شود.

چندین ویژگی امنیتی مورد بحث در مدل CK وجود دارد که با موقعیت‌های مهاجم در دنیای واقعی مرتبط است. یکی از این ویژگی‌ها، به نام «امنیت کلید شناخته شده[2]» تضمین می‌کند که یک کلید نشست هرگز با کلیدهای نشستِ نشست‌های دیگر مرتبط نیست. این بدان معنی است که حتی اگر یک مهاجم اطلاعاتی در مورد یک کلید نشست خاص داشته باشد، نمی‌تواند کلید نشست مورد استفاده در نشست دیگر را به دست آورد. ویژگی دیگری که با عنوان پرسمان «آسیب» شناخته می‌شود، با «اشتراک کلید ناشناخته[3]» سر و کار دارد. این زمانی اتفاق می‌افتد که طرف A معتقد است که یک کلید نشست را با طرف B به اشتراک می‌گذارد، اما در واقع، آن کلید را با طرف C به اشتراک می‌گذارد. در این مورد، اگر یک کلید نشست بین طرف‌های A و B به اشتراک گذاشته شود، احتمال آسیب از طرف C وجود دارد. مدل CK همچنین به «رازمانی پیش‌سوی کامل[4]» می‌پردازد، بدین معنی که حتی اگر مهاجم به طور فعال با پروتکل نشست تعامل داشته باشد، نمی‌تواند کلید مخفی بلندمدت شرکت‌کنندگان پروتکل را به دست آورد. با این‌حال، مدل CK به طور خاص به «جعل هویت کلید» نمی‌پردازد، بلکه به عدم دسترسی مهاجم به کلید مخفی بلندمدت شرکت‌کنندگان پروتکل، قبل از منقضی شدن نشست اشاره دارد.

۳.۲ مدل Canetti-Krawczyk (eCK) توسعه‌یافته

در مدل eCK [۳]، شناسه نشست به صورت

$$sid = (role, ID, ID, comm_1, ..., comm_n)$$

تعریف می‌شود که ID و ID^* بیانگر شناسه شرکت‌کننده‌هاست. $role$ می‌تواند آغازگر یا پاسخ‌دهنده اجرای پروتکل باشد و $comm_n$، به عنوان n-امین ارتباط بین شرکت‌کنندگان پروتکل است. sid برای اشاره به نشست تبادل کلید تکمیل‌شده استفاده می‌شود که توسط طرف A در ارتباط با طرف B انجام شده است. از طرف دیگر، sid^* نشان‌دهنده نشست تطابقی با sid است که توسط طرف B اجرا شده است. سه نوع پرسمان مهم: آشکارکننده کلید نشست، آشکارکننده کلید بلندمدت و آشکارکننده کلید موقت به صورت زیر تعریف شده‌اند:

- آشکارکننده کلید نشست که هدفش فاش کردن کلید نشست پایان‌یافته است،
- آشکارکننده کلید بلندمدت که هدفش فاش کردن کلید مخفی بلندمدت شرکت‌کنندگان پروتکل است،
- آشکارکننده کلید موقت که هدفش فاش کردن کلیدمخفی موقت نشست است.

در مدل eCK یک نشست در معرض افشا قرار دارد اگر و تنها اگر مهاجم یکی از موارد زیر را اجرا کند:

[1] Probabilistic Polynomial Time

[2] Known Key Security [3] Unknown Key Share [4] Perfect Forward Secrecy

- پرسمان آشکارکننده کلید نشست روی شناسه یک نشست یا شناسه نشست تطابقی آن،
- شناسه نشست تطابقی وجود نداشته باشد و مهاجم پرسمان آشکارکننده کلید بلندمدت را روی طرف B یا هر دو پرسمان آشکارکننده کلید نشست موقت و بلندمدت را روی طرف A اجرا کند،
- شناسه نشست تطابقی موجود باشد و مهاجم هر دو پرسمان آشکارکننده موقت و بلندمدت را روی طرف A یا B اجرا کند،
- بخش A یا B دچار آسیب می‌شود در صورتی که هر دو کلید موقت و بلندمدت فاش شوند.

برای این‌که یک پروتکل تبادل کلید در مدل eCK امن باشد، باید شرایط زیر برقرار باشد:

(۱) دو طرف آسیب‌ندیده، یک نشست تطابقی eCK جدید و کلید نشست یکسانی داشته باشند.

(۲) هیچ مهاجم زمانی چندجمله‌ای احتمالاتی وجود نداشته باشد که بتواند با احتمال قابل توجه در بازی تشخیص‌ناپذیری موفق شود.

مدل eCK بر چندین ویژگی امنیتی که مربوط به طراحی‌های مهاجم در دنیای واقعی هستند، تمرکز دارد. یکی از این ویژگی‌ها عبارت است از «آشکارکننده کلید موقت» که به مهاجم اجازه می‌دهد تا با فاش کردن کلید مخفی موقت مورد استفاده در یک نشست مشخص، وضعیت داخلی شرکت‌کننده پروتکل را افشا کند، مشروط بر این‌که کلید مخفی بلندمدت فاش‌نشده باقی بماند. این پرسمان مشابه حملات بدافزارهایی است که پودمان‌های سخت‌افزاری [۱] را هدف قرار می‌دهند تا کلید مخفی بلندمدت را جدا از کلید موقت ذخیره کنند و هدف آن سرقت کلیدهای موقت است [۱۱]. از سوی دیگر، پرسمان «آشکارکننده کلید بلندمدت» در مدل eCK به مسائل «اشتراک کلید ناشناخته» و «جعل هویت کلید [۲]» می‌پردازد. این پرسمان به مهاجم امکان می‌دهد تا کلید مخفی بلندمدت شرکت‌کنندگان پروتکل را قبل از منقضی شدن نشست کشف کند. با این حال، هنگامی که نشست فعال شود، مهاجم تنها درصورتی می‌تواند کلید مخفی بلندمدت هر دو شرکت‌کننده را داشته باشد که آن‌ها به طور فعال پروتکل نشست را مختل نکنند، بنابراین در مدل امنیتی eCK تنها به «رازمانی پیش‌سوی ضعیف [۳]» می‌پردازد. پرسمان «آشکارکننده کلید نشست [۴]» در مدل eCK به «امنیت کلید شناخته شده [۵]» می‌پردازد و به حملاتی اشاره دارد که از کلیدهای نشست‌های قبلی استفاده می‌کنند. امنیت کلید شناخته شده الزام می‌کند که هیچ کلید نشستی با کلیدهای نشست‌های قبلی یا بعدی ارتباطی نداشته باشد. درنتیجه، اگر کلیدهای نشست قبلی در معرض خطر قرار گیرند، مهاجم از سایر کلیدهای نشست‌ها بی‌اطلاع می‌ماند. حال از لحاظ ویژگی‌های امنیتی، به بررسی مدل CK و eCK می‌پردازیم:

ویژگی جعل هویت کلید در مدل CK موضوعیت ندارد چرا که مهاجم مجاز نیست کلید مخفی بلندمدت شرکت‌کنندگان در پروتکل را قبل از اتمام نشست به دست آورد. در مدل eCK، تنها «رازمانی پیش‌سوی کامل–ضعیف» ارائه شده است، زیرا مهاجم ممکن است کلید مخفی بلندمدت هر دو شرکت کننده پروتکل را به دست آورد در صورتی که فعالانه در نشست مداخله نداشته باشد. یک تفاوت اساسی بین مدل CK و مدل eCK این است که مدل CK بر خلاف مدل eCK به مفهوم نشت کلیدهای مخفی موقت نمی‌پردازد.

۳ مدل‌های امنیتی برای پروتکل‌های تبادل کلید در برخورد با نشتی

این بخش، یک بررسی جامع از مدل‌های امنیتی مربوط به پروتکل‌های تبادل کلید ارائه و نشت اطلاعات مرتبط با هر پروتکل را بررسی می‌کند. علاوه بر این، به مقایسه بین توانایی‌های مهاجم در ارتباط با دنیای واقعی می‌پردازد.

۱.۳ مدل LR-CK

مدل LR-CK توسیعی از مدل امنیتی CK است که ویژگی‌های امنیتی مدل CK را به ارث می‌برد. این مدل که توسط دودیس [۶] و همکاران معرفی شده است، یک پروتکل تبادل کلید احراز اصالت شده را ارائه می‌کند که در چارچوب مدل نشت نسبی ساخته شده است [۴]. مدل LR-CK توسط آکاویا [۷] و همکاران رسمیت یافته است و به طور ضمنی وجود یک حمله نشتی را فرض می‌کند که بخشی از کلید مخفی را بدون توجه به اندازه آن فاش می‌کند [۱۲]. مدل امنیتی LR-CK در واقع برای محافظت در برابر حملات کانال جانبی نهان که به عنوان «حملات حافظه [۸]» نیز شناخته می‌شود، طراحی شده است. این حملات از اطلاعات به دست آمده با مشاهده اجرای الگوریتم‌های رمزنگاری در یک سامانه رایانه‌ای سوء استفاده می‌کنند و اگر مقدار اندکی از اطلاعات کلید مخفی از طریق حمله حافظه فاش شود، می‌تواند مزیتی برای مهاجم در شکستن سامانه رمزنگاری باشد. LR-CK مفهوم اندازه محدود نشت کلید مخفی را که توسط پارامتر نشتی تعریف شده است، به عنوان شرط لازم و کافی برای ایمن در نظر گرفتن پروتکل معرفی می‌کند. دودیس و همکاران یک پروتکل تبادل کلید احراز اصالت شده دیفی– هلمن (DH [۹]) را بر اساس یک طرح امضای نشت‌تاب طراحی کردند [۴]. آن‌ها در ابتدا نتایج آلون [۱۰] و همکاران را دنبال کردند و پروتکلی به نام «eSig-DH» پیشنهاد کردند که از یک طرح امضای نشت‌تاب برای احراز اصالت استفاده می‌کند و با امضای پیام دریافت‌شده از شریک پروتکل خود، آن را احراز می‌کند [۱۳]. پروتکل دیگری که آن‌ها پیشنهاد می‌کنند «Enc-DH» است که بر روی یک طرح رمزگذاری کلید عمومی ایمن با حمله متن رمزی منتخب نشت‌تاب متکی است (جدول ۱). در این ساختار، هر دو بخش پروتکل یکدیگر را با رمزگشایی دقیق کلیدعمومی موقت DH که با کلید عمومی بلندمدت رمزگذاری شده است، احراز هویت می‌کنند. این رویکرد، احراز

[1] Hardware module [2] Key Compromise Impersination [3] Weak Forward Secrecy [4] Session Key Reveal Ouery [5] Known Key Security

[6] Dodis [7] Akavia [8] Memory attack [9] Authenticated Diffie- Hellman Key Exchange Protocol [10] Alwen

جدول ۱. پروتکل Enc-DH

$\mathbb{G}$ گروه تصمیم دیفی- هلمن از مرتبه q با مولد g و (pk_i, pk_j) به عنوان کلیدهای عمومی رمزگذاری هستند		
Initiator $\mathcal{P}_i(\mathsf{sk}_i)$		Responder $\mathcal{P}_j(\mathsf{sk}_j)$
$a \xleftarrow{\mathbb{S}} \mathbb{Z}_p, \alpha = g^a$		
$C_i = \mathsf{Enc}_{\mathsf{pk}_j}^{\mathcal{P}_i}(\alpha)$		
register session $(\mathcal{P}_j, \alpha)$	$\mathcal{P}_i, C_i$	
	$\longrightarrow$	
		$b \xleftarrow{\mathbb{S}} \mathbb{Z}_q, \beta = g^b$
		$\hat{\alpha} = \mathsf{Dec}_{\mathsf{pk}_j}^{\mathcal{P}_i}(C_i)$
	$\mathcal{P}_i, \mathcal{P}_j, C_j$	$C_j = \mathsf{Enc}_{\mathsf{pk}_i}^{\mathcal{P}_i, \mathcal{P}_j}(\hat{\alpha}, \beta)$
		register session $(\mathcal{P}_i, \hat{\alpha}, \beta)$
	$\longleftarrow$	
$(\hat{\alpha}, \hat{\beta}) = \mathsf{Dec}_{\mathsf{pk}_i}^{\mathcal{P}_i, \mathcal{P}_j}(C_j)$		
output peer $= \mathcal{P}_j, \mathsf{sid} = (\alpha, \hat{\beta})$,		
output session key $\gamma_i = \hat{\beta}^a$		
delet a	$\mathcal{P}_i, \hat{\beta}$	
	$\longrightarrow$	
mark session complete		output peer $= \mathcal{P}_i, \mathsf{sid} = (\hat{\alpha}, \beta)$
		output session key $\gamma_i = \hat{\alpha}^b$
		delet b
		mark session complete

اصالت را از طریق رمزگذاری کلید عمومی فراهم می‌کند و تضمین می‌کند که تنها پروتکل اصلی با کلید مخفی صحیح می‌تواند متن رمزگذاری شده با کلید عمومی مربوطه را رمزگشایی کند. توجه به این نکته مهم است که مدل LR-CK اجازه پرسمان نشتی را در طول اجرای نشست چالش نمی‌دهد.

۲.۳ مدل LR-eCK

موریاما [۱] و اوکاموتو [۲] یک مدل نشت‌تاب برای پروتکل تبادل کلید احراز اصالت شده به نام LR-eCK معرفی کردند که توسیعی از مدل امنیتی eCK است [۱۰]. آن‌ها در مقاله خود بیان کردند که روش NAXOS شرح داده‌شده در [۳] برای دست‌یابی به امنیت eCK ضروری نیست. هدف روش NAXOS پنهان کردن توان کلید عمومی موقت X با محاسبه $X = g^{\tilde{x}}$ است، که در آن $\tilde{x}$ نشان دهنده چکیده‌ساز کلید خصوصی موقت x و کلید خصوصی ایستای a، یعنی $\tilde{x} = H(x, a)$ است. توجه به این نکته مهم است که مقدار چکیده‌ساز شده $\tilde{x}$ ذخیره نمی‌شود، اما هر موقع که نیاز باشد قابل محاسبه است. با این حال، حمله‌ای وجود دارد که به عنوان حمله کانال جانبی تحلیل توان شناخته می‌شود و به مهاجم اجازه نفوذ می‌دهد. در نتیجه، این روش در دنیای واقعی که در آن حملات کانال جانبی تحلیل توان امکان‌پذیر است، ایمن نیست. برای غلبه بر این ضعف، موریاما و اوکاموتو مدل LR-eCK را بدون روش NAXOS پیشنهاد کردند [۵]. تفاوت اصلی بین این دو در اجزای مورد استفاده برای کلید عمومی و خصوصی ایستا است. مدل LR-eCK مبتنی بر مدل امنیتی eCK است که نشت کلید خصوصی موقت یا بلندمدت را به صورت جداگانه و نه همزمان بررسی می‌کند. علاوه بر این، مدل LR-eCK تحت یک تنظیم نشتی چالش-مستقل [۳] طراحی شده است، به این معنی که پرسمان‌های نشتی در طول نشست چالش در نظر گرفته نمی‌شوند. این مدل متکی بر خانواده توابع شبه تصادفی مستقل جفتی [۴]، خانواده توابع چکیده‌ساز برخوردتاب [۵] و فرض تصمیم دیفی-هلمن (DDH [۶]) بدون پیشگوی تصادفی [۷] است. مجموعه پرسمان‌های موجود برای مهاجم همانند مدل امنیتی eCK است به اضافه یک پرسمان خاص که بخش استقرار [۸] نام دارد و به مهاجم امکان می‌دهد یک کلید عمومی ایستا را از طرف یک بخش پروتکل ثبت کند طوری که آن بخش تحت کنترل مهاجم باشد.

[1] Moriyama [2] Okamoto

[3] Challenge-independent leakage setting [4] Pair-wise independent pseudo-random function familly [5] Collision resistant hash function familly [6] Decision Diffie-Hellman assumption [7] Random oracle [8] Establish Party

۱.۲.۳ پروتکل Mo ایمن در مدل LR-eCK

در طراحی این پروتکل از روش NAXOS استفاده نشده است تا از آسیب حملات کانال جانبی در امان باشد. گیریم $k \in \mathbb{N}$ یک پارامتر امنیتی و $\{\mathbb{G}\}_k \xleftarrow{U} \mathbb{G}$ گروهی از مرتبه اول q باشد طوری که $|q| = k$. فرض کنیم $(g_1, g_2) \xleftarrow{U} \mathcal{G}^2$، H_k خانواده توابع چکیده‌ساز برخوردتاب و $h_1, h_2, h_3 \xleftarrow{R} H_k$ شاخص تابع چکیده‌ساز برخوردتاب $H_i := H_{h_i}^{k,\mathcal{D}_H,\mathcal{R}_H}$ به ازای $i = \{1,2,3\}$ باشد طوری که $\mathcal{DH} := (\pi_k)^2 \times \mathcal{G}^{18}$، $\mathcal{R}_H := \mathbb{Z}_q$ و π_k بیانگر فضای کلید عمومی ایستای تأیید شده باشد. F خانواده‌ی توابع π–شبه تصادفی [1] و $F := F^{k,\Sigma_F,\mathcal{D}_F,\mathcal{R}_F}$ طوری که

$$\Sigma_F := \mathbb{G}, \quad \mathcal{D}_F := (\pi_k)^2 \times \mathbb{G}^{12}, \quad \mathcal{R}_F := \{0,1\}^k.$$

$\mathbb{G}$، g_1، g_2، H_1، H_2، H_3 و F را به‌عنوان پارامترهای پروتکل مفروض در نظر می‌گیریم. آلیس [2] کلید مخفی ایستای

$$(a_1, \ldots, a_8, a) \xleftarrow{U} (\mathbb{Z}_q)^9$$

را انتخاب و کلید عمومی ایستای

$$(A_1, A_2, A_3, A_4, A_5, A_6) := (g_1^{a_1} g_2^{a_2}, g_1^{a_3} g_2^{a_4}, g_1^{a_5} g_2^{a_6}, g_1^{a_7} g_2^{a_8}, g_1^{a}, g_2^{a}).$$

را محاسبه می‌کند. باب [3] نیز به طور مشابه، کلید مخفی ایستای $(b_1, \ldots, b_8, b) \xleftarrow{U} (\mathbb{Z}_q)^9$ را انتخاب و کلید عمومی ایستای

$$(B_1, B_2, B_3, B_4, B_5, B_6) := (g_1^{b_1} g_2^{b_2}, g_1^{b_3} g_2^{b_4}, g_1^{b_5} g_2^{b_6}, g_1^{b_7} g_2^{b_8}, g_1^{b}, g_2^{b})$$

را محاسبه می‌کند. در مرحله اجرا، آلیس گام‌های زیر را برای ایجاد کلید نشست با باب انجام می‌دهد:

(۱) کلید مخفی موقت $(x, x_3) \xleftarrow{U} (\mathbb{Z}_q)^2$ را انتخاب و کلید عمومی موقت را محاسبه می‌کند.

(۲) کلید عمومی موقت $(X_1, X_2, X_3) := (g_1^x, g_2^x, g_1^{x_3})$ را محاسبه می‌کند.

(۳) $(\hat{B}, \hat{A}, X_1, X_2, X_3)$ را به باب ارسال می‌کند.

با دریافت $(\hat{B}, \hat{A}, X_1, X_2, X_3)$، باب بررسی می‌کند که

$$(X_1, X_2, X_3) \in \mathbb{G}^3.$$

اگر $(X_1, X_2, X_3) \in \mathbb{G}^3$ برقرار باشد، باب مراحل زیر را اجرا می‌کند:

(۱) کلید مخفی موقت $(y, y_3) \xleftarrow{U} (\mathbb{Z}_q)^2$ را انتخاب می‌کند.

(۲) کلید عمومی موقت $(Y_1, Y_2, Y_3) := (g_1^y, g_2^y, g_1^{y_3})$ را محاسبه می‌کند.

(۳) $(\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$ را به آلیس ارسال می‌کند.

با دریافت $(\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$، آلیس بررسی می‌کند که $(\hat{B}, \hat{A}, X_1, X_2, X_3)$ را به باب فرستاده است یا خیر و نیز $(Y_1, Y_2, Y_3) \in \mathbb{G}^3$ را بررسی می‌کند. درصورت تایید، آلیس

$$K_A := Y_1^{a_1 + c a_2} Y_2^{a_3 + c a_4} Y_3^{x_3} B_1^{x} B_2^{dx} B_3^{a} B_4^{ea} B_5^{a_5 + e a_7} B_6^{a_6 + e a_8}$$

را محاسبه می‌کند. باب نیز به‌طریق مشابه

$$K_B := X_1^{b_1 + d b_2} X_2^{b_3 + d b_4} X_3^{y_3} A_1^{y} A_2^{cy} A_3^{b} A_4^{eb} A_5^{b_5 + e b_7} A_6^{b_6 + e b_8}$$

را محاسبه می‌کند طوری که $c := H_1(s)$، $d := H_2(s)$، $e := H_3(s)$ و $s := (\hat{A}, \hat{B}, X_1, X_2, X_3, Y_1, Y_2, Y_3)$ به عنوان بخش باقیمانده‌ی شناسه آغازگر هست. سرانجام، آلیس کلید نشست را به صورت $SK_A := F_{K_A}(s)$ و باب به صورت $SK_B := F_{K_B}(s)$ محاسبه می‌کند (جدول ۲).

۳.۳ مدل AFL-eCK(·)

مدل نشت پس از وقوع eCK (AFL-eCK(·) [4]) که توسط آلاواتوگودا [5] معرفی شده است، می‌تواند به دو گونه افراز شود: نشت محدود پس از وقوع eCK (BAFL-eCK [6]) و نشت مداوم پس از وقوع eCK (CAFL-eCK [7]) [۶]. در طراحی این دو مدل، تفاوت‌هایی در وضعیت تازگی نشست پروتکل وجود دارد که به مجوز تابع نشت بستگی دارد. با این حال، آن‌ها از نظر مفاهیمی چون پرسمان‌های مهاجم شباهت‌هایی نیز دارند. مدل BAFL-eCK مقدار کلی اطلاعات لو رفته کلیدهای مخفی موقت یا بلندمدت را در کل فرایند اجرای پروتکل محدود می‌کند. از سوی دیگر، CAFL-eCK مقدار نامحدودی از اطلاعات نشتی را برای این کلیدها امکان‌پذیر می‌کند، با این تفاوت که میزان افشا در هر اجرای پروتکل ثابت باقی می‌ماند.

۱.۳.۳ مدل BAFL-eCK

مدل BAFL-eCK که توسط آلاواتوگودا توسعه یافته‌است، بر روی نشت کلید مخفی بلندمدت که پس از فعال شدن نشست آزمایش رخ می‌دهد، تمرکز دارد و به مهاجم اجازه می‌دهد تا پس از ایجاد کلید نشست، درخواست ارسال کند، اما با محدودیت مدل تقسیم حالت [۶]. در این مدل، حالت مخفی یک سامانه رمزنگاری به بخش‌هایی تقسیم می‌شود و مهاجم می‌تواند به طور مستقل در هر قسمت تقسیم شده، نشتی مورد نظر خود را به دست آورد، البته نه نشت سراسری از کل کلید مخفی. اجرای رمزنگاری‌های اولیه در تقسیم حالت به این معنی است که مرحله اجرا به مراحل متوالی تقسیم می‌شود و در هر مرحله، یک بخش از کلید مخفی استفاده می‌شود. یک تابع نشت در هر یک از مراحل تقسیم شده برای

[1] Pseudo-random function [2] Alice [3] Bob

[4] After-the-fact leakage eCK model [5] Alawatugoda [6] Bounded- after- the-fact leakage eCK [7] Continuous- after- the- fact leakage eCK model

جدول ۲. پروتکل Mo

Alice	Bob
$(a_1,\ldots,a_8,a) \xleftarrow{U} (Z_q)^9$	$(b_1,\ldots,b_8,b) \xleftarrow{U} (Z_q)^9$
$(A_1,A_2,A_3,A_4,A_5,A_6) := (g_1^{a_1}g_2^{a_2}, g_1^{a_3}g_2^{a_4}, g_1^{a_5}g_2^{a_6}, g_1^{a_7}g_2^{a_8}, g_1^{a}, g_2^{a})$	$(B_1,B_2,B_3,B_4,B_5,B_6) := (g_1^{b_1}g_2^{b_2}, g_1^{b_3}g_2^{b_4}, g_1^{b_5}g_2^{b_6}, g_1^{b_7}g_2^{b_8}, g_1^{b}, g_2^{b})$
$(x,x_3) \xleftarrow{U} (Z_q)^2$	
$(X_1,X_2,X_3) := (g_1^{x}, g_2^{x}, g_1^{x_3})$	
$\xrightarrow{(\hat{B},\hat{A},X_1,X_2,X_3)}$	
	$(X_1,X_2,X_3) \in G^3?$
	$(y,y_3) \xleftarrow{U} (Z_q)^2$
	$(Y_1,Y_2,Y_3) := (g_1^{y}, g_2^{y}, g_1^{y_3})$
$\xleftarrow{(\hat{A},\hat{B},X_1,X_2,X_3,Y_1,Y_2,Y_3)}$	
$(Y_1,Y_2,Y_3) \in G^3?$	
$c := H_1(s), d := H_2(s), e := H_3(s)$	$c := H_1(s), d := H_2(s), e := H_3(s)$
$s := (\hat{A},\hat{B},X_1,X_2,X_3,Y_1,Y_2,Y_3)$	$s := (\hat{A},\hat{B},X_1,X_2,X_3,Y_1,Y_2,Y_3)$
$K_A := Y_1^{a_1+ca_3}Y_2^{a_2+ca_4}Y_3^{x_3}B_1^{x}B_2^{dx}B_3^{a}B_4^{ea}B_5^{a_5+ea_7}B_6^{a_6+ea_8}$	$K_B := X_1^{b_1+db_3}X_2^{b_2+db_4}X_3^{y_3}A_1^{y}A_2^{cy}A_3^{b}A_4^{eb}A_5^{b_5+eb_7}A_6^{b_6+eb_8}$
$SK_A := F_{K_A}(s)$	$SK_B := F_{K_B}(s)$

شبیه‌سازی نشت به کار گرفته می‌شود. پرسمان زیر در مدل BAFL-eCK تابع نشت را دربر دارد:

- پرسمان «ارسال» که به مهاجم اجازه می‌دهد تا پروتکل را مطابق مشخصات پروتکل اصلی اجرا کند و خروجی را به همراه تابع نشت به دست آورد.

با صدور این پرسمان که شامل تابع نشت است، مهاجم می‌تواند اطلاعاتی محدود در مورد کلید مخفی بلندمدت به دست آورد، به طور ویژه مقدار کل کلید مخفی بلندمدت به دست آمده توسط مهاجم می‌تواند به یک پارامتر نشت در هر بخش کلید مخفی محدود شود. در این مدل دو نوع پرسمان وجود دارد: پرسمان آسیب و پرسمان ارسال. پرسمان آسیب به مهاجم اجازه می‌دهد تا کل کلید مخفی بلندمدت پروتکل اصلی را بررسی کند. از سوی دیگر، پرسمان ارسال، مهاجم را قادر می‌سازد تا مقدار محدودی از اطلاعات لو رفته در مورد کلید مخفی بلندمدت را با تابع نشتی تعبیه شده در پرسمان «ارسال» بررسی کند. توجه به این نکته مهم است که پرسمان «آشکارکننده کلید موقت [1]» در پروتکل BAFL-eCK تصادفی بودن در فرآیند امضای نشست را فاش نمی‌کند. این با این اصل مطابقت دارد که طرح‌های امضای نشت‌تاب نباید تصادفی بودن را به طور کامل افشا کنند. این حالت که پرسمان «آشکارکننده کلید موقت» نمی‌تواند تصادفی بودن را در محاسبه امضا فاش کند، این مدل امنیتی را به wBAFL-eCK تضعیف می‌کند. این بحث مهم در کار آلاواتوگودا نادیده گرفته شده بود [۶]. یانگ [2] و همکاران در مقاله خود عنوان کردند که درصورت بروز نشتی کامل در فرآیند تصادفی بودن امضا، نشست‌های ناهماهنگ می‌توانند کلیدهای نشست مشابه را محاسبه کنند [۱۴]. مدل BAFL-eCK برای کاهش حملات راه‌انداز سرد [3]، که نوعی حمله کانال جانبی است، طراحی شده است. در حمله راه‌انداز سرد، مهاجمان به سامانه رمزنگاری دسترسی فیزیکی پیدا کرده و با یک راه‌اندازی مجدد، کلیدهای مخفی موقت یا بلندمدت پروتکل را بازیابی می‌کنند. BAFL-eCK بر اساس مدل eCK ساخته شده است و ویژگی‌های امنیتی آن را به ارث می‌برد. این مدل به طور خاص به نشت کلیدهای مخفی موقت یا بلندمدتی که ممکن است در حملات مورد سوء استفاده قرار گیرند، می‌پردازد. پرسمان آسیب به مهاجم اجازه می‌دهد تا از طریق یک حمله بدافزاری که پروتکل اصلی را هدف قرار می‌دهد، کلید مخفی بلندمدت را به دست آورد و این باعث افشای غیرمجاز یا سرقت کلید مخفی بلندمدت می‌شود.

۲.۳.۳ پروتکل π ایمن در مدل BAFL-eCK

در این پروتکل الگوریتم‌های KeyGen، Enc، Dec به ترتیب مسئول تولید کلید، رمزگذاری و رمزگشایی در سامانه رمزنگاری کلید عمومی PKE [4] هستند. به طور مشابه الگوریتم‌های Sign، KG، Vfy به ترتیب برای تولید امضا، تولید کلید و تایید امضای طرح امضای نشت‌تاب SIG به کار می‌روند. برای تولید کلید نشست به طول k از تابع استخراج‌کننده کلید امن نشست KDF [5] استفاده می‌شود. پروتکل π نوعی پروتکل توافق کلید دیفی-هلمن هست [۱۵] طوری که G گروهی از مرتبه اول q با مولد g می باشد. بسیار مهم است که q با اندازه فضای پیام M مطابقت داشته باشد. پس از تبادل مقادیر عمومی، هر دو شرکت‌کننده یک مقدار مخفی مشترک را با استفاده از رویکرد دیفی-هلمن محاسبه می‌کنند. پس از آن، کلید مخفی مشترک ms با استفاده از تابع استخراج‌کننده کلید

[1] Ephemeral Key Reveal [2] Yang

[3] Cold boot [4] Public-Key Encryption [5] Key derivation function

جدول ۳. پروتکل π

(initiator) A		(responder) B
راه‌انداز اولیه		
$sk_A, vk_A \xleftarrow{\$} kG(1^k)$		$sk_B, vk_B \xleftarrow{\$} kG(1^k)$
$s_A, p_A \xleftarrow{\$} \text{keyGen}(1^k)$		$s_B, p_B \xleftarrow{\$} \text{keyGen}(1^k)$
اجرای پروتکل		
$r_A \xleftarrow{\$} \hat{C}$		If $\text{Vfy}(vk_A, X_A, \sigma_A) = \text{"true"}$ {
$\widetilde{r_A} \leftarrow \underline{\text{Dec}}(S_A, r_A)$		$r_B \xleftarrow{\$} \hat{C}$
$X_A \leftarrow g^{\widetilde{r_A}}$		$\widetilde{r_B} \leftarrow \underline{\text{Dec}}(s_B, r_B)$
$\sigma_A \xleftarrow{\$} \underline{\text{sign}}(sk_A, (A, B, X_A))$	$\xrightarrow{A,B,X_A,\sigma_A}$	$X_B \leftarrow g^{\widetilde{B}}$
	$\xrightarrow{A,B,X_B,\sigma_B}$	$\sigma_B \xleftarrow{\$} \underline{\text{sign}}(sk_B, (B, A, S_B))$
$1 - 1\text{If}\text{Vfy}(vk_B, (B, A, X_B), \sigma_B) = \text{"}true\text{"}\{$		
$\widetilde{r_a} \leftarrow \underline{\text{Dec}}(s_A, r_A)$		$ms \leftarrow \text{KDF}(X_A^{\widetilde{r_B}}, \perp, k, \perp)$
$ms \leftarrow \text{KDF}(X_B^{\widetilde{r_A}}, \perp, k, \perp)$	K کلید نشست است	$k \leftarrow \text{PRF}(m_s, A\|X_A\|\sigma_A B\|X_B\|\sigma_B)$
$k \leftarrow \text{PRF}(ms, A\|X_A\|\sigma_A\|B\|X_B\|\sigma_B)$		}
}		

KDF، محاسبه می‌شود و درنهایت تابع شبه‌تصادفی PRF برای محاسبه کلید نشست به کار گرفته می‌شود. در جدول ۳ که ساختار کلی پروتکل π را نمایش می‌دهد، زیرمحاسباتی که منجر به نشت اطلاعات می‌شوند، خط کشیده شده است.

۳.۳.۳ مدل CAFL-eCK

مدل نشت مداوم پس از وقوع eCK (CAFL-eCK) که توسط آلاواتوگودا و همکاران پیشنهاد شده است [۷]، شبیه به BAFL-eCK است اما با یک ویژگی متمایز. در این مدل مهاجم این توانایی را دارد که به طور مداوم اطلاعات لو رفته در مورد هر بخش از کلید مخفی را به دست آورد. با این‌حال، محدودیتی در میزان نشت در هر رخداد وجود دارد که توسط یک پارامتر نشت خاص تعیین می‌شود. آن‌ها اولین پروتکل تبادل کلید ایمن در برابر نشت مداوم را معرفی کردند که از روش استخراج‌کننده ترکیب داخلی توسط زیمبوسکی [۱] و فاوست [۲] استفاده می‌کند [۱۶]. ایمن بودن پروتکل تبادل کلید تحت مدل امنیتی CAFL-eCK ثابت شده است. این مدل دارای یک سامانه ذخیره‌سازی حالت تقسیم نشت‌تاب است که در آن عناصر با استفاده از یک روش رمزگذاری تصادفی به دو بخش تقسیم می‌شوند. این رمزگذاری‌ها را می‌توان به شیوه‌ی نشت‌تاب به‌روزرسانی کرد و امکان استفاده مجدد را فراهم کرد. در نتیجه، پروتکل به تنظیمات اولیه نشت‌تاب مداوم دست می‌یابد. هر کدام از کلیدهای رمزنگاری به بخش‌های متوالی تقسیم می‌شوند و در مراحل مختلف به طور جداگانه مورد استفاده قرار می‌گیرند. در دنیای واقعی، مدل امنیتی CAFL-eCK می‌تواند به عنوان چارچوبی برای پروتکل تبادل کلید استفاده شود. این چارچوب بر کاهش حملات کانال جانبی نشت مداوم مانند حمله تحلیل توان [۳] تمرکز دارد. هر زمان که محاسباتی در سامانه‌های رمزنگاری انجام شود، این نوع حملات به طور مداوم اطلاعات محرمانه داخلی را درز می‌کنند.

۴.۳.۳ پروتکل P_2 ایمن در مدل CAFL-eCK

این پروتکل از دو مرحله راه‌انداز اولیه و اجرای پروتکل به صورت زیر تشکیل شده است: ابتدا $a \xleftarrow{\$} \mathbb{Z}_q^*$ به نوان کلید مخفی بلندمدت انتخاب و به دنبال آن کلید مخفی به صورت $(a_L^\circ, a_R^\circ) \leftarrow \text{Encode}_{\mathbb{Z}_q^*}^{n,1}(a)$ رمزگذاری می‌شود. سپس کلید عمومی بلندمدت $A = g^a$ با استفاده از (a_L°, a_R°) محاسبه می‌شود. در آخر a از حافظه پاک می‌شود. تهدید بالقوه در فرایند تولید کلید این است که حتی اگر کلید مخفی بلندمدت a از حافظه پاک شود، ممکن است به درستی پاک نشود و در حین تولید کلید، دست مهاجم بیفتد. به منظور جلوگیری از چنین ضعفی، دو مقدار $a_L^\circ \xleftarrow{\$} (\mathbb{Z}_q^*)^n$ و $a_R^\circ \xleftarrow{\$} (\mathbb{Z}_q^*)^{n\times 1}$ به صورت تصادفی انتخاب و از آن‌ها برای رمزگذاری کلید مخفی بلندمدت a پروتکل اصلی استفاده می‌شود. همان‌طور که بیان شد، a_L°، a_R° برای محاسبه کلید عمومی بلندمدت A طی دو مرحله $a' \leftarrow g^{a_L^\circ}$ و $A \leftarrow a'^{a_R^\circ}$ به کار می‌رود. بدین ترتیب، می‌توان از نشت مستقیم کلید مخفی رمزگذاری نشده a در فرایند تولید کلید در هر مقطع زمانی جلوگیری کرد. به علاوه از حاصلضرب دو بردار تصادفی a_L° و a_R° به صورت $a = a_L^\circ \cdot a_R^\circ$، عدد تصادفی $a \in \mathbb{Z}_q^*$ حاصل می‌شود. این روش، معکوس روشی است که در آن ابتدا $a \xleftarrow{\$} \mathbb{Z}_q^*$ انتخاب و سپس رمزگذاری می‌شود. در طی فرایند اجرای پروتکل هر دوی a_L° و a_R°

[1] Dziembowski [2] Faust

[3] Power analysis attack

جدول ۴. پروتکل $P_۲$

آلیس(Alice)		باب(Bob)
$a_L^{\circ} \xleftarrow{\$} (\mathbb{Z}_q^*)^n, a_R^{\circ} \xleftarrow{\$} (\mathbb{Z}_q^*)^{n\times ۱}$ $a' \leftarrow g^{a_L^{\circ}}, A \leftarrow (a')^{a_R^{\circ}}$	راه‌انداز اولیه	$b_L^{\circ} \xleftarrow{\$} (\mathbb{Z}_q^*)^n, b_R^{\circ} \xleftarrow{\$} (\mathbb{Z}_q^*)^{n\times ۱}$ $b' \leftarrow g^{b'_L}, B \leftarrow (b')^{b_R^{\circ}}$
$x \xleftarrow{\$} \mathbb{Z}_q^*, X \leftarrow g^x$	اجرای پروتکل $\xleftarrow{\text{Alice},X}$ $\xrightarrow{\text{Bob},Y}$	$y \xleftarrow{\$} \mathbb{Z}_q^*, Y \leftarrow g^y$
$T_۱ \leftarrow B^{a'_L}, Z_۱ \leftarrow T_۱^{a'_R}$ $Z_۲ \leftarrow B^x$ $T_۲ \leftarrow Y^{a'_L}, Z_۳ \leftarrow T_۲^{a'_R}$ $Z_۴ \leftarrow Y^x$ $\left(a_L^{j+۱}, a_R^{j+۱}\right) \leftarrow \text{Refresh}_{\mathbb{Z}_q^*}^{n,۱}(a'_L, a'_R)$ $k \leftarrow H(Z_۱, Z_۲, Z_۳, Z_۴, \text{Alice}, X, \text{Bob}, Y)$	K کلید نشست است	$T_۳ \leftarrow A^{b'_L}, Z'_۱ \leftarrow T_۳^{b'_R}$ $T_۴ \leftarrow X^{b'_L}, Z'_۲ \leftarrow T_۴^{b'_R}$ $Z'_۳ \leftarrow A^y$ $Z'_۴ \leftarrow X^y$ $\left(b_L^{j+۱}, b_R^{j+۱}\right) \leftarrow \text{Refresh}_{\mathbb{Z}_q^*}^{n,۱}(b'_L, b'_R)$ $k \leftarrow H(Z'_۱, Z'_۲, Z'_۳, Z'_۴, \text{Alice}, X, \text{Bob}, Y)$

به طور مداوم تازه‌سازی شده و رمزگذاری‌های تازه‌شده a_L^j و a_R^j برای محاسبات توان به کار گرفته می‌شوند. نشت یک کلید مخفی بلندمدت به طور مستقیم از خود کلید اتفاق نمی‌افتد، بلکه از دو رمزگذاری کلید مخفی بلندمدت که با تابع نشتی $\mathbf{f} = (f_{۱j}, f_{۲j})$ نشان داده می‌شود، رخ می‌دهد. در محاسبات توان و عملیات تازه‌سازی، حداکثر نشتی $\boldsymbol{\lambda} = (\lambda_۱, \lambda_۲)$ از هر بخش، به طور مستقل مجاز است. پس از آن، دو بخش کلید مخفی بلندمدت رمزگذاری و تازه‌سازی شده و در نشست پروتکل بعدی، نشتی λ-کراندار دیگری امکان‌پذیر می‌شود. همین امر، امکان نشتی مداوم را فراهم می‌کند. پروتکل $P_۲$ با جزئیات بیشتر در جدول ۴ نمایش داده شده است.

۴.۳ مدل GCL-eCK

وو [۱] و همکاران یک پروتکل توافق کلید احراز اصالت شده نشت‌تاب کارآمد را در مدل نشت مداوم eCK ارائه می‌کنند [۸]. آن‌ها کار آلاواتوگودا و همکاران را بهبود بخشیدند و از یک روش تازه‌سازی کلید سازگار با روش استخراج ترکیب داخلی برای به‌روز کردن کلیدهای مخفی بلندمدت بعد از اجرای هر پروتکل استفاده کردند [۷، ۱۶]. در مدل GCL-eCK، وو و همکاران از روش کورسازی چندگانه به عنوان یک جایگزین برای روش استخراج‌کننده ترکیب داخلی زمان-ناکارآمد [۲] استفاده کردند. این مدل به مهاجم امکان می‌دهد تا مقدار مشخصی از نشت اطلاعات را در هر بخش نشست پروتکل فاش کند. با این حال، مهاجم قادر است مقدار نامحدودی از نشت اطلاعات را در کل چرخه سامانه رمزنگاری جمع آوری کند. برای دستیابی به تنظیمات نشت بدون محدودیت در چرخه سامانه رمزنگاری، یک طرح رمزنگاری معمولاً شامل چندین دور محاسباتی است طوری که در هر دور، کلیدهای مخفی بلندمدت درگیر به روز می‌شوند. نشت کلیدهای مخفی بلندمدت در هر دور، مستقل از یکدیگر است.

علاوه بر این، مقدار اطلاعات نشت در هر دور محاسباتی به بخش خاصی محدود می‌شود. این رویکرد با تازه کردن کلیدها و محدودکردن مقدار اطلاعاتی که در هر دور به دست می‌آید به کاهش نشت کمک می‌کند. یک پرسمان خاص مدل GCL-eCK برای اعمال تنظیمات نشت کلی بدون محدودیت وجود دارد:

- پرسمان نشت [۳] مهاجم را قادر می‌سازد تا اطلاعات نشت جزئی را در مورد کلیدهای مخفی موقت و بلندمدت مورد استفاده در هر دو حالت تازه‌سازی کلید و توافق کلید به دست آورد.

مدل GCL-eCK بر طرح‌های حمله‌ای در دنیای واقعی که شامل تنظیمات نشت مداوم است، تمرکز دارد. این مدل به طور موثر انواع مختلفی از حملات کانال جانبی را که به طور مداوم پارامترهای حساس را در طول محاسبات سامانه رمزنگاری نشت می‌کند، بررسی می‌کند. در مدل GCL-eCK، پرسمان نشتی به طور خاص برای مقابله با حملات نشت جزئی مداوم بر روی کلیدهای مخفی موقت و بلندمدت تعریف شده است. این حملات این توانایی را دارند که به تدریج حتی مقادیر کمی از اطلاعات مخفی را در طول اجرای مداوم سامانه رمزنگاری فاش کنند. این مدل برای تامین امنیت در زمینه تنظیمات نشت مداوم طراحی شده است.

۱.۴.۳ پروتکل LR-AKA ایمن در مدل GCL-eCK

در این بخش به مرور پروتکل LR-AKA می‌پردازیم که از دو مرحله راه‌انداز اولیه و ساخت کلید نشست تشکیل شده است. مرحله ساخت کلید نشست نیز دارای دو زیرمحله بروز کردن و توافق کلید است.

مرحله راه‌اندازی اولیه: بر اساس مقدار امنیتی κ، سامانه در ابتدا مقادیر عمومی $\{G, G_T, \hat{e}, p, g, h\}$ را انتخاب می‌کند که در آن G گروه دوری ضربی با مولد g، p عدد اولی بزرگ و G_T گروه ضربی دوری از مرتبه p، h مولد تصادفی از گروه G و $\hat{e}$ نگاشت زوج‌نگار دوخطی می‌باشند.

[1] Wu [2] Time-inefficient inner product axtractor

[3] Leak Query

آلیس و باب جفت کلید خصوصی و عمومی خود را به صورت زیر تولید می‌کنند:

- آلیس مقادیر تصادفی a و $\alpha_0 \in Z_p^*$ را انتخاب می‌کند و دو جفت کلید خصوصی اولیه $((SA_{0,1}, SA_{0,2}) = (g^{\alpha_0}, g^{a-\alpha_0}))$ و $((XA_{0,1}, XA_{0,2}) = (\hat{e}(SA_{0,1}, h), \hat{e}(SA_{0,2}, h)))$ را محاسبه می‌کند. داریم $X_A = XA_{0,1} \cdot XA_{0,2} = \hat{e}(g^a, h)$ و $SA = SA_{0,1} \cdot SA_{0,2} = g^a$. به علاوه آلیس زوج کلید عمومی را به صورت $(TA, PA) = (\hat{e}(g, SA), \hat{e}(g, g)^{XA})$ محاسبه می‌کند.
- به طور مشابه، باب نیز دو مقدار تصادفی $b, \beta_0 \in Z_p^*$ را برمی‌گزیند و دو جفت کلید خصوصی $(SB_{0,1}, SB_{0,2})$ و $(XB_{0,1}, XB_{0,2})$ و نیز یک جفت کلید عمومی $(TB, PB) = (\hat{e}(g, SB), \hat{e}(g, g)^{XB})$ را محاسبه می‌کند طوری که $SB = SB_{0,1} \cdot SB_{0,2} = g^b$ و $XB = XB_{0,1} \cdot XB_{0,2} = \hat{e}(g^b, h)$.

مرحله ساخت کلید نشست: برای ساخت کلید نشست مشترک بین نشست i–ام آلیس و j–ام باب خواهیم داشت:

- تازه‌سازی کلید: آلیس مقدار تصادفی $\alpha_i \in Z_p^*$ را انتخاب و جفت کلید خصوصی خود را با

$$(SA_{i,1} = SA_{i-1,1} \cdot g^{\alpha i}, SA_{i,2} = SA_{i-1,2} \cdot g^{-\alpha i})$$

و

$$(XA_{i,1} = XA_{i-1,1} \cdot \hat{e}(g^{\alpha_i}, h), \\ XA_{i,2} = XA_{i-1,2} \cdot \hat{e}(g^{-\alpha_i}, h))$$

به‌روز می‌کند. به صورت مشابه، باب نیز مقدار تصادفی $\beta_j \in Z_p^*$ را انتخاب و اقدام به به‌روز کردن جفت کلید خصوصی خود

$$(SB_{j,1} = SB_{j-1,1} \cdot g^{\beta j}, SB_{j,2} = SB_{j-1,2} \cdot g^{-\beta j})$$

و

$$(XB_{j,1} = XB_{j-1,1} \cdot \hat{e}(g^{\beta_j}, h), \\ XB_{j,2} = XB_{j-1,2} \cdot \hat{e}(g^{-\beta_j}, h))$$

می‌کند. این روش به‌روز کردن کلید به روش کورسازی چندگانه مشهور است. حال داریم:

$$SA_{i,1} \cdot SA_{i,2} = g^a = SA = SA_{0,1} \cdot SA_{0,2} \quad (۱)$$
$$XA_{i,1} \cdot XA_{i,2} = \hat{e}(g^a, h) = XA = XA_{0,1} \cdot XA_{0,2} \quad (۲)$$
$$SB_{j,1} \cdot SB_{j,2} = g^b = SB = SB_{0,1} \cdot SB_{0,2} \quad (۳)$$
$$XB_{j,1} \cdot XB_{j,2} = \hat{e}(g^b, h) = XB = XB_{0,1} \cdot XB_{0,2} \quad (۴)$$

- توافق کلید: در این مرحله، آلیس کلید مخفی موقت x را انتخاب و $X = g^x$ را محاسبه می‌کند. سپس X را به باب ارسال می‌کند. در مقابل، باب نیز کلید مخفی موقت خود y را انتخاب و $Y = g^y$ را محاسبه و به آلیس ارسال می‌کند. حال آلیس برای محاسبه کلید نشست $SK_{A,i}$ با استفاده از کلید عمومی باب (TB, PB)، Y و جفت کلید خصوصی خود $(SA_{i,1}, SA_{i,2})$ و $(XA_{i,1}, XA_{i,2})$ به صورت زیر عمل می‌کند:

$$KA_{i,1} = TB^x \quad (۱)$$
$$KA_{i,2} = \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) \quad (۲)$$
$$KA_{i,3} = Y^x \quad (۳)$$
$$KA_{i,4} = (PB^{XA_{i,1}})^{XA_{i,2}} \quad (۴)$$
$$SK_{A,i} = KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4} \quad (۵)$$

به طور مشابه، باب نیز جفت کلید خصوصی خود $(SB_{j,1}, SB_{j,2})$ و $(XB_{j,1}, XB_{j,2})$ را برای محاسبه کلید نشست $SK_{B,j}$ به صورت زیر به کار می‌گیرد:

$$KB_{j,1} = TA^y \quad (۱)$$
$$KB_{i,2} = \hat{e}(X, SB_{j,1}) \cdot \hat{e}(X, SB_{j,2}) \quad (۲)$$
$$KB_{j,3} = Y^x \quad (۳)$$
$$KB_{j,4} = (PA^{XB_{j,1}})^{XB_{j,2}} \quad (۴)$$
$$SK_{B,j} = KB_{j,1} \oplus KB_{j,2} \oplus KB_{j,3} \oplus KB_{j,4} \quad (۵)$$

چون $SK_{A,i} = KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4}$ و $SK_{B,j} = KB_{j,1} \oplus KB_{j,2} \oplus KB_{j,3} \oplus KB_{j,4}$ برقرار است، لذا بررسی درستی $SK_{A,i} = SK_{B,j}$ به صورت زیر است:

(آ)

$$\begin{aligned} KA_{i,1} &= TB^x = \hat{e}(g,g)^{bx} = \hat{e}(g^x, g^b) \\ &= \hat{e}(X, g^b) = \hat{e}(X, SB_{j,1} \cdot SB_{j,2}) \\ &= \hat{e}(X, SB_{j,1}) \cdot \hat{e}(X, SB_{j,2}) = KB_{j,2} \end{aligned}$$

(ب)

$$\begin{aligned} KA_{i,2} &= \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2}) \\ &= \hat{e}(Y, SA_{i,1} \cdot SA_{i,2}) = \hat{e}(Y, g^a) = \hat{e}(g^y, g^a) \\ &= \hat{e}(g, g^a)^y = TA^y = KB_{j,1} \end{aligned}$$

(ج)

$$KA_{i,3} = Y^x = g^{yx} = g^{xy} = X^y = KB_{j,3}$$

(د)

$$\begin{aligned} KA_{i,4} &= (PB^{XA_{i,1}})^{XA_{i,2}} = (PB^{XA_{i,1} \cdot XA_{i,2}}) \\ &= \hat{e}(g,g)^{XB.XA} = PA^{XB} = (PA^{XB_{j,1} \cdot XB_{j,2}} \\ &= KB_{j,4}. \end{aligned}$$

بنابراین، $SK_{A,i} = SK_{B,j}$.

در این پروتکل به جای استفاده از روش زمان‌بر استخراج ترکیب داخلی برای تازه‌سازی کلید، از روش کورکننده ضربی استفاده شده است. این روش شامل تقسیم کلید خصوصی به دو بخش است که نشت هر بخش مستقل از دیگری است. هنگامی که این دو مؤلفه فعلی برای ساختن یک کلید نشست استفاده می‌شوند، باید به‌روزرسانی شوند و به دو بخش جدید کلید خصوصی برای استفاده مجدد در آینده تبدیل شوند (جدول ۵).

جدول ۵. پروتکل LR- AKA

مرحله راه‌انداز	
مقادیر عمومی = $\{G, G_T, \hat{e}, p, g, h\}$	
باب	آلیس
جفت کلید خصوصی: $(SA_{\circ,1}, SA_{\circ,2}) = (g^{\alpha_\circ}, g^{a-\alpha_\circ})$	**جفت کلید خصوصی:** $(SB_{\circ,1}, SB_{\circ,2}) = (g^{\beta_\circ}, g^{a-\beta_\circ})$
جفت کلید خصوصی: $(XA_{\circ,1}, XA_{\circ,2}) = (\hat{e}(SA_{\circ,1}, h), \hat{e}(SA_{\circ,2}, h))$	**جفت کلید خصوصی:** $(XB_{\circ,1}, XB_{\circ,2}) = (\hat{e}(SB_{\circ,1}, h), \hat{e}(SB_{\circ,2}, h))$
جفت کلید عمومی: $(TA, PA) = (\hat{e}(g, SA), \hat{e}(g, g)^{XA})$	**جفت کلید عمومی:** $(TB, PB) = (\hat{e}(g, SB), \hat{e}(g, g)^{XB})$
$X_A = XA_{\circ,1} \cdot XA_{\circ,2} = \hat{e}(g^a, h)$ و $SA = SA_{\circ,1} \cdot SA_{\circ,2} = g^a$	$XB = XB_{\circ,1} \cdot XB_{\circ,2} = \hat{e}(g^b, h)$ و $SB = SB_{\circ,1} \cdot SB_{\circ,2} = g^b$
مرحله ساخت کلید نشست	
تازه‌سازی کلید	
باب	آلیس
$\beta_j \in Z_p^*$	$\alpha_i \in Z_p^*$
$(SB_{j,1} = SB_{j-1,1} \cdot g^{\beta j}, SB_{j,2} = SB_{j-1,2} \cdot g^{-\beta j})$	$(SA_{i,1} = SA_{i-1,1} \cdot g^{\alpha i}, SA_{i,2} = SA_{i-1,2} \cdot g^{-\alpha i})$
$(XB_{j,1} = XB_{j-1,1} \cdot \hat{e}(g^{\beta_j}, h), XB_{j,2} = XB_{j-1,2} \cdot \hat{e}(g^{-\beta_j}, h))$	$(XA_{i,1} = XA_{i-1,1} \cdot \hat{e}(g^{\alpha_i}, h), XA_{i,2} = XA_{i-1,2} \cdot \hat{e}(g^{-\alpha_i}, h))$
توافق کلید	
باب	آلیس
$y \in Z_p^*$	$x \in Z_p^*$
$Y = g^y$	$X = g^x$
	$\xrightarrow{X}$
	$\xleftarrow{Y}$
$KB_{j,1} = TA^y$	$KA_{i,1} = TB^x$
$KB_{i,2} = \hat{e}(X, SB_{j,1}) \cdot \hat{e}(X, SB_{j,2})$	$KA_{i,2} = \hat{e}(Y, SA_{i,1}) \cdot \hat{e}(Y, SA_{i,2})$
$KB_{j,3} = X^y$	$KA_{i,3} = Y^x$
$KB_{j,4} = (PA^{XB_{j,1}})^{XB_{j,2}}$	$KA_{i,4} = (PB^{XA_{i,1}})^{XA_{i,2}}$
$SK_{B,j} = KB_{j,1} \oplus KB_{j,2} \oplus KB_{j,3} \oplus KB_{j,4}$	$SK_{A,i} = KA_{i,1} \oplus KA_{i,2} \oplus KA_{i,3} \oplus KA_{i,4}$

فرآیند تازه‌سازی کلید تضمین می‌کند که پروتکل پیشنهادی LR-AKA ویژگی نشتی نامحدود را حفظ کند. این بدان معنی است که یک مهاجم فقط می‌تواند نشت جزئی اطلاعات را در مورد دو بخش فعلی به دست آورد. وو و همکاران امنیت پروتکل LR-AKA را در مدل نشت مداوم eCK، ثابت کردند [۸]. علاوه بر این، تجزیه و تحلیل عملکرد نشان داد که پروتکل LR-AKA هم برای دستگاه‌های تلفن همراه و هم برای رایانه‌های شخصی مناسب است.

۵.۳ مدل CLR-eCK

چن [1] و همکاران، یک مدل امنیتی به نام نشت‌تاب چالش محور-eCK را برای تبادل کلید احراز اصالت شده معرفی کردند. هدف این مدل این است که هر دو نشتی در کلیدهای مخفی بلندمدت و موقت را بررسی کند [۹]. در این مدل نشت جزئی کلید مخفی (بلندمدت یا موقت) و نشتی کامل کلید دیگر (موقت یا بلندمدت) مجاز است. این یک مدل بدون حالت تقسیم است که امکان انجام پرسمان‌های نشتی را قبل، در طول و بعد از نشست چالش فراهم می‌کند، اینجاست که مفهوم نشت وابسته به چالش معنا می‌یابد. در طرح‌های دنیای واقعی، ممکن است مهاجمانی وجود داشته باشند که بتوانند یک کلید مخفی موقت یا بلندمدت را فاش کنند و در عین حال کلید مخفی موقت یا بلندمدت دیگر را به طور نسبی فاش کنند. آن‌ها همچنین مدل نشت نسبی معرفی شده توسط آکاویا [2] و همکاران را نیز در نظر می‌گیرند [۱۲]. این چارچوب را می توان به عنوان یک نسخه جایگزین از مدل تبادل کلید احراز اصالت شده دانست که توسط اوکاموتو [3] و همکاران ارائه شده است [۱۰]. در این مدل از اصول رمزنگاری اولیه مانند توابع شبه تصادفی، خانواده توابع شبه تصادفی مستقل جفتی و استخراج‌کننده‌های تصادفی قوی استفاده شده است. بنابراین آن‌ها چارچوبی کلی را مطرح کردند که تحت فرض تصمیم

[1] Chen

[2] Akavia [3] Okamoto

دیفی–هلمن بدون نیاز به یک پیشگوی تصادفی نمونه‌سازی شده است. مدل CLR-eCK مدل امنیتی eCK را توسعه می‌دهد و ویژگی‌های امنیتی آن را به ارث می‌برد و نیز اطلاعات نشتی بیشتری را در مقایسه با مدل eCK به دست می‌دهد تا مهاجم پرسمان‌های نشت را در طول فعال‌سازی نشست چالش ارسال کند. برای درک مفهوم نشت وابسته به چالش، مدل CLR-eCK دو پرسمان مخصوص به خود را معرفی می‌کند:

- نشت کلید بلندمدت، که به مهاجم اجازه می‌دهد تا تابع نشت دلخواه کلید مخفی بلندمدت یک بخش را قبل از به دست آوردن کلید مخفی موقت درخواست کند،
- نشت کلید موقت، که به مهاجم اجازه می‌دهد تا تابع نشت دلخواه کلید مخفی موقت را قبل از به دست آوردن کلید مخفی بلندمدت درخواست کند.

علاوه بر این، مهاجم می‌تواند تا زمانی که محدودیت برقرار است، توابع نشت (نشت کلید موقت یا نشت کلید بلندمدت) را در حین و حتی پس از مرحله چالش انتخاب کند.

۱.۵.۳ پروتکل CLR-eCK secure AKA ایمن در مدل CLR-eCK

فرض کنیم k پارامتر امنیتی سامانه و $\mathbb{G}$ گروهی از مرتبه عدد اول p با مولد تصادفی g باشد. فرض کنیم $\mathcal{SPHF}$ بیانگر یک SPHF ۲–هموار روی $\mathcal{L} \subset \mathcal{X}$ و به روی مجموعه $\mathcal{Y}$ باشد طوری که مسئله عضویت زیرمجموعه بین $\mathcal{L}$ و $\mathcal{X}$ مشکل باشد. فضای کلید چکیده‌ساز را با $\mathcal{HK}$، فضای کلید طرح‌ریزی را با $\mathcal{HP}$، فضای ورودی کمکی را با $\mathcal{AUX}$ و فضای شاهد را با $\mathcal{W}$ نمایش داده و دو تابع چکیده‌ساز برخوردتاب $H_1 : \{0,1\}^* \to \mathcal{AUX}$ و $H_2 : \mathbb{G}^2\mathcal{Y}$ را در نظر می‌گیریم. حال فرض کنیم $\lambda_1 = \lambda_1(k)$ و $\lambda_2 = \lambda_2(k)$ به ترتیب کران‌هایی بر میزان نشت کلید مخفی بلندمدت و موقت باشند. Ext_1، Ext_2 و Ext_3 را به صورت زیر به عنوان استخراج‌کننده‌های قوی در نظر می‌گیریم:

$$\mathrm{Ext}_1 : \mathcal{HK} \times \{0,1\}^{t_1(k)} \to \{0,1\}^{l_1(k)},$$
$$\mathrm{Ext}_2 : \{0,1\}^{u(k)} \times \{0,1\}^{t_2(k)} \to\to \{0,1\}^{l_2(k)},$$
$$\mathrm{Ext}_3 : \mathcal{Y} \times \{0,1\}^{t_3(k)} \to \{0,1\}^{l_3(k)}$$

به ترتیب، استخراج‌کننده قوی با حالت متوسط $(|\mathcal{HK}|\lambda_1, \epsilon_1)$، $(k\lambda_2, \epsilon_2)$ و $(|\mathcal{Y}|\lambda_1, \epsilon_3)$ است طوری که $\epsilon_1 = \epsilon_1(k)$، $\epsilon_2 = \epsilon_2(k)$ و $\epsilon_3 = \epsilon_3(k)$ ناچیز هستند. فرض کنیم $\hat{F}$ و $\bar{F}$ خانواده توابع شبه تصادفی (PRF [1]) و نیز $\tilde{F}$ خانواده‌ای از πPRF به صورت زیر باشد:

$$\hat{F}^{k,\sum_{\hat{F},\mathcal{D}_{\hat{F}},\mathcal{R}_{\hat{F}}},\sum_{\hat{F}}} = \{0,1\}^{l_1(k)},$$
$$\mathcal{D}_{\hat{F}} = \{0,1\}^{u(k)}, \mathcal{R}_{\hat{F}} = \mathcal{W} \times \mathbb{Z}_p,$$
$$\bar{F}^{k,\sum_{\bar{F},\mathcal{D}_{\bar{F}},\mathcal{R}_{\bar{F}}},\sum_{\bar{F}}} = \{0,1\}^{l_2(k)},$$
$$\mathcal{D}_{\bar{F}} = \{0,1\}^{t_1(k)}, \mathcal{R}_{\bar{F}} = \mathcal{W} \times \mathbb{Z}_p,$$
$$\tilde{F}^{k,\sum_{\tilde{F},\mathcal{D}_{\tilde{F}},\mathcal{R}_{\tilde{F}}},\sum_{\tilde{F}}} = \{0,1\}^{l_3(k)},$$
$$\mathcal{D}_{\tilde{F}} = (\Lambda_k)^2 \times \mathcal{L}^2 \times \mathbb{G}^2 \times \{0,1\}^{2t_2(k)}, \mathcal{R}_{\tilde{F}} = \{0,1\}^{l_4(k)}.$$

فرض کنیم $\hat{F} \leftarrow \hat{F}^{k,\sum_{\hat{F},\mathcal{D}_{\hat{F}},\mathcal{R}_{\hat{F}}}}$، $\bar{F} \leftarrow \bar{F}^{k,\sum_{\bar{F},\mathcal{D}_{\bar{F}},\mathcal{R}_{\bar{F}}}}$ و $\tilde{F} \leftarrow \tilde{F}^{k,\sum_{\tilde{F},\mathcal{D}_{\tilde{F}},\mathcal{R}_{\tilde{F}}}}$ برقرار باشند در این صورت، پارامتر سامانه $(\mathrm{param}, \mathbb{G}, p, g, H_1, H_2, \mathrm{Ext}_1, \mathrm{Ext}_2, \mathrm{Ext}_3, \hat{F}, \bar{F}, \tilde{F})$ می‌باشد طوری که $\mathrm{param} \leftarrow \mathrm{SPHFSetup}(1^k)$.

تحلیل درستی: با توجه به $K_{\mathcal{A}_1} = Y^x = X^y = K_{\mathcal{B}_1} = g^{xy}$ داریم $K_{\mathcal{A}_1} = K_{\mathcal{B}_1}$. از طرفی با توجه به ویژگی $SPHF$ داریم:

$$\begin{aligned} K_{\mathcal{A}_2} &= \mathrm{ProjHash}(\mathrm{param}, \mathcal{L}, lpk_{\mathcal{B}}, W_{\mathcal{A}}, \omega_{\mathcal{A}}, aux) \\ &= \mathrm{Hash}(\mathrm{param}, \mathcal{L}, \mathrm{lsk}_{\mathcal{B}}, W_{\mathcal{A}}, aux) = K_{\mathcal{B}_2}, \\ K_{\mathcal{A}_3} &= \mathrm{Hash}(\mathrm{param}, \mathcal{L}, lsk_{\mathcal{A}}, W_{\mathcal{B}}, aux) \\ &= \mathrm{ProjHash}(\mathrm{param}, \mathcal{L}, lpk_{\mathcal{A}}, W_{\mathcal{B}}, \omega_{\mathcal{B}}, aux) = K_{\mathcal{B}_3}. \end{aligned}$$

بنابراین، می‌توان

$$s_{\mathcal{A}} = \mathrm{Ext}_3(H_2(K_{\mathcal{A}_1}) \oplus K_{\mathcal{A}_2} \oplus K_{\mathcal{A}_3}, t_{\mathcal{A}} \oplus t_{\mathcal{B}}) =$$
$$s_{\mathcal{B}} = \mathrm{Ext}_3(H_2(K_{\mathcal{B}_1}) \oplus K_{\mathcal{B}2} \oplus K_{\mathcal{B}_3}, t_{\mathcal{A}} \oplus t_{\mathcal{B}})$$

را به دست آورد که $\mathrm{SK}_{\mathcal{A}} = \mathrm{SK}_{\mathcal{B}}$ می‌باشد. جدول ۶ بیانگر جزئیات بیشتر است.

۴ مقایسه مدل‌ها بر اساس نشتی

جدول ۷ مقایسه ویژگی‌های امنیتی ذکر شده برای هر مدل امنیتی بررسی شده در بخش ۲ را خلاصه می‌کند. همه مدل‌های امنیتی در جدول ۷ با مدل نشت نسبی توسط آکاویا و همکاران رسمیت یافته‌اند و به طور ضمنی مفهوم حملات نشتی را معرفی می‌کند که بخشی از کلید مخفی را بدون توجه به اندازه آن فاش می‌کند [۱۲]. مدل CAFL-eCK در AFL-eCK(.)، یک نوع نشت مداوم با نشت کلی بدون محدودیت در اجرای پروتکل است، اما با یک مقدار ثابت برای هر رخداد. این تنظیم نشت زمانی جالب است که از رمزنگاری‌های اولیه ذخیره‌سازی نشت‌تاب برای نمونه‌سازی یک پروتکل عمومی برای انواع نشت مداوم در AFL-(.)eCK، استفاده شود. در مدل CAFL-eCK، روش تازه‌سازی کلید به نام استخراج‌کننده ترکیب داخلی به کار گرفته شده است. این روش از یک طرح رمزگذاری استفاده می‌کند تا کلید را به بخش‌هایی تقسیم کند و سپس آن را برای استفاده مجدد به‌روز کند. به طور مشابه، مدل GCL-eCK یک پروتکل عمومی برای تنظیم نشت مداوم در مدل eCK با استفاده از ایده ذخیره‌سازی تقسیم‌کننده به نام روش کورسازی چندگانه برای بروز کردن کلید ارائه می‌کند. روش کورسازی چندگانه، کلید مخفی را به بخش‌هایی با نشت مستقل از هم تقسیم می‌کند تا به هنگام درگیر شدن در اجرا،

[1] Pseudo- random function

جدول ۶. پروتکل AKE secure CLR-eCK

$\mathcal{A}$	$\mathcal{B}$
تولید کلید بلندمدت	
$hk \xleftarrow{\$} \text{HashKG}(\text{param}, \mathcal{L})$,	$hk' \xleftarrow{\$} \text{HashKG}(\text{param}, \mathcal{L})$
$hp \xleftarrow{\$} \text{ProjKG}(\text{param}, \mathcal{L}, hk)$,	$hp' \xleftarrow{\$} \text{ProjKG}(\text{param}, \mathcal{L}, hk')$,
$r_{\mathcal{A}_1} \xleftarrow{\$} \{0,1\}^{t_1(k)}, r_{\mathcal{A}_2} \xleftarrow{\$} \{0,1\}^{t_2(k)}$	$r_{\mathcal{B}_1} \xleftarrow{\$} \{0,1\}^{t_1(k)}, r_{\mathcal{B}_2} \xleftarrow{\$} \{0,1\}^{t_2(k)}$,
$lsk_{\mathcal{A}} = hk, lpk_{\mathcal{A}} = (hp, r_{\mathcal{A}_1}, r_{\mathcal{A}_2})$.	$lsk_{\mathcal{B}} = hk', lpk_{\mathcal{B}} = (hp', r_{\mathcal{B}_1}, r_{\mathcal{B}_2})$.
اجرای نشست	
$esk_{\mathcal{A}} \xleftarrow{\$} \{0,1\}^{u(k)}, t_{\mathcal{A}} \xleftarrow{\$} \{0,1\}^{t_2(k)}$,	$esk_{\mathcal{B}} \xleftarrow{\$} \{0,1\}^{u(k)}, t_{\mathcal{B}} \xleftarrow{\$} \{0,1\}^{t_2(k)}$,
$\widehat{lsk}_{\mathcal{A}} = \text{Ext}_1(lsk_{\mathcal{A}}, r_{\mathcal{A}_1})$,	$\widehat{lsk}_{\mathcal{B}} = \text{Ext}_1(lsk_{\mathcal{B}}, r_{\mathcal{B}_1})$,
$\widehat{esk}_{\mathcal{A}} = \text{Ext}_2(esk_{\mathcal{A}}, r_{\mathcal{A}_2})$,	$\widehat{esk}_{\mathcal{B}} = \text{Ext}_2(esk_{\mathcal{B}}, r_{\mathcal{B}_2})$,
$(\omega_{\mathcal{A}}, x) = \hat{F}_{\widehat{lsk}_{\mathcal{A}}}(esk_{\mathcal{A}} + \bar{F}_{\widehat{\text{elsk}}_{\mathcal{A}}}(r_{\mathcal{A}_1})$,	$(\omega_{\mathcal{B}}, y) = \hat{F}_{\widehat{lsk}_{\mathcal{B}}}(esk_{\mathcal{B}} + \bar{F}_{\widehat{\text{elsk}}_{\mathcal{B}}}(r_{\mathcal{B}_1})$,
$W_{\mathcal{A}} = \text{WordG}(\text{param}, \mathcal{L}, \omega_{\mathcal{A}}), X = g^x$,	$W_{\mathcal{B}} = \text{WordG}(\text{param}, \mathcal{L}, \omega_{\mathcal{B}}), Y = g^y$,
except state all Erase$(esk_{\mathcal{A}}, W_{\mathcal{A}}, X, t_{\mathcal{A}})$.	except state all Erase$(esk_{\mathcal{B}}, W_{\mathcal{B}}, Y, t_{\mathcal{B}})$.
$\xrightarrow{(\hat{B}, \hat{A}, W_{\mathcal{A}}, X, t_{\mathcal{A}})}$	
$\xleftarrow{(\hat{A}, \hat{B}, W_{\mathcal{B}}, Y, t_{\mathcal{B}})}$	
خروجی کلید نشست	
Set sid $= (\hat{A}, \hat{B}, W_{\mathcal{A}}, X, t_{\mathcal{A}}, W_{\mathcal{B}}, Y, t_{\mathcal{B}})$	Set sid $= (\hat{A}, \hat{B}, W_{\mathcal{A}}, X, t_{\mathcal{A}}, W_{\mathcal{B}}, Y, t_{\mathcal{B}})$
$aux = H_1(\text{sid}, k_{\mathcal{A}_1} = Y^x$,	$aux = H_1(\text{sid}, k_{\mathcal{A}_1} = X^y$,
$K_{\mathcal{A}_2} = \text{ProjHash}(\text{param}, \mathcal{L}, lpk_{\mathcal{B}}, W_{\mathcal{A}}, \omega_{\mathcal{A}}, aux)$,	$K_{\mathcal{B}_2} = \text{ProjHash}(\text{param}, \mathcal{L}, lpk_{\mathcal{B}}, W_{\mathcal{A}}, \omega_{\mathcal{A}}, aux)$
$k_{\mathcal{A}_3} = \text{Hash}(\text{param}, \mathcal{L}, lsk_{\mathcal{A}}, W_{mathcalB}, aux)$,	$k_{\mathcal{B}_3} = \text{ProjHash}(\text{param}, \mathcal{L}, lpk_{\mathcal{A}}, W_{mathcalB}, aux)$,
$s_{\mathcal{A}} = \text{Ext}_3(H_2(K_{\mathcal{A}_1}) \oplus k_{\mathcal{A}_2} \oplus k_{\mathcal{A}_3}, t_{\mathcal{A}} \oplus t_{\mathcal{B}})$,	$s_{\mathcal{B}} = \text{Ext}_3(H_2(K_{\mathcal{B}_1}) \oplus k_{\mathcal{B}_2} \oplus k_{\mathcal{B}_3}, t_{\mathcal{A}} \oplus t_{\mathcal{B}})$,
$SK_{\mathcal{A}} = \tilde{F}_{s_{\mathcal{A}}}(\text{sid})$.	$SK_{\mathcal{B}} = \tilde{F}_{s_{\mathcal{B}}}(\text{sid})$.

جدول ۷. خواص امنیتی مدل‌های امنیتی بر اساس تنظیم نشت

مدل امنیتی	LR-CK	LR-eCK	CAFL-eCK	BAFL-eCK	GCL-eCK	CLR-eCK
مدل پایه	مدل CK	مدل eCK	مدل eCK	مدل eCK	مدل eCK	مدل eCK
مدل‌های نشت	نشت نسبی	نشت نسبی	نشت نسبی	نشت نسبی	نشت نسبی	نشت نسبی
مدل حالت ـ تقسیم	خیر	خیر	بلی	بلی	بلی	خیر
مرحله ارسال پرسمان	قبل مرحله چالش	قبل مرحله چالش	قبل، حین و بعد از مرحله چالش	قبل، حین و بعد از مرحله چالش	قبل، حین و بعد از مرحله چالش	قبل، حین و بعد از مرحله چالش
میزان نشت	محدود	محدود	نامحدود	محدود	نامحدود	محدود
نمونه پروتکل تبادل کلید	Enc-DH	پروتکل MO	پروتکل P_2	پروتکل π	LR-AKA	CLR-eCK AKA
مقاومت در برابر حمله کانال جانبی	حمله کانال جانبی نهان، حملات حافظه	تحلیل توان	نشت مداوم در تحلیل توان، تابش EM	حمله راه انداز سرد، حملات بدافزاری	نشت مداوم اطلاعات پارامترهای درگیر در محاسبات	حمله شامل نشت کلیدهای مخفی موقت و بلندمدت

جدول ۸. مقایسه توانمندی پرسمان مهاجم در مدل‌های امنیتی

پرسمان توانایی مهاجم	LR-CK	LR-eCK	CAFL-eCK	BAFL-eCK	GCL-eCK	CLR-eCK
مهاجم مجاز به فعال‌سازی و کنترل روابط بخش‌های پروتکل است	مهاجم مجاز به کنترل برنامه‌ریزی پروتکل، آغاز پروتکل و ارسال پیام‌های پروتکل است	ارسال پرسمان: مهاجم طرف پروتکل را با پیام‌های پروتکل فعال می‌کند، کنترل فعال‌سازی نشست‌ها	ارسال پرسمان تعبیه شده با تابع نشت	ارسال پرسمان تعبیه شده با تابع نشت	پرسمان ارسال: پیام‌های پروتکل را دریافت کرده و نتایج مربوطه را برای اجرای پروتکل بر اساس پیام ارسال می‌کند	پرسمان ارسال: ارسال پیام پروتکل برای آغازگر نشست از طرف شریک نشست و دریافت پاسخ
مهاجم مجاز به افشای کلیدهای نشست است	پرسمان کلید نشست: مهاجم مجاز به دریافت کلید نشست تولید شده توسط نشست است	پرسمان افشای کلید نشست: پرسمان کلید نشست نشست خاتمه یافته	پرسمان افشای کلید نشست: کلید نشست نشست خاتمه یافته تحویل مهاجم می‌شود	پرسمان افشای کلید نشست: کلید نشست نشست خاتمه یافته تحویل مهاجم می‌شود	پرسمان افشا: مهاجم می‌تواند کلید نشست فعال را به دست آورد	پرسمان افشای کلید نشست: پرسمان کلید نشست نشست خاتمه یافته
مهاجم می‌تواند کلیدهای مخفی بلندمدت بخش های پروتکل با آسیب زدن به آن‌ها به دست آورد	مهاجم نمی‌تواند کلید مخفی بلندمدت را قبل اتمام نشست به دست آورد	پرسمان افشای کلید ایستا: مهاجم کلید خصوصی ایستای پروتکل اصلی را به دست می‌آورد	پرسمان آسیب: کلید مخفی بلندمدت شرکت‌کنندگان پروتکل تحویل مهاجم می‌شود	پرسمان آسیب: کلید مخفی بلندمدت شرکت‌کنندگان پروتکل تحویل مهاجم می‌شود	پرسمان آسیب: مهاجم می‌تواند کلید خصوصی پروتکل اصلی را به دست آورد	افشای کلید بلندمدت: مهاجم می‌تواند کلید مخفی بلندمدت پروتکل اصلی را به دست آورد
به دست آوردن وضعیت‌های نشست یا کلیدهای مخفی موقت نشست هدف یا نشست شریک آن	پرسمان افشای وضعیت نشست: مهاجم کل وضعیت داخلی نشست را به دست می‌آورد	افشای کلید موقت: پرسمان کلید خصوصی موقت نشست	افشای کلید موقت: کلید مخفی موقت نشست تحویل مهاجم می‌شود	افشای کلید موقت: کلید مخفی موقت نشست تحویل مهاجم می‌شود	نشت مخفی موقت: مهاجم این پرسمان را برای به دست آوردن کلید مخفی موقت نشست صادر می‌کند	افشای کلید موقت: پرسمان کلید مخفی موقت نشست
ایجاد کلید عمومی برای شرکت کننده پروتکل	-	پرسمان ایجاد طرف پروتکل: مهاجم می‌تواند یک کلید عمومی بلندمدت برای پروتکل اصلی ثبت کند	-	-	-	پرسمان ایجاد طرف پروتکل: مهاجم می‌تواند یک کلید عمومی بلندمدت برای پروتکل اصلی ثبت کند

کلید را بروز کند و به یک تنظیم کلی بدون محدودیت برسد. این روش کارآیی بهتری را در مقایسه با روش زمان‌بر استخراج ترکیب داخلی ارائه می‌دهد. مدل‌های AFL-eCK(.)، GCL-eCK و CLR-eCK به مفهومی دست می‌یابند که توسط چن به نام نشت وابسته به چالش پیشنهاد شده است که در آن مهاجم می‌تواند درخواست‌هایی را در طول و بعد از فعال شدن نشست چالش ارسال کند [۹]. هدف تمام مدل‌های امنیتی پیشنهادی در جدول ۷ توجه به حملات احتمالی، از جمله حملات کانال جانبی مانند تشعشعات EM [1]، حمله حافظه نهان، تحلیل توان، حملات راه‌انداز سرد و حملات حافظه است. محققان در مقاله‌های بالا تنظیمات نشت را برای جلوگیری از نشت اطلاعات ناشی از حملات کانال جانبی مورد مطالعه و مدل‌سازی قرار داده‌اند. جدول ۸ قابلیت‌های مهاجم را از نظر ارسال درخواست برای مدل های امنیتی نشت‌تاب مقایسه می‌کند. دشمن به‌عنوان یک مهاجم زمانی چندجمله‌ای احتمالاتی (PPT) مدل شده است که می‌تواند هر پیام ارتباطی بین شرکت کنندگان پروتکل را کنترل کند و این قدرت را دارد که فعال‌سازی‌های بخشی را برنامه‌ریزی کند تا مقصدهای تحویل پیام‌ها را در طول یک نشست کنترل کند. در مدل AFL-eCK(.)، پرسمان فعال‌سازی با یک تابع نشت تعریف شده است تا اطلاعات نشت را به دست آورد. این قابلیت می‌تواند مربوط به مهاجمی باشد که کنترل کامل ارتباطی را مانند حمله مرد در میانه در دست دارد. پرسمان‌های ارسال شده در جدول ۸ باید از قاعده‌ای پیروی کنند که در آن مهاجم فقط محدود به ارسال مجموعه‌ای از پرسمان‌های احتمالی

[1] EM radiation

متوالی برای حفظ تازگی نشست است. این محدودیت برای هر دو مدل پایه CK و eCK اعمال می‌شود و مدل‌های امنیتی پیشنهادی در جدول ۸ این مدل‌های پایه را در حالی که با قاعده تازگی نشست مطابقت دارند، گسترش می‌دهند. مهاجم همچنین می‌تواند کلیدهای نشست را برای اجرای پروتکل در مدل‌های امنیتی پیشنهادی فاش کند. این قابلیت به حملات میان‌گذاری[1] در دنیای واقعی مربوط است که در آن نشت کلیدهای مخفی از یک نشست می‌تواند به دریافت کلید دیگری در نشست های دیگر کمک کند. مهاجم می‌تواند کلیدهای نشست را به دلخواه برنامه‌ریزی کند. قابلیت یادگیری کلیدهای مخفی بلندمدت شرکت کنندگان پروتکل با آسیب زدن به شرکت کنندگان پروتکل در مدل های امنیتی در جدول ۸ ذکر شده است. پرسمان خاص مورد استفاده در بین مدل‌های امنیتی پیشنهادی متفاوت است. در مدل LR-CK، مهاجم مجاز به ارسال پرسمان و یادگیری کلید مخفی بلندمدت قبل از اتمام نشست نیست چرا که این محدودیت بر اساس مدل پایه (CK) است. جدول ۸ همچنین پرسمان توانایی مهاجم[2] را برای وضعیت‌های نشست یادگیری یا کلیدهای مخفی موقت نشست هدف یا نشست شریک آن در اجرای پروتکل ارائه می‌دهد. در مدل LR-CK، مهاجم می‌تواند تمام اطلاعات وضعیت داخلی (به استثنای مخفی بودن بلندمدت) نشست را دریافت کند که به آن پرسمان آشکارکننده وضعیت نشست گفته می‌شود. از سوی دیگر، در مدل‌های LR-eCK، AFL-eCK، GCL-eCK و CLR-eCK به جای این‌که به مهاجم اجازه دهند تا اطلاعات وضعیت داخلی پروتکل را مانند مدل CK به طور کامل فاش کند، یک پرسمان جدید به نام آشکارکننده کلید موقت، به طور خاص تنها کلید مخفی موقت را در یک نشست مشخص فاش می‌کند. در مدل‌های LR-eCK و CLR-eCK، هر دو مدل امنیتی به صراحت پرسمان «ایجاد بخش» را بیان می‌کنند که به مهاجم توانایی ایجاد یک کلید عمومی برای شرکت‌کننده پروتکل را می‌دهد.

۵ نتیجه‌گیری

در این مقاله، مدل‌های پایه Dolev-Yao، CK و eCK مورد بحث قرار گرفته‌است. هر مدل پایه قابلیت‌های متفاوتی را ارائه می‌کند که می‌تواند حداکثر تعداد حملات شناخته‌شده را برطرف کند. این مقاله به بررسی شش پروتکل می‌پردازد که اساس همه آن‌ها مدل‌های CK یا eCK هستند. مفاهیم امنیتی مختلفی برای هر پروتکل وجود دارد و مهاجمان تا حد امکان قوی مدل‌سازی می‌شوند تا بیشترین تعداد حملات نشتی را پوشش دهند. قوی‌ترین مدل امنیتی برای طرح نشتی وجود ندارد، بلکه تنها مناسب‌ترین مدل برای تنظیمات نشتی خاص مطرح است. اگر مهاجم بتواند داده‌های خارج از محدوده خود را افشا کند، مدل امنیتی ممکن است ناامن شود. هدف از این کار درک بهتر این است که کدام مدل امنیتی نشت برای یک تنظیم نشتی خاص مناسب‌تر است.

[1] Interleaving Attacks [2] Adversary capability query

مراجع

[1] Dolev, Danny and Yao, Andrew. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[2] Canetti, Ran and Krawczyk, Hugo. Analysis of key-exchange protocols and their use for building secure channels. in *International conference on the theory and applications of cryptographic techniques*, pp. 453–474. Springer, 2001.

[3] LaMacchia, Brian, Lauter, Kristin, and Mityagin, Anton. Stronger security of authenticated key exchange. in *International conference on provable security*, pp. 1–16. Springer, 2007.

[4] Dodis, Yevgeniy, Haralambiev, Kristiyan, López-Alt, Adriana, and Wichs, Daniel. Efficient public-key cryptography in the presence of key leakage. in *Advances in Cryptology-ASIACRYPT 2010: 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings 16*, pp. 613–631. Springer, 2010.

[5] Okamoto, Tatsuaki. Authenticated key exchange and key encapsulation in the standard model. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 474–484. Springer, 2007.

[6] Alawatugoda, Janaka, Stebila, Douglas, and Boyd, Colin. Modelling after-the-fact leakage for key exchange. in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pp. 207–216, 2014.

[7] Alawatugoda, Janaka, Stebila, Douglas, and Boyd, Colin. Continuous after-the-fact leakage-resilient eck-secure key exchange. in *IMA international conference on cryptography and coding*, pp. 277–294. Springer, 2015.

[8] Wu, Jui-Di, Tseng, Yuh-Min, and Huang, Sen-Shan. Efficient leakage-resilient authenticated key agreement protocol in the continual leakage eck model. *IEEE Access*, 6:17130–17142, 2018.

[9] Chen, Rongmao, Mu, Yi, Yang, Guomin, Susilo, Willy, and Guo, Fuchun. Strongly leakage-resilient authenticated key exchange. in *Cryptographers' Track at the RSA Conference*, pp. 19–36. Springer, 2016.

[10] Moriyama, Daisuke and Okamoto, Tatsuaki. An eck-secure authenticated key exchange protocol without ran-

dom oracles. *KSII Transactions on Internet & Information Systems*, 5(3), 2011.

[11] Kasslin, Kimmo. Kernel malware: The attack from within. *Computer Security Journal*, 23(1):43, 2007.

[12] Akavia, Adi, Goldwasser, Shafi, and Vaikuntanathan, Vinod. Simultaneous hardcore bits and cryptography against memory attacks. in Reingold, Omer, ed. , *Theory of Cryptography*, pp. 474–495, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[13] Alwen, Joël, Dodis, Yevgeniy, and Wichs, Daniel. Leakage-resilient public-key cryptography in the bounded-retrieval model. in *Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pp. 36–54. Springer, 2009.

[14] Yang, Zheng and Li, Shuangqing. On security analysis of an after-the-fact leakage resilient key exchange protocol. *Information Processing Letters*, 116(1):33–40, 2016.

[15] Diffie, Whitfield and Hellman, Martin E. New directions in cryptography. in *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pp. 365–390. 2022.

[16] Dziembowski, Stefan and Faust, Sebastian. Leakage-resilient cryptography from the inner-product extractor. in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 702–721. Springer, 2011.

[17] Alawatugoda, Janaka. On the leakage-resilient key exchange. *Journal of Mathematical Cryptology*, 11(4):215–269, 2017.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2024, Volume 21, Number 1 (pp. 49–65)
https://monadi.isc.org.ir

Review of the security models with emphasis on leakage-resilient key exchange protocols

Nasser Zarbi [1], Ali Zaeembashi [1,*] and Nasour Bagheri [2]

[1] *Department of Science, Shahid Rajaee Teacher Training University, Tehran, Iran*
[2] *Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: December 16, 2023
Accepted: February 20, 2024
Published Online: March 13, 2024

Keywords:
Key exchange protocol
Leakage-resilient
Security model
Adversary

Type: Review paper

ABSTRACT

Leakage-resilient cryptography aims to design key exchange protocols to withstand leakage attacks. These protocols are examined using a leakage-resilient security model to determine whether they possess the claimed security properties. The security analysis focuses on how the leakage-resilient security model has evolved to meet increasing security requirements and cover a broader range of attacks. By studying and analyzing the presented security properties of these models, potential vulnerabilities in protocol design can be effectively addressed. This article delves into various leakage-resilient security models based on two models, CK and eCK, and provides examples of secure key exchange protocols defined within these models. Additionally, it explores the relationship between adversaries' capabilities in these models and different attack schemes in the real world. By offering insights into various leakage-resilient security models, leakage attacks, and the development of secure protocols, it contributes to advancing knowledge in this field.

* Corresponding author
Email addresses: nasser_zarbi@yahoo.com (Nasser Zarbi), azaeembashi@sru.ac.ir (Ali Zaeembashi), Nbagheri@sru.ac.ir (Nasour Bagheri)