جواب سوال ۱

جواب سوال ۲

بخش اول

وقتی یک پیام m ابتدا توسط الگوریتم فشرده‌سازی $\mathrm{C(m)}$ فشرده می‌شود و سپس با استفاده از رمزنگاری بلوکی با اندازه بلوک ۸ بیت پد شده و رمزگذاری می‌گردد، طول نهایی پیام رمزگذاری شده مستقیماً به طول پیام فشرده شده $\mathrm{C(m)}$ بستگی دارد. مهاجم که تنها تعداد بلوک‌های ارسال شده را مشاهده می‌کند، می‌تواند اطلاعاتی درباره طول $\mathrm{C(m)}$ به دست آورد. از آنجایی که طول $\mathrm{C(m)}$ نشان‌دهنده تعداد و طول رشته‌های ۱ متوالی در پیام اصلی m است، مهاجم می‌تواند استنتاج‌هایی درباره الگوی تکراری در m انجام دهد.

با یک مثال درباره‌ی این مورد بیشتر توضیح می‌دهم. **مثال:**

پیام $m_۱$: ۱۱۱۱۱۱۱۱۱۱۰۱۱۱۱۱۰۱۰۱۱۱

$C(m_۱) =$ ۱۰۱۰۰۱۰۱۰۰۰۱۰۰۱۱

طول $C(m_۱) =$ ۱۶ بیت $\rightarrow$ ۲ بلوک ۸ بیتی

پیام $m_۲$: ۱۱۱۱۱۰۱۱۱۱۱۰۱۱۱۱۱۰۱۱۱۱

$C(m_۲)$ ممکن است طول بیشتری داشته باشد (فرضاً ۲۴ بیت) $\rightarrow$ ۳ بلوک ۸ بیتی

در این حالت، اگر مهاجم مشاهده کند که دو بلوک ارسال شده‌اند، احتمال می‌دهد پیام $m_۱$ با الگوی تکراری مناسب فشرده شده است. اگر سه بلوک ارسال شود، ممکن است پیام $m_۲$ با الگوی کمتر تکراری فشرده شده باشد.

مهاجم می‌تواند از تفاوت‌های فشرده‌سازی بین دو پیام با طول اصلی برابر استفاده کند تا تشخیص دهد کدام پیام رمزگذاری شده است. اگر دو پیام $m_۱$ و $m_۲$ دارای طول اصلی یکسان باشند اما الگوهای متفاوتی از رشته‌های ۱ متوالی داشته باشند، فشرده‌سازی آن‌ها نیز متفاوت خواهد بود. پیام با الگوی فشرده‌تر (بیشتر رشته‌های تکراری و طولانی‌تر ۱ها) منجر به $C(m)$ کوتاه‌تر و در نتیجه تعداد بلوک‌های کمتر خواهد شد.

با یک مثال درباره‌ی این مورد بیشتر توضیح می‌دهم. **مثال:**

پیام $m_۱$: ۱۱۱۱۱۱۱۱۱۱۰۱۱۱۱۱۰۱۰۱۱۱

$C(m_۱) =$ ۱۰۱۰۰۱۰۱۰۰۰۱۰۰۱۱

تعداد بلوک‌های رمزگذاری شده: ۲ بلوک

پیام $m_۲$: ۱۱۱۱۱۰۱۱۱۱۱۰۱۱۱۱۱۰۱۱۱۱

$C(m_۲) =$ ۱۰۱۰۱۰۱۰۰۱۰۱۰۱۰۱

تعداد بلوک‌های رمزگذاری شده: ۳ بلوک

در این حالت، اگر مهاجم مشاهده کند که تعداد بلوک‌های رمزگذاری شده ۲ بلوک است، نتیجه می‌گیرد که پیام $m_۱$ رمزگذاری شده است. اگر ۳ بلوک مشاهده کند، نتیجه می‌گیرد که پیام $m_۲$ رمزگذاری شده است.

این روش به مهاجم اجازه می‌دهد تا با تحلیل طول پیام‌های رمزگذاری شده و مقایسه آن‌ها با الگوهای مختلف فشرده‌سازی، اطلاعاتی درباره محتوای پیام اصلی بدست آورد. این نوع حمله نشان‌دهنده آسیب‌پذیری‌های احتمالی در ترکیب فشرده‌سازی و رمزنگاری است که پس از حملات CRIME در پروتکل‌های جدید مانند TLS 1.3 حذف شده‌اند تا از چنین تهدیداتی جلوگیری شود.

بخش دوم

در این بخش، مهاجم توانایی افزودن بیت‌های دلخواه به ابتدای پیام‌های اصلی را دارد و با انجام این کار، می‌خواهد تعداد ۱های رشته اول پیام m را کشف کند. برای این منظور، مهاجم از ویژگی‌های الگوریتم فشرده‌سازی بی‌اتلاف $\mathrm{C(m)}$ استفاده می‌کند که تعداد ۱ های هر رشته را به صورت یک مقدار ۴ بیتی نمایش می‌دهد و محدودیت‌هایی در تعداد ۱ های متوالی دارد.

روش حمله

افزودن بیت‌های کنترل‌شده به ابتدای پیام

مهاجم می‌تواند به تعداد دلخواه بیت‌های ۱ یا ۰ را به ابتدای پیام m اضافه کند. هدف این است که با تنظیم تعداد بیت‌های ۱ اضافه شده، تغییراتی در فشرده‌سازی رخ دهد که به کمک آن بتوان تعداد ۱های رشته اول m را تعیین کرد.

تعیین نقطه تغییر در فشرده‌سازی

از آنجا که الگوریتم فشرده‌سازی $\mathrm{C(m)}$ فقط تا پانزده ۱ متوالی را به درستی فشرده می‌کند و اگر تعداد ۱ ها بیشتر از ۱۵ شود، فقط پانزده ۱ اول را فشرده می‌کند و بقیه را نادیده می‌گیرد، مهاجم می‌تواند با افزایش تدریجی تعداد ۱ های اضافه شده، نقطه‌ای را پیدا کند که در آن تعداد ۱ های متوالی به ۱۵ برسد.

تحلیل تعداد بلوک‌های رمزگذاری شده

پس از افزودن بیت‌های کنترل‌شده و فشرده‌سازی پیام، پیام فشرده شده به بلوک‌های ۸ بیتی تقسیم می‌شود و رمزگذاری می‌گردد. تعداد بلوک‌های رمزگذاری شده به طور مستقیم به طول پیام فشرده شده بستگی دارد. مهاجم با مشاهده تعداد بلوک‌ها می‌تواند تشخیص دهد که آیا تعداد ۱ های متوالی افزوده شده به m به ۱۵ رسیده یا نه.

گام‌های دقیق حمله

ابتدا یک سری فرضیات اولیه داریم.

- فرض کنید پیام اصلی m با k عدد ۱ در ابتدای آن شروع می‌شود.
- مهاجم قصد دارد تعداد ۱های k را تعیین کند.

افزودن بیت‌های ۱ کنترل‌شده مهاجم به ترتیب از $t = ۰$ تا $t = ۱۵$ بیت ۱ به ابتدای پیام m اضافه می‌کند، یعنی پیام‌های زیر را ارسال می‌کند:

- $t = ۰ : m$
- $t = ۱ : ۱m$
- $t = ۲ : ۱۱m$
- ...
- $t = ۱۵ : ۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱m$

مشاهده تعداد بلوک‌های رمزگذاری شده برای هر مقدار t ، مهاجم تعداد بلوک‌های رمزگذاری شده را مشاهده می‌کند.

تجزیه و تحلیل تغییرات

- اگر $t + k \leq$ ۱۵ باشد: تعداد ۱های متوالی در پیام فشرده شده برابر با $t + k$ است و فشرده‌سازی به طور بهینه انجام می‌شود.
- اگر $t + k >$ ۱۵ باشد: فشرده‌سازی فقط پانزده ۱ اول را در نظر می‌گیرد و بقیه نادیده می‌گیرد، که منجر به افزایش طول پیام فشرده شده و در نتیجه افزایش تعداد بلوک‌های رمزگذاری شده می‌شود.

تعیین تعداد ۱ های اولیه (k)

مهاجم به دنبال کمترین مقداری از t است که در آن $t + k >$ ۱۵ می‌شود. این نقطه تغییر نشان‌دهنده آن است که $k = ۱۶ - t$. بنابراین، با یافتن مقداری از t که در آن تعداد بلوک‌های رمزگذاری شده افزایش می‌یابد، مهاجم می‌تواند مقدار k را محاسبه کند.

حالا برای مثال بدین شکل داریم:

فرض کنید پیام اصلی m با $k =$ ۱۰ عدد ۱ شروع می‌شود:

افزودن $t =$ ۰ بیت ۱

- پیام تغییر یافته: m (۱۰ عدد ۱ متوالی)
- تعداد ۱های فشرده شده: ۱۰
- تعداد بلوک‌های رمزگذاری شده: به عنوان مثال، ۲ بلوک

افزودن $t =$ ۵ بیت ۱

- پیام تغییر یافته: m۱۱۱۱۱ (۵ + ۱۰ = ۱۵ عدد ۱ متوالی)
- تعداد ۱های فشرده شده: ۱۵
- تعداد بلوک‌های رمزگذاری شده: همچنان ۲ بلوک

افزودن $t =$ ۶ بیت ۱

- پیام تغییر یافته: m۱۱۱۱۱۱ (۶ + ۱۰ = ۱۶ عدد ۱ متوالی)
- تعداد ۱های فشرده شده: فقط ۱۵ عدد ۱ اول
- تعداد بلوک‌های رمزگذاری شده: افزایش به ۳ بلوک

با مشاهده افزایش تعداد بلوک‌ها زمانی که $t =$ ۶ است، مهاجم می‌داند که $t+k =$ ۱۶ از این رو $k =$ ۱۶$-$۶ $=$ ۱۰

نتیجه‌ی بخش دوم

با استفاده از افزودن بیت‌های کنترل‌شده به ابتدای پیام و تحلیل تغییرات در تعداد بلوک‌های رمزگذاری شده، مهاجم قادر است تعداد ۱ های متوالی در ابتدای پیام اصلی m را به طور دقیق تعیین کند. این روش نشان‌دهنده آسیب‌پذیری ترکیب فشرده‌سازی و رمزگذاری در پروتکل‌های امنیتی مشابه TLS 1.2 است که منجر به حملاتی مانند CRIME شده است. به همین دلیل، در نسخه‌های جدیدتر مانند TLS 1.3 ، قابلیت فشرده‌سازی حذف شده تا از چنین حملاتی جلوگیری شود.

بخش سوم

حمله CRIME (Compression Ratio Info-leak Made Easy) یک حمله جانبی (side-channel attack) علیه پروتکل‌های امنیتی مانند TLS است که از ویژگی فشرده‌سازی داده‌ها سوء استفاده می‌کند. در این حمله، مهاجم با استفاده از قابلیت فشرده‌سازی و مشاهده تغییرات در طول داده‌های رمزگذاری‌شده، می‌تواند اطلاعات حساس مانند کوکی‌های کاربر را به‌دست آورد.

شرایط لازم برای موفقیت حمله CRIME

فعال بودن فشرده‌سازی در ارتباط TLS

پروتکل TLS باید قابلیت فشرده‌سازی داده‌ها را فعال کرده باشد. در نسخه‌های قدیمی‌تر مانند TLS 1.2 این ویژگی وجود دارد، در حالی که در نسخه‌های جدیدتر مانند TLS 1.3 به منظور جلوگیری از حملات مشابه CRIME، فشرده‌سازی حذف شده است.

توانایی مهاجم در تزریق داده به جریان اطلاعات

مهاجم باید بتواند داده‌هایی را به جریان اطلاعاتی که بین کاربر و سرور رد و بدل می‌شود، اضافه کند. این معمولاً از طریق ضعف‌های موجود در مرورگرها یا کاربردهای مبتنی بر وب انجام می‌گیرد.

قابلیت مشاهده طول داده‌های رمزگذاری‌شده

مهاجم باید بتواند طول داده‌های رمزگذاری‌شده را قبل و بعد از فشرده‌سازی مشاهده کند. این اطلاعات به او کمک می‌کند تا تغییرات در فشرده‌سازی را تحلیل کند.

نحوه انجام حمله CRIME

تزریق داده‌های کنترل‌شده

مهاجم داده‌های خاصی را به درخواست‌های HTTP ارسال می‌کند که در کنار کوکی‌های کاربر قرار می‌گیرند. هدف این است که داده‌های تزریق‌شده با بخش‌های مخفی (مانند کوکی‌ها) تکراری شوند.

تحلیل تغییرات در طول فشرده‌سازی

زمانی که داده‌های تزریق‌شده با کوکی‌ها تکراری باشند، الگوریتم فشرده‌سازی می‌تواند این تکرارها را به‌طور موثرتری فشرده کند، که منجر به کاهش طول داده‌های رمزگذاری‌شده می‌شود. مهاجم با مقایسه طول‌های مختلف داده‌های رمزگذاری‌شده برای حدس‌های مختلف، می‌تواند تشخیص دهد که کدام حدس‌ها صحیح هستند.

بازسازی کوکی‌ها به صورت مرحله‌ای

مهاجم به صورت تدریجی و با حدس‌های ترتیبی، بخش‌های مختلف کوکی را بازسازی می‌کند. هر بار که یک حدس صحیح باشد، کاهش قابل‌توجهی در طول داده‌های فشرده‌شده مشاهده می‌شود که به مهاجم نشان می‌دهد که آن بخش از کوکی صحیح حدس زده شده است.

حالا در یک مثال این مورد رو بررسی می‌کنم.

فرض کنید کوکی کاربر شامل رشته‌ای مخفی مانند session id=ABC123 است:

تزریق داده‌های کنترل‌شده توسط مهاجم

مهاجم داده‌هایی مانند A، AB، ABC و ... را به ابتدای درخواست‌های HTTP اضافه می‌کند.

تحلیل تغییرات طول داده‌های رمزگذاری‌شده

هر بار که یک بخش از حدس‌های مهاجم با کوکی واقعی مطابقت داشته باشد (مثلاً A با A یا AB با AB)، فشرده‌سازی بهینه‌تر انجام شده و طول داده‌های رمزگذاری‌شده کاهش می‌یابد.

بازسازی کامل کوکی

مهاجم با تحلیل این تغییرات طول، به تدریج می‌تواند کل کوکی را بازسازی کند.

حمله CRIME نشان‌دهنده آسیب‌پذیری‌های ترکیب فشرده‌سازی و رمزنگاری در پروتکل‌های امنیتی است. با حذف قابلیت فشرده‌سازی در نسخه‌های جدیدت

جواب سوال ۴