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

مدل‌سازی و تحلیل کمّی پروتکل‌های امنیتی مبتنی بر بررسی مدل احتمالی و ابزار PRISM

مجتبی اکبرزاده، محمد عبداللهی ازگمی

آزمایشگاه مهندسی کارایی و اتکاءپذیری، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران

azgomi@iust.ac.ir ،m_akbarzade@comp.iust.ac.ir

چکیده

روشهای صوری و فنون درستی‌یابی، از ابزارهای تحلیل سیستم‌ها هستند. هدف این مقاله، ارائه روشی برای درستی‌یابی و تحلیل پروتکل‌های امنیتی است. ویژگی اصلی روش ارائه شده در این مقاله بهره‌گیری از فنون بررسی مدل احتمالی برای تحلیل پروتکل‌های امنیتی است. با استفاده از فنون بررسی مدل احتمالی، پروتکل‌های امنیتی را می‌توان به‌صورت کمّی مورد بررسی قرار داد. مزیت بسیار مهم درستی‌یابی کمّی پروتکل، قابلیت بررسی دقیق یک پروتکل در برابر مهاجمین مختلف و همچنین امکان مقایسه چندین پروتکل با یکدیگر در مقابل مهاجمین مختلف است. در این مقاله برای بررسی مدل از ابزار PRISM که توانایی بررسی مدل احتمالی را دارد استفاده شده است. در این روش ابتدا ویژگی‌ها و مشخصات عامل‌های پروتکل و مهاجمین با استفاده از زبان پروتکل‌های امنیتی بیان شده و مشخصات هر عامل با استفاده از زبان ابزار PRISM مدل می‌شود و سپس بررسی احتمالی مدل انجام شده و احتمال وجود رخنه در پروتکل مورد بررسی قرار می‌گیرد. به‌عنوان مطالعه موردی، پروتکلهای نیدهام-شرودر و TMN با استفاده از روش پیشنهادی مدلسازی و تحلیل کمّی شده که نتایج حاصله در مقاله ارائه شده است.

واژه‌های کلیدی

پروتکل‌های امنیتی، درستی‌یابی، بررسی مدل احتمالی، روشهای صوری.

۱- مقدمه

یکی از چالش‌های دنیای امروزی ایجاد امنیت در سیستم‌های کامپیوتری است که با وجود مهاجمان و حمله‌های امنیتی گوناگون، روشهای مختلفی برای حصول امنیت به‌کار گرفته می‌شود. پروتکل‌های امنیتی نقش بسیار مهمی در ایجاد امنیت دارند. هدف از پروتکل‌های امنیتی دستیابی به امنیت است که برای رسیدن به این هدف باید این پروتکل‌ها بسیار دقیق و ماهرانه طراحی شوند. همچنین، با افزایش روند ایجاد و طراحی پروتکل‌های امنیتی، درستی‌یابی آنها یکی از زمینه‌های تحقیقاتی مهم است. برای رسیدن به این مهم احتیاج به چارچوبی است که به وسیله‌ی آن بتوان پروتکل‌های امنیتی را توصیف و تحلیل نمود.

یکی از پرکاربردترین روشهای تجزیه و تحلیل پروتکل‌های امنیتی، فنون مدل‌سازی صوری است. مزایای استفاده از روشهای صوری، مبتنی بر ریاضی بودن و همچنین قابلیت درستی‌یابی خودکار این روشها است.

در درستی‌یابی پروتکل‌های امنیتی دو مرحله‌ی اصلی وجود دارد. یکی از این مراحل، شناسایی خصوصیت‌ها یا ویژگی‌های پروتکل و توصیف آنها با یک زبان صوری است. برای این منظور، روشهای مختلفی در دسترس است. در اینجا برای توصیف ویژگی‌های پروتکل از زبان پروتکل امنیتی (SPL)[1] و برای مدل‌سازی آن از ابزار PRISM بهره‌گرفته شده است. در ادامه مدل بدست آمده مورد بررسی مدل احتمالی[2] قرار گرفته و احتمال وجود رخنه در پروتکل مشخص می‌شود.

ادامه این مقاله به این صورت سازماندهی شده است که در بخش (۲)، به مرور و بررسی کارهای مشابه در به‌کارگیری روشهای صوری برای تجزیه و تحلیل پروتکل‌های امنیتی پرداخته می‌شود.

[1] Security Protocol Language (SPL)
[2] Probabilistic Model Checking

در بخش (۳)، به نحوه‌ی توصیف پروتکل‌های امنیتی خواهیم پرداخت. در بخش (۴)، روش بررسی مدل احتمالی با استفاده از ابزار PRISM ارائه خواهد شد. در بخش (۵)، نحوه‌ی مدل‌سازی و درستی‌یابی پروتکل بیان می‌شود. در بخش (۶)، به بررسی موردی پروتکل TMN پرداخته می‌شود. سرانجام در بخش (۷)، نتیجه‌گیری ارائه خواهد شد.

۲- کارهای مرتبط

استفاده از روشهای صوری برای درستی‌یابی سیستم‌های کامپیوتری بسیار متداول است. روشهای صوری فراوانی مانند جبر فرایندی، فنون استقرایی، روشهای مبتنی بر منطق و شبکه‌های پتری وجود دارد که هر یک از این روشها برای مدل‌سازی سیستم‌ها استفاده شده، اما در هر حال هر کدام دارای مزایا و معایبی هستند.

کارهای تحقیقاتی فراوانی در این زمینه صورت گرفته است که منجر به ایجاد ابزارها و روشهای مختلف شده است. یکی از چارچوب‌های ارائه شده برای بیان ویژگی‌های پروتکل از زبان UMLsec استفاده نموده است [1]. در این چارچوب برای بررسی مدل از ابزارهایی مانند AVISPA که این ابزار از روشهای بررسی مدل متفاوتی استفاده می‌نماید بهره‌گرفته شده است. همچنین از ابزارهای FOCUS, CSP که در حال حاضر در دسترس طراحان پروتکل‌های امنیتی است برای درستی‌یابی پروتکل‌های امنیتی استفاده شده است [2]. روشهایی نیز بر اساس شبکه پتری ارائه شده‌اند که در آنها از شبکه‌های پتری سطح بالا برای برای مدل‌سازی پروتکل‌ها بهره‌گرفته شده است [3].

تفاوت این روشها و چارچوب‌ها در فنون صوری و ابزارهای مختلفی است که در آنها به‌کار گرفته شده است، مانند روشهایی که از جبر فرایندی برای توصیف صوری پروتکل‌های امنیتی بهره گرفته‌اند. در بعضی از چارچوب‌ها نیز از روشهای صوری مبتنی بر منطق یا روشهای استقرایی نیز استفاده شده است [4, 5, 6].

۳- توصیف پروتکل‌های امنیتی

در یک پروتکل امنیتی، عامل‌های مختلفی وجود دارند. برای مثال در پروتکل نیدهام- شرودر[1] (NS) دو نقش اصلی وجود دارد، که عبارتند از آغازگر[2] و پاسخگو[3] که این عامل‌ها از کلیدهای خصوصی و عمومی برای فرستادن و دریافت پیغام بهره می‌گیرند. فرایند پروتکل امنیتی NS که A آغازگر و B پاسخگوست، به‌صورت زیر است [7]:

۱) $A \rightarrow B: \{m, A\}Pub(B)$ عامل آغازگر A یک پیغام حاوی مقصود فعلی[4] (m) و نام خود را که با کلید عمومی عامل B رمزگذاری شده است را به عامل B می‌فرستد.

۲) $B \rightarrow A: \{m, n\}Pub(A)$ سپس عامل B یک پیغام حاوی مقصود فعلی (m) که از عامل A در مرحله قبل دریافت کرده و یک مقصود فعلی (n) جدید را ایجاد نموده و این پیغام را با کلید عمومی عامل A رمزگذاری کرده و برای عامل A می‌فرستد.

۳) $A \rightarrow B: \{n\}Pub(B)$ سرانجام عامل A مقصود فعلی n را که در مرحله قبل دریافت نموده با استفاده از کلید عمومی عامل B رمزگذاری نموده و برای آن می‌فرستد. بر اساس این پروتکل یک ارتباط تصدیق شده ایجاد می‌شود.

در ادامه بیان چگونگی پروتکل NS به وسیله زبان پروتکل امنیتی پرداخته خواهد شد. در اینجا مجموعه‌های نامتناهی زیر را در نظر می‌گیریم:

$N = \{n, n1, n2, ...\}$
$G = \{A, B, ...\}$
$I = \{i, ...\}$

از این مجموعه‌ها به ترتیب برای مقادیر مقصود فعلی و نام عامل‌ها و شناسه‌های فرایندها استفاده می‌شود. همچنین سه مجموعه متغیرهای زیر را نیز برای مقوصد فعلی و نام عامل‌ها و شناسه‌های فرایندها مورد استفاده قرار می‌دهیم:

$Vn = \{u, v, w, x, y, z, ...\}$
$Vq = \{X, Y , Z , ...\}$
$VM = \{\Psi, \Psi\_, \Psi1, . . .\}$

از نماد برداری $\vec{X}$ برای بیان لیست $X_1, X_2, X_3, ...$ که معمولاً به‌صورت $\{X_1, X_2, X_3, ...\}$ بیان می‌شود استفاده می‌کنیم. در ادامه از e برای نشان دادن یک عبارت، k برای کلید، M برای پیام‌ها و P برای نشان دادن فرایندها در گرامر استفاده شده است:

$e ::= n \mid ... \mid A \mid ... \mid y \mid ... \mid Y \mid ...$
$k ::= Pub(e) \mid Priv(e) \mid Key(e1, e2)$
$M ::= e \mid k \mid M1, M2 \mid \{M\}k \mid \Psi$
$p ::= out\ new\ y\ M.p \mid in\ pat\ y\ \Psi M.p \mid \|_{i \in I}\ pi$

در اینجا Pub(e) برای بیان کلید عمومی، Priv(e) کلید خصوصی و Key(e1, e2) کلید متقارن مشترک بین e_1 و e_2 مورد استفاده قرار می‌گیرد. همچنین برای بیان رمزگذاری پیام M به‌وسیله کلید K از عبارت {M}k استفاده می‌شود.

[1] Needham-Schroder (NS)
[2] Initiator
[3] Responder
[4] Nonce

۳-۱ توصیف فرایند امنیتی در زبان پروتکل امنیتی

در SPL، فرایند $\vec{y}$ out new M.p یک عمل خروجی است که نشان‌دهنده‌ی فرستادن پیام توسط فرایند P است. در این فرایند یک مقدار متمایز جدید مانند $\vec{n} = \{n_1, n_2, ...\}$ ایجاد شده و در متغیر $\vec{y} = \{y_1, y_1, ...\}$ قرار می‌گیرد. سپس پیغام $M\{\vec{n}/\vec{y}\}$ به شبکه فرستاده می‌شود[۸].

فرایند $\vec{y}$ in pat ΨM.p یک عمل ورودی است که نشان‌دهنده‌ی دریافت پیام توسط فرایند P است. در این فرایند عامل منتظر یک پیغام ورودی مطابق با الگوی M است و با دریافت پیغام متناظر به متغیرهای محلی $\vec{y}, \vec{\psi}$ مقداردهی می‌شوند. همچنین عملیات ‖iϵI pi بیان‌کننده‌ی اجرای موازی چندین فرایند است.

۳-۲ توصیف پروتکل به وسیله زبان پروتکل امنیتی

پروتکلی که در قسمت‌های قبل بیان شد که شامل دو عامل آغازگر و پاسخگو بود به وسیله زبان SPL به‌صورت زیر بیان می‌شود [9]:

```
Init(A, B) = out new{y}{y, A}Pub(B).
             in pat{z}{y, z}Pub(A).
             out {z}Pub(B).

Resp(B) = in pat{x, Z}{x, Z}Pub(B).
          out new {v}{x, v}Pub(Z).
          in pat{v}{v}Pub(B).
```

در اینجا Resp (B) و Init (A, B) به صورت موازی در حال کار هستند که در زیر نحوه‌ی اجرای آنها مورد بررسی قرار می‌گیرد.

عامل A مقصود فعلی y را همراه با نام A ترکیب نموده و به وسیله کلید عمومی عامل B رمزگذاری کرده و برای عامل B ارسال می‌کند. عامل B منتظر دریافت پیامی است که مطابق با الگوی مورد نظرش شامل دو پارامتر Z و X است.

```
Init(A, B) = out new{y}{y, A}Pub(B)
Resp(B) =  in pat{x, Z}{x, Z}Pub(B)
```

عامل B یک مقصود فعلی جدید v را همراه با مقصود فعلی که در گام قبل دریافت نموده است به وسیله کلید عمومی عامل A رمزگذاری نموده و برای عامل A ارسال می‌کند. عامل A منتظر دریافت پیامی مطابق با الگوی مورد نظرش است که شامل دو پارامتر Z و Y است.

```
Resp(B) = out new {v}{x, v}Pub(Z)
Init(A, B) = in pat{z}{y, z}Pub(A)
```

عامل A پس از بررسی مقصود فعلی، آن را برای عامل B ارسال می‌نماید و B نیز منتظر دریافت مقصود فعلی خود است.

```
Init(A, B) = out {z}Pub(B)
Resp(B) = in pat{v}{v}Pub(B)
```

در این قسمت چهار نوع از مهاجمان را بر اساس ویژگی‌های رفتاری که از آنها وجود دارد به‌وسیله زبان پروتکل امنیتی توصیف کرده‌ایم. در ادامه نحوه عملکرد آنها بیان شده است:

```
Spy1 =in pat{Φ1, Φ2}{Φ1, Φ2}. out{Φ1}. out{Φ2}
Spy2=in pat{Ψ1}{Ψ1} in pat{Ψ2}{Ψ2}. out{Ψ1, Ψ2}
Spy3=in pat{Y }{Y }. in pat{Ψ'}{Ψ'}.out{Ψ'}Pub(Y )
Spy4=in pat{X} Priv(X). in pat{Φ'}{Φ'}Pub(X). out {Φ'}
SPY = ! _iϵ{1..4}Spy i
```

چون در این سیستم اجرا به صورت نامحدود بوده است نحوه اجرای سیستم به صورت موازی در زیر بیان می‌شود:

```
INIT = ‖A, BϵG  Init(A, B)
RESP =‖BϵG Resp(B)
NS = INIT‖ RESP ‖ SPY.
```

در این قسمت پروتکل امنیتی را با استفاده از زبان صوری بیان نمودیم و رفتارهای نقش‌های موجود در پروتکل به‌صورت کامل مشخص شد. در ادامه با توجه به ویژگی‌های بدست آمده رفتارهای عامل‌های پروتکل با استفاده از زبان PRISM مدل می‌شود و سپس بررسی مدل صورت می‌گیرد.

۴- بررسی مدل احتمالی

بررسی مدل احتمالی، تکنیکی صوری برای بررسی سیستم‌هایی است، که رفتاری تصادفی را به نمایش می‌گذارند. این مدل مبتنی بر ساخت و تحلیل مدل ریاضی سیستم است. مدل‌ها زبان‌های سطح بالا دارد که با این زبان برای بررسی کننده مدل، مشخصه‌های مورد نظر بیان می‌شوند. عموماً این مدل از چندین حالت تشکیل می‌شود که نشان‌دهنده پیکربندی‌های احتمالی سیستم است. انتقال‌دهنده‌ها ممکن است بین حالات اتفاق بیفتند و اطلاعاتی درباره زمان و احتمال وقوع انتقال را شامل می‌شوند [10].

۴-۱ مدل‌های احتمالی صوری

در بررسی مدل‌های احتمالی، مدل‌های صوری چندین نوع هستند که عبارتند از:

۱) زنجیره‌های مارکوف با زمان گسسته[1] (DTMC)

۲) زنجیره‌های مارکوف با زمان پیوسته[2] (CTMC)

۳) فرایندهای تصمیم مارکوف[3] (MDP)

برای بررسی و ارزیابی خودکار یک مدل احتمالی از تکنیک‌های صوری، الگوریتم‌ها و ابزارهای نرم‌افزاری استفاده می‌شود که فضای حالت محدود شده‌ای از سیستم احتمالی را برای ارزیابی خودکار در نظر می‌گیرد. چندین بررسی‌کننده مدل وجود دارد که برای بیان مدل‌ها در بررسی کننده‌های مدل باید از منطقی که آن بررسی‌کننده پشتیبانی می‌کند استفاده کرد. البته این منطق مختص خود آن بررسی کننده است که عمومی‌ترین منطق‌ها، CTL و PCTL است و می‌توان از آنها برای بیان خصوصیات مدل‌ها استفاده کرد.

[1] Discrete-Time Markov Chain (DTMC)
[2] Continuous-Time Markov Chain (CTMC)
[3] Markov Decision Process (MDP)

۲-۴ بررسی کننده مدل PRISM

ابزار PRISM، یک ابزار بررسی مدل‌های احتمالی است که به صورت مستقیم، از سه نوع مدل صوری که در بخش قبل بیان شد پشتیبانی می‌کند [11].

زبان ورودی این ابزار، زبانی بر مبنای حالت است و برای مدل‌های ترکیبی از متغیرهای مشترک و همگامی گذرها کمک گرفته است. برای توصیف خصوصیت در مدل‌های MDP و DTMC از توسعه احتمالی CTL استفاده می‌شود. این خصوصیت، ممکن است احتمال رخ دادن یک اتفاق خاص باشد که محاسبه احتمال یک شرط در DTMC یا احتمال کمینه/بیشینه در MDP را شامل می‌شود. بیان خصوصیت‌های مدل CTMC نیز بر مبنای CSL بیان می‌گردد. معماری PRISM از ساختار MTBDD برای نگهداری ماتریس‌های انتقال حالات و بعضی ساختارهای مورد نیاز دیگر، به‌صورت کارآمد استفاده می‌کند [12, 13].

۵- مطالعه موردی ۱: توصیف پروتکل NS با PRISM

در این بخش به بیان چگونگی مدل‌سازی پروتکل‌های امنیتی با استفاده از ابزار PRISM و مدل DTMC پرداخته می‌شود. در اینجا برای مدل کردن پروتکل هر عامل به‌عنوان یک ماژول مدل می‌شود. سپس ماژول‌ها از طریق میانگیر که به‌صورت سراسری تعریف شده‌اند با هم ارتباط برقرار می‌کنند.

۵-۱ مدل‌سازی پروتکل NS

پروتکل NS در بخش قبلی به‌صورت کامل توصیف شد. در اینجا چگونگی مدل‌سازی این پروتکل شرح داده می‌شود. در این پروتکل دو عامل اصلی وجود دارد که به‌عنوان دو ماژول مدل شده است. همچنین میانگیر نیز به‌صورت سراسری بیان می‌شود.

همانطور که در شکل (۱) نشان داده شده است، ماژول A1 مدل مربوط به آغازگر است که سه گام دارد و ماژول B2 نیز مدل عامل پاسخگو است.

```
probabilistic

global buf_noncel : [100..110];
global buf_nonce2  : [100..110];
global buf_send :[1..10];
global buf_res :[1..10];
global id_agent : [1..10];
global buf_empt : bool init true;

const   int agent_numl= 1;
const int anl=123;
module Al
        sl : [0..3] init 0;
        an2 : [100..110] ;
        aag_num : [1..10] ;
        []   (sl=0) & (buf_empt = true) -> 1 :(sl'=sl+1) & (buf_send'=agent_numl)& (buf_res' = 2) & (buf_noncel'=anl)&(buf_empt' = false);
        []   (sl=1) & (buf_empt = false)& (buf_res=agent_numl) & (buf_noncel=anl) -> 1   : (sl'=sl+1) & (an2' = buf_nonce2) & (aag_num'=buf_send) & (buf_empt' = true);
        []   (sl=2) & (buf_empt = true) -> 1     : (sl'=sl+1) & (buf_send'=agent_numl)& (buf_res' = aag_num) & (buf_noncel'=an2)& (buf_empt' = false);
endmodule


const int agent_num2= 2;
const int bnl=125;
module B2
        s2 : [0..3] init 0;
        bn2 : [100..110];
        bag_num : [1..10];
        []   (s2=0) & (buf_empt = false)& (buf_res=agent_num2)  -> 1     : (s2'=s2+1) & (bn2' = buf_noncel) & (bag_num'=buf_send) & (buf_empt' = true);
        []   (s2=1) & (buf_empt = true) -> 1     : (s2'=s2+1) & (buf_send'=agent_num2)& (buf_res' = bag_num) & (buf_noncel'=bn2)& (buf_nonce2' =bnl)&(buf_empt' = false);
        []   (s2=2) & (buf_empt = false) & (buf_res=agent_num2) & (buf_nonce2=bnl) -> 1  : (s2'=s2+1) & (buf_empt' = true);
endmodule
```

شکل ۱: توصیف پروتکل NS به‌صورت DTMC با استفاده از ابزار PRISM

```
const int agant_num3= 3;
module I1
        SI1 : [0..3] init 0;
        I1n : [100..110];
        I1ag_num : [1..10];
        []   (SI1=0) & (buf_empt = false)& (buf_res=agant_num3)  -> 1    : (SI1'=SI1+1) & (I1n' = buf_noncel) & (I1ag_num'=buf_send) & (buf_empt' = true);
        []   (SI1=1) & (buf_empt = true) -> 0.5  : (SI1'=SI1+1) & (buf_send'=I1ag_num)& (buf_res' = 5) & (buf_noncel'=I1n)&(buf_empt' = false)
                                        +0.5     : (SI1'=SI1+1) & (buf_send'=I1ag_num)& (buf_res' = 6) & (buf_noncel'=I1n)&(buf_empt' = false);
        []   (SI1=2) & (buf_empt = true) -> 0.5  : (SI1'=SI1+1) & (id_agent'=2)& (buf_res' = 5)& (buf_empt' = false)
                                        +0.5     : (SI1'=SI1+1) & (id_agent'=2)& (buf_res' =6)& (buf_empt' = false)     ;
endmodule

const int agant_num5= 5;
module I2
        SI3 : [0..3] init 0;
        I3n : [100..110];
        I3ag_num : [1..10];
        I3ag_num2 : [1..10];
        []   (SI3=0) & (buf_empt = false)& (buf_res=agant_num5)  -> 1    : (SI3'=SI3+1) & (I3n' = buf_noncel) & (I3ag_num'=buf_send) & (buf_empt' = true);
        []   (SI3=1) & (buf_empt = false)& (buf_res=agant_num5)  -> 1    : (SI3'=SI3+1) & (I3ag_num2'=id_agent) & (buf_empt' = true);
        []   (SI3=2) & (buf_empt = true) -> 1    : (SI3'=SI3+1) & (buf_send'=I3ag_num) &(buf_res'=I3ag_num2)& (buf_noncel'=I3n)&(buf_empt' = false)     ;
endmodule
```

شکل ۲: مدل مهاجم با استفاده از ابزار PRISM

۲-۵ مدل‌سازی مهاجم

در مدل‌سازی پروتکل‌های امنیتی علاوه‌بر عامل‌های درگیر در پروتکل، مهاجمانی وجود دارند که دارای اهمیت فراوانی هستند و باید آنها نیز مدل شوند. چندین نوع از مهاجمان و چگونگی عملکرد آنها در بخش ۳ به صورت کامل و با استفاده از زبان پروتکل امنیتی توصیف شده‌اند که در اینجا نحوه‌ی مدل‌سازی آنها بیان می‌شود.

در این قسمت به شرح مدل دو مهاجم پرداخته می‌شود. همانطور که در شکل (۲) نشان داده شده است، هر مهاجم به عنوان یک عامل در نظر گرفته شده و به‌صورت یک ماژول مدل می‌شود. دو ماژول I1 و I2 مدل مربوط به مهاجم‌هایی است که توصیف آنها با استفاده از زبان پروتکل امنیتی به صورت زیر است.

Spy1 =in pat{Φ1, Φ2}{Φ1, Φ2}. out{Φ1}. out{Φ2}
Spy2=in pat{Ψ1}{Ψ1} in pat{Ψ2}{Ψ2}. out{Ψ1, Ψ2}

مهاجم I1 پیامی را دریافت نموده، تجزیه می‌کند و سپس ارسال می‌نماید. همچنین مهاجم I2 دو پیام را دریافت نموده، ترکیب می‌کند و ارسال می‌نماید. پس از مدل‌سازی مهاجمین باید آنها را به مدل پروتکل اضافه کرد. ارتباط این مهاجمین با یکدیگر و با عامل‌های پروتکل با استفاده از میانگیر برقرار می‌شود. در ادامه نحوه‌ی ارزیابی وجود رخنه در پروتکل و همچنین نتایج بدست آمده بیان می‌شود.

۳-۵ ارزیابی پروتکل NS

در اینجا به بررسی پروتکل و احتمال وجود رخنه در آن پرداخته می‌شود. برای ارزیابی یک پروتکل دو حالت بسیار مهم است. یکی از این حالات زمانی است که پروتکل به اتمام رسیده است. برای بدست آوردن احتمال این حالت باید احتمال تمامی حالاتی که فرایند عامل‌های اصلی پروتکل و تبادلات پیام مربوط به پروتکل به پایان رسیده باشد را بدست آورد. برای بدست آوردن احتمال این حالت در این مدل از فرمول زیر به زبان PRISM استفاده شده است:

P=? [true U (s1=3) & (s2=3)]

حالت دیگری که بسیار دارای اهمیت است حالتی است که در پروتکل رخنه وجود دارد. احتمال تمامی حالاتی که پروتکل به‌صورت کامل به اتمام رسیده و یکی از مهاجمین نیز توانسته کار خود را به پایان رساند نشان دهنده‌ی وجود رخنه در پروتکل است. احتمال این حالت نیز با استفاده از فرمول زیر بدست می‌آید.

P=? [true U (s1=3) & (s2=3) & ((SI1=3) | (SI2=3) | (SI3=3) | (SI4=3))]

در اینجا پروتکل با وجود تعداد مهاجمین متفاوتی مورد بررسی قرار گرفته است. در جدول (۱) نتایج بدست آمده از بررسی پروتکل نشان داده شده که در آن احتمال به اتمام رسیدن پروتکل و وجود رخنه نیز بیان شده است.

جدول ۱: نتایج بررسی پروتکل NS

احتمال وجود رخنه	احتمال موفقیت مهاجم	احتمال اتمام پروتکل	تعداد مهاجم	تعداد گذر	تعداد حالات
۰/۶۹۱	۰/۲۵۰۵۸۱	۰/۳۶۲۲۴۵	۴	۱۷۱	۱۳۹
۰/۷۱۳	۰/۱۵۸۰۴۶	۰/۲۱۸۹۲۴	۶	۸۱۱	۶۲۲

۶- مطالعه موردی ۲: درستی‌یابی پروتکل TMN

در این بخش به بررسی موردی پروتکل TMN که پروتکل تبادل کلید رمزنگاری، در سیستم‌های ارتباط سیّار است پرداخته می‌شود. ابتدا پروتکل همانطور که در بخش قبلی بیان شد مدل‌سازی شده و سپس ارزیابی می‌شود.

۱-۶ مدل‌سازی پروتکل TMN

پروتکل TMN دارای سه عامل اصلی است. دو عامل A و B که قصد برقراری ارتباط با یکدیگر را دارند و عامل R که کارگزار است و در تبادل کلید نقش مهمی دارد. نحوه‌ی انجام این پروتکل به‌صورت زیر است:

1. $A \rightarrow R: B, K_R^{Pb}(K_{AR})$
2. $R \rightarrow B: A$
3. $B \rightarrow R: A, K_R^{Pb}(K_{AB})$
4. $R \rightarrow A: B, K_{AR}(K_{AB})$

این پروتکل دارای چهار گام است. در این قسمت همانطور که در بخش ۵ بیان شد ابتدا باید جزئیات پروتکل به‌صورت کامل به وسیله زبان پروتکل‌های امنیتی بیان و سپس با استفاده از PRISM هر کدام از عامل‌ها مدل شود. همانگونه که در بخش قبل نیز نشان داده شد هر کدام از عامل‌ها به صورت یک ماژول مدل می‌شود. این ماژول فعالیت‌های عامل را که با استفاده از زبان پروتکل امنیتی توصیف شده است شامل می‌شود. عامل‌ها با استفاده از میانگیر سراسری با یکدیگر در ارتباط هستند. مدل پروتکل TMN که شامل سه ماژول است در شکل ۳ نشان داده شده است.

```
probabilistic
global buf_key  : [100..110];
global buf_send :[1..10];
global buf_res :[1..10];
global id_agent : [1..10];
global buf_empt : bool init true;

const   int agent_num1= 1;
const int kaj=103;
module A1
        s1 : [0..2] init 0;
        aag_num : [1..10] ;
        []   (s1=0) & (buf_empt = true) -> 1 :(s1'=s1+1) & (buf_send'=agent_num1)& (buf_res' = 3)& (id_agent'=2)& (buf_key'=kaj)&(buf_empt' = false);
        []   (s1=1) & (buf_empt = false)& (buf_res=agent_num1)  -> 1     : (s1'=s1+1) & (buf_empt' = true);
endmodule

const int agent_num2= 2;
const int kab=105;
module B2
        s2 : [0..2] init 0;
        bag_num : [1..10];
        bag_num2 : [1..10];
        []   (s2=0) & (buf_empt = false) & (buf_res=agent_num2)  -> 1    : (s2'=s2+1)  & (bag_num'=id_agent)& (bag_num2'=buf_send) & (buf_empt' = true);
        []   (s2=1) & (buf_empt = true) -> 1     : (s2'=s2+1) & (buf_send'=agent_num2)& (buf_res' = bag_num2) & (id_agent'=bag_num)& (buf_key' =kab)&(buf_empt' = false);
endmodule

const int agent_num3= 3;
module R3
        s3 : [0..4] init 0;
        Rag_num1 : [1..10];
        Rag_num2 : [1..10];
        []   (s3=0) & (buf_empt = false)& (buf_res=agent_num3)  -> 1     : (s3'=s3+1) & (Rag_num1' = id_agent) & (Rag_num2'=buf_send) & (buf_empt' = true);
        []   (s3=1) & (buf_empt = true) -> 1     : (s3'=s3+1) & (buf_send'=agent_num3)& (buf_res' =Rag_num1) & (id_agent'=Rag_num2)&(buf_empt' = false);
        []   (s3=2) & (buf_empt = false)& (buf_res=agent_num3)  -> 1     : (s3'=s3+1) & (Rag_num1' = id_agent) & (Rag_num2'=buf_send) & (buf_empt' = true);
        []   (s3=3) & (buf_empt = true) -> 1     : (s3'=s3+1) & (buf_send'=agent_num3)& (buf_res' =Rag_num1) & (id_agent'=Rag_num2)&(buf_empt' = false);
endmodule
```

شکل ۳: مدل پروتکل TMN با استفاده از ابزار PRISM

۲-۶ ارزیابی پروتکل TMN

پس از مدل‌سازی پروتکل، مدل‌های مهاجم‌هایی که در بخش‌های قبلی بیان شده است به پروتکل اضافه می‌شود. در این مرحله مدل‌سازی به اتمام می‌رسد. در ادامه همانند مراحل ارائه شده در بخش قبل به بررسی احتمال حالات مورد نظر پرداخته و با استفاده از آنها پروتکل مورد ارزیابی قرار می‌گیرد.

احتمال حالات مورد نظر مانند به اتمام رسیدن پروتکل و وجود رخنه با استفاده از فرمول‌های زبان PRISM محاسبه می‌شود.

نتایج بررسی پروتکل TMN در جدول ۲ نشان داده شده است. یکی از مزیت‌های بسیار مهم این روش وجود امکان مقایسه دو پروتکل مختلف از نظر امنیتی است.

جدول ۲: نتایج بررسی پروتکل TMN

احتمال وجود رخنه	احتمال موفقیت مهاجم	احتمال اتمام پروتکل	تعداد مهاجم	تعداد گذر	تعداد حالات
۰/۴۹۸	۰/۳۳۲۸۱۳	۰/۶۶۸۲۲۳	۴	۸۵	۷۷

۷- نتیجه‌گیری

در این مقاله یک راهکار برای درستی‌یابی کمّی پروتکل‌های امنیتی ارائه شد. در این روش ابتدا پروتکل به وسیله زبان پروتکل‌های امنیتی که یک زبان صوری است توصیف می‌شود و به‌وسیله آن جزئیات یک پروتکل به‌صورت کامل و دقیق بیان می‌شود. سپس پروتکل با استفاده از ابزار PRISM مدل‌سازی می‌شود. در مرحله نهایی فضای حالت مدل بدست آمده، سپس بررسی احتمالی مدل صورت می‌گیرد و احتمال حالت‌های مورد نظر بدست می‌آید.

مزیت بسیار مهم این روش بررسی و درستی‌یابی کمّی پروتکل است. با ارزیابی کمّی پروتکل‌ها قابلیت بررسی دقیق یک پروتکل در برابر مهاجمین مختلف و همچنین امکان مقایسه چندین پروتکل با یکدیگر در مقابل مهاجمین وجود دارد که در روشهای قبلی امکان مقایسه چندین پروتکل در برابر حملات وجود نداشت.

به‌عنوان مطالعه موردی، در این مقاله با مدل‌سازی و بررسی مدل احتمالی دو پروتکل NS و TMN مشخص شد که هر دو پروتکل دارای رخنه هستند. ارزیابی کمّی و بررسی مدل احتمالی دارای این مزیت است که با استفاده از آن می‌توان دو پروتکل را در مقابل حملات امنیتی با یکدیگر مقایسه نمود. برای این منظور، با ارزیابی کمّی این دو پروتکل می‌توان نتیجه گرفت که علی‌رغم وجود رخنه در هر دو پروتکل، امنیت در پروتکل TMN نسبت به پروتکل NS بیشتر است.

در آینده قصد داریم از این روش برای مطالعه موردی پروتکل‌های مختلفی بهره بگیریم. همچنین قصد داریم که یک راهکار برای درستی‌یابی پروتکل‌های امنیتی بر مبنای یک روش مدل‌سازی سطح بالاتر، مانند شبکه‌های پتری سطح بالا، ارائه نمایم.

مراجع

[1] A. Armando, et al., "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," Computer Aided Verification, LNCS 3576, 2005.

[2] M. Olszewski and L. Cyra, "An Integrated Framework for Security Protocol Analysis," Proceedings of ASIACCS'08, March 18-20, Tokyo, Japan ACM, 2008.

[3] R. Bouroulet, R. Devillers, H. Klaudel, E. Pelz and F. Pommereau, "Modeling and Analysis of Security Protocols Using Role-Based Specifications and Petri Nets, " Proceedings of Petri Nets'08, LNCS 5062, Springer-Verlag Berlin Heidelberg, pp. 72–91, 2008.

[4] G. Baldi, A. Bracciali, G. Ferrari and E. Tuosto, "A Coordination-based Methodology for Security Protocol Verification," Proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP'05), Italy, Electronic Notes in Theoretical Computer Science, Volume 121, Elsevier, pp. 23-46, 2005.

[5] S. Andova, C. Cremers, K. Gjøsteen, S. Mauwd, S.F. Mjolsnes and S. Radomirovi, "A Framework for Compositional Verification of Security Protocols," Information and Computation, Volume 206, Issues 2-4, pp. 425-459, 2008.

[6] S. Wang and Y. Zhang, "A Logic Programming Based Framework for Security Protocol Verification," LNAI 4994, Springer-Verlag, pp. 638–643, 2008.

[7] F. Crazzolara and G. Winskel, "Events in Security protocol," Proceedings of ACM Conf. on Computer and Communications Security, ACM Press, 2001.

[8] E. Best, R. Devillers, M. Koutny, "Petri Net Algebra," EATCS Monographs on TCS, Springer, 2001.

[9] R. Bouroulet, H. Klaudel and E. Pelz, "Modelling and Verification of Authentication Using Enhanced Net Semantics of Security Protocol Language," Proceedingd of ACSD'06, IEEE Computer Society, pp. 179-188, 2006.

[10] M. Kwiatkowska, G. Norman and D. Parker, "PRISM: Probabilistic Symbolic Model Checker," TOOLS 2002, Lecture Notes in Computer Science, Volume 2324, Springer, pp 200-204, 2002.

[11] PRISM Web Site, www.cs.bham.ac.uk/~dxp/prism.

[12] PRISM Manual, Version 3.1, www.cs.bham.ac.uk/~dxp/prism

[13] M. Kwiatkowska, G. Norman and D. Parker, "Quantitative Analysis with the Probabilistic Model Checker PRISM," QAPL, 2005.