بررسی روش‌های به‌کارگیری کلپتوگرافی

هادی سلیمانی[1*] و فرخ لقا معظمی[2]

[1] استادیار پژوهشکده فضای مجازی، دانشگاه شهید بهشتی، تهران، ایران
h_soleimany@sbu.ac.ir

[2] استادیار پژوهشکده فضای مجازی، دانشگاه شهید بهشتی، تهران، ایران
f_moazemi@sbu.ac.ir

چکیده

با توجه به پیشرفت‌های گسترده در حوزه فناوری اطلاعات و ارتباطات، صورت مسأله‌ها و چالش‌های جدیدی در حوزه امنیت افتا مطرح شده است که جامعه علمی کشور را ملزم می‌کند تا با بررسی دقیق و موشکافانه ضمن فهم این موضوعات، تلاش کند با ارائه طرح‌های نوین به این نیازهای مهم و حیاتی پاسخ دهد. برهمین اساس هدف این مقاله، مطالعه و برجسته‌کردن یکی از این موضوعات اساسی است که طی چند سال اخیر در حوزه رمزنگاری کاربردی مطرح شده و کمتر در داخل کشور به آن پرداخته شده است. در این مقاله ما دسته‌ای خاص و کاربردی از روش‌های ایجاد درب‌های پشتی در سامانه‌های رمزنگاری را که به کلپتوگرافی[1] مشهور هستند، بررسی و ضمن مطالعه دقیق روش‌های کلپتوگرافی ارائه‌شده، کاربردهای احتمالی آنها را بررسی می‌کنیم. هدف از این مقاله روشن‌ساختن زوایای جدید کلپتوگرافی با بررسی مفاهیم نوین ارائه‌شده طی سالیان اخیر است.

واژگان کلیدی: کلپتوگرافی، درب پشتی، رمزنگاری مخرب

۱- مقدمه

کاربرد عملی کلپتوگرافی درحال حاضر شامل هدف‌گذاری سامانه‌های امنیتی کشورهای پیشرفته به‌منظور نظارت گسترده بر فعالیت‌های روزمره مردم در کشور مبدأ و یا کشورهای دیگر است [۱]. در این روش آژانس‌های امنیتی با ایجاد یک درب پشتی مخفی ویژه در یک الگوریتم رمزنگاری تلاش می‌کنند که به اطلاعات رمزشده دست پیدا کنند. در کلپتوگرافی موضوع اساسی این است که چگونه این درب‌های پشتی طراحی شوند که نخست قابل کشف نباشند و دوم‌این‌که در صورت کشف درب پشتی توسط شخص سوم، امکان استفاده از آن برای دیگران وجود نداشته باشد. این مبحث در مجامع علمی کمتر مورد توجه بود تا اینکه در سال‌های اخیر و پس از افشاگری‌های ادوارد اسنودن[2] مشخص شد که سرویس امنیت ملی آمریکا (NSA)[3] از این روش به‌منظور شنود گسترده اطلاعات استفاده کرده است. برپایه یافته‌های اخیر علمی مولد شبه‌تصادفی Dual-EC که توسط مؤسسه ملی فناوری و استانداردهای آمریکا (NIST)[4] به آن استاندارد پردازش اطلاعات فدرال (FIPS)[5] اعطا شده بود، دارای درب پشتی است[6] [۲] [۳] [۴] [۵]. این مسأله سبب شد که طی سالیان گذشته بار دیگر بحث کلپتوگرافی و روش‌های مقابله با آن به یکی از موضوعات مورد توجه جامعه رمزنگاری تبدیل شود [۶] [۷] [۸].

[1] Kleptography
[2] Edward Snowden
[3] National Security Agency
[4] National Institute of Standards and Technology
[5] Federal Information Processing Standards

[6] لازم به ذکر است که استاندارد FIPS بسیار معتبر بوده و توسط همه مؤسسات غیرنظامی و پیمانکاران دولتی مورد استفاده واقع می‌شود. این استاندارد همچنین توسط مؤسسه استانداردهای ملی آمریکا (ANSI)، مؤسسه مهندسان برق و الکترونیک (IEEE) و سازمان بین‌المللی استانداردسازی (ISO) به‌عنوان یک مرجع مورد استفاده قرار می‌گیرد.

*نویسندهٔ مسؤول

کلپتوگرافی به معنای سرقت اطلاعات به‌صورت امن و به‌طورکامل پنهان است. یک حمله کلپتوگرافی زمانی موفقیت‌آمیز است که غیرقابل کشف باشد. این نوع از حملات به‌طورعمومی در پیاده‌سازی جعبهٔ سیاه اولیه‌ها و یا پروتکل‌های رمزنگاری قابل استفاده هستند [۹]. بدین معنی که الگوریتم یا پروتکل رمزنگاری در یک دستگاه پیاده‌سازی شده و کاربر در حال استفاده از آن است؛ اما از جزئیات پیاده‌سازی و نحوه عملکرد دستگاه آگاه نیست. کلپتوگرافی‌ها به‌گونه‌ای طراحی می‌شوند که در خروجی ابزارهای رمزنگاری (جعبه سیاه‌ها) هیچ اثری از وجود حمله از خود به جای نمی‌گذارند. به عبارت دیگر تنها با مشاهدهٔ نحوهٔ کارکرد دستگاه و دیدن خروجی، نمی‌توان متوجه وجود کلپتوگرافی شد.

خطر استفاده از جعبهٔ سیاه رمزنگاری به‌طورتقریبی دو دهه است که به‌صورت عمومی مطرح شده است. اگرچه امروزه همچنان از جعبهٔ سیاه رمزنگاری به‌صورت گسترده‌ای در ابزارهای رمزنگاری استفاده می‌شود. درحقیقت بحث درباره خطر کلپتوگرافی از اواسط دهه ۱۹۹۰ میلادی آغاز شده بود اما امروزه توجه خیلی ویژه‌ای به این امر شده است. افشاگری‌های اخیر از تلاش‌های آژانس امنیت ملی آمریکا برای خراب‌کاری محصولات و استانداردهای رمزنگاری نشان می‌دهد حملات کلپتوگرافی یک خطر جدی و لازم است اقدامات متقابلی برای جلوگیری از این حملات انجام شود. درحقیقت آژانس امنیت ملی آمریکا درب‌های پشتی در پروتکل‌های رمزنگاری ایجاد می‌کند و با همکاری شرکت‌های سازنده محصولات رمزنگاری این درب‌های پشتی را روی محصولات قرار می‌دهد.

اطلاعاتی که عناصر مهاجم در رمزنگاری به‌دنبال ربودن آنها هستند، شامل کلید خصوصی در سامانه‌های رمزنگاری کلید عمومی، کلید امضا، کلید سامانه‌های رمزنگاری متقارن و اطلاعات محرمانه است. به عبارت دقیق‌تر کلپتوگرافی به راه‌های پژوهش برای به‌دست‌آوردن این اطلاعات به‌صورت امن و غیر قابل کشف اختصاص دارد. کلپتوگرافی درحقیقت یک مطالعه رسمی رمزنگاری از طراحی درب‌های پشتی در اولیه‌ها و پروتکل‌های رمزنگاری است که نخستین‌بار در سال ۱۹۹۶ در مجموعه مقالات منسجمی توسط آدام یوآنگ[1] و موتی یانگ[2] مطرح شد [۹] [۱۰] [۱۱] [۱۲].

ما در این مقاله ابتدا مروری خواهیم داشت بر روش‌هایی که تاکنون به‌منظور کسب اطلاعات محرمانه کاربران مورد استفاده قرار گرفته شده است؛ سپس در بخش دوم کلپتوگرافی را تعریف می‌کنیم. پس از آن روش‌های مختلف ارائه‌شده را به‌صورت مصداقی بررسی خواهیم کرد. درنهایت در بخش چهارم جمع‌بندی خود را ارائه می‌کنیم.

۲- رمزنگاری مخرب

رمزنگاری به‌عنوان پایه و اساس تأمین امنیت فضای تبادل اطلاعات و ارتباطات همواره مورد توجه نهادهای امنیتی کشورهای پیشرفته به‌منظور به‌کنترل‌درآوردن اطلاعات کشورهای دیگر بوده است. برهمین اساس آژانس‌های امنیتی کشورهای پیشرفته با توجه به مقتضیات زمان و زمینه‌های فناورانه راه‌کارهای متفاوتی را به‌منظور ورود به حریم خصوصی افراد و دسترسی به اطلاعات حساس دیگران در پیش گرفته‌اند. در این بخش به‌اختصار برخی از این روش‌ها را یاد می‌کنیم.

۲-۱- ممنوعیت و محدودیت به‌کارگیری رمزنگاری

رمزنگاری تا اوایل دهه هفتاد میلادی، علمی مخفی به حساب می‌آمد که تلاش می‌شد از گسترش آن جلوگیری شود. رشد سریع فناوری و افزایش درخواست‌های نهادهای تجاری (همچون بانک‌ها) سبب شد تا آژانس امنیت ملی آمریکا استفاده از رمزنگاری توسط نهادهای غیرنظامی را بپذیرد.

۲-۲- تضعیف الگوریتم‌های رمزنگاری

نخستین الگوریتم رمز استاندارد، DES است که به‌عنوان رمز استاندارد توسط NIST معرفی شد. این الگوریتم رمزنگاری توسط IBM طراحی شده بود. طول کلید DES پس از فشارهای آژانس امنیت ملی آمریکا به ۵۶ بیت کاهش پیدا کرد که این امر سبب شد از روزهای آغازین معرفی آن با اعتراضات متعددی مواجه شود. براساس مستندات حاضر طول کلید این الگوریتم توسط NSA کاهش پیدا کرده بود. اعمال محدودیت روی طول کلید بار دیگر در رمزنگاری به‌کاررفته در سامانه‌های مخابراتی موبایل تکرار شد. الگوریتم رمز A5/2 به کشورهای غیراروپایی و در حال توسعه تحمیل شد که مشخص شد نه‌تنها طول کلید نامناسب است، بلکه ضعف‌های

[1] Adam Young

[2] Moti Yung

بسیار چشم‌گیری نیز در آن وجود دارد. الگوریتم رمز A5/1 در کشورهای توسعه‌یافته به کار رفت، اما طول کلید با دخالت آژانس امنیتی انگلستان از ۱۲۸ بیت به ۴۸ بیت کاهش پیدا کرد. با واکنش آلمان غربی و فشار آن کشور طول کلید A5/1 درنهایت ۵۶ بیت در نظر گرفته شد. به‌عنوان مثالی دیگر می‌توان به الگوریتم رمز KeeLoq اشاره کرد که به‌صورت گسترده در سامانه‌های امنیتی خودروها و همچنین درب‌های خودکار پارکینگ‌ها استفاده شده است. طول کلید Keeloq تنها ۶۴ بیت در نظر گرفته شده است و هم اکنون با توجه به ضعف‌های موجود در الگوریتم این سامانهٔ رمزنگاری حملات عملی بر روی آن وجود دارد.

۲-۳- سپردن کلید[1]

روش دیگری که به کار رفته قراردادن کلید در یک مکان سوم است که در صورت نیاز و با توجه به درخواست یک مرجع فرادستی (همچون دادگاه) به آن مراجعه شده و متون رمزشده توسط افرادی خاص رمزگشایی می‌شوند. به‌طور مصداقی می‌توان به مورد زیر اشاره کرد:

آژانس امنیت ملی آمریکا الگوریتم رمز قالبی Skipjack را طراحی و بر روی یک تراشه خاص Clipper chip پیاده‌سازی کرد. تراشه بدین صورت کار می‌کرد که در کارخانه به هر تراشه یک کلید اختصاص داده می‌شد و به‌طور موازی کلید یادشده در یک پایگاه داده ذخیره می‌شد. این امر سبب ایجاد اعتراضاتی به آژانس ملی آمریکا شد. این تراشه در سال ۱۹۹۳ ارائه و در سال ۱۹۹۶ به کلی پروژه تولیدش متوقف شد.

۲-۴- استفاده از تروژان‌های سخت‌افزاری

به‌طورمعمول دولت‌ها، شرکت‌ها و افراد مختلف مجبور هستند که محصولات الکترونیکی خود را در مرحله پیاده‌سازی برون‌سپاری کنند. این امر می‌تواند به‌خاطر نداشتن فناوری پیاده‌سازی و یا به‌صرفه‌نبودن تأمین هزینه‌های اقتصادی راه‌اندازی کارخانه‌های مدارات مجتمع باشد. در هرصورت این حقیقت می‌تواند زمینه‌ساز سوءاستفاده صاحبان فناوری و کارخانه‌های تولید و پیاده‌سازی مدارات مجتمع شود؛ چون به آنها اجازه می‌دهد که با دست‌کاری نحوه پیاده‌سازی و جاگذاری تروجان‌های سخت‌افزاری، اطلاعات حساسی را از آنچه که پیاده‌سازی‌شده به‌صورت غیرمحسوس و مخفیانه (از دید سفارش‌دهنده) در اختیار خودشان قرار دهند. این مسأله کماکان جزء مشکلات امنیتی جدی است که با توجه به پیچیده‌شدن مدارات مجتمع و رشد خیره‌کننده فناوری در این حوزه چالش‌های بیشتری را برای همه و به‌خصوص کشورهایی که دارای صنعت پیاده‌سازی نیستند، ایجاد کرده است.

۲-۵- درب پشتی[2]

درب پشتی در یک سامانهٔ رمزنگاری روشی است که برای عبور از سامانهٔ احراز اصالت توسط یک کاربر غیرمجاز استفاده می‌شود تا (از طریق یک ارتباط دور) بتواند به متن آشکار و یا اطلاعاتی مخفی دست پیدا کند؛ درحالی‌که تلاش مهاجم برای کسب این اطلاعات از دید کاربر و دیگران مخفی باشد. درب پشتی می‌تواند بخشی مخفی از یک قسمت از برنامه یا برنامه‌ای مجزا باشد که بتواند سیستم را از طریق یک روت‌کیت[3] تحت کنترل قرار دهد. روت کیت‌ها اغلب در سطح سیستم‌عامل فعالیت کرده و با تغییراتی که در سیستم‌عامل یا منابع آن انجام می‌دهند، به مقاصد خود دست پیدا می‌کنند.

تعداد زیادی از کرم‌های رایانه‌ای[4] مانند Sobig و یا Mydoom یک درب پشتی بر روی رایانهٔ هدف نصب می‌کنند (به‌طورمعمول رایانه‌هایی که ویندوز دارند) که حمله‌کننده را قادر می‌کند از طریق رایانه آلوده‌شده رایانه‌های اسپم ارسال کند. در هنگام نصب و استفاده از محصولات شرکت‌های رایانه‌ای همیشه احتمال نصب غیرقانونی روت کیت وجود دارد. به‌عنوان نمونه مشخص شد که شرکت سونی در سال ۲۰۰۵ دو روت‌کیت را به بهانه جلوگیری از رونوشت‌های غیرمجاز بر روی دستگاه‌های بیش از ده میلیون کاربر نصب کرده است. نکته جالب این بود که یکی از این روت کیت‌ها در فهرست مجوز شرایط کاربر (EULA)[5] یاد شده بود و دیگری بدون یاد در فهرست نصب می‌شد.

۲-۶- کلپتوگرافی

زمانی که یک درب پشتی یا تروژان سخت‌افزاری روی یک محصول و یا پروتکل رمزنگاری گذاشته می‌شود، این خطر وجود دارد که افراد دیگر نیز آن را کشف و از آن بهره‌برداری کنند. برای جلوگیری از این خطر در حملات کلپتوگرافی از درب‌های پشتی نامتقارن استفاده می‌شود. درب‌های پشتی نامتقارن به‌گونه‌ای طراحی می‌شوند که فقط به حمله‌کننده یا

[1] Key escrow

[2] Back door
[3] Rootkit
[4] Computer worms
[5] End-user license agreement

به عبارت دیگر فقط کسی که اطلاعات خصوصی یا به عبارت دیگر کلید درب پشتی را دارد، امکان بهره‌برداری از این درب پشتی را می‌دهند. هدف از این مقاله بررسی این‌گونه از درب‌های پشتی است که در بخش بعدی به‌تفصیل مفاهیم اولیه و انواع آن را بررسی خواهیم کرد.

۳- مروری بر دسته‌بندی‌ها و مفاهیم به‌کار رفته

۳-۱- مفهوم کلپتوگرافی

مفهوم کلپتوگرافی به‌صورت رسمی و در یک چهارچوب علمی برای نخستین‌بار توسط آدام یوآنگ و موتی یانگ در کنفرانس CRYPTO سال ۱۹۹۶ ارائه شد [9]. ایده اصلی مطرح‌شده در مقاله به این شکل است که چگونه در مرحله پیاده‌سازی یک الگوریتم امن می‌توان الگوریتم را به‌نحوی بازتعریف کرد که دارای درب پشتی باشد؛ به‌گونه‌ای که نخست افراد استفاده‌کننده متوجه درب پشتی نشوند و دوم اگر وجود آن کشف شد، افراد دیگر قادر به استفاده از درب پشتی نباشند. آنها در این مقاله و کارهای بعدی خود مثال‌های متعددی در مورد رمزهای عموماً کلید عمومی ارائه کردند. به‌عنوان مثال نحوه خاصی از تولید کلیدخصوصی و عمومی سبب ایجاد درب پشتی در سامانهٔ رمزنگاری RSA می‌شود.

کلپتوگرافی به معنی سرقت اطلاعات به‌صورت امن و به‌طورکامل پنهان است. یک حمله کلپتوگرافی حمله‌ای است که در آن یک سازنده مخرب از یک رمزگذار نامتقارن استفاده می‌کند تا یک درب پشتی رمزنگاری ایجاد کند. در این حالت دراصطلاح، رمزنگاری بر علیه رمزنگاری استفاده می‌شود. یک درب پشتی یک کانال اضافه برای ایجاد ارتباط با دنیای بیرون رمزنگاری نیست؛ همچنین در آن احتیاجی به ارسال اطلاعات اضافه نیست؛ بلکه درب پشتی درون ارتباطات درنظرگرفته‌شده تعبیه شده است. بنابراین کلپتوگرافی زیرشاخه‌ای از crypto virology (کاربرد رمزنگاری در بدافزارها) است [۱۳]. اگرچه هدف یک حمله کلپتوگرافی فقط شکل کلی یک نرم‌افزار نیست؛ بلکه هدف سامانهٔ رمزنگاری است که استفاده می‌شود. مثال زیر یک حمله کلپتوگرافی را شرح می‌دهد:

یک جعبه سیاه یک جفت کلید نامتقارن تولید می‌کند، که یکی از آنها کلید عمومی و دیگری کلید خصوصی است. کلید خصوصی که برای از رمز خارج‌کردن و امضا استفاده می‌شود؛ به‌طور انحصاری باید داخل جعبهٔ سیاه قرار بگیرد تا از استفاده‌های نادرست و رونوشت برداشتن محافظت شود و کلید عمومی می‌تواند بدون هیچ نگرانی به‌صورت عمومی اعلام شود. همان‌طور که همه پذیرفته‌اند، هیچ کس نمی‌تواند از روی کلید عمومی کلید خصوصی را به‌دست آورد- یا شاید آنها می‌توانند؟

درحقیقت چنین اشتقاقی زمانی امکان‌پذیر است که روند تولید کلید به روش خاصی دست‌کاری شده باشد. یک درب پشتی رمزنگاری می‌تواند در روند ساخته‌شدن سامانهٔ رمزنگاری در شرکت سازنده روی ابزارها تعبیه شود و به حمله کننده این امکان را بدهد که بدون جلب توجه شخص سوم بتواند به کلید خصوصی دسترسی داشته باشد؛ یعنی در این حالت فقط شرکت سازنده و حمله‌کننده از وجود چنین درب پشتی مطلع هستند. از آنجا که تولید کلید عمومی آشکار نیست، هیچ ارتباط اضافه و یا غیر منتظره‌ای ایجاد نمی‌شود و هیچ خطایی در حین استفاده از سامانهٔ رمزنگاری رخ نمی‌دهد. تأثیر این حمله بسیار شدید است؛ چون با داشتن یک رونوشت از کلید خصوصی حمله‌کننده می‌تواند امضاهای جعلی تولید و داده‌های رمزشده را از رمز خارج کند؛ اما آیا واقعاً چنین کاری امکان‌پذیر است یا اینکه کلیدهای رمزنگاری در یک جعبهٔ سیاه مهر و موم‌شده تولید شده‌اند و هیچ دسترسی غیرمجازی به محتویات داخل آن داده نمی‌شود.

در یک نسخه ساده‌تر از این حمله می‌توان تولید اعداد تصادفی درون سامانه‌های رمزنگاری را دست‌کاری کرد. روند تولید کلید در سامانه‌های رمزنگاری از یک مولد شبه‌تصادفی استفاده می‌کند و سیدی[1] را تولید می‌کند که حمله‌کننده از آن مطلع است، به جای آنکه واقعا از یک تابع تصادفی استفاده کند. با دانستن این سید حمله‌کننده می‌تواند کلید را خارج از جعبهٔ سیاه تولید کند. برهمین اساس برخی پژوهش‌گران به این چالش پرداخته‌اند [۱۴] [۱۵]. ضمن آنکه دست‌کاری از این نوع تا زمانی‌که جلوی آن توسط یک سازوکار امنیتی گرفته نشود می‌تواند در یک مهندسی معکوس مشخص شود. چون سید در یک تولیدکننده اعداد شبه‌تصادفی درون کد منبع فیکس شده است با مهندسی معکوس می‌توان کلید خصوصی را خارج از جعبهٔ سیاه تولید کرد. از دید حمله‌کننده بسیار مطلوب است که یک دسترسی منحصر به‌فرد به سازوکار حمله وجود داشته باشد. حملات پیشرفته‌تر کلپتوگرافی در برابر مهندسی معکوس برای به‌دست‌آوردن چنین اطلاعاتی نیز محافظ هستند.

[1] seed

۳-۲- حمله SETUP

همان‌طور که پیش از این اشاره شد، کلپتوگرافی نخستین‌بار در سال ۱۹۹۶ در کنفرانس CRYPTO توسط آدام یوانگ و موتی یانگ مطرح شد. در این مقاله، آنها پژوهش‌گران را به فرصت‌های متعددی که در آن امکان حمله به سامانه‌های رمزنگاری دارای جعبهٔ سیاه وجود دارد، معطوف کردند. آنها برای نخستین‌بار مفهوم (SETUP) Secretly Embedded Trapdoor with Universal Protection را معرفی کردند و نحوه پیاده‌شدن این حمله را روی تولید کلید RSA تشریح کردند.

ویژگی متمایز حمله یادشده این است که فقط از طریق مهندسی معکوس می‌توان آن را شناسایی کرد و اگر شناسایی شود، شناسایی‌کننده نمی‌تواند از آن استفاده کند. مهندسی معکوس فقط می‌تواند کلید عمومی حمله‌کننده را به‌دست بیاورد و هیچ اطلاعاتی در مورد کلید خصوصی حمله‌کننده نمی‌تواند به‌دست آورد. از آنجا که این حمله بر اساس سامانه‌های رمزنگاری نامتقارن بنا نهاده شده است، می‌تواند از دید حمله‌کننده امن باشد.

۴- چند مثال کاربردی از کلپتوگرافی

بررسی تمامی حملات ارائه‌شده در این مقاله امکان‌پذیر نیست. ما در ادامه دو مثال مهم و کاربردی را بررسی خواهیم کرد:

۴-۱- ایجاد کلپتوگرافی در فرآیند تولید کلید RSA

۴-۱-۱- معرفی RSA و تولید کلیدهای آن

الگوریتم رمزنگاری RSA جزء پرکاربردترین سامانه‌های رمزنگاری کلید عمومی است. روش تولید کلیدهای عمومی و خصوصی در RSA به‌صورت زیر است:

۱- دو عدد نخست p و q به‌صورت تصادفی انتخاب و مقدار $n = p \times q$ محاسبه می‌شود.

۲- مقدار e به‌صورت تصادفی انتخاب می‌شود به‌گونه‌ای که $gcd(e, \phi(n)) = 1$.

۳- مقدار d به‌صورت $d = e^{-1} \bmod \phi(n)$ محاسبه می‌شود.

مقادیر $(n; e)$ به‌عنوان کلید عمومی و مقدار d به‌عنوان کلید خصوصی در نظر گرفته می‌شود. کلید خصوصی توسط کاربر به‌صورت محرمانه نگهداری و کلید عمومی اعلام عمومی می‌شود. کلید عمومی به‌منظور رمزنگاری و کلید خصوصی به‌منظور رمزگشایی به کار می‌رود.

بدیهی است که تولید کلید درعمل به‌شکل بالا توسط کاربر کار چالش‌برانگیزی است؛ بنابراین به‌طورعمومی کاربران از نرم‌افزارهایی استفاده می‌کنند که عملیات بالا را برای آنها انجام می‌دهند. این نرم‌افزارها به‌طورمعمول به‌صورت جعبهٔ سیاه هستند؛ بدین معنی که نحوه پیاده‌سازی تولیدکننده کلید RSA برای کاربر قابل مشاهده نیست. در چنین مواردی می‌توان نشان داد که با پیاده‌سازی مخرب تولید کلیدهای RSA، کاربر را می‌توان فریب داد. یعنی کاربر فکر کند که نرم‌افزار درحقیقت به‌صورت تصادفی در حال تولید کلیدها است و این درحالی است که نحوه پیاده‌سازی به‌گونه‌ای است که مهاجم با مشاهده کلید عمومی کاربر می‌تواند اطلاعاتی درخصوص کلید خصوص وی به‌دست آورد. در ادامه دو سناریوی[1] مختلف را بررسی می‌کنیم:

۴-۱-۲- ایجاد کلپتوگرافی (سناریوی نخست)

فرض کنیم که مهاجم دارای کلید عمومی $(N; E)$ و خصوصی D خاص خود است. هدف مهاجم این است که تولید کلید RSA را به‌نحوی پیاده‌سازی کند که کاربر با دیدن خروجی فکر کند که کلیدها به‌صورت تصادفی و صحیح تولید شده‌اند؛ اما درحقیقت کلید عمومی کاربر شامل اطلاعاتی درخصوص کلید خصوصی وی است. همچنین این درب پشتی باید به‌نحوی باشد که اگر کاربر متوجه وجود آن شد، نتواند از آن برای حمله به یک کاربر دیگر استفاده کند. بدین منظور مهاجم به‌صورت زیر پیاده‌سازی را انجام می‌دهد:

۱- دو عدد نخست p و q به‌صورت تصادفی انتخاب می‌شوند و مقدار $n = p \times q$ محاسبه می‌شود.

۲- مقدار e را به‌صورت $e = p^E \ (mod\ N)$ محاسبه می‌کند (به‌جای آنکه واقعا به‌صورت تصادفی آن را تولید کند!). درصورتی‌که $\gcd(e; \phi(n)) \neq 1$ به گام نخست برمی‌گردد و گام نخست را دوباره تکرار می‌کند.

۳- مقدار d را به‌صورت عادی تولید می‌کند. یعنی $d = e^{-1} \ mod\ \phi(n)$ محاسبه می‌شود.

درصورتی‌که کاربر از نرم‌افزاری استفاده کند که به‌صورت بالا کلید عمومی و خصوصی را تولید می‌کند، مهاجمی که درب پشتی بالا را در الگوریتم قرار داده است می‌تواند به‌راحتی از روی مقدار کلید عمومی کاربر (یعنی e) مقدار کلید خصوصی کاربر (یعنی d) را به‌صورت $p =$

[1] scenario

$e^D \ (mod\ N)$ محاسبه کند؛ بدون آنکه کاربر متوجه شود. نکته مهم آن است که اگر با مهندسی معکوس بتوان متوجه شد که نحوه تولید کلید RSA به‌صورت مخرب پیاده‌سازی شده است، نمی‌توان از این درب پشتی برای حمله به سایر کاربران استفاده کرد. دلیل این است که مقدار کلید خصوصی مهاجم که به‌صورت مخرب الگوریتم را پیاده‌سازی کرده (یعنی D) نامعلوم است.

روش بالا برخلاف مزیت‌هایی که برای مهاجم دارد، دارای این نقطه‌ضعف نیز هست که مقدار کلید عمومی تولیدشده برای کاربر با عدد نخست تولیدشده رابطه بسیار قوی دارد و چه بسا کاربر بتواند با برخی آزمون‌های آماری متوجه وجود درب پشتی شود. برهمین اساس، در ادامه یک سناریوی دیگر را بررسی می‌کنیم که این ضعف را ندارد.

۴-۱-۳- ایجاد کلپتوگرافی (سناریوی دوم)

فرض کنیم که مهاجم دارای کلید عمومی $(N; E)$ و خصوصی D خاص خود است. همان‌طور که پیش از این گفته شد، هدف مهاجم این است که تولید کلید RSA را به‌نحوی پیاده‌سازی کند که کاربر با دیدن خروجی فکر کند که کلیدها به‌صورت تصادفی و صحیح تولید شده‌اند، اما درحقیقت کلید عمومی کاربر شامل اطلاعاتی درخصوص کلید خصوصی وی باشد. مشابه سناریوی نخست، این درب پشتی باید به‌نحوی باشد که اگر کاربر متوجه وجود آن شد، نتواند از آن برای حمله به یک کاربر دیگر استفاده کند. بدین منظور مهاجم به‌صورت زیر پیاده‌سازی را انجام می‌دهد:

۱- مقدار e به‌صورت تصادفی تولید می‌شود (دقت کنید که نظم تولید کلیدها متفاوت است!).

۲- مقدار s به‌صورت تصادفی تولید می‌شود (توجه شود که در حالت عادی چنین مقداری نباید تولید شود؛ بلکه در این پیاده‌سازی به منظور ایجاد یک درب پشتی این مقدار تولید می‌شود).

۳- مقدار p به‌صورت $p = H(s)$ تولید می‌شود که در آن H یک تابع درهم‌ساز است که ورودی دلخواه را به k بیت منتقل می‌کند. اگر p یک عدد مرکب بود و یا آنکه $\gcd(p-1; e) \neq 1$ در این صورت به گام یک رفته و دوباره این گام تکرار می‌شود.

۴- مقدار t به‌صورت تصادفی تولید می‌شود (دوباره توجه شود که در حالت عادی چنین مقداری نباید تولید شود. بلکه در این پیاده‌سازی به‌منظور ایجاد یک درب پشتی این مقدار تولید می‌شود).

۵- مقدار $c = s^E \ (mod\ N)$ محاسبه می‌شود. به عبارت دیگر مقدار s توسط کلید عمومی مهاجم یعنی $(N; E)$ رمز می‌شود.

۶- مقادیر $(q; r)$ به‌نحوی محاسبه می‌شود که $(c||t) = pq + r$ اگر q یک عدد مرکب بود و یا درصورتی‌که $\gcd(q-1; e) \neq 1$ در این صورت به گام یک رفته و دوباره گام ۱ تکرار می‌شود.

۷- مقدار $(n = p.q; e)$ به‌عنوان کلید عمومی و مقدار $d = e^{-1}\ mod\ \phi(n)$ به‌عنوان کلید خصوصی به کاربر داده می‌شود.

می‌توان نشان داد که مهاجمی که به‌صورت خراب‌کارانه درب پشتی بالا را در درون نرم‌افزار قرار داده است، می‌تواند به‌راحتی اقدام به بازیابی کلید کند. به‌منظور بازیابی مهاجم به‌صورت زیر اقدام می‌کند:

۱- از آنجاکه مقدار کلید کاربر به‌صورت عمومی اعلام می‌شود، مهاجم از مقادیر $(n; e)$ آگاهی دارد. برهمین اساس مهاجم k بیت نخست مقدار n را برابر u قرار می‌دهد.

۲- مقادیر c_1 و c_2 را به‌صورت $c_1 = u$ و $c_2 = u + 1$ محاسبه می‌کند.

۳- مقادیر c_1 و c_2 را با کلید خصوص خود رمزگشائی کرده و معادل رمزگشائی‌شده آنها را به‌ترتیب تحت عنوان s_1 و s_2 ذخیره می‌کند.

۴- حال مقادیر p_1 و p_2 را به‌صورت $p_1 = H(s_1)$ و $p_2 = H(s_2)$ محاسبه می‌کند. یکی از مقادیر p_1 و p_2 عدد n را می‌شمارد.

همان‌طورکه مشاهده می‌شود، مهاجم می‌تواند به‌راحتی به مقادیر اعداد نخست دست یابد و امنیت سیستم را به طور کامل زیر سؤال ببرد. مشابه آنچه مشاهده شد، نکته مهم آن است که اگر با مهندسی معکوس بتوان متوجه شد که نحوه تولید کلید RSA به‌صورت مخرب پیاده‌سازی شده است، نمی‌توان از این درب پشتی برای حمله به سایر کاربران استفاده کرد. دلیل این است که مقدار کلید خصوصی مهاجم که به‌صورت مخرب الگوریتم را پیاده‌سازی کرده نامعلوم است.

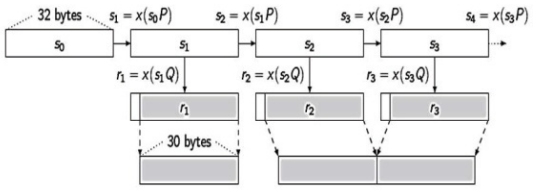
۴-۲- کلپتوگرافی در استاندارد Dual_EC_DRBG

الگوریتم Dual_EC_DRBG، یک مولد شبه‌تصادفی در رمزنگاری است که توسط NIST تحت شماره NIST SP 800-90A به‌عنوان استاندارد به جامعه معرفی شد. این مولد شبه‌تصادفی به‌صورت گسترده برای تولید نانس و کلید

خصوصی مورد نیاز در پروتکل‌های بسیار پرکاربرد به کار می‌رفت. در سال‌های اخیر مشخص شده که آژانس امنیت آمریکا یک درب پشتی در این الگوریتم قرار داده است. درب پشتی به‌کاررفته در الگوریتم Dual_EC_DRBG از نوع کلپتوگرافی است. نکته مهم این است که برخلاف انواع کلپتوگرافی معرفی‌شده در مقالات، این درب پشتی در طراحی و نه پیاده‌سازی الگوریتم رمزنگاری تعبیه شده است. ما در ادامه ابتدا الگوریتم Dual_EC_DRBG را معرفی و سپس نحوه ایجاد درب پشتی در این الگوریتم را بررسی می‌کنیم.

۴-۲-۱- معرفی الگوریتم Dual_EC_DRBG

مولد شبه‌تصادفی Dual_EC_DRBG یک ساختار عمومی و انعطاف‌پذیر دارد. این مولد از یک خم بیضوی و محاسبات روی خم برای تولید دنباله بیت تصادفی استفاده می‌کند. روش تولید دنباله بیت تصادفی به‌شرح زیر است (شکل ۱):



(شکل-۱): ساختار مولد شبه‌تصادفی Dual_EC_DRBG

مقداردهی اولیه:

ابتدا یک خم بیضوی انتخاب و بر روی خم بیضوی دو نقطه P و Q به صورت تصادفی انتخاب می‌شود. الگوریتم یک ورودی ۳۲ بایتی را به‌عنوان مقدار اولیه و یا سید دریافت می‌کند که آن را با s_0 نمایش می‌دهیم (این مقدار توسط کاربر تعیین می‌شود)؛ سپس نقطه P در s_0 بر روی خم بیضوی ضرب می‌شود و مقدار مولفه افقی نقطه جدید به عنوان ۳۲ بایت جدید برای مقدار میانی که با s_1 نمایش می‌دهیم، در نظر گرفته می‌شود.

تولید نانس:

با استفاده از مقدار میانی s_1 مقدار ۳۲ بایتی جدید تولید می‌شود. نحوه تولید این مقدار جدید بدین گونه است که نقطه Q در s_1 بر روی خم بیضوی ضرب می‌شود و مقدار مؤلفه افقی نقطه جدید به‌عنوان ۳۲ بایت جدید که با r_1 نمایش می‌دهیم، در نظر گرفته می‌شود. دو بایت ابتدایی دور ریخته می‌شود و ۳۰ بایت باقی‌مانده به‌عنوان نانس در خروجی ظاهر می‌شود.

گفتنی است که مقدار نانسی که توسط کاربر به کار می‌رود اعلام عمومی می‌شود.

تولید کلید تصادفی:

براساس تعداد بیت‌های تصادفی مورد نیاز کاربر، فرآیند زیر تکرار می‌شود (در هربار تکرار ۳۲ بایت تولید می‌شود): نقطه P در s_i بر روی خم بیضوی ضرب می‌شود و مقدار مؤلفه افقی نقطه جدید به‌عنوان ۳۲ بایت جدید برای مقدار میانی که با s_{i+1} نمایش می‌دهیم، در نظر گرفته می‌شود؛ سپس نقطه Q در s_{i+1} بر روی خم بیضوی ضرب می‌شود و مقدار مؤلفه افقی نقطه جدید به‌عنوان ۳۲ بایت جدید که با r_{i+1} نمایش می‌دهیم، در نظر گرفته می‌شود. دو بایت ابتدایی دور ریخته و سی بایت مابقیه به‌عنوان دنباله تصادفی در خروجی ظاهر می‌شود.

۴-۲-۲- امنیت الگوریتم Dual_EC_DRBG

امنیت الگوریتم Dual_EC_DRBG براساس سختی مسأله لگاریتم گسسته بر روی خم‌های بیضوی است. بدین معنا که با داشتن یک خم بیضوی E و عضو تولیدکننده G و عنصر T مسأله لگاریتم گسسته، یافتن عدد صحیح $1 \le d \le \#E$ است به‌طوری‌که $G + G + \cdots + G = [d]G = T$ در زمان چندجمله‌ای و به‌صورت ساده امکان‌پذیر نیست. درصورتی‌که مسأله لگاریتم گسسته بر روی خم‌های بیضوی یک مسأله سخت باشد، مهاجم با دیدن نانس (یعنی سی بایت از r_1) نمی‌تواند مقدار میانی s_1 را به‌دست آورد، چون این امر مستلزم حل مسأله لگاریتم گسسته بر روی خم بیضوی به‌کاررفته است. درنتیجه برخلاف دیدن نانس نمی‌تواند به مقدار کلید خصوصی کاربر دسترسی پیدا کند.

۴-۲-۳- کلپتوگرافی موجود در Dual_EC_DRBG

در این بخش روش تعبیه یک درب پشتی از نوع کلپتوگرافی در Dual_EC_DRBG را بررسی می‌کنیم. همان‌طورکه گفته شد، نقاط P و Q باید به‌صورت *تصادفی* انتخاب شوند. در فرآیند استانداردسازی، آژانس امنیت ملی آمریکا با دخالت خود (که بعدها مشخص شد) به سازمان فناوری و استاندارد تحمل کرد که نقاط P و Q روی خم به‌صورت تصادفی انتخاب نشود و براساس استاندارد دو نقطه خاص برای نقاط P و Q در نظر گرفته شد. این نقاط به‌صورت هوشمندانه انتخاب شده‌اند که در ادامه روش انتخاب آنها و همچنین نحوه به‌کارگیری آنها را به‌عنوان درب پشتی توضیح خواهیم داد:

۱. **نحوه انتخاب نقاط:** به جای انتخاب نقاط P و Q روی خم به‌صورت تصادفی، ابتدا یک عدد تصادفی مانند d انتخاب و نقطه Q نیز به‌صورت تصادفی روی خم انتخاب می‌شود؛ سپس نقطه P به شکل $P = dQ$ تولید می‌شود (نه به‌صورت تصادفی) و ادعا می‌شود که نقطه P به شکل تصادفی تولید شده است.

۲. **نحوه به‌کارگیری درب پشتی:** اگر کسی مقدار d را داشته باشد، می تواند از روی مقدار r_1 مقدار s_2 را به‌دست آورد:

$$s_2 = x(s_1 P) = x(s_1 dQ) = x(dr_1)$$

گفتنی است که مقدار r_1 را می توان به‌راحتی حدس زد. چون سی بایت آن به‌عنوان نانس (nonce) اعلام عمومی می شود. دو بایت باقیمانده را می توان حدس زد و به‌ازای تمامی مقادیر ممکن یعنی 2^{16} حالت، با سعی و خطا کلید مخفی تولیدشده را پیدا کرد.

نکته مهم این است که درب پشتی تعبیه‌شده از نوع کلیپتوگرافی است؛ بدین معنا که کسی به جز آژانس امنیت ملی آمریکا نمی‌تواند از این ضعف استفاده کند (حتی هم‌اکنون که این درب پشتی کشف شده است!). دلیل این امر این است که برای استفاده از ضعف ایجاد در استاندارد یادشده، باید مقدار d در دسترس باشد و این در حالی است که این مقدار محرمانه است و کسی آن را ندارد. همچنین کسی نمی‌تواند با درنظرگرفتن نقاط P و Q مقدار d را به‌دست آورد چون سختی حل این مسأله برابر با سختی حل مسأله لگاریتم گسسته بر روی خم‌های بیضوی است.

۵- جمع‌بندی

در این مقاله مفاهیم مرتبط با کلیپتوگرافی را معرفی کردیم. همان‌گونه که یاد شد، هدف از کلیپتوگرافی ایجاد یک درب پشتی قوی در سامانه‌های رمزنگاری است که حمله‌کننده به اطلاعات مخفی مورد نیاز دسترسی پیدا کند؛ به‌گونه‌ای که فقط برای طراح حمله یا به عبارت دیگر کسی که کلید خصوصی درب پشتی را دارد در پیاده‌سازی‌های جعبه سیاه قابل کشف باشد. پس از آن به‌صورت مصداقی دو نمونه مهم از کلیپتوگرافی‌های شناخته‌شده را با یاد جزئیات بررسی و مشاهده کردیم که خطر این‌گونه از درب‌های پشتی بسیار جدی است. برهمین اساس پژوهش‌های گسترده در این حوزه می‌تواند در تأمین امنیت فضای سایبری تأثیر به‌سزایی داشته باشد.

۶- مراجع

[1] A. L. Young and M. Yung, Malicious cryptography - exposing cryptovirology, Wiley,2004.

[2] S. Checkoway, R. Niederhagen , A. Everspaugh, M. Green, T. Lange, T. Ristenpart, D. J. Bernstein, J. Maskiewicz, H. Shacham and M. Fredrikson, "On the Practical Exploitability of Dual EC in TLS Implementations," in *23rd USENIX Security Symposium*, 2014.

[3] S. Checkoway, J. Maskiewicz, C. Garman , J. Fried, S. Cohney , M. Green, N. Heninger , R. P. Weinmann, E. Rescorla and H. Shacham, "A Systematic Analysis of the Juniper Dual EC Incident," in *SIGSAC*, 2016

[4] J. Woodage and D. Shumow, "An Analysis of the NIST SP 800-90A Standard," IACR Cryptology ePrint Archive, 2018.

[5] K. Q. Ye, M. Green, N. Sanguansin, L. Beringer, A. Petcher and A. W. Appel, "Verified Correctness and Security of mbedTLS HMAC-DRBG," in *CCS* , 2017.

[6] Q. Tang, M. Yung and H. S. Zhou, "Generic Semantic Security against a Kleptographic Adversary," in *SIGSAC*, 2017.

[7] J. P. Degabriele, K. G. Paterson, J. C. N. Schuldt and J. Woodage, "Backdoors in Pseudorandom Number Generators: Possibility and ImpossibilityResults," in *CRYPTO 2016*, 2016.

[8] J. P. Degabriele , P. Farshim and B. Poettering, "A More Cautious Approach to Security Against Mass Surveillance," in *FSE* , 2015.

[9] M. Yung, "The Dark Side of "Black-Box" Cryptography, or: Should We Trust Capstone?," in *CRYPTO*, 1996.

[10] A. L. Young and M. Yung, "Kleptography: Using Cryptography Against Cryptography," in *EUROCRYPT* , 1997.

[11] A. L. Young and M. Yung, "The Prevalence of Kleptographic Attacks on Discrete-Log Based Cryptosystems," in *CRYPTO,* 1997.

[12] A. L. Young and M. Yung, "RSA-Based Auto-recoverable Cryptosystems," in *PKC 2000*.

[13] A. L. Young and M. Yung, "Cryptovirology: Extortion-Based Security Threats and Countermeasures," in *IEEE Symposium on Security and Privacy*, 1996.

[14] P. Soni and S. Tessaro, "Public-Seed Pseudorandom Permutations," in *EUROCRYPT 2017*, 2017.

[15] B. Auerbach , M. Bellare and E. Kiltz, "Public-Key Encryption Resistant to Parameter Subversion and Its Realization from Efficiently-Embeddable Groups," in *PKC 2018*, 2018.

هادی سلیمانی کارشناسی را در رشته مخابرات از دانشگاه علم وصنعت ایران در سال ۱۳۸۷ اخذ کرد. تحصیلات خود در مقطع کارشناسی‌ارشد در دانشگاه امام حسین (ع) در گرایش مخابرات رمز در سال ۱۳۸۹ و سپس در مقطع دکترا در دانشکده علوم کامپیوتر دانشگاه آلتو فنلاند درسال ۱۳۹۴ به پایان رساند. همچنین طی یک دورهٔ کوتاه‌مدت پسادکترا در گروه رمزنگاری دانشگاه DTU دانمارک درخصوص تحلیل و طراحی رمزهای قالبی نوین مشغول به پژوهش شد. وی هم‌اکنون، ضمن همکاری با پژوهشکده‌ها و مراکز پژوهشی مختلف در حوزه رمزنگاری و امنیت اطلاعات، به‌عنوان استادیار گروه امنیت شبکه و رمزنگاری پژوهشکده فضای مجازی دانشگاه شهید بهشتی مشغول به کار است. زمینه‌های پژوهشی مورد علاقهٔ ایشان تحلیل و طراحی اولیه‌های رمزنگاری متقارن و همچنین پیاده‌سازی امن است.

فرخ لقا معظمی در سال ۱۳۸۳ در مقطع کارشناسی ریاضی از دانشگاه الزهرا فارغ‌التحصیل شد و تحصیلات خود در مقطع کارشناسی‌ارشد را در دانشگاه صنعتی شریف در سال ۱۳۸۵ و سپس در مقطع دکترا در دانشکده ریاضی دانشگاه الزهرا درسال ۱۳۹۲ به پایان رساند. همچنین طی یک دوره پسا دکتری در دانشکده ریاضی دانشگاه شریف درخصوص تحلیل سامانه‌های رمزنگاری مشغول به پژوهش شد. وی هم‌اکنون، به‌عنوان استادیار گروه امنیت شبکه و رمزنگاری پژوهشکده فضای مجازی دانشگاه شهید بهشتی مشغول به کار است. زمینه‌های پژوهشی مورد علاقه ایشان پروتکل‌های رمزنگاری و سامانه‌های رمزنگاری مشبکه مبنا است.