# وارسی پروتکل‌های رمزنگاری بدون فرض وجود صریح نفوذی

بهروز ترک لادانی سعید جلیلی

torkla_b@net1cs.modares.ac.ir sjalili@modares.ac.ir

دانشگاه تربیت مدرس

دانشکدهٔ فنی مهندسی، گروه کامپیوتر

**چکیده**

پروتکل‌های رمزنگاری از دو دید قابل تحلیل هستند، اول اینکه اهداف امنیتی متصوّر برای پروتکل از طریق اجرای آن تحقّق یابد و دیگر اینکه با فرض وجود عوامل مخلّ امنیت در روند اجرای آن، به این اهداف ضربه ای وارد نشود. اخیراً چارچوبی برای بیان روش‌های تحلیل به صورتی که هر دو دید فوق را پوشش دهد ارائه شده است. در این مقاله نشان داده شده است که چگونه می توان بر اساس این چارچوب و مدل محاسباتی ارائه شده در آن، یک روش تحلیل برای وارسی تحقّق ویژگیهای امنیتی در پروتکل‌ها بدون تصریح روی نقش نفوذی ایجاد نمود. برای این منظور، یک روش نمونه برای تحلیل پروتکل‌های توافق کلید ارائه شده است. این روش مشتمل بر یک مدل محاسباتی از پروتکل، زبانی برای بیان ویژگی‌های عوامل، ومجموعه ای از قواعد استنتاج راجع به پروتکل است. پروتکل‌های توافق کلید دیفی‌هلمن، الجمال، و *MTI/A0* نیز با این روش تحلیل شده‌اند.

**کلمات کلیدی:** تحلیل پروتکل‌های رمزنگاری، وارسی صوری پروتکل‌های رمزنگاری، تحلیل منطقی پروتکل‌ها، پروتکل‌های توافق کلید.

## ۱) مقدمه

بکارگیری پروتکل‌های رمزنگاری روش اصلی برای ایجاد ویژگی‌های امنیتی در شبکه‌های ارتباطی و سیستم‌های توزیع شده است. دو نگرانی عمده در طراحی این پروتکل‌ها وجود دارد. اول این که پروتکل طراحی شده در نهایت اهداف امنیتی مورد نظر (مثل تصدیق اصالت عوامل و پیام‌ها، توزیع یا توافق روی کلیدها و...) را به درستی برآورده کند و دیگری اینکه با فرض وجود نفوذی به همراه قابلیت‌های ارتباطی و محاسباتی در بستر اجرای پروتکل، تمهیدات امنیتی در نظرگرفته شده، امنیت پروتکل را تضمین کند و راهی را برای انجام حملات مختلف به پروتکل باز نگذارد. بر این اساس روش‌های صوری ارئه شده برای وارسی پروتکل‌های رمزنگاری را می‌توان به دو دستهٔ روش‌های تحلیل منطقی پروتکل و روش‌های ساخت سناریوهای حمله برعلیه پروتکل تقسیم نمود*[2]*. در رویکرد اول، هدف این است که با مدل کردن منطق حاکم بر اجزاء پروتکل (توابع رمزنگاری و تکنیک‌های طراحی پروتکل ) روشی برای اثبات ایجاد ویژگی‌های امنیتی در پروتکل فراهم گردد. منطق‌های *BAN [3]* و *GNY [4]* نمونه هایی از این روش ها هستند. در رویکرد دوم، پروتکل به همراه یک عامل نفوذی در نظر گرفته می‌شود و سعی می‌شود با مدل کردن چنین سیستمی، راهی برای کشف سناریوهای نقض ویژگی‌های امنیتی فراهم شود. *[5]* و *[6]* ، نمونه هایی از این روش ها هستند. در *[7]* روش های مختلف در این رویکرد، مرور و با یکدیگر مقایسه شده‌اند.

در *[1]* چارچوبی برای وارسی پروتکل‌های رمزنگاری ارائه شد. این چارچوب را می‌توان به عنوان یک دستهٔ کلی از روش‌های تحلیل در نظر گرفت که به صورت $Anal(\mathcal{L},\mathcal{M},\mathcal{PIR},\mathcal{EIR})$ نشان داده می‌شود و در آن $\mathcal{L}$ زبانی است که از طریق آن ویژگیهای عوامل مجاز و نفوذی در پروتکل بیان می شود، $\mathcal{M}$ یک مدل محاسباتی از پروتکل ونحوهٔ اجرای آن است، $\mathcal{PIR}$[1] و $\mathcal{EIR}$[2] نیز به ترتیت مجموعه اصول برای انجام استنتاج‌های عوامل مجاز و نفوذی است.توصیف پروتکل در مدل محاسباتی $\mathcal{M}$ و اصول لازم در مجموعه‌های $\mathcal{PIR}$ و $\mathcal{EIR}$ به زبان $\mathcal{L}$ بیان می شود. در*[1]* هدف این است که با تغییر مؤلفه‌های $\mathcal{L}$ ، $\mathcal{M}$ ، $\mathcal{PIR}$ و $\mathcal{EIR}$ ، بتوان روش‌های تحلیل مختلفی رادر این چارچوب بیان نمود.

---

[1] *Principal Inference Rules*

[2] *Eavesdropper Inference Rules*

ما در این مقاله با بکارگیری چارچوب کلی و مدل محاسباتی ارائه شده در *[1]* ، یک روش برای تحلیل پروتکل‌های رمزنگاری بدون فرض وجود صریح نفوذی ارائه نموده‌ایم. در این روش یک مجموعه دانش برای پروتکل تعریف می‌شود. این مجموعه دانش حاوی تمامی مسندات صحیح راجع به عوامل پروتکل است، که به زبان $\mathcal{L}$ بیان می‌شود. مقدار اولیه این مجموعه دانش، فرضیات اولیه پروتکل است و در طی قدم‌های متوالی در مسیر اجرای پروتکل، با کمک مجموعه قوانین استنتاج عوامل $(\mathcal{PIR})$، این مجموعه به صورت یکنوا[1] گسترش می‌یابد. تحلیل پروتکل از طریق تعریف مسندات هدف درمورد ویژگی‌های پروتکل و سعی در اثبات ایجاد آنها در مجموعه دانش پروتکل انجام می‌شود. مجموعه $\mathcal{EIR}$ نیز در این روش تهی در نظر گرفته می‌شود. تحلیل پروتکل در این وضعیت معادل رویکرد تحلیل منطقی پروتکل است و با کمک آن می‌توان به همان توانایی‌های روش تحلیل منطقی دست یافت. در ادامه مقاله، ابتدا مدل محاسباتی مطرح شده در *[1]* به اختصار بیان می‌شود. سپس چگونگی تحلیل پروتکل بدون فرض وجود صریح نفوذی با کمک این مدل محاسباتی بیان می‌گردد. در ادامه با ارائهٔ یک زبان توصیف $\mathcal{L}$ و یک مجموعهٔ $\mathcal{PIR}$ مناسب برای تحلیل پروتکل‌های توافق کلید مبتنی بر لگاریتم گسسته، سه پروتکل توافق کلید دیفی هلمن[2]، الجمال[3] و *MTI/A0 [ 6]* را به عنوان نمونه مورد تحلیل قرار می‌دهیم.

## ۲) مدل محاسباتی پروتکل

در این قسمت به تشریح مختصر مدل محاسباتی مطرح شده در*[1]* می پردازیم. در این مدل محاسباتی، پروتکل به عنوان مجموعه‌ای از نقشها درنظر گرفته می شود. هر نقش مشتمل بر چندین قدم پی‌درپی است و هر قدم رشته‌ای از توانایی‌ها است. توانایی نشان دهندهٔ ریزترین عملیات پایه[4]‌ای است که بایستی توسط یک نقش انجام شود. یک توانایی دارای دو مؤلفه است : مجموعه پیش‌شرطهای آن توانایی و مجموعه پس‌شرطهای آن توانایی. در صورت فراهم بودن پیش شرطهای یک توانایی، آن توانایی قابل اجراست و با اجرای آن توانایی، پس‌شرطهای آن نیز فراهم می‌شوند. برای هر پروتکل یک مسیر اجرا تعریف می‌شود که این مسیر اجرا حاصل تلفیق نحوی قدمهای نقشهای پروتکل است.

هر عامل در پروتکل برای انجام صحیح پروتکل نیاز به انجام یک سری عملیات دارد. ریزترین عملیات پایه‌ای که در یک پروتکل توسط یک عامل انجام می‌شود را یک توانایی می‌نامیم. هر توانایی برای قابل انجام بودن مستلزم وجود بعضی شرایط در پروتکل است ودر صورت انجام، ممکن است شرایط را تغییر دهد.

**تعریف توانایی :** دوتایی $<pre\ ,\ post>$ که در آن *pre* و *post* هر کدام مجموعه‌ای از مسندات با پارامترهای متغیر[5] در زبان $\mathcal{L}$ هستند، تشکیل یک توانایی $\alpha$ را می‌دهند. توانایی $\alpha$ را به صورت ظاهری $L_\alpha : \frac{pre}{post}$ نشان می‌دهیم که $L_\alpha$ برچسب نام این توانایی است.

*Pre* و *post* به ترتیب مجموعه پیش‌شرطها و پس‌شرطهای توانایی هستند و اگر $\alpha$ یک توانایی باشد، از $pre(\alpha)$ و $post(\alpha)$ به ترتیب برای نشان دادن این دو مجموعه استفاده می‌کنیم. در ساخت مجموعه‌های *pre* و *post* بایستی از مسندات قابل تعریف در زبان $\mathcal{L}$ استفاده کرد. وجود دو مسند خاص زیر در زبان $\mathcal{L}$ الزامی‌است و به کمک آنها دو توانایی اصلی ارسال و دریافت پیام در پروتکل تعریف می‌شود:

- مسندات وضعیت به صورت $S_1(...),S_2(...),...$
- مسند پیام به صورت $\lhd(...)$

*Send* و *Receive* دو توانایی پایه هستند که دلالت بر ارسال یا دریافت یک پیام از/ به شبکه دارند. پیش‌شرط توانایی *Receive* و پس‌شرط توانایی *Send* بایستی حتماً مشتمل بر مسندات یکتای وضعیت و مسند پیام باشند.

**تعریف قدم :** رشته‌ای از توانایی‌ها به همراه اندیس $\omega$ ، نشان‌دهندهٔ یک قدم است. قدم $S$ را به صورت $(\alpha_1.\alpha_2...\alpha_n):\omega_S$ نشان می‌دهیم که در آن $\alpha_1.\alpha_2...\alpha_n$ رشتهٔ توانایی‌های قدم $S$ و $\omega_S$ دنیایی است که قدم $S$ در آن انجام می‌شود.

اگر $S$ یک قدم باشد، از $\alpha_i(S)$ برای رجوع به توانایی $i$ ام قدم $S$ استفاده می‌کنیم و $|S|$ طول قدم $S$ برحسب تعداد توانایی‌های آن را نشان می‌دهد. پیش‌شرط اولین توانایی و پس‌شرط آخرین توانایی قدم $S$ به ترتیب نشان‌دهندهٔ ورودی و خروجی قدم $S$ است. ورودی قدم $S$ را با $in(S)$ و خروجی این قدم را با $out(S)$ نشان می‌دهیم:

---

[1] *Monotonic*

[2] *Diffie-Hellman*

[3] *ELGamal*

[4] *Primitive Operation*

[5] *Non ground*

$in(S) = pre(\alpha_1(S))$ , $out(S) = post(\alpha_{|S|}(S))$

مجموعهٔ پیش‌شرطها و مجموعه پس شرطهای قدم $S$ را به ترتیب با $prem(S)$ و $Conseq(S)$ نشان می‌دهیم :

$$prem(S) = \bigcup_{i=1}^{|S|} pre(\alpha_i(S)) \quad , \quad Conseq(S) = \bigcup_{i=1}^{|S|} post(\alpha_i(S))$$

توانایی‌های اول و آخر قدم $S$ را به ترتیب با $first(S)$ و $last(S)$ نشان می‌دهیم.

قدم $S_0$ را که مجموعه پیش‌شرطهای آن تهی است، قدم مقداردهی اولیه می‌نامیم: $prem(S_0) = \phi$

**تعریف قابلیت دسترسی دنیای قدم‌ها به یکدیگر :** اگر $S_i$ و $S_j$ دو قدم باشند و $\omega_{S_i}$ و $\omega_{S_j}$ به ترتیب دنیاهای این دو قدم باشند، $\omega_{S_j}$ از $\omega_{S_i}$ قابل دسترس است اگر تابع جایگذاری[1] مثل $\theta$ وجود داشته باشد به صورتی که $out(S_i)\theta = in(S_j)\theta$ ، یعنی $out(S_i)$ و $in(S_j)$ قابل یکسان‌سازی[2] باشند. در این حالت $\omega_{S_j}$ دنیای بعدی $\omega_{S_i}$ است، یعنی $\omega_{S_j} = succ(\omega_{S_i})$ .

**تعریف نقش :** رشتهٔ محدودی از یک سری قدمها یک نقش را تشکیل می‌دهند.

اگر $\rho$ یک نقش به صورت $S_1 : \omega_{S_1}.S_2 : \omega_{S_2} \ldots S_m : \omega_{S_m}$ باشد، $S_i(\rho)$ نشان دهندهٔ قدم $i$ ام نقش $\rho$ است.

**تعریف پروتکل :** پروتکل $P$ یک چهارتایی به صورت $< \Lambda, \rho^*, \delta_0, S_0 >$ است که در آن $\Lambda$ مجموعه‌ای از نقشهاست، $\rho^* \in \Lambda$ نقش آغازگر پروتکل است. $\delta_0$ تابع جایگذاری اولیه است که بایستی روی نقشهای $\Lambda$ اعمال شود و $S_0$ یک قدم مقداردهی اولیه است.

**تعریف مسیر اجرای پروتکل :** اگر $P =< \Lambda, \rho^*, \delta_0, S_0 >$ یک پروتکل باشد، $R(P)$ مسیر اجرای پروتکل $P$ ، رشته محدودی از قدمهای درهم شده از نقش‌های مختلف $P$ است که به صورت زیر ساخته می‌شود:

- قدم اول رشته، خاص شدهٔ قدم مقداردهی اولیهٔ پروتکل براساس $\delta_0$ است $(S_0\delta_0)$.
- قدم دوم رشته، اولین قدم نقش آغازگر پروتکل ( $\rho^*$ ) است که تابع جایگذاری $\delta_0$ نیز روی آن اعمال شده است $(S_1(\rho^*)\delta_0)$.
- قدمهای سوم تا آخر این مسیر اجرا، سایر قدمهای نقشهای $\Lambda$ هستند، به صورتی که دنیای هر قدم از دنیای قدم بلافاصله قبل از آن قابل دسترس است و تمام توابع جایگذاری اعمال شده روی عناصر زیر رشتهٔ سمت چپ آنها نیز روی آنها اعمال شده است.

$$\forall i \geq 3, \exists \delta_i \ . \ out(S_i\delta_0\delta_1 \ldots \delta_{i-1})\delta_i = in(S_{i+1}\delta_0\delta_1 \ldots \delta_{i-1})\delta_i$$

- هیچ دو قدم پی‌درپی در مسیر اجرا متعلق به یک نقش نیستند.
- رعایت تقدم و تأخر قدمهای هر نقش، در رشتهٔ اجرا نیز به قوت خود باقی است.

مسیر اجرایی $R(P)$ را به صورت دقیق‌تر می‌توان چنین توصیف نمود :

$$R(P) = (t_0\delta_0).(t_1\delta_0\delta_1).(t_2\delta_0\delta_1\delta_2). \ \ldots \ .(t_n\delta_0\delta_1 \ldots \delta_n)$$

$$t_0 = S_0$$

$$t_1 = S_1(\rho^*)$$

$$\forall i \geq 1, \exists \rho', \rho'' \in \Lambda, \exists k', k'', \delta_i \bullet S_{k'}(\rho') = t_{i-1}, S_{k''}(\rho'') = t_i,$$
$$out(t_{i-1}\delta_0\delta_1 \ldots \delta_{i-1})\delta_i = in(t_i\delta_0\delta_1 \ldots \delta_{i-1})\delta_i,$$
$$\rho' \neq \rho''$$

$$\forall i, j \exists \rho' \in \Lambda, \exists k', k'' \bullet S_{k'}(\rho') = t_i, S_{k''}(\rho') = t_j,$$
$$k' < k'' \Rightarrow i < j$$

طول یک مسیر اجرا را بنا به تعریف برابر تعداد قدمهای آن در نظر می‌گیریم و با $|R(P)|$ نشان می‌دهیم.

---

[1] *Substitution function*

[2] *Unifiable*

### ۳) تحلیل پروتکل با فرض نفوذی غیرصریح

در مدل محاسباتی مطرح شده در بخش قبل، مسیر اجرای یک پروتکل را تعریف نمودیم. این مسیر اجرا رشته‌ای از قدم‌های مختلف از نقش‌های پروتکل است که از طریق یک رابطهٔ نحوی با یکدیگر مرتبط شده‌اند. می‌توانیم نقش نفوذی را در اجرای پروتکل نادیده بگیریم که در این صورت مجموعه قوانین استنتاج نفوذی ($\mathcal{EIR}$) نیز تهی خواهد بود. هدف این است که تعبیری از اجرای پروتکل روی مسیر اجرای آن بدست آوریم. برای این منظور، برای پروتکل یک مجموعه دانش تعریف می‌کنیم. این مجموعه دانش حاوی تمامی مسندات صحیح راجع به عوامل پروتکل است. همگی این مسندات متعلق به زبان $\mathcal{L}$ هستند. مقدار اولیه این مجموعه دانش همان فرضیات اولیهٔ پروتکل است و در طی مسیر اجرا با به فعلیت رسیدن هر توانایی، باکمک مجموعه قوانین استنتاج عوامل ($\mathcal{PIR}$)، این مجموعه دانش به صورت یکنوا گسترش می‌یابد. در ادامه به تشریح دقیق‌تر این تعبیر از اجرای پروتکل می‌پردازیم.

قدم‌های متوالی یک مسیر اجرا را می توان با رشتهٔ توانایی‌های خاص شدهٔ آنها جایگزین نمود و مسیر اجرا را به صورت رشته ای از این توانایی‌ها در نظر گرفت. اگر $R(P)$، مسیر اجرای پروتکل $P$، رشته ای از قدم‌ها به صورت زیر باشد:

$$R(P)=(S_0\delta_0).(S_1\delta_0\ \delta_1 \quad (S_m\delta_0\delta_1 \quad \delta_m)$$

اگر $r_i = (S_i\ \delta_0\delta_1\ \ldots\ \delta_i)$ در نظر بگیریم، $R(P)$ را می توان به صورت زیر نشان داد :

$$R(P) = r_0 \,.\, r_1 \quad {}_m$$

اگر هر $r_i$ را نیز به صورت رشته ای از توانایی‌های خاص شدهٔ $\beta_{i1} \,.\, \beta_{i2} \,.\, \ldots \,.\, \beta_{iq}$ در نظر بگیریم، که در آن $\beta_{ij}=\alpha_{ij}\ \delta_0\delta_1\ \ldots\ \delta_i$ و $\alpha_{ij}$ توانایی $i$ از مسیر اجرا است ، در این صورت مسیر اجرای $R(P)$ به صورت زیر خواهد بود:

$$R(P) =\beta_0.\,\beta_1 \quad \beta_n \quad , \quad n = \sum_{i=1}^{|R(P)|} |\, r_i \,|$$

**تعریف دانش پروتکل:** اگر $P$ یک پروتکل باشد و $R(P) =\beta_0.\,\beta_1.\ \ldots\ .\,\beta_n$ رشتهٔ توانایی‌های خاص شدهٔ این مسیر اجرا باشد، دانش پروتکل مجموعه ای از مسندات در زبان $\mathcal{L}$ است که به صورت زیر روی مسیر اجرای $R(P)$ گسترش می یابد:

- $K_0(P)$ مجموعه دانش اولیهٔ پروتکل $P$ برابر تهی است $(K_0(P) = N)$.
- اگر $Ki(P)$ دانش پروتکل بعد از انجام توانایی $\beta_i$ از مسیر اجرا باشد و $K_{i-1}(P) \;|=_{PIR}\; pre(\beta_i)$ (یعنی پیش شرط توانایی $\beta_i$ از روی دانش کسب شدهٔ پروتکل تابحال به پیمانهٔ قواعد $\mathcal{PIR}$ قابل حصول باشد ) در این صورت داریم :

$$K_i(P) = K_{i-1}(P) \cup post(\beta_i)$$

**تعریف اجرای یک توانایی:** اگر $\beta_i$ توانایی $i$ از مسیر اجرای پروتکل $P$ باشد، $e_i$ جملهٔ اجرایی توانایی $\beta_i$ است:

$$e_i : \quad IF \quad K_{i-1}(P) \;|=_{PIR}\; pre(\beta_i) \quad THEN$$
$$K_i(P) = K_{i-1}(P) \cup post(\beta_i)$$

که درآن $K_i(P)$، دانش حاصل از اجرای پروتکل بعد از اجرای توانایی $i$ از مسیر اجرا است.

توانایی $\beta_i$ قابل اجرا است ، اگر پیش شرط جملهٔ اجرایی آن درست باشد. جملهٔ اجرایی $e_i$ شرایط لازم برای به فعلیت رسیدن یک توانایی و همچنین نتایج حاصل از اجرای آن را بیان می کند.

با توجه به نیاز به تعریف جملهٔ اجرا یی در سطح یک قدم ، با اتکا به تعاریف بالا ، در زیر جملهٔ اجرایی قدم $r_i$ راتعریف می کنیم.

**تعریف اجرای یک قدم :** اگر $r_i$ قدم $i$ از یک مسیر اجرا برابر رشتهٔ $\beta_{i1} \,.\, \beta_{i2} \,.\, \ldots \,.\, \beta_{im}$ از توانایی‌های خاص شده باشد، این قدم قابل اجرا است اگر تمام توانایی‌های این رشته قابل اجرا باشند ودر این صورت، دانش حاصل از اجرای این قدم $K_{im}(P)$ است و با $K : T_i$ نشان می دهیم. $E_i$ جملهٔ اجرایی قدم $r_i$ به صورت زیر تعریف می شود:

$$E_i : \quad IF \quad executable(r_i) \quad THEN$$
$$K : \omega_i = K_{last(r_i)}(P)$$

که در آن : $executable(r_i) \equiv \bigwedge_{l=1}^{m} (K_{l-1}(P) \;|=_{PIR}\; pre(\beta_{li}))$

قدم $r_i$ از مسیر اجرای $R(P)$ قابل اجرا است اگر پیش شرط جملهٔ اجرایی آن درست باشد.

**تعریف وارسی اجراپذیری پروتکل** : پروتکل $R(P)$ قابل اجرا است اگر تمام قدم‌های آن قابل اجرا باشد.

**تعریف وارسی وجود ویژگیهای امنیتی :** فرض کنیم $G$ مجموعه ای از مسندات در زبان $\mathcal{L}$ باشد که توصیف کنندهٔ ویژگی خاصی در پروتکل است. پروتکل $P$ در قدم $i$ از اجرا ، ویژگی توصیف شده با $G$ را داراست اگر $G$ با کمک قواعد استنتاج عوامل $(\mathcal{PIR})$ از دانش پروتکل در آن قدم قابل حصول باشد : $K : \omega_i \mid=_{PIR} G$

### ٤) تحلیل چند پروتکل نمونه

در این بخش نشان می‌دهیم که چگونه می‌توان بر اساس روش تحلیل مطرح شده در بخش قبل به وارسی ویژگی های امنیتی در پروتکل‌ها پرداخت. برای این منظور چند پروتکل توافق کلید را در نظر گرفته‌ایم. در ادامه، ابتدا این پروتکل‌ها را معرفی می‌کنیم، سپس برای تکمیل اجزاء چارچوب تحلیل مطرح شده، ضمن ارائه یک زبان $\mathcal{L}$ برای بیان ویژگی‌های پروتکل، یک مجموعه اصول استنتاجی برای عوامل $(\mathcal{PIR})$ تعریف می‌کنیم و سپس به بیان نحوهٔ انجام تحلیل و نتایج حاصل می‌پردازیم.

### ٤-١) پروتکل‌های توافق کلید

پروتکل‌های توافق کلید آن دسته از پروتکل‌های توزیع کلید هستند که طرفین ارتباط برای دستیابی به یک کلید مشترک، یک سری اجزاء سازنده کلید را با یکدیگر تبادل می‌کنند و با انجام برخی محاسبات، هر دو طرف به کلیدهای مشترکی دست پیدا می‌کنند که برای عوامل دیگر قابل محاسبه نیست. پروتکل‌های دیفی‌هلمن، الجمال و *MTI/A0* نمونه‌هایی از این پروتکل‌ها هستند که اساس آنها بر سختی محاسبهٔ لگاریتم گسسته به پیمانه یک عدد اول بزرگ است.

- **پروتکل توافق کلید دیفی‌هلمن :** در پروتکل دیفی‌هلمن فرض بر این است که طرفین روی یک عدد اول بزرگ $n$ و عدد صحیح $g$ توافق کرده‌اند به صورتی که $g$ یک مولد میدان $n$ است $(g \in Z_P^*)$. این دو عدد لازم نیست مخفی باشند. بعد از این فرض اولیه، پروتکل به صورت زیر انجام می‌شود : عامل $A$ (آغازگر) یک عدد صحیح بزرگ مثل $x$ را به صورت تصادفی و مخفی انتخاب کرده، پیام $X=g^x \bmod n$ را برای $B$ (مخاطب) می‌فرستد. $B$ نیز در جواب، عدد صحیح بزرگ، تصادفی و مخفی $y$ را انتخاب کرده، پیام $Y=g^y \bmod n$ را برای $A$ پس می‌فرستد. بعد از این دو قدم، عامل $A$، مقدار $k' = Y^x \bmod n$ و عامل $B$ مقدار $k'' = X^y \bmod n$ را محاسبه می‌کنند. $k'$ و $k''$ هر دو مساوی $k = g^{xy} \bmod n$ هستند. هیچکس حتی باشنود مقادیر $X$ و $Y$ از روی خط و دارا بودن مقادیر $g$ و $n$ قادر به محاسبهٔ مقدار $k$ نیست مگر اینکه با محاسبه لگاریتم گسسته، مقادیر مخفی $x$ و $y$ را بازیابی کند که البته این کاردارای سختی محاسباتی است. به این ترتیب $A$ و $B$ هر دو به صورت مستقل یک مقدار مشترک مخفی را محاسبه کرده‌اند.
- **پروتکل توافق کلید الجمال :** در این پروتکل نیز طرفین مبادله ابتدا روی دو مقدار عمومی یعنی عدد اول بزرگ $n$ و عدد صحیح $g$ ( مولد میدان $n$ ) توافق می‌کنند. علاوه براین، عامل $B$ (مخاطب)، عدد صحیح و تصادفی $b$ را به عنوان کلید خصوصی خود در نظر گرفته، مقدار $g^b \bmod n$ را به عنوان کلید عمومی متناظر آن منتشر می‌کند. این پروتکل تنها شامل یک قدم است : عامل $A$(آغازگر) عدد تصادفی و مخفی $x$ را انتخاب کرده، $g^x \bmod n$ را برای $B$ می‌فرستد. عامل $B$ این پیام را دریافت کرده، بادارا بودن مقدار خصوصی $b$، مقدار $k' = (g^x)^b \bmod n$ را می‌سازد. عامل $A$ نیز با دارا بودن مقادیر $g^b \bmod n$ (کلید عمومی $B$) و $x$ (مقدار مخفی انتخاب شده توسط وی )، مقدار $k'' = (g^b)^x \bmod n$ را می‌سازد. مقادیر $k'$ و $k''$ مساوی و برابر $k=g^{bx} \bmod n$ است. به این ترتیب، طرفین به یک کلید مشترک دست یافته‌اند که کسی غیر از آنها قادر به تولید آن نیست.
- **پروتکل توافق کلید *MTI/A0* :** در این پروتکل نیز همانند دو پروتکل قبلی، عدد اول بزرگ $n$ و مولد میدان $g$ به صورت عمومی در اختیار طرفین قرار دارد. عامل $A$ مقدار مخفی $a$ را به عنوان کلید خصوصی انتخاب کرده، مقدار $Z_A=g^a \bmod n$ را به عنوان کلید عمومی خود منتشر می‌کند. عامل $B$ نیز به صورت متناظر دارای کلید خصوصی $b$ و کلید عمومی $Z_B=g^b \bmod n$ است. در قدم اول پروتکل، عامل $A$ مقدار مخفی $x$ را انتخاب کرده، $g^x \bmod n$ را برای $B$ ارسال می‌کند. عامل $B$ نیز مقدار مخفی $y$ را انتخاب کرده، $g^y \bmod n$ را برای $A$ پس می‌فرستد. طرفین $A$ و $B$ در نهایت به ترتیب مقادیر $k' = (g^y)^a z_B^x \bmod n$ و $k'' = (g^x)^b z_A^y \bmod n$ را ساخته، به عنوان کلید در نظر می‌گیرند. این دو مقدار مساوی و برابر $k=g^{ay+bx} \bmod n$ خواهد بود.

### ٤-٢) زبان توصیف $\mathcal{L}$

زبان توصیفی که برای استفاده در روش تحلیل پروتکل‌های توافق کلید در نظر می‌گیریم، یک زبان نوع دار مرتبه اول است. برخی نمادهای منطق *GNY* در این زبان اقتباس شده است. دو نوع پایه‌ای مورد استفاده، فرمول (*Formula*) و عامل (*Principal*) هستند. فرمول نامی است که برای

رجوع به یک رشتهٔ بیتی (که در اجرا مقدار خاصی خواهد یافت) در نظر گرفته می‌شود و عامل، نامی است که برای رجوع به طرفین دخیل در پروتکل بکار می‌رود. بابکارگیری این دو نوع، اجزاء زبان $\mathcal{L}$ به صورت زیر ساخته می‌شوند :

- **فرمول‌ها :** فرض کنیم $X$ و $Y$ از نوع فرمول باشند. دو نوع خاص ازفرمول‌ها یعنی $n$ (یک عدد اول بزرگ) و (یک مولد میدان$n$) رانیز در نظر می‌گیریم. مقادیر زیر نیز فرمول هستند :

  $\boldsymbol{X+Y}$: حاصل جمع دو فرمول $X$ و $Y$ ( با خواص جابجایی و شرکت پذیری ) .

  $\boldsymbol{X.Y}$: حاصل ضرب دو فرمول $X$ و $Y$ (با خواص جابجایی و شرکت پذیری ) .

  $\boldsymbol{g^x \bmod n}$: مقدار $g$ به توان فرمول $X$ و به پیمانه $n$ .

  $\boldsymbol{(X,Y)}$ : اتصال دو فرمول ( با خاصیت جابجایی و شرکت پذیری) .

  $X \sim\!\succ C$ : فرمول $X$ به همراه دانش ضمنی $c$ ($c$ یک مسند است) .

- **عبارات :** اگر $P$ و $Q$ دو عامل باشند و $X$ یک فرمول باشد، یک عبارت به یکی از دو شکل زیر خواهد بود :

  $P \xleftarrow{X} Q$ : فرمول $X$ یک مقدار مخفی مشترک بین عوامل $P$ و $Q$ است.

  $P \overset{\smile X}{\longmapsto} Q$ : فرمول $X$ یک مقدار مخفی برای عامل $Q$ است.

- **مسندات :** اگر $X$ یک فرمول، $P$ یک عامل و $S$ یک عبارت باشد، مسندات به یکی از اشکال زیر ساخته می‌شوند:

  $\lhd (X)$ : فرمول $X$ روی شبکه قرار گرفته است

  $P \ni X$ : عامل $P$ مالک فرمول $X$ است.

  $P \Delta X$ : عامل $P$ توانایی ساختن فرمول $X$ را دارد.

  $P \mid\!\equiv S$ : عامل $P$ ، عبارت $S$ را قبول دارد.

### ٤-٣) مجموعه قواعد استنتاج عوامل

در این بخش مجموعه‌ای از قواعد استنتاج برای استفاده در تحلیل پروتکل‌های توافق کلید مطرح شده در بخش ۳-۱ به عنوان $\mathcal{PIR}$ ارائه می‌شود. این مجموعه مشتمل بر چهار بخش قواعد پیام‌های شبکه، قواعد مالکیت، قواعد تعبیر پیام و قواعد ساخت پیام است. قواعد پیام‌های شبکه و قواعد مالکیت به ترتیب تشابه زیادی با قواعد *Being-told* و *Possessions* در منطق *GNY [4]* دارند. لازم به ذکر است که این مجموعه قواعد، یک مجموعهٔ الزاماً کامل نیست و صرفاً برای نشان دادن قابلیت بکارگیری روش تحلیل روی پروتکل‌های توافق کلید ارائه شده است.

- **قواعد پیام‌های شبکه**

در صورتی که پیامی متشکل از دو جزء $X$ و $Y$ روی شبکه قرار گیرد، هر کدام از اجزاء نیز روی شبکه قرار گرفته است.

$$N_1 : \frac{\lhd (X,Y)}{\lhd (X)}$$

قرار گرفتن پیامی به همراه یک دانش ضمنی روی شبکه قرار گرفتن پیام به تنهایی را نیز نتیجه می‌دهد.

$$N_2 : \frac{\lhd (X \sim\!\succ C)}{\lhd (X)}$$

- **قواعد مالکیت**

اگر پیامی روی شبکه قرار گیرد. هر عاملی مثل $P$ می‌تواند آن را گرفته مالک شود.

$$P_1 : \frac{\lhd (X)}{P \ni X}$$

اگر عامل $P$ پیامی شامل چند جزء را مالک شود، اجزاء آ ن را نیز مالک است.

$$P_2 : \frac{P \ni (X,Y)}{P \ni X}$$

- **قواعد تعبیر پیام**

$$I_1 : \frac{\lhd (g^X \bmod n \sim\!\succ P \xleftarrow{g^z \bmod n} Q), P\Delta(g^z \bmod n), P \mid\!\equiv \overset{\smile Z}{\longmapsto} Q}{P \mid\!\equiv P \xleftarrow{g^z \bmod n} Q}$$

اگر پیامی مثل $g^X \bmod n$ به همراه این دانش ضمنی که $g^Z \bmod n$ مقدار مخفی مشترکی برای $P$ و $Q$ است روی شبکه قرار گیرد و $P$ توانایی ساخت $g^Z \bmod n$ را داشته باشد و $P$ به اینکه $Z$ یک مقدار مخفی برای $Q$ است معتقد باشد، در این صورت $P$ به اینکه $g^Z \bmod n$ مقدار مخفی مشترکی بین وی و $Q$ است متقاعد خواهد شد.

اگر عامل $P$، مخفی بودن پیامی مثل $X$ نزد $Q$ را قبول داشته باشد، مخفی بودن مقدار مجموع $X$ با هر چیز دیگر را نیز قبول دارد.

$$I_2 : \frac{P \mid\!\equiv \overset{\smile X}{\longmapsto} Q}{P \mid\!\equiv \overset{\smile X+Y}{\longmapsto} Q}$$

اگر عامل $P$، مخفی بودن پیامی مثل $X$ نزد $Q$ را قبول داشته باشد، مخفی بودن مقدارحاصل ضرب $X$ با هر چیز دیگر را نیز قبول دارد.

$$I_3 : \frac{P \mid\equiv \overset{X}{\smile\!\!\!\longmapsto} Q}{P \mid\equiv \overset{X.Y}{\smile\!\!\!\longmapsto} Q}$$

- **قواعد ساخت پیام**

اگر عامل $P$ توانایی ساخت $g^X \bmod n$ و $g^Y \bmod n$ را داشته باشد و پیمانهٔ $n$ را نیز مالک باشد، توانایی ساخت $g^{X+Y} \bmod n$ را نیز خواهد داشت.

$$C_1 : \frac{P\Delta(g^X \bmod n), P\Delta(g^Y \bmod n), P \ni n}{P\Delta(g^{X+Y} \bmod n)}$$

عامل $P$ مقادیر $g$، $n$ و $X$ را داشته باشد، توانایی ساخت $g^X \bmod n$ را دارد.

$$C_2 : \frac{P \ni g, P \ni n, P \ni X}{P\Delta(gX \bmod n)}$$

اگر عامل $P$ مالک فرمولی باشد، توانایی ساخت آن را نیز دارد.

$$C_3 : \frac{P \ni X}{P\Delta X}$$

اگر عامل $P$، مقادیر $g^X \bmod n$، $Y$، $X$ را داشته باشد، توانایی ساخت $g^{XY} \bmod n$ را خواهد داشت.

$$C_4 : \frac{P \ni g^X \bmod n, P \ni n, P \ni Y}{P\Delta(g^{X.Y} \bmod n)}$$

## ٤-٤)تحلیل پروتکل‌ها

تابحال با در نظر گرفتن مدل محاسباتی مطرح شده در بخش ۲ به همراه زبان توصیف $\mathcal{L}$ و قواعد $\mathcal{PIR}$، به یک روش خاص تحلیل برای پروتکل‌های توافق کلید دست یافته‌ایم. در این قسمت سعی می‌کنیم پروتکل‌های نمونه توافق کلید را با این روش تحلیل نماییم.

- **پروتکل توافق کلید دیفی‌هلمن**

شکل ۱ توصیف پروتکل دیفی‌هلمن و مسیر اجرای آن بر اساس مدل محاسباتی مطرح شده را نشان می‌دهد. توصیف پروتکل شامل نقش‌های پروتکل به همراه قدم‌های هر نقش، قدم اولیه پروتکل ($S_0$) و قدم جایگذاری اولیه ($\delta_0$) است. مسیر اجرای پروتکل نیز به کمک تعریف توابع جایگذاری $\delta_1$ و $\delta_2$ بین قدم‌هایی که رابطه دسترسی پذیری دارند نشان داده شده است. قدم مقدار دهی اولیه پروتکل دیفی‌هلمن، پیش فرض‌های این پروتکل یعنی اینکه آغازگر و مخاطب پیمانه اول $n$ و مولد میدان $g$ را در اختیار دارند را نشان می‌دهد .

پروتکل دارای دو نقش آغازگر ($\rho_I$) و مخاطب ($\rho_R$) است. قدم اول نقش آغازگر شامل سه توانایی است. توانایی اول نشان دهنده ایجاد یک مقدار تصادفی مخفی ($x$) توسط آغازگر است. توانایی دوم نشان دهندهٔ ساخت $g^x \bmod n$ توسط آغازگر و توانایی سوم نشان دهندهٔ ارسال آن به مخاطب است. پیامی که آغازگر به عنوان خروجی قدم اول خود روی شبکه می‌گذارد مشتمل بر این دانش ضمنی است که از طریق توان رسانی فرمول ( $g^x \bmod n$) بافرمولی مثل $Y$ به پیمانهٔ $n$ می‌توان مقدار مخفی مشترک مناسبی برای آغازگر و مخاطب ایجاد نمود. مقدار $Y$ در اینجا یک مقدار متغیر است. مخاطب نیز در توانایی اول از قدم اول خود انتظار دارد در وضعیتی که با مسند $S_1$ تطبیق پیدا می کند پیامی مثل $\mathcal{M}$ از شبکه دریافت کند که از طریق توان رسانی آن با مقدار مخفی خود ($y$) به پیمانهٔ $n$ بتواند مقدار مخفی مشترکی با فرستندهٔ پیام ایجاد نماید. سایرتوانایی‌های مخاطب در قدم اول نشان دهندهٔ ساخت مقدار مخفی مشترک، ساخت پیام جهت ارسال به مخاطب، و ارسال آن است. مخاطب نیز در قدم دوم خود انتظار دریافت پیامی را دارد که باتوان رسانی آن با مقدار مخفی خود (یعنی $x$) بتواند به مقدار مخفی مشترک دست یابد. توانایی اول از این قدم، این موضوع را نشان می‌دهد. توانایی دوم این قدم نیز نشان دهندهٔ ساخت مقدار مخفی مشترک توسط آغازگر است.

با دقت در پروتکل مشخص می‌شود که پروتکل دیفی‌هلمن اجرا پذیر است به این معنی که پیش شرط‌های تمامی توانایی‌های آن در مسیر اجرا با کمک ( استنتاج روی دانش تجمعی حاصل از توانایی‌های ما قبل آن به پیمانه $\mathcal{PIR}$ قابل حصول است، یعنی :

$$K_0(P) = \Phi \mid\!=_{PIR} pre(\alpha_1(S_0)\delta_0) = \Phi \ , \ K_1(P) = \{a \ni g, a \ni n\}$$

$$K_1(P) \mid\!=_{PIR} pre(\alpha_2(S_0)\delta_0) = \Phi \ , \ K_2(P) = K_1(P) \cup \{b \ni g, b \ni n\}$$

$$K_2(P) \mid\!=_{PIR} pre(\alpha_1(S_1(\rho_I))\delta_0) = \Phi \ , \ K_3(P) = K_2(P) \cup \{a \ni x, a \mid\equiv \overset{x}{\smile\!\!\!\longmapsto} a\}$$

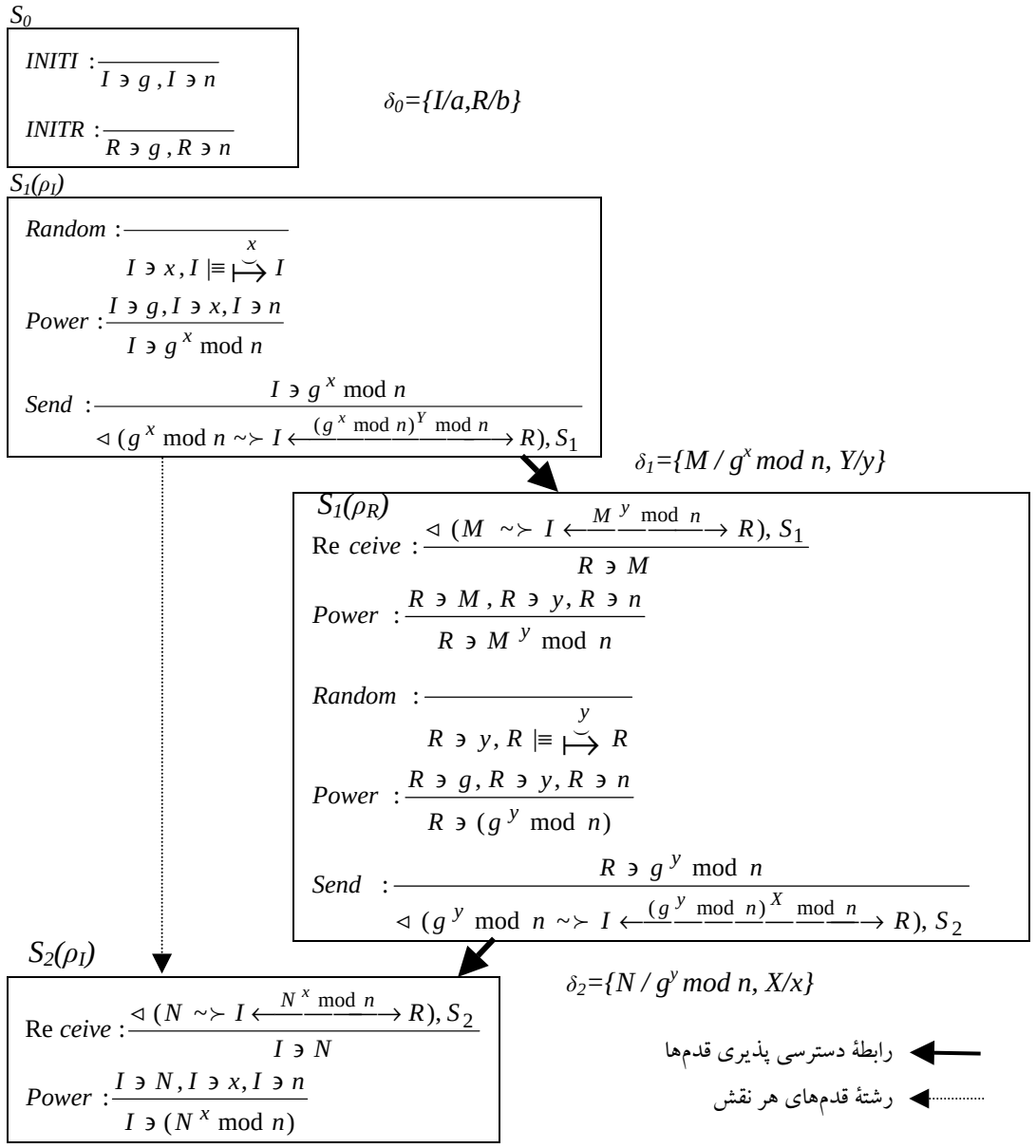
$$\vdots$$

$$K_5(P) \mid\!=_{PIR} pre(\alpha_1(S_1(\rho_R))\delta_0\delta_1) = \{\lhd (g^x \bmod n \sim\!\!\succ a \xleftarrow{(g^x \bmod n)^y \bmod n} b, S_1\},$$

$$K_6(P) = K_5(P) \cup \{b \ni g^x \bmod n\}$$

$$\vdots$$

$$k_{11}(P) \mid\!=_{PIR} pre(\alpha_2(s_2(\rho_I))\delta_0\delta_1\delta_2) \qquad , k_{12}(P) = k_{11}(P) \cup \{a \ni (g^y \bmod n)^x \bmod n\}$$




**شکل ۱) توصیف پروتکل دیفی‌هلمن**

هدف از پروتکل دیفی‌هلمن این است که طرفین پس ازانجام پروتکل مقدار $g^{xy} \bmod n$ را به عنوان یک مقدار مخفی مشترک قبول کنند. برای وارسی این ویژگی بایستی اهداف زیر در طی مسیر اجرا محقق شوند :

$$K : \omega_2 \models_{PIR} \quad b \models a \xleftarrow{g^{x.y} \bmod n} b$$

$$K : \omega_3 \models_{PIR} \quad a \models a \xleftarrow{g^{x.y} \bmod n} b$$

یعنی به ترتیب پس از قدم‌های ۲و۳ از مسیراجرا بتوان اعتقاد عوامل $b$ و $a$ به مناسب بودن مقدار $g^{xy} \bmod n$ به عنوان یک مقدار مشترک مخفی را از دانش پروتکل در این قدم‌ها و تحت قواعد $\mathcal{PIR}$ استنتاج کرد. البته هیچکدام از این دو نتیجه گیری قابل انجام نیست. در مورد نتیجه‌گیری اول، بر اساس قانون $I_1$ از مجموعه قواعد $\mathcal{PIR}$ بایستی سه مسند زیر درست باشند :

$$\lhd (g^X \bmod n \rightarrow a \xleftarrow{g^{x.y} \bmod n} b)$$

$$a \Delta g^{x.y} \bmod n$$

$$a \models \overset{x.y}{\smile\!\!\rightarrow} b$$

مسند اول از دانش پروتکل در قدم دوم، بلاواسطه قابل حصول است. مسند دوم نیز به کمک دانش پروتکل و بکارگیری قاعدهٔ $C_4$ از $\mathcal{PIR}$ قابل حصول است ولی مسند سوم را نمی‌توان استنتاج کرد. این امر بدین معنی است که عامل $a$ نمی‌تواند متقاعد شود که یک جزء مخفی برای $b$ در فرمول $x.y$ وجود دارد. این امر زیاد هم دور از انتظار نیست، زیرا مقدار $x$ که اصولاً متعلق به عامل $b$ نیست، در مورد مقدار $y$ نیز تنها اطلاع $a$ در مورد آن، پیام $g^x \bmod n$ است که برای $a$ یک پیام کاملاً نامشخص و تصادفی تلقی می‌شود. هر عاملی ( از جمله $b$ ) می‌توانسته با توان رسانی $g$ با $x$ ، این پیام را ساخته و ارسال کند. در واقع اصالت پیام $g^{xy}$ و تعلق آن به $b$ برای $a$ محرز نمی‌شود.

در مورد هدف دوم مطرح شده نیز تحلیلی به صورت فوق جوابگو است. لازم به یاد آوری است که دلیل انجام پذیری حمله فردی در میان[1] *[8]* به پروتکل دیفی‌هلمن نیز وجود همین مسئله است.

- **پروتکل توافق کلید الجمال**

پروتکل توافق کلید الجمال فقط شامل یک مرحله ارسال پیام از طرف آغازگر به مخاطب است. توصیف پروتکل در شکل ۲ نشان داده شده است. نکته حائز اهمیت و متفاوت با پروتکل قبلی در اینجا، قدم مقدار دهی اولیه است. همانطور که گفتیم در پروتکل الجمال، مخاطب دارای یک زوج کلید عمومی و خصوصی است که در ساخت کلید مشترک از آنها نیز استفاده می‌شود. در قدم مقدار دهی اولیهٔ پروتکل، پس شرط توانایی *INITI* محتوی دانش اولیهٔ عامل آغازگر است. آغازگر علاوه بر دارا بودن $g$ و $n$ ، مقدار کلید عمومی مخاطب یعنی $g^s \bmod n$ را در اختیار دارد و به مخفی بودن s به عنوان کلید خصوصی مخاطب در نزد وی معتقد است. به همین ترتیب پس شرط *INITR* نیز تعلق مقدار s به مخاطب واعتقاد وی به مخفی بودن آن برای خودش را شامل می‌شود. به راحتی می‌توان نشان داد که پروتکل الجمال با توصیف فوق قابل اجرا است. همچنین رسیدن به دو هدف زیر بایستی در مورد پروتکل بررسی شود:

$$K : \omega_1 \models_{PIR} \quad a \models a \xleftarrow{g^{sx} \bmod n} b$$

$$K : \omega_2 \models_{PIR} \quad b \models a \xleftarrow{g^{sx} \bmod n} b$$

$S_0$

$$INITI : \frac{}{I \ni g, I \ni n, I \ni (g^s \bmod n), I \models \overset{s}{\underset{\smile}{\mapsto}} R}$$

$$INITR : \frac{}{R \ni g, R \ni n, R \ni s, R \models \overset{s}{\underset{\smile}{\mapsto}} R}$$

$\delta_0 = \{I/a, R/b\}$

$S_1(\rho_1)$

$$Random : \frac{}{I \ni x, I \models \overset{x}{\underset{\smile}{\mapsto}} I}$$

$$Power : \frac{I \ni g, I \ni x, I \ni n}{I \ni g^x \bmod n}$$

$$Power : \frac{I \ni (g^s \bmod n), I \ni x}{I \ni g^{sx} \bmod n}$$

$$Send : \frac{I \ni g^x \bmod n}{\lhd (g^x \bmod n \sim\succ I \xleftarrow{(g^x \bmod n)^s \bmod n} R), S_1}$$

$\delta_1 = \{M / g^x \bmod n\}$

$S_1(\rho_R)$

$$\text{Re}\,ceive : \frac{\Delta(M \rightarrow I \xleftarrow{M^S \bmod n} R), S_1}{R \ni M}$$

$$Power : \frac{I \ni M, I \ni S, I \ni n}{I \ni (M^S \bmod n)}$$

**شکل ۲) توصیف پروتکل توافق کلید الجمال**

[1] *Man-In-The-Middle*

از دو هدف فوق، هدف اول با کمک دانش پروتکل در قدم اول و مجموعه قواعد $\mathcal{PIR}$ قابل اثبات است ولی هدف دوم را نمی‌توان اثبات کرد. در مورد هدف اول، وجود کلید خصوصی عامل $b$ (یعنی $s$) در توان $g^{sx} \bmod n$ و اعتقاد عامل $a$ به مخفی بودن آن برای $b$، منجر به تحقق این هدف می‌شوددر واقع چون پیام $g^{sx} \bmod n$ به نحوی توسط عامل $b$ امضاء شده است، همین امضاء اصالت و تعلق آن به $b$ را برای $a$ محرز می‌کند. در مورد هدف دوم، عامل $b$ هیچ راهی برای احراز اصالت پیام $g^{sx} \bmod n$ ندارد و به همین دلیل نمی‌تواند آن را به عنوان کلید مشترک خود با عامل $b$ قبول کند.

- **پروتکل توافق کلید *MTI/A0***

همانطور که قبلا نیز گفتیم، پروتکل توافق کلید *MTI/A0*، یک پروتکل دو مرحله‌ای است که تبادل پیام‌ها در آن دقیقاً شبیه پروتکل دیفی‌هلمن است ولی نحوهٔ ساخت کلید مشترک متفاوت می‌باشد. طرفین توافق دارای زوج کلیدهای عمومی خصوصی هستند و کلید مشترک نهایی به نحوی ساخته می‌شود که امضاء هر دو طرف ( رد پای کلید خصوصی در توان $g$) روی کلید مشترک باشد. در شکل ۳ قدم مقدار دهی اولیه از توصیف پروتکل نشان داده شده است. همانطور که در شکل می‌بینیم، به عنوان پیش فرض‌های پروتکل، فرض شده است که هر کدام از عوامل آغازگر و مخاطب، علاوه بر دارا بودن مقادیر $g$ و $n$، کلید عمومی طرف مقابل و کلید خصوصی خود را نیز دارا هستند. همچنین هر کدام از طرفین، مخفی بودن کلید خصوصی طرف مقابل برای وی و مخفی بودن کلید خصوصی خودش را باور دارد.

$S_0$

$$INITI: \frac{}{I \ni g, I \ni n, I \ni (g^t \bmod n), I \mid\equiv \overset{t}{\longmapsto} R, I \ni s, I \mid\equiv \overset{s}{\longmapsto} I}$$

$$INITR: \frac{}{R \ni g, R \ni n, R \ni (g^s \bmod n), R \mid\equiv \overset{s}{\longmapsto} I, R \ni t, R \mid\equiv \overset{t}{\longmapsto} R}$$

$\delta_0=\{I/a,R/b\}$

**شکل ۳) قدم مقدار دهی اولیهٔ پروتکل *MTI/A0***

توصیف مسیر اجرای پروتکل با اندکی تفاوت، شبیه توصیف مسیر اجرای پروتکل دیفی‌هلمن است و ما در اینجا به توصیف آن نمی پردازیم. براحتی می‌توان مشاهده کرد که پروتکل *MTI/A0*، اجرا پذیر است و هر دو هدف زیر نیز در مورد آن اثبات می‌گردد :

$$K: \omega_2 \mid=_{PIR} \quad b \mid\equiv a \xleftarrow{g^{sy+tx} \bmod n} b$$

$$K: \omega_3 \mid=_{PIR} \quad a \mid\equiv a \xleftarrow{g^{sy+tx} \bmod n} b$$

نتایج حاصل از سه پروتکل فوق در جدول ۱ نشان داده شده است. تحلیل این پروتکل‌ها، اهمیت امضاء شدن پیام‌ها در تبادلات توافق کلید و تضمین اصالت آنها را نشان می‌دهد.

**جدول ۱) نتایج تحلیل پروتکل‌های توافق کلید**

| | اجرا پذیری | $a \mid\equiv a \xleftarrow{g^Z \bmod n} b$ * | $b \mid\equiv a \xleftarrow{g^Z \bmod n} b$ * |
|---|---|---|---|
| دیفی‌هلمن | × | - | - |
| الجمال | × | × | - |
| *MTI/A0* | × | × | × |

* مقدار $Z$ برای پروتکل‌های دیفی‌هلمن، الجمال و *MTI /A0* به ترتیب برابر است با : $x.y$، $s.x$ و $s.y+t.x$.

## ۵) جمع‌بندی و نتیجه‌گیری

در *[1]* چارچوبی برای بیان روش‌های تحلیل پروتکل‌های رمزنگاری ارائه شد. این چارچوب دارای چهار جزء است : زبانی برای توصیف ویژگی‌های پروتکل، یک مدل محاسباتی از پروتکل و نحوهٔ اجرای آن و دو مجموعه از قواعد استنتاج، یکی برای پروتکل و دیگری برای نفوذی. هدف از این چارچوب این است که به عنوان یک دستهٔ کلی از روش‌های تحلیل، قابلیت بیان روش‌های تحلیل منطقی و روش‌های ساخت حمله را داشته باشد.

ما در این مقاله نشان دادیم که چگونه می توان با بکارگیری این چارچوب و مدل محاسباتی مطرح شده در *[1]* یک روش تحلیل ارائه نمود. برای این کار یک زبان توصیف برای بیان ویژگیهای یک سری از پروتکل‌های توافق کلید به همراه مجموعه ای از قواعد استنتاجی خاص اینگونه پروتکل‌ها ارائه شد. این مجموعه شامل قوانینی برای استنتاج راجع به پیام‌های دریافتی توسط عوامل پروتکل، مالکیت‌های عوامل پروتکل، تعبیر پیام‌های دریافتی و چگونگی ساخت پیامها ومقادیر مخفی عوامل است.

در روش ارائه شده، فرض کرده ایم که نفوذی دارای نقش صریحی نیست و بنابر این مجموعه قواعد استنتاجی نفوذی تهی در نظر گرفته می شود. برای انجام تحلیل، یک مجموعه دانش برای پروتکل تعریف می شود که حاوی تمامی مسندات صحیح راجع به عوامل پروتکل است. مقدار اولیهٔ این مجموعه دانش، فرضیات اولیهٔ پروتکل است و در طی مسیر اجرا با به فعلیّت رسیدن توانایی‌های عوامل پروتکل، وبا کمک قواعد استنتاجی عوامل، این مجموعه گسترش می یابد. تحلیل پروتکل‌ها در این روش به معنی بررسی اجرا پذیری پروتکل و بررسی قابل اثبات بودن بعضی ویژگیهای امنیتی از طریق دانش پروتکل و مجموعه قواعد استنتاج پروتکل است. سه پروتکل توافق کلید دیفی‌هلمن، الجمال و *MTI/A0* با کمک این روش تحلیل شدند. نتایج تحلیل، قابل قبول نبودن کلیدهای ایجاد شده برای طرفین در پروتکل دیفی‌هلمن و برای مخاطب در پروتکل الجمال، به دلیل عدم احراز اصالت پیام‌های دریافتی را نشان می دهد. لازم به ذکر است که مجموعه قواعد منطق‌های *BAN* و *GNY* قابلیت تحلیل پروتکل‌های توافق کلید را ندارند.

مزیتی که بکارگیری چارچوب کلی *[1]* دارد این است که می توان به کمک آن به سهولت برای هر دسته از پروتکل‌های رمزنگاری با تکیه بر ویژگیهای خاص آنها، روش‌های تحلیل خاص منظوره ایجاد نمود. این امر به ویژه در مورد پروتکل‌های تجارت الکترونیک به دلیل نیاز به ویژگیهای خاص در هر مجموعه پروتکل حائز اهمیت است. کار در زمینهٔ ارائهٔ روش‌های دیگر تحلیل به ویژه روش‌های ساخت حمله و روش‌های ترکیبی به کمک چارچوب مذکور ادامه دارد.

## ٦) مراجع


۱. ترک لادانی، بهروز و جلیلی، سعید، *چارچوبی برای وارسی پروتکل‌های رمزنگاری* ، مجموعه مقالات اولین کنفرانس رمز ایران، دانشگاه امام حسین(ع) ، آبان ۱۳۸۰.

2. S. Gritzalis, N. Nickitakos and P. Georgiadis, *Formal Methods for the Analysis and Design of Cryptographic Protocols: A State of the Art Review*, Communication and Multimedia Security, Vol. 3, S. Katsicas(Ed.) IFIP, Chapman & Hall 1997.

3. M. Burrows, M. Abadi and R. Needham, *A Logic of Authentication*, Proc. Royal Society, Series A, Vol. 246, No. 1871, pp. 233-271, 1989.

4. L. Gong, R. Needham and R. Yahalom, *Reasoning About Belief in Cryptographic Protocols*, Proc. IEEE 1990 symposium on Security and Privacy, Oakland, California, pp. 234-248, May 1990.

5. C. Meadows, *Applying Formal Methods to the Analysis of a Key Management Protocol*, The Journal of Computer Security, Vol. 1, No. 1, Jan. 1992.

6. P. Ryan and S. Shneider, *Modeling and Analysis of Security Protocols*, Pearson Education, 2001.

7. C. Meadows, *Invariant Generation Techniques in Cryptographic Protocol Analysis*, Proceedings of the 13th Computer Security Foundations Workshop, IEEE Computer Society Press, July 2000.

8. A. Menezes, P. C. Van Oorschot, S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.