ششمین کنفرانس بین المللی انجمن رمز ایران
۱۵ و ۱۶ مهر ماه ۱۳۸۸، دانشگاه اصفهان
انجمن رمز ایران
دانشگاه اصفهان

معرفی و تحلیل یک پروتکل توافق کلید چندتایی جدید

محمد سبزی‌نژادفراش[۱]، محمود گردشی[۲]، مجید بیات[۳]

[۱و۳] تهران، پژوهشکده پردازش هوشمند علائم، گروه رمز و امنیت اطلاعات
{m.sabzinejad, m.bayat}@rcisp.com
[۲] تهران، دانشگاه امام حسین(ع)، مرکز تحقیقات فتح
mgardeshi@ihu.ac.ir

چکیده

پروتکل‌های توافق کلید چندتایی در هر بار اجرا، ساخت چند کلید مشترک را برای کاربران امکان پذیر می‌کنند. ایده اولیه این پروتکل‌ها بر اساس استفاده از امضای بدون تابع درهمساز برای تایید هویت هر کاربر استوار است. به همین دلیل (یعنی عدم استفاده از تابع درهمساز) بیشتر طرح‌های ارائه شده در این زمینه از امنیت لازم برخوردار نیستند. ما در این مقاله مروری جامع بر این پروتکل‌ها خواهیم داشت و بر روی برخی از آنها حملاتی ارائه می‌کنیم. سپس برای بهبود امنیت طرح‌های موجود، یک پروتکل پیشنهادی ارائه می‌دهیم و نشان خواهیم داد که این پروتکل از نظر امنیتی بهتر از همه طرح‌های مورد ارزیابی است.

واژه‌های کلیدی

رمزنگاری، پروتکل‌های توافق کلید، پروتکل‌های توافق کلید چندتایی، طرح‌های امضا

۱- مقدمه

برای برقراری ارتباط امن در شبکه‌های ناامن، استفاده از رمزنگاری امری اجتناب ناپذیر است. در کاربردهای رمزنگاری، کلید محرمانه نقش اساسی دارد بطوریکه به خطر افتادن این کلید، امنیت تمام سیستم رمزنگاری را به خطر می‌اندازد. بنابراین مدیریت و چگونگی تبادل کلید محرمانه در رمزنگاری از اهمیت زیادی برخوردار است.

یکی از ابزارهای مهم برای مدیریت و تبادل کلید محرمانه، پروتکل‌های ساخت کلید[۱] است. این پروتکل‌ها به کاربران امکان می‌دهند تا در یک شبکه ناامن بتوانند کلید محرمانه‌ای را به اشتراک گذاشته و از آن برای برقراری ارتباط امن در شبکه استفاده کنند. یک دسته مهم از این پروتکل‌ها، پروتکل‌های توافق کلید هستند. در این نوع پروتکل‌ها کاربران با استفاده از پیام‌هایی که برای یکدیگر ارسال می‌کنند یک کلید مشترک می‌سازند.

اولین پروتکل توافق کلید با استفاده از رمزنگاری کلید عمومی توسط دیفی[۲] و هلمن[۳] [5] ارائه شد. اما این پروتکل در برابر حمله مردی در میانه[۴] آسیب پذیر است زیرا طرفین پروتکل هویت همدیگر را وارسی[۵] نمی‌کنند. راه‌کار معمول برای حل این مشکل، استفاده از یک زوج کلید عمومی و خصوصی معتبر توسط هر کاربر است. رایج‌ترین روش‌های موجود برای این کار، زیرساخت کلید عمومی[۶] (رمزنگاری کلید عمومی مبتنی بر گواهینامه) و رمزنگاری کلید عمومی مبتنی بر شناسه[۷] است.

در سال ۱۹۹۵ منزس[۸] و همکارانش [11] یک پروتکل توافق کلید بنام MQV ارائه دادند که در آن برای امضای کلیدهای دیفی- هلمن از توابع درهمساز[۹] استفاده نمی‌شود. این پروتکل در استانداردهای بین بین المللی ANSI X9.42 [2]، ANSI X9.63 [3] و IEEE 1363 [10] استاندارد شده است. با استفاده از ایده MQV (یعنی امضای

[1] Key Establishment Protocols
[2] Diffie
[3] Hellman
[4] Man in the Middle Attack
[5] Verification
[6] Public Key Infrastructure (PKI)
[7] Identity-Based Cryptography
[8] Menezes
[9] Hash Functions

بدون تابع درهمساز)، هارن[1] و لین[2] در سال ۱۹۹۸ [6] ایده پروتکل-پروتکل‌های توافق کلید چندتایی[3] را مطرح کردند. در این پروتکل‌ها کاربران در هر نشست، چند کلید مشترک می‌سازند. این ایده از این جهت مورد توجه است که در هر بار اجرای پروتکل چند کلید مشترک ساخته می‌شود در حالیکه برای ساخت این تعداد کلید با استفاده از پروتکل‌های توافق کلید معمولی، باید چند بار پروتکل اجرا شود که این سبب بالا رفتن هزینه محاسباتی و ارتباطی می‌شود.

Yen و Joye [16] نشان دادند که پروتکل هارن- لین در برابر حمله جعل امضا آسیب پذیر است و برای حل مشکل آن یک طرح پیشنهادی ارائه دادند. اما Wu و همکارانش [17] نشان دادند که این طرح نیز همان مشکل طرح هارن- لین را دارد و خود یک پروتکل بهبود یافته ارائه دادند ولی برخلاف ایده اولیه هارن- لین، از تابع درهمساز در آن استفاده کردند، اما با این وجود نیز مشکل حمله جعل هنوز پا برجا بود.

هارن و لین [7] با توجه به حملات ارائه شده بر روی طرح اولیه خود، به منظور برطرف کردن مشکل، یک تغییر در امضای آن بوجود آوردند و مدعی شدند که در اینصورت پروتکل در برابر حمله جعل مقاوم است اما طولی نکشید که Zhou و همکارانش [18] نشان دادند که این امضای جدید نیز در برابر حمله جعل آسیب پذیر است.

ما در این مقاله مروری جامع بر پروتکل‌های توافق کلید چندتایی انجام می‌دهیم و ضمن بیان نقاط ضعف عموم آنها، یک طرح پیشنهادی بهبود یافته به همراه تحلیل غیر رسمی آن ارائه خواهیم داد.

۲- مفاهیم مقدماتی

در تحلیل امنیتی پروتکل‌های توافق کلید به روش غیر رسمی، امنیت پروتکل در برابر حملات موجود مورد ارزیابی قرار می‌گیرد. مهم‌ترین این ویژگی‌های امنیتی [4] عبارتند از:

امنیت کلید شناخته شده[4]: این ویژگی بیان می‌دارد که اگر مهاجم به یک کلید نشست دسترسی پیدا کرد توانایی بدست آوردن کلیدهای نشست بعدی را نداشته باشد.

امنیت پیشرو[5]: این ویژگی امنیتی بیان می‌دارد که درصورت آشکار شدن کلیدهای خصوصی طولانی‌مدت کاربران، امنیت کلیدهای نشست قبلی به خطر نیفتد. اگر با آشکار شدن کلید خصوصی طولانی مدت یکی از کاربران، محاسبه کلیدهای نشست قبلی ممکن نباشد اما با داشتن کلیدهای خصوصی طولانی‌مدت هر دو طرف چنین امکانی وجود داشته باشد گوییم پروتکل دارای امنیت پیشرو جزئی[6] است. در صورتی که با داشتن کلیدهای خصوصی طولانی‌مدت هر دو طرف نیز محاسبه کلیدهای نشست قبلی امکان پذیر نباشد گفته می‌شود پروتکل دارای امنیت پیشرو کامل[7] است.

امنیت در مقابل جعل هویت با کلید آشکار شده[8]: درصورت آشکار شدن کلید خصوصی طولانی مدت طرف A ، مهاجمی که این کلید را در اختیار دارد نتواند خود را بجای طرف B به A معرفی کند.

امنیت کلید ناشناخته[9]: فرض کنید A و B در حال اجرای اجرای پروتکل توافق کلید هستند. مهاجم فعال C نباید بتواند به نحوی در اجرای پروتکل دخالت کند تا بعد از اتمام پروتکل، A بر این باور باشد که با B توافق کلید انجام داده اما B معتقد است با C یک کلید محرمانه مشترک ساخته است.

حمله «مردی در میانه»: فرض کنید A و B در حال اجرای پروتکل توافق کلید هستند. در این حمله مهاجم فعال C به نحوی در اجرای پروتکل دخالت می‌کند تا سبب شود طرفین پروتکل روی کلیدهای متفاوتی توافق کنند.

علاوه بر ویژگی‌هایی که ذکرشد دو ویژگی اساسی دیگر نیز برای پروتکل‌های توافق کلید وجود دارد که عبارتند از:

تأیید کلید ضمنی[10]: یک پروتکل توافق کلید دارای ویژگی تأیید کلید ضمنی است اگر طرفین پروتکل مطمئن باشند که فقط طرف مقابل آنها توانایی محاسبه کلید نشست را دارد.

تأیید کلید تضمینی[11]: یک پروتکل توافق کلید دارای ویژگی تأیید کلید تضمینی است اگر طرفین پروتکل مطمئن باشند که طرف مقابل آنها کلید نشست را محاسبه کرده‌است.

پروتکل‌های توافق کلید از جنبه کارآمدی نیز مورد ارزیابی قرار می‌گیرند. کارآمدی یک پروتکل از جهت محاسباتی و ارتباطی مورد بررسی قرار می‌گیرد. کارآمدی محاسباتی پروتکل به حجم محاسبات مورد نیاز برای اجرای پروتکل توسط هر کاربر بستگی دارد. میزان پیام‌های تبادل شده بین طرفین پروتکل در هر بار اجرا نیز کارآمدی ارتباطی پروتکل را تعیین می‌کند.

بنابراین هدف اصلی طراحان پروتکل‌های توافق کلید، طراحی یک پروتکل توافق کلید امن (دارای ویژگی‌های امنیتی بالا باشد) و کارآمد (دارای حداقل هزینه محاسباتی و ارتباطی) است. یک ویژگی مهم پروتکل‌های توافق کلید چندتایی این است که از جهت

[1] Harn
[2] Lin
[3] Multiple Key Agreement Protocols
[4] Known-Key Security
[5] Forward Secrecy
[6] Partial Forward Secrecy
[7] Perfect Forward Secrecy
[8] Key-Compromise Impersonation
[9] Unknown Key Security
[10] Implicit Key Authentication
[11] Explicit Key Authentication

کارآمدی بهتر از پروتکل‌های توافق کلید معمولی هستند زیرا با انجام یک نشست بجای یک کلید چند کلید تولید می‌کنند.

۳- مروری بر پروتکل‌های توافق کلید چندتایی

در این بخش قصد داریم مرور مختصری بر روی پروتکل‌های توافق کلید چندتایی مبتنی بر گواهینامه داشته باشیم. در این پروتکل‌ها، طرفین طی دو بار گذر اطلاعات، همدیگر را تایید کرده و سپس چند کلید مشترک می‌سازند. از آنجایی که چالش امنیتی این طرح‌ها ناشی از طرح امضای مورد استفاده است، لذا ما فقط به بررسی امضاهای آنها می‌پردازیم. این امضاها به همراه وارسی در جدول(۳-۲) نشان داده شده‌اند. نمادگذاری مورد استفاده در این مقاله در جدول(۳-۱) آمده است.

جدول(۳-۲) دارای چهار ستون است که از سمت راست، در ستون اول نام طرح، ستون دوم کلیدهای عمومی کوتاه مدت طرف A، ستون سوم رابطه امضا و ستون چهارم وارسی امضا قرار دارد. در پروتکل‌های مورد بررسی به ازای n مقدار تصادفی، تعداد n^2 یا n^2-1 کلید نشست ساخته می‌شود. در جدول(۳-۲) پروتکل-های توافق کلید چندتایی را به ازای دو مقدار تصادفی t_{A1},t_{A2} مورد بررسی قرار داده‌ایم که به ساخت چهار کلید مشترک (در برخی از طرح‌ها سه کلید مشترک) منجر می‌شوند.

در جدول (۳-۳) ضعف پروتکل‌های مطرح شده در جدول (۳-۲) مورد بررسی قرار گرفته است. برخی از این ضعف‌ها برگرفته از مراجعی است که در جدول به آنها ارجاع داده شده است. ما در ادامه برخی ضعف‌ها که توسط مولفین این مقاله مطرح شده است را مورد بررسی قرار می‌دهیم.

جدول۳-۱: نمادها

g	مولد زیرگروه ضربی از مرتبه q
x_A, x_B	کلیدهای خصوصی طولانی مدت A و B
y_A, y_B	کلیدهای عمومی طولانی مدت A و B
r_A, r_B	کلیدهای خصوصی کوتاه‌مدت (مقادیر تصادفی) A و B
t_A, t_B	کلیدهای عمومی کوتاه مدت (مقادیر تصادفی) A و B
s_A, s_B	امضاهای کاربران A و B
$K_{AB} = g^{x_A x_B}$	کلید محرمانه طولانی مدت مشترک A و B
K	کلید نشست

جدول۳-۲: الگوریتم‌های امضا و وارسی پروتکل‌های توافق کلید چندتایی*

طرح	کلیدهای موقت	امضا	وارسی امضا
HL98 [6]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - g^{t_{A1}t_{A2}}(r_{A1}+r_{A2})$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{g^{t_{A1}} \cdot t_{A2}}$
YJ [16]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - (t_{A1} \cdot t_{A2})(r_{A1}+r_{A2})$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{t_{A1} \cdot t_{A2}}$
WHH [17]**	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - H(t_{A1} \cdot t_{A2})(r_{A1}+r_{A2})$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{h(t_{A1} \cdot t_{A2})}$
HL01 [7]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - t_{A1}r_{A1} - t_{A2}r_{A2}$	$y_A \overset{?}{=} g^{s_A} \cdot t_{A1}^{t_{A1}} \cdot t_{A2}^{t_{A2}}$
ZFL [18]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - (t_{A1}+t_{A2})(r_{A1}+r_{A2})$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1}+t_{A2})}$
YSH [15]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A = x_A - (t_{A1} \oplus t_{A2})(r_{A1}+r_{A2})$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1} \oplus t_{A2})}$
Tseng [14]	$t_A = g^{r_{A1}+r_{A2}}$ $t_{A1/2} = y_B^{r_{A1/2}}$	$s_A \cdot t_A = x_A - t_{A1}(r_{A1}+r_{A2})$	$t_A = t_{A1}^{x_B^{-1}} \cdot t_{A2}^{x_B^{-1}}$, $y_A \overset{?}{=} t_A^{t_{A1}} \cdot g^{s_A t_A}$
Shao [12]	$t_A = g^{r_{A1}+r_{A2}}$ $t_{A1/2} = y_B^{r_{A1/2}}$	$s_A = x_A \cdot t_A - (t_{A1}+t_{A2})(r_{A1}+r_{A2})$	$t_A = t_{A1}^{x_B^{-1}} \cdot t_{A2}^{x_B^{-1}}$, $y_A^{t_A} \overset{?}{=} t_A^{(t_{A1}+t_{A2})} \cdot g^{s_A}$
HC [8]	$t_{A1/2} = g^{r_{A1/2}}$	$s_A \oplus K_{AB} = (t_{A1}-t_{A2})x_A - (t_{A1} \oplus t_{A2})(r_{A1}+r_{A2})$	$y_A^{(t_{A1}-t_{A2})} = g^{s_A \oplus K_{AB}} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1} \oplus t_{A2})}$
HCH [9]	$t_{A1/2} = g^{r_{A1/2}}$	$h_A = (y_B)^{r_{A1}} \cdot t_{A2}$ $s_A = x_A - h_A(r_{A1}+r_{A2})$	$h_A = (t_{A1})^{x_B} \cdot t_{A2}$ $y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{h_A}$

* چهارچوب اجرای تمام طرح‌های موجود در این جدول همانند پروتکل (۴-۱) است با این تفاوت که رابطه امضا و وارسی تغییر می‌کند.

** تابع $H()$ یک تابع درهمساز است. در این طرح برخلاف ایده اولیه هارن - لین (یعنی ایده "عدم استفاده از تابع درهمساز") رفتار شده است.

جدول۳-۳: ضعف پروتکل‌های توافق کلید چندتایی

طرح	تعداد کلید نشست ساخته شده*	ضعف
HL98 [6]	n^2-1	جعل امضا [16]
YJ [16]	n^2-1	جعل امضا [17]
WHH [17]	n^2-1	جعل امضا [15]
HL01 [7]	n^2-1	جعل امضا [18]
ZFL [18]	n^2-1	جعل امضا [15]
YSH [15]	n^2-1	حمله کلید ناشناخته [15]
Tseng [14]	n^2	جعل امضا [12] و حمله جعل هویت با کلید آشکار شده [۱]
Shao [12]	n^2	حمله کلید ناشناخته [13]
HC [8]	n^2	حمله جعل هویت با کلید آشکار شده [۱]
HCH [9]	n^2	با آشکار شدن کلید خصوصی طرفین و یک کلید نشست، سه کلید دیگر نیز آشکار می‌شوند. [۱]

* علت اینکه در برخی از طرح‌های موجود در جدول تعداد کلیدهای مشترک ساخته شده n^2-1 کلید است، ایجاد امنیت در برابر حمله کلید شناخته شده است. این مطلب در [7] مورد بررسی قرار گرفته است.

۳-۱ حمله جعل هویت با کلید آشکار شده بر روی پروتکل Tseng

طرح Tseng [14] در جدول (۳-۲) نشان داده شده است. در حمله جعل هویت با کلید آشکار شده، مهاجم E با در اختیار داشتن کلید خصوصی A قصد دارد خود را بجای B به A معرفی کند. برای اعمال این حمله بر روی پروتکل Tseng ، مهاجم مقادیر (t_{B1},t_{B2},s_B) را بصورت زیر محاسبه می‌کند:

$$t_{B2}=\left(y_B^{t_{B1}^{-1}}\right)^{x_A} \quad , \quad t_{B1}=y_A$$

$$s_B\cdot t_B=-t_{B1} \quad , \quad t_B=g\cdot y_B^{t_{B1}^{-1}}$$

مهاجم پس از محاسبه مقادیر (t_{B1},t_{B2},s_B) آنها را برای A ارسال می‌کند. A نیز به محض دریافت این مقادیر تایید امضا را بصورت زیر انجام می‌دهد:

$$t_{b1}=t_{B1}^{x_A^{-1}}=y_A^{x_A^{-1}}=g$$

$$t_{b2}=t_{B2}^{x_A^{-1}}=\left(\left(y_B^{t_{B1}^{-1}}\right)^{x_A}\right)^{x_A^{-1}}=y_B^{t_{B1}^{-1}}$$

$$t_B=t_{b1}\cdot t_{b2}=g\cdot y_B^{t_{B1}^{-1}}$$

$$y_B=t_B^{t_{B1}}\cdot g^{s_Bt_B}=\left(g\cdot y_B^{t_{B1}^{-1}}\right)^{t_{B1}}\cdot g^{s_Bt_B}$$
$$=g^{t_{B1}}\cdot y_B\cdot g^{-t_{B1}}=y_B$$

A در پایان کلیدهای نشست را بصورت زیر محاسبه می‌کند:

$$K_1=t_{b1}{}^{r_{A1}}=g^{r_{A1}} \quad , \quad K_3=t_{b2}{}^{r_{A1}}=\left(y_B^{t_{B1}^{-1}}\right)^{r_{A1}}=t_{A1}^{t_{B1}^{-1}}$$

$$K_2=t_{b1}{}^{r_{A2}}=g^{r_{A2}} \quad , \quad K_4=t_{b2}{}^{r_{A2}}=\left(y_B^{t_{B1}^{-1}}\right)^{r_{A2}}=t_{A2}^{t_{B1}^{-1}}$$

همانطور که در بالا مشاهده می‌شود مهاجم بدلیل در اختیار داشتن مقادیر (t_{B1},t_{A1},t_{A2}) می‌تواند کلیدهای نشست K_3 و K_4 را محاسبه کند. بنابراین پروتکل Tseng در برابر حمله جعل هویت با کلید آشکار شده ناامن است.

۳-۲ حمله جعل هویت با کلید آشکار شده بر روی پروتکل HC

طرح HC [8] در جدول (۳-۲) نشان داده شده است. در حمله جعل هویت با کلید آشکار شده، مهاجم با در اختیار داشتن کلید خصوصی کاربر A قصد دارد خود را بجای کاربر B به A معرفی کند. برای اعمال این حمله بر روی پروتکلHC ، مهاجم می‌تواند مقادیر (t_{B1},t_{B2}) را مساوی انتخاب کند. در اینصورت $t_{A1}-t_{A2}=0$ و $t_{A1}\oplus t_{A2}=0$ خواهد شد. پس رابطه‌ی امضا

همانطور که در جدول(۳-۲) نشان داده شده است بصورت زیر خواهد بود

$$s_B \oplus K_{AB} = (t_{B1} - t_{B2})x_B - (t_{B1} \oplus t_{B2})(r_{B1} + r_{B2}) = (0)x_B - (0)(r_{B1} + r_{B2}) = 0$$

بنابراین مهاجم مقادیر مساوی (t_{B1}, t_{B2}) را می‌تواند با $s_B = K_{AB}$ امضا کند و از آنجایی که کلید خصوصی طولانی مدت کاربر A را در اختیار دارد محاسبه $s_B = K_{AB} = y_B^{x_A}$ برای وی مشکل نخواهد بود. پس نتیجه می‌گیریم که اگر مهاجم کلید خصوصی کوتاه مدت کاربر A را در اختیار داشته باشد با انتخاب $t_{B1} = t_{B2}$ و $s_B = K_{AB}$ براحتی می‌تواند خود را بجای B به A معرفی کند.

ضعف دیگر امضای طرح HC این است که اگر مهاجم کلید خصوصی طولانی‌مدت دو طرف پروتکل را داشته باشد براحتی می تواند مقادیر زیر را محاسبه کند:

$$(r_{A1} + r_{A2}) = \left((t_{A1} - t_{A2})x_A - s_A \oplus K_{AB}\right)(t_{A1} \oplus t_{A2})^{-1}$$
$$(r_{B1} + r_{B2}) = \left((t_{B1} - t_{B2})x_B - s_B \oplus K_{AB}\right)(t_{B1} \oplus t_{B2})^{-1}$$

و سپس با استفاده از آن مقادیر زیر را بدست بیاورد:

$$t_{B1}^{(r_{A1}+r_{A2})} = g^{r_{A1}r_{B1}} \cdot g^{r_{A2}r_{B1}} = K_1 \cdot K_3$$
$$t_{B2}^{(r_{A1}+r_{A2})} = g^{r_{A1}r_{B2}} \cdot g^{r_{A2}r_{B2}} = K_2 \cdot K_4$$
$$t_{A1}^{(r_{B1}+r_{B2})} = g^{r_{A1}r_{B1}} \cdot g^{r_{A1}r_{B2}} = K_1 \cdot K_2$$
$$t_{A2}^{(r_{B1}+r_{B2})} = g^{r_{A2}r_{B1}} \cdot g^{r_{A2}r_{B2}} = K_3 \cdot K_4$$

در این صورت مهاجم اگر یک کلید از چهار کلید نشست را بدست آورد به راحتی می‌تواند سه تای دیگر را نیز بدست آورد.

۳-۳ امن‌ترین پروتکل توافق کلید چندتایی

پروتکل HCH برای ساخت تعداد n^2 کلید نشست مورد استفاده قرار می‌گیرد. همانطور که در جدول(۳-۳) مشاهده می‌شود این پروتکل از نظر امنیتی بهترین پروتکلی است که ما مورد بررسی قرار داده‌ایم. تنها ضعف جزئی این طرح این است که با آشکار شدن کلیدهای خصوصی طولانی مدت طرفین پروتکل و یک کلید نشست، سه کلید دیگر آشکار می‌شوند (برای توضیح بیشتر به بخش ۵-۲-۴ از [۱] مراجعه شود).

۴- معرفی یک پروتکل توافق کلید چندتایی جدید

همانطور که در جدول (۳-۳) مشاهده می‌شود هیچکدام از پروتکل‌های توافق کلید چندتایی موجود خالی از اشکال نیست. بنابراین طراحی یک پروتکل توافق کلید چندتایی امن بدون استفاده از تابع درهمساز، به دقت و تحقیق بیشتری نیاز دارد. ما در این بخش قصد داریم به چنین هدفی دست پیدا کنیم.

۴-۱ توصیف پروتکل پیشنهادی

طرح پیشنهادی در پروتکل(۴-۱) نشان داده شده است. توصیف این پروتکل به شرح زیر است:

- A به عنوان آغازکننده اعداد تصادفی $r_{A1}, r_{A2} \in_R Z_n^*$ را انتخاب می‌کند. سپس مقادیر $t_{A1} = g^{r_{A1}}$، $t_{A2} = g^{r_{A2}}$ و $t_A = t_{A1}^{r_{A1}}$ را محاسبه و با استفاده از رابطه زیر

$$s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)$$

A (x_A, Y_A)	B (x_B, Y_B)
$r_{A1}, r_{A2} \in_R Z_q^*$	$r_{B1}, r_{B2} \in_R Z_q^*$
$t_{A1} = g^{r_{A1}}$, $t_{A2} = g^{r_{A2}}$	$t_{B1} = g^{r_{B1}}$, $t_{B2} = g^{r_{B2}}$
$t_A = t_{A1}^{r_{A1}}$	$t_B = t_{B1}^{r_{B1}}$
$s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)$	$s_B = x_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}^2 + r_{B2}\right)$
$(t_{A1}, t_{A2}, t_A, s_A) \rightarrow$	
$\leftarrow (t_{B1}, t_{B2}, t_B, s_B)$	
$y_B \overset{?}{=} g^{s_B} \cdot \left(t_B \cdot t_{B2}\right)^{\left(t_{B1}^{x_B} \cdot t_{B2}\right)}$	$y_A \overset{?}{=} g^{s_A} \cdot \left(t_A \cdot t_{A2}\right)^{\left(t_{A1}^{x_B} \cdot t_{A2}\right)}$
$K_{ij} = t_{Bi}^{r_{Aj}} = g^{r_{Aj}r_{Bi}}$	$K_{ij} = t_{Aj}^{r_{Bi}} = g^{r_{Aj}r_{Bi}}$
$i \,.\, i \in \{1.2\}$	$i, j \in \{1,2\}$

پروتکل(۴-۱): پروتکل توافق کلید چندتایی پیشنهادی

امضا می‌کند. A در پایان این مرحله مقادیر $(t_{A1}, t_{A2}, t_A, s_A)$ را برای B ارسال می‌کند.

- B نیز اعداد تصادفی $r_{B1}, r_{B2} \in_R Z_n^*$ را انتخاب می‌کند. سپس مقادیر $t_{B1} = g^{r_{B1}}$، $t_{B2} = g^{r_{B2}}$ و $t_B = t_{B1}^{r_{B1}}$ را محاسبه و با استفاده از رابطه زیر

$$s_B = x_B - \left(y_B^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}^2 + r_{B2}\right)$$

امضا می‌کند. B در پایان این مرحله مقادیر $(t_{B1}, t_{B2}, t_B, s_B)$ را برای A ارسال می‌کند.

- A به محض دریافت مقادیر $(t_{B1}, t_{B2}, t_B, s_B)$ ابتدا صحت امضا را بصورت زیر

$$y_B \overset{?}{=} g^{s_B} \cdot \left(t_B \cdot t_{B2}\right)^{\left(t_{B1}^{x_A} \cdot t_{B2}\right)}$$

بررسی می‌کند. در صورتی که این تساوی برقرار نباشد A اجرای پروتکل را رها می‌کند و درغیر این صورت کلیدهای نشست را بصورت زیر محاسبه می‌کند:

$$K_{ij} = t_{Bi}^{\,r_{Aj}} = g^{r_{Aj} r_{Bi}} \quad : i, j \in \{1,2\}$$

- B نیز به محض دریافت مقادیر (t_{A1}, t_{A2}, s_A) ابتدا صحت امضا را بصورت زیر:

$$y_A \overset{?}{=} g^{s_A} \cdot \left(t_A \cdot t_{A2}\right)^{\left(t_{A1}^{x_B} \cdot t_{A2}\right)}$$

بررسی می‌کند، در صورتی که این تساوی برقرار نباشد B اجرای پروتکل را رها می‌کند و درغیر این صورت کلیدهای نشست را بصورت زیر محاسبه می‌کند:

$$K_{ij} = t_{Aj}^{\,r_{Bi}} = g^{r_{Aj} r_{Bi}} \quad : i, j \in \{1,2\}$$

۲-۴ تحلیل امنیتی پروتکل پیشنهادی

امنیت کلید شناخته شده: در حمله کلید شناخته شده مهاجم با در اختیار داشتن کلیدهای نشست قبل قصد دارد امنیت نشست‌های بعدی را به خطر اندازد. در پروتکل پیشنهادی، مهاجم با در اختیار داشتن کلیدهای نشست

$$K_{ij} = t_{Bi}^{\,r_{Aj}} = g^{r_{Aj} r_{Bi}}$$

هیچ اطلاعات مفیدی برای تهدید امنیت نشست‌های بعدی بدست نخواهد آورد، زیرا برای ساخت کلید نشست از پارامترهای موقت r_{A1} و r_{A2} استفاده شده است که در هر نشست تغییر می‌کنند. بنابراین پروتکل پیشنهادی امنیت کلید شناخته شده دارد.

امنیت کلید ناشناخته: در این حمله، مهاجم فعال C قصد دارد بگونه‌ای در روند اجرای پروتکل بین A و B دخالت کند که در پایان، B تصور داشته باشد با C نشست انجام داده درحالیکه A بر این باور است که با B توافق کلید انجام داده است. مهاجم C درصورت اعمال چنین حمله‌ای بر روی پروتکل پیشنهادی، پیام ارسالی $(t_{A1}, t_{A2}, t_A, s_A)$ از طرف A را سد می‌کند. برای اینکه بتواند مقادیر (t_{A1}, t_{A2}) را با استفاده از کلید خصوصی خود بصورت زیر امضا کند

$$s_C = x_C - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)$$

باید مقادیر تصادفی (r_{A1}, r_{A2}) را در اختیار داشته باشد در حالیکه آنها را ندارد زیرا برای بدست آوردن آنها با حل مسئله لگاریتم گسسته مواجه خواهد شد. بنابراین مهاجم C نمی‌تواند حمله کلید ناشناخته را بر روی پروتکل پیشنهادی اعمال کند.

حمله جعل هویت با کلید آشکار شده: در این حمله مهاجم فعال C با در اختیار داشتن کلید خصوصی طولانی‌مدت A تلاش می‌کند خود را بجای B به A معرفی کند. در پروتکل پیشنهادی، اگر مهاجم کلید خصوصی طولانی‌مدت x_A را در اختیار داشته باشد برای اینکه بتواند خود را بجای B به A معرفی کند باید قادر باشد امضای

$$s_B = x_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1} t_{B1} + r_{B2} t_{B2}\right)$$

را به ازای (t_{B1}, t_{B2}) دلخواه محاسبه کند، اما بدلیل نداشتن کلید خصوصی x_B قادر به محاسبه این مقدار نخواهد بود. پس پروتکل پیشنهادی در برابر حمله جعل هویت با کلید آشکار شده امن است.

امنیت پیشرو کامل: این ویژگی امنیتی بیان می‌دارد که درصورت آشکار شدن کلیدهای خصوصی طولانی‌مدت کاربران امنیت کلیدهای نشست قبلی به خطر نیفتد. در پروتکل پیشنهادی، اگر مهاجم کلیدهای خصوصی طولانی مدت x_A و x_B را در اختیار داشته باشد نمی‌تواند کلید نشست را محاسبه کند. زیرا برای محاسبه کلید نشست

$$K_{ij} = t_{Bi}^{\,r_{Aj}} = g^{r_{Aj} r_{Bi}}$$

پارامترهای موقت r_{Ai} یا r_{Bj} به ازای $i, j \in \{1,2\}$ را باید در اختیار داشته باشد. این پارامترها را با استفاده از مقادیر t_{Ai} یا t_{Bj} نمی‌تواند بدست آورد زیرا با حل مسئله لگاریتم گسسته مواجه خواهد شد. حال ببینیم آیا مهاجم با استفاده از s_A یا s_B می‌تواند پارامترهای موقت r_{Ai} یا r_{Bj} را بدست آورد. فرض می‌کنیم مهاجم s_A را در اختیار داشته باشد. در اینصورت با رابطه زیر مواجه خواهد بود

$$s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right) \qquad (۱-۴)$$

پارامترهای معلوم برای مهاجم عبارتند از $(t_{A1}, t_{A2}, t_A, x_A, x_B, s_A)$. او با استفاده از این معلومات رابطه (۱-۴) را به رابطه زیر تبدیل می‌کند

$$\left(t_{A1}^{x_B} \cdot t_{A2}\right)^{-1}\left(x_A - s_A\right) = \left(r_{A1}^2 + r_{A2}\right) \qquad (۲-۴)$$

اکنون سمت چپ تساوی (۲-۴) برای مهاجم معلوم است. بنابراین به تعداد q (مرتبه گروه ضربی که پروتکل روی آن اجرا می‌شود) زوج معتبر (r_{A1}, r_{A2}) در تساوی (۲-۴) صدق می‌کنند. بنابراین برای یافتن یکی از پارامترهای r_{A1} یا r_{A2} باید تمام اعضای گروه ضربی بررسی شوند. پس مهاجم با در اختیار داشتن کلیدهای خصوصی طرفین پروتکل نمی‌تواند کلیدهای نشست قبل را بدست آورد. بنابراین پروتکل پیشنهادی دارای امنیت پیشرو کامل است.

۳-۴ نگاهی دقیق‌تر به پروتکل پیشنهادی

اگر x_A و x_B را در هم ضرب کنیم ، خواهیم داشت:

$$
\begin{aligned}
x_A x_B &= \left(s_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)\right) \\
&\quad \cdot \left(s_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}^2 + r_{B2}\right)\right) \\
&= s_A s_B - \left(s_A r_{B1}^2 + s_A r_{B2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right) \\
&\quad - \left(r_{A1}^2 s_B + r_{A2} s_B\right)\left(y_B^{r_{A1}} \cdot t_{A2}\right) \\
&\quad + \begin{pmatrix} r_{A1}^2 r_{B1}^2 + r_{A1}^2 t_{B2} r_{B2} \\ + t_{A2} r_{A2} r_{B1}^2 + t_{A2} r_{A2} t_{B2} r_{B2} \end{pmatrix}\left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right)
\end{aligned}
$$

اگر دو طرف رابطه بالا را به توان g برسانیم خواهیم داشت:

$$
\begin{aligned}
g^{x_A x_B} &= K_{AB} = g^{\left(s_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)\right)\left(s_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}^2 + r_{B2}\right)\right)} \\
&= g^{s_A s_B} \cdot g^{-\left(s_A r_{B1}^2 + s_A r_{B2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right)} \cdot g^{-\left(r_{A1}^2 s_B + r_{A2} s_B\right)\left(y_B^{r_{A1}} \cdot t_{A2}\right)} \\
&\quad \cdot \left(g^{r_{A1}^2 r_{B1}^2} \cdot g^{r_{A1}^2 r_{B2}} \cdot g^{r_{A2} r_{B1}^2} \cdot g^{r_{A2} r_{B2}}\right)^{\left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right)} \\
&= g^{s_A s_B} \cdot \left(t_{B1}^{r_{B1}} \cdot t_{B2}\right)^{-s_A \cdot \left(t_{B1}^{x_A} \cdot t_{B2}\right)} \cdot \left(t_{A1}^{r_{A1}} \cdot t_{A2}\right)^{-s_B \left(t_{A1}^{x_B} \cdot t_{A2}\right)} \\
&\quad \cdot \left(K_1^{r_{A1} r_{B1}} \cdot K_2^{r_{A1}} \cdot K_3^{r_{B1}} \cdot K_4\right)^{\left(t_{B1}^{x_A} \cdot t_{B2}\right)\left(t_{A1}^{x_B} \cdot t_{A2}\right)}
\end{aligned}
$$

همانطور که در رابطه بالا مشاهده می‌شود طرفین تساوی به کلید خصوصی کاربران بستگی دارد. بنابراین در صورتی که مهاجم چهار کلید یک نشست را در اختیار داشته باشد با استفاده از رابطه بالا نمی‌تواند مقدار K_{AB} را محاسبه کند مگر اینکه کلید خصوصی طولانی‌مدت یکی از طرفین را در اختیار داشته باشد.

حال اگر فرض کنیم مهاجم کلید خصوصی دو طرف پروتکل را داشته باشد، در اینصورت براحتی می‌تواند طرف چپ تساوی بالا را محاسبه کند. در طرف راست این تساوی، کلیدهای $K_1,....,K_4$ برای مهاجم مجهول هستند. اما از آنجایی که به توان کلیدهای خصوصی کوتاه مدت r_{Ai} یا r_{Bj} رسیده‌اند، بنابراین اگر مهاجم سه تا از کلیدهای $K_1,....,K_4$ را داشته باشد، نمی‌تواند کلید چهارم را بدست آورد. پس پروتکل ما ضعف پروتکل HCH را ندارد به همین دلیل از همه پروتکل‌های تبادل کلید چندتایی امن‌تر است.

پروتکل پیشنهادی در مقایسه با طرح HCH، از لحاظ هزینه محاسباتی به هر طرف پروتکل یک توان رسانی اضافه می‌کند و از لحاظ هزینه ارتباطی هر طرف پروتکل باید یک پارامتر اضافی ارسال کند.

۵- نتیجه گیری

در این مقاله به بررسی اجمالی پروتکل‌های توافق کلید چندتایی پرداخته شد و حملاتی را به برخی طرح‌های موجود ارائه دادیم. در بررسی این پروتکل‌ها مشاهده شد که هنوز یک پروتکل چندتایی بدون استفاده از توابع درهمساز کاملا امن ارائه نشده است. به همین دلیل ما یک طرح پیشنهادی ارائه دادیم که تمام ضعف‌های طرح‌های قبل را برطرف می‌کند. این طرح از لحاظ هزینه محاسباتی و هزینه ارتباطی نیز سربار قابل توجهی نسبت به طرح‌های قبلی ندارد.

مراجع

[۱] سبزی‌نژادفراش، محمد، بررسی پروتکل‌های توافق کلید مبتنی بر خم‌بیضوی و زوج‌سازی‌های دوخطی و مقایسه‌ی آنها، پایان نامه کارشناسی ارشد، دانشگاه امام حسین(ع)، تهران، بهمن ۱۳۸۷.

[2] ANSI X9.42, Agreement of Symmetric Algorithm Keys Using Diffie–Hellman, Working Draft, May 1998.

[3] ANSI X 9.63, Elliptic Curve Key Agreement and Key Transport Protocols, Working Draft, July 1998.

[4] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In Proc. of Sixth IMA International Conference on Cryptography and Coding, pages 30 – 45. Cirencester, UK, 1997.

[5] W. Diffiee, M. Hellman. New Directions in Cryptography. In IEEE Transaction on Information Theory, IT-22 (6), pp. 644-654, 1976.

[6] L. Harn, H.-Y. Lin, An authenticated key agreement protocol without using one-way function. In: Proceedings of eighth information security conference, Taiwan, May 1998; p. 155–60.

[7] L. Harn, H.-Y. Lin, Authenticated key agreement without using one-way hash function. Electron Lett 2001;37(10):629–30.

[8] H. Huang and C. Chang. Enhancement of an Authenticated Multiple-Key Agreement Protocol Without Using Conventional One-Way Function. In CIS 2005, Part II, LNAI 3802, pp. 554 – 559, 2005. Springer-Verlag (2005).

[9] C.-J. Huang, S.-H. Chang and W.-H. Hsu, Authenticated Key Agreement Protocol for Exchanging n2 Keys without Using One-way Hash Function, In NCS 全國計算機會議 DSpace at FCUniversity, available in: (dspace.lib.fcu.edu.tw/bitstream/2377/3190/3/ce07ncs001999000185.pdf), (2006).

[10] IEEE P1363, Standards Specifications for Public-Key Cryptosystems, Working Draft, July 1998.

[11] A.J. Menezes, M. Qu, and S.A. Vanstone. Some key agreement protocols providing implicit authentication. In: Proceeding of the second workshop on selected areas in cryptography (SAC'95), 1995; pp. 22–32.

[12] Z. Shao, Security of Robust Generalized MQV Key Agreement Protocol Without Using One-way Hash Functions, Computer Standards and Interfaces, Vol. 25, (2003) 431–436.

[13] K.-A. Shim, Vulnerabilities of generalized MQV key agreement protocol without using one-way

hash functions, In: Computer Standards & Interfaces, 29, (2007), 467–470.

[14] Y.-M. Tseng, Robust Generalized MQV Key Agreement Protocol without Using One-way Hash Functions. Computer Standards and Interfaces, Vol. 24, (2002) 241–246

[15] H.-T. Yeh, H.-M. Sun, T. Hwang, Improved authenticated multiple-key agreement protocol, in: Proceedings of the 11th National Conference on Information Security, TaiNan, Taiwan,May 2001, pp.229–231 .

[16] S.-M. Yen, M. Joye, Improved authenticated multiple-key agreement protocol, Electronics Letters 1998;34 (18):1738–1739

[17] T.-S. Wu, W.-H. He, C.-L. Hsu, Security of authenticated multiple-key, Electronics Letters 35 (5) (1999) 391–392.

[18] H.-S. Zhou, L. Fan and J.-H. Li, Remarks on unknown key-share attack on authenticated multiple-key agreement protocol, In Electronics Letters 2003;39 (17):1248–1249.