نشریه علمی "پدافند الکترونیکی و سایبری"
سال هفتم، شماره ۳، پاییز ۱۳۹۸، ص ۱۱۹-۱۱۳

حمله‌ای جدید به شبکه مخلوط مرکب جیکوبسون

محمود سلماسی‌زاده[1*]، سید امیر مرتضوی[2]، جواد مهاجری[3]

۱- دانشیار، پژوهشکده الکترونیک، دانشگاه صنعتی شریف، ۲- استادیار، دانشگاه تبریز، ۳- استادیار، پژوهشکده الکترونیک، دانشگاه صنعتی شریف

(دریافت: ۹۷/۱۰/۲۳، پذیرش: ۹۷/۱۲/۱۴)

چکیده

شبکه مخلوط مرکب جیکوبسون شبکه‌ای مخلوط است که با استفاده توأم از رمزنگاری متقارن و غیرمتقارن، گمنام‌سازی پیام‌های طولانی را به‌صورتی بسیار کارا ممکن می‌سازد. در این مقاله، حمله‌ای جدید به شبکه مخلوط مرکب جیکوبسون ارائه می‌شود که ویژگی صحت این شبکه مخلوط را نقض می‌کند. نشان خواهیم داد که با استفاده از این حمله درصورت تبانی یکی از فرستنده‌ها با اولین سرور مخلوط‌کننده، این سرور قادر خواهد بود که پیام تمامی فرستنده‌ها را با پیام‌های دلخواه خود جایگزین کند.

کلیدواژه‌ها: رمز شبکه مخلوط، شبکه مخلوط مرکب، گمنامی، صفردانشی، تسهیم راز

۱- مقدمه

امروزه شبکه مخلوط[2] ابزاری رمزنگاشتی برای ایجاد یک کانال ناشناس میان گروهی از فرستنده‌ها و گیرنده‌ها است. با استفاده از شبکه مخلوط هویت فرستنده‌ها برای گیرنده‌ها ناشناس باقی خواهد ماند. مهم‌ترین هدف از طراحی شبکه مخلوط حفظ گمنامی فرستنده‌ها است. به‌طور معمول یک شبکه مخلوط گروهی از پیام‌های رمزشده را به‌عنوان ورودی قبول می‌کند و به‌عنوان خروجی گروهی از پیام‌های آشکار را تولید و در اختیار گیرنده‌ها قرار می‌دهد به نحوی که گمنامی فرستنده‌ها حفظ شود.

ایده اولیه شبکه مخلوط توسط چام در [۱] مطرح شد و از آن زمان به بعد به یکی از ابزارهای قوی برای ایجاد گمنامی تبدیل شده است. برای آشنایی با روش‌های ایجاد کانال گمنامی می‌توان به [۳-۲] مراجعه کرد. در هر شبکه مخلوط سه هستار مشارکت دارند که عبارتند از: سرورهای مخلوط‌کننده[3]، گیرنده‌ها و فرستنده‌ها.

روند کلی عملکرد یک شبکه مخلوط به‌شرح زیر است:

فرستنده‌ها ابتدا پیام‌های خود را رمزگذاری می‌کنند و آن را به اولین سرور مخلوط کننده تحویل می‌دهند. این سرور عملیات مخلوط‌کردن را بر روی پیام‌های ورودی انجام می‌دهد و خروجی خود را به سرور مخلوط‌کننده بعدی تحویل می‌دهد. عملیات مخلوط‌کردن معمولاً شامل مراحل زیر است:

۱- انجام عملیات رمزنگاشتی نظیر الگوریتم‌های رمزگذاری یا رمزگشایی بسته به طراحی شبکه مخلوط.
۲- جای‌گشت‌های تصادفی بر روی مجموعه پیام‌های ورودی به‌منظور حذف ارتباط میان پیام‌های ورودی و خروجی.

خروجی هر سرور مخلوط‌کننده، ورودی سرور مخلوط‌کننده بعدی است که این سرور نیز عملیات مخلوط‌کردن را بر روی ورودی‌های دریافتی انجام می‌دهد. خروجی آخرین سرور شبکه مخلوط، پیام‌های رمزگشایی شده هستند که به گیرنده‌ها تحویل داده می‌شوند.

از جمله ویژگی‌های مطلوب شبکه‌های مخلوط می‌توان به ویژگی‌هایی نظیر صحت[4]، حفظ حریم خصوصی فرستنده‌ها، پایداری[5] و وارسی‌پذیری عمومی[6] اشاره کرد. هدف اصلی در طراحی شبکه مخلوط برآوردن این ویژگی‌ها، همزمان با تأمین کارایی بالا است. صحت در شبکه مخلوط به این معنی است که مجموعه پیام‌های خروجی شبکه مخلوط برابر با مجموعه پیام‌های آشکار متناظر با متون رمزشده ورودی به شبکه مخلوط باشد. شبکه مخلوط پایدار نامیده می‌شود اگر حتی در صورت عدم مشارکت تعدادی از سرورهای مخلوط‌کننده نیز عملکرد نهایی شبکه مخلوط درست باشد. یک شبکه مخلوط وارسی‌پذیر نامیده

* رایانامه نویسنده مسئول salmasi@sharif.ir

[2] Mix-net
[3] Mixing servers
[4] Correctness
[5] Robustness
[6] Public Verifiability

می‌شود، هرگاه هر سرور مخلوط‌کننده قادر به اثبات صحت عملکرد خود به دیگران باشد. این کار عمدتاً با استفاده از اثبات‌های صفر دانشی[1] انجام می‌شود.

از شبکه‌های مخلوط برای ایجاد یک کانال گمنام‌ساز استفاده می‌شود. به‌طور مثال در تعدادی از پروتکل‌های رأی‌گیری الکترونیکی[2] برای ایجاد یک کانال گمنام‌ساز میان رأی‌دهنده‌ها و نهادهای شمارش آرا، از شبکه‌های مخلوط استفاده می‌شود [۵-۴].

در سال ۱۹۸۱ چام اولین شبکه مخلوط معرفی به نام شبکه مخلوط رمزگشا را معرفی کرد [۱]. در شبکه مخلوط رمزگشا[3]، هر یک از فرستنده‌ها پیام‌های خود را با استفاده از کلید عمومی سرورهای مخلوط‌کننده به صورت مکرر رمزگذاری می‌کند و هر سرور مخلوط‌کننده با استفاده از الگوریتم رمزگشایی یک لایه از متن رمزگذاری شده را رمزگشایی و جابجا می‌کند، به این ترتیب خروجی آخرین سرور مخلوط‌کننده پیام‌های اصلی با ترتیب‌های متفاوت خواهد بود. این شبکه فاقد ویژگی پایداری است. زیرا برای عملکرد بدون نقص شبکه مخلوط رمزگشا باید تمامی سرورهای مخلوط‌کننده درست‌کار باشند، که این امر مهمترین نقطه ضعف این نوع از شبکه‌های مخلوط است. چرا که اگر حتی یکی از سرورهای مخلوط‌کننده از انجام عملیات خود سرباز بزند عملکرد کل شبکه مخلوط مختل می‌شود.

برای دست‌یابی به شبکه مخلوط پایدار در برابر سرورهای مخلوط‌کننده خرابکار، پارک و همکاران در مرجع [۶] شبکه مخلوط جدیدی را پیشنهاد کردند که در آن عملیات رمزگشایی با عملیات بازرمزگذاری[4] جایگزین می‌شود. مفهوم شبکه مخلوط بازرمزگذار اولین بار در [۶] معرفی شد. پس از معرفی چندین آسیب‌پذیری این شبکه مخلوط در مرجع [۷] شبکه مخلوط جدیدی در [۸] معرفی شد که در برابر حملات مزبور مقاوم است.

اولین شبکه مخلوط با وارسی‌پذیری عمومی در [۹] مطرح شد که در آن هر کسی می‌تواند صحت عملیات مخلوط کردن را بررسی کند ولی این طرح کارایی پایین و پیچیدگی بالایی دارد. بعد از آن شبکه‌های مخلوط با وارسی‌پذیری عمومی و کارایی بالایی در [۱۳-۱۰] معرفی شدند. همچنین ساکو در [۱۴] و نف در [۱۵] دو شبکه مخلوط با وارسی‌پذیری عمومی و کارایی بالا پیشنهاد دادند. در [۱۶] شبکه مخلوطی با وارسی‌پذیری عمومی و وارسی‌پذیری فرستنده‌ها ارائه شد. در [۱۷] برای مقاوم کردن شبکه مخلوط ار روش RPC[5] استفاده شده است که در آن به جای اثبات صحت کلی عملکرد سرورهای مخلوط‌کننده، هر سرور شواهدی قوی از صحت عملکرد خود را ارائه می‌دهد.

شبکه‌های مخلوط معمولاً از سامانه‌های رمزگذاری کلید عمومی نظیر الجمال و RSA برای رمزگشایی و رمزگذاری مجدد استفاده می‌کنند. استفاده از سامانه‌های رمزگذاری کلید عمومی در طراحی شبکه‌های مخلوط، منجر به بروز مشکلاتی نظیر محدود شدن طول پیام‌ها و کاهش کارایی (به دلیل کارایی بالاتر سامانه‌های رمزگذاری کلید متقارن) می‌شود. برای مخلوط‌کردن پیام‌های با اندازه بزرگ باید این پیام را به قسمت‌های کوچکتر تقسیم کرده و هر قسمت را به صورت مجزا مخلوط و در انتها دوباره این پیام‌ها را به یکدیگر متصل کنیم. برای رسیدن به کارایی بالاتر و پذیرش پیام‌های با طول بزرگ‌تر شبکه‌های مخلوط با استفاده توأم از سامانه‌های رمزگذاری متقارن و نامتقارن طراحی شدند و به شبکه‌های مخلوط مرکب[6] معروف شدند. در شبکه‌های مخلوط مرکب، پیام‌های ورودی فرستنده‌ها با الگوریتم‌های رمزگذاری متقارن رمزگذاری می‌شوند که سبب افزایش کارایی طرح می‌شود.

شبکه‌های مخلوط مرکب مقاوم محدودیت طول ورودی پیام‌ها را ندارند و در برابر تعدادی سرور خطاکار نیز مقاوم هستند. آبی و همکاران در [۱۸] شبکه مخلوط مرکب مقاومی را معرفی کردند که در برابر $\sqrt{n}$ سرور خطاکار از n سرور مخلوط‌کننده مقاوم هستند. جیکوبسون و همکاران در [۱۹] شبکه مخلوط مرکبی را معرفی کردند که در برابر نیمی از سرورهای مخلوط‌کننده خطاکار مقاوم هستند. شبکه‌های مخلوط معرفی شده در [۲۰-۱۸] توانایی مخلوط‌کردن پیام‌های طولانی را به صورتی کارا دارند ولی دارای ویژگی وارسی‌پذیری عمومی نیستند.

در [۲۱] ادعا شده است که از هر سامانه رمزنگاری با امنیت تمایزناپذیری تحت حمله متن‌رمزشده منتخب[7]، CCA، می‌توان یک شبکه مخلوط با امنیت اثبات‌پذیر ارائه کرد. در [۲۲] تحلیلی صوری[8] از شبکه مخلوط معرفی شده در [۱۷] انجام یافته است. همچنین اولین شبکه مخلوط بر اساس رمزنگاری شناسه‌بنیاد[9] در [۲۰] معرفی شده است.

همزمان با طراحی شبکه‌های مخلوط با ویژگی‌های گوناگون،

[1] Zero Konwledge
[2] E-voting
[3] Decryption mix-net
[4] Reencryption
[5] Randomized Partial Checking
[6] Hybrid Mix-Net
[7] Indistinguishability under Chosen Ciphertext Attack
[8] Formal analysis
[9] Identity-based

تحلیل‌های مختلفی نیز بر روی این شبکه‌های اعمال شده است. در [۲۳] حمله‌ای به شبکه مخلوط مرکب جیکوبسون [۱۹] ارائه شده است که می‌تواند گمنامی فرستنده‌ها را نقض کند. در [۲۴] پنج حمله مختلف بر شبکه مخلوط [۲۵] پیشنهاد شده است. در [۲۶] تحلیل امنیتی از شبکه مخلوط [۱۷] ارائه شده است.

در این مقاله ابتدا در بخش ۲ مروری بر شبکه مخلوط مرکب جیکوبسون خواهیم داشت. سپس در بخش ۳ حمله جدیدی بر شبکه مخلوط جیکوبسون معرفی می‌کنیم. این حمله ویژگی صحت این شبکه مخلوط را نقض می‌کند.

۲- مروری بر شبکه مخلوط جیکوبسون

در این بخش به صورت مختصر شبکه مخلوط مرکب جیکوبسون [۱۹] را معرفی خواهیم کرد. توصیفات ذکرشده در این بخش از همین مرجع برگرفته شده است ولی از ذکر بعضی از جزییات غیرمرتبط با حمله پیشنهادی صرف‌نظر شده است. برای آشنایی بیشتر با جزییات این طرح می‌توان به [۱۹] مراجعه کرد.

۲-۱- مشخصه‌های طرح

در این قسمت به معرفی نهادها و نمادهای به‌کاررفته در این شبکه مخلوط می‌پردازیم.

نهادهای مشارکت‌کننده در شبکه مخلوط مرکب جیکوبسون عبارتند از: N فرستنده با نمادهای $P_1, \dots, P_N$ و n سرورمخلوط‌کننده با نمادهای $S_1, \dots, S_n$ نمایش داده می‌شود و یک بولتن عمومی که برای نمایش پیام‌های عمومی استفاده می‌شود. هر فرستنده ابتدا پیام خود را رمزگذاری می‌کند و در بولتن عمومی[1] به نمایش می‌گذارد و سپس سرورهای مخلوط‌کننده عملیات مخلوط کردن را انجام می‌دهند و با استفاده از بولتن عمومی پیام‌های خروجی خود را به اطلاع سرور بعدی می‌رسانند.

همچنین p, q نشان‌دهنده اعداد اول بزرگی هستند که در رابطهٔ $q|p-1$ صدق می‌کنند. Z_p^* گروه ضربی به پیمانه p و G_q زیرگروهی از مرتبه q از گروه Z_p^* است. مولد G_q با نماد g نمایش داده می‌شود. کد احراز اصالت پیام MAC برای احراز اصالت پیام استفاده می‌شود. کد احراز اصالت پیام m با نماد $\mathrm{MAC}_z(m)$ نشان داده می‌شود که در آن، z کلید مشترک فرستنده و وارسی‌کننده است. اگر $k \in G_q$ کلیدی به اشتراک گذاشته میان دو نهاد باشد در این‌صورت، عبارات $E_k[m] = c$ و $D_k[c] = m$ به‌ترتیب نشان‌دهندهٔ فرایندهای رمزگذاری و رمزگشایی متقارن پیام m هستند.

اثبات صفردانشی برای نشان دادن این که در چهارتایی $(a, b, y, z) \in G_q^4$ روابط $\log_a b = \log_y z = x$ برقرار بوده و اثبات‌کننده مقدار x را می‌داند با نماد EQDL[a,b,y,z] نشان داده می‌شود.

تعریف ۱: (صفردانشی)
صفردانشی الگوریتمی است که در آن یک طرف به‌عنوان اثبات‌کننده اطلاع خود از رازی را به طرف دیگری به نام وارسی‌کننده[2] به اثبات می‌رساند. با اجرای پروتکل صفردانشی، وارسی‌کننده هیچ اطلاعی راجع به راز را به‌دست نمی‌آورد.

۲-۲- تولید کلید

هر سرور مخلوط‌کننده S_i, $1 \le i \le n$ سه مقدار مخفی و تصادفی $\alpha_i, \beta_i, \gamma_i \in Z_q$ را به‌عنوان کلیدهای مخفی خود انتخاب می‌کند و سپس سه‌تایی (Y_i, K_i, Z_i) را محاسبه و به‌عنوان کلید عمومی خود منتشر می‌کند که در آن، $Y_i = Y_{i-1}^{\alpha_i}$ و $K_i = Y_{i-1}^{\beta_i}$ و $Z_i = Y_{i-1}^{\gamma_i}$ است. در این محاسبات فرض می‌شود که $Y_0 = g$ است.

در مرحله راه‌اندازی این طرح هر سرور مخلوط‌کننده با استفاده از طرح‌های اثبات صفردانشی ارائه شده در [۲۸-۲۷] آگاهی خود از کلیدهای مخفی را به سایر نهادها نشان می‌دهد. سپس کلیدهای مخفی مشترک سرورها با استفاده از یک طرح تسهیم راز وارسی‌پذیر $(t+1, n)$ بین سرورها تسهیم می‌شود. در این رابطه $n = 2t + 1$ است. نمونه‌ای از طرح‌های تسهیم راز وارسی‌پذیر در [۲۹] ارائه شده است.

۲-۳- سرور شبیه‌سازی‌شده

با همکاری تمامی سرورها و استفاده از روش‌های تسهیم راز وارسی‌پذیر، VSS[3]، سرور شبیه‌سازی شده S_{n+1} با کلیدهای خصوصی $\beta_{n+1}, \gamma_{n+1}$ و کلیدهای عمومی K_{n+1}, Z_{n+1} شبیه‌سازی می‌شود و این مقادیر میان سایر سرورها به اشتراک گذاشته می‌شود.

۲-۴- رمزگذاری

در این بخش الگوریتم رمزگذاری که فرستنده‌ها برای رمزگذاری پیام‌های خود استفاده می‌کنند، معرفی خواهد شد. این الگوریتم رمزگذاری بر اساس الگوریتم‌های رمزگذاری متقارن پیام‌ها را لایه

[1] Bulletin Board

[2] Verifier

[3] Verifiable Secret Sharing

به لایه با کلیدهای سرورها به ترتیب معکوس رمزگذاری می‌کند.

۲-۴-۱- فرانمای کلید

فرانمای کلید[1] الگوریتمی برای اشتراک‌گذاری کلید میان سرورها و فرستنده‌ها است که در آن هر سرور با استفاده از پیام‌های دریافتی از سرور قبلی می‌تواند کلید به اشتراک‌گذاری شده با فرستنده‌ها را بازیابی کند. این روش در حقیقت یک نوع سامانه رمزگذاری کلید عمومی است که در آن، هر فرستنده $N+1$ کلید تصادفی را انتخاب و رمزگذاری می‌کند که این متن رمزشده با y_0 نشان داده می‌شود.

هر فرستنده مانند P_j کلید مخفی خود $\rho \in Z_q$ را به‌صورت تصادفی انتخاب و سپس مقادیر زیر را برای هر سرور محاسبه می‌کند:

$$\begin{aligned} &k_i = K_i^{\rho}, \quad 0 \le i \le n+1 \\ &z_i = Z_i^{\rho}, \quad 1 \le i \le n+1\#1 \\ &y_0 = Y_0^{\rho}. \end{aligned} \tag{۱}$$

سرور S_i با دریافت پیام y_{i-1} از سرور قبلی سه‌تایی $(y_i, k_i, z_i) = \left(y_{i-1}^{\alpha_i}, y_{i-1}^{\beta_i}, y_{i-1}^{\gamma_i}\right)$ را محاسبه می‌کند که به نوبه خود برابر $Y_i^{\rho}, K_i^{\rho}, Z_i^{\rho}$ است.

۲-۴-۲- رمزگذاری پیام

هر فرستنده با استفاده از کلیدهای به اشتراک‌گذاشته با سرورهای مخلوط‌کننده، معرفی شده در بخش ۲-۴، پیام خود را در این مرحله رمزگذاری می‌کند. برای مثال فرستنده jام پیام m_j را به شکل زیر رمزگذاری می‌کند:

$$\begin{cases} c_n = \mathrm{E}_{k_{n+1}}[m_j] \\ c_i = \mathrm{E}_{k_{i+1}}[c_{i+1} \,||\, \mu_{i+1}] & 0 \le i \le n-1 \\ \mu_i = \mathrm{MAC}_{z_{i+1}}[c_i \,||\, I] & 0 \le i \le n \end{cases} \tag{۲}$$

در رابطه بالا کلید k_j کلید مشترک سرور iام و فرستنده برای رمزگذاری، و کلید z_i کلید مشترک سرور iام و فرستنده برای استفاده در کد احراز اصالت پیام (MAC) است. همچنین I رشته بیت عمومی و آشکاری است که در ابتدای هر عملیات مخلوط‌سازی با همکاری تمامی سرورها ایجاد می‌شود و نقش یک تک‌شمار[2] را بازی می‌کند.

متن رمزشده فرستنده jام را با نماد $\left\{c_0^{(j)}, \mu_0^{(j)}, y_0^{(j)}\right\}$ نشان می‌دهیم و فرستنده‌های دیگر نیز به‌طور مشابه پیام‌های خود را تشکیل می‌دهند. برای امنیت طرح، طول تمامی متن‌های رمزشده باید برابر باشد. مجموعه پیام‌های ارسال شده به سرور اول با $\left\{c_0^{(j)}, \mu_0^{(j)}, y_0^{(j)}\right\}_{j=1}^{N}$ نمایش داده می‌شود.

۲-۵- عملیات مخلوط کردن

در این شبکه مخلوط مرکب، تمامی تراکنش‌ها میان نهادهای شبکه مخلوط مرکب با استفاده از بولتن عمومی انجام می‌شود.

سرور S_i، مجموعه $\left\{c_{i-1}^{(j)}, \mu_{i-1}^{(j)}, y_{i-1}^{(j)}\right\}_{j=1}^{N}$ را به‌عنوان ورودی می‌پذیرد و مراحل زیر را بر روی آن انجام می‌دهد.

الف- تولید کلید: سرور S_i با استفاده از کلیدهای خصوصی خود کلیدهای به اشتراک‌گذاشته‌شده خود با فرستنده‌ها را به‌صورت زیر محاسبه می‌کند:

$$\begin{aligned} \tilde{y}_i^{(j)} &= (y_{i-1}^{(j)})^{\alpha_i} \\ \tilde{k}_i^{(j)} &= (y_{i-1}^{(j)})^{\beta_i} \\ \tilde{z}_i^{(j)} &= (y_{i-1}^{(j)})^{\gamma_i}, \end{aligned} \tag{۳}$$

که در آن، $1 \le j \le N$ است.

ب- وارسی کد احراز اصالت پیام

سرور S_i صحت کد احراز اصالت پیام را با رابطه زیر وارسی می‌کند:

$$\mu_i^{(j)} = MAC_{\tilde{z}_i^{(j)}}\left[c_{i-1}^{(j)} \,\|\, I\right] \tag{۴}$$

اگر نتیجه وارسی کد احراز اصالت پیام منفی باشد، سرور S_i الگوریتم شکایت (رجوع به بخش ۲-۶) را اجرا می‌کند.

ج- رمزگشایی پیام

سرور S_i متن رمزشده فرستنده jام را به شکل زیر رمزگشایی می‌کند:

$$\left(\tilde{c}_i^{(j)} \,\|\, \tilde{\mu}_i^{(j)}\right) \leftarrow D_{\tilde{k}_i^{(j)}}\left[c_{i-1}^{(j)}\right]. \tag{۵}$$

د- جایگشت تصادفی

سرور S_i یک جایگشت تصادفی را بر روی $\left\{\tilde{c}_i^{(j)}, \tilde{\mu}_i^{(j)}, \tilde{y}_i^{(j)}\right\}_{j=1}^{N}$ اعمال می‌کند و نتیجه نهایی یعنی $\left\{c_i^{(j)}, \mu_i^{(j)}, y_i^{(j)}\right\}_{j=1}^{N}$ را بر روی بولتن عمومی ارسال می‌کند.

ه- اثبات صفردانشی

سرور S_i باید صحت عملکرد خود را به سرور S_{i+1} ثابت کند. اگر فرض کنیم که $P_i = \prod_{j=1}^{N} y_i^{(j)}$ باشد، در این صورت سرور S_i برای اثبات برقراری رابطه $P_i = P_{i-1}^{\alpha_i}$ از اثبات صفردانشی $\mathrm{EQDL}[P_{i-1}, P_i, Y_{i-1}, Y_i]$ استفاده می‌کند. اگر سرور S_{i+1} در اثبات مغایرتی پیدا کند، الگوریتم شکایت اجرا خواهد شد.

[1] Key scheduling
[2] Nounce

الگوریتم شکایت در بخش ۲-۶ بیان خواهد شد.

خروجی سرور S_n برابر $\left\{c_n^{(j)}, \mu_n^{(j)}, y_n^{(j)}\right\}_{j=1}^{N}$ است. در این مرحله سرور شبیه‌سازی‌شده S_{n+1} با کمک سایر سرورها و الگوریتم تسهیم راز ساخته می‌شود (برای نحوه ساخت سرور شبیه‌سازی‌شده به بخش ۲-۳ مراجعه شود). ابتدا کلید $z_{n+1}^{(j)}$ برای $1 \le j \le N$ تولید می‌شود و با استفاده از این کلید ارزیابی تابع MAC انجام می‌شود و در صورت صحت ارزیابی، کلید $k_{n+1}^{(j)}$ محاسبه و رمزگشایی توسط رابطه $m_j \leftarrow D_{k_{n+1}^{(j)}}\left[c_n^{(j)}\right]$ انجام می‌شود.

۲-۶- الگوریتم شکایت

الگوریتم Verify_Complaint(i, j) زمانی توسط سرور S_i اجرا می‌شود که ورودی $\left\{c_{i-1}^{(j)}, \mu_{i-1}^{(j)}, y_{i-1}^{(j)}\right\}_{j=1}^{N}$ نادرست باشد. با همکاری سایر سرورها این شکایت بررسی خواهد شد و سه حالت ممکن است رخ بدهد:

۱- سرور S_i خطاکار است.
۲- سرور S_{i-1} خطاکار است.
۳- متن رمزشده دارای شکل نادرستی است.

۳- حمله‌ای کارا بر شبکه مخلوط مرکب جیکوبسون

در این بخش حمله‌ای جدید بر شبکه مخلوط مرکب جیکوبسون ارائه می‌شود. این حمله ویژگی صحت شبکه مخلوط را نقض می‌کند. در این حمله اولین سرور مخلوط‌کننده خطاکار و یکی از فرستنده‌ها با یکدیگر تبانی می‌کنند تا در خروجی پیام‌های دلخواهی را بدون شناسایی توسط سایر سرورها تولید کنند.

۳-۱- شرح حمله جدید

در این حمله فرض بر این است که اولین سرور S_1 و فرستنده Nام، P_N، با یکدیگر تبانی می‌کنند ولی هر کدام از فرستنده‌ها نیز می‌توانند به جای P_N در تبانی شرکت کنند.

در این حمله اولین سرور مخلوط‌کننده می‌تواند پیام‌های دریافتی خود را حذف و با پیام‌های دلخواهی جایگزین کند، بدون اینکه توسط سایر سرورها شناسایی شود. نسخه‌ای تا حدودی شبیه به این حمله در [۲۴] به شبکه مخلوط دیگری ارائه شده است ولی این حمله منجر به نتیجه خیلی قویتری نسبت به حملات قبلی می‌شود. پیام‌های ورودی به اولین سرور شامل $\left\{c_0^{(j)}, \mu_0^{(j)}, y_0^{(j)}\right\}_{j=1}^{N}$، همچنین $y_0^{(j)} = Y_0^{\rho(j)}$ است. در این عبارت مقدار $\rho(j)$، کلید مخفی فرستنده jام است. فرستنده خطاکار N ام پیام ورودی خود را با استفاده از رابطه (۶) تشکیل می‌دهد:

$$y_0^{(N)} = \prod_{j=1}^{N-1} y_0^{-(j)} Y_0^{\rho(N)}. \qquad (۶)$$

در رابطه بالا مقدار $\rho(N)$ کلید مخفی فرستنده Nام و مقدار $y_0^{-(j)}$ معکوس عبارت $y_0^{(j)}$ (متن ارسالی فرستنده jام) به پیمانه p و $c_0^{(N)}, \mu_0^{(N)}$ مقادیری تصادفی و دلخواهی انتخاب می‌شوند تا ساختار کلی پیام ارسالی درست باشد. سپس فرستنده Nام متن رمزشده خود را به شکل $\left\{c_0^{(N)}, \mu_0^{(N)}, y_0^{(N)}\right\}_{j=1}^{N}$ تشکیل می‌دهد.

به خاطر این‌که پیام‌های ورودی به بولتن عمومی ارسال می‌شوند و برای همه نهادها آشکار هستند، فرستنده P_N می‌تواند عبارت $y_0^{(N)}$ را تشکیل دهد. برای اعمال این حمله، فرستنده Nام با سرور اول تبانی می‌کنند و فرستنده N ام مقدار کلید مخفی خود یعنی $\rho(N)$ را در اختیار سرور اول قرار می‌دهد.

در این مرحله سرور اول، خود را به‌جای فرستنده‌ها می‌گذارد و به‌جای فرستنده‌ها پیام‌های انتخابی خود را برگزیده و فرایند زیر را دنبال می‌کند:

اولین سرور مخلوط‌کننده می‌تواند پیام‌های ورودی را به نحوی تغییر دهد که از مراحل وارسی عبور کند. اولین سرور مقادیر تصادفی و یکنواخت $m_1', m_2', \dots, m_N'$ و $\rho'(j) \in Z_q$ را برای $1 \le j \le N$ انتخاب می‌کند به‌نحوی که:

$$\sum_{j=1}^{N} \rho'(j) = \rho(N) \bmod q. \qquad (۷)$$

سرور اول مخلوط‌کننده مقادیر زیر را برای $1 \le j \le N$ محاسبه می‌کند:

$$\begin{aligned} &\rho''(j) = \rho'(j)\alpha_1 \\ &k_i^{(j)} = K_i^{(j)\rho'(j)} \quad 2 \le i \le n+1 \\ &z_i^{(j)} = Z_i^{(j)\rho'(j)} \quad 2 \le i \le n+1 \\ &y_1^{(j)} = Y_0^{\rho''(j)}, \end{aligned} \qquad (۸)$$

سپس پیام m_j' را به شکل زیر رمزگذاری می‌کند:

$$\begin{aligned} &c_n = E_{k_{n+1}^{(j)}}[m_i'] \\ &c_i = E_{k_{i+1}^{(j)}}[c_{i+1} \parallel \mu_{i+1}] \quad 2 \le i \le n-1 \\ &\mu_i = MAC_{z_{i+1}^{(j)}}[c_i \parallel I] \quad 2 \le i \le n, \end{aligned} \qquad (۹)$$

این سرور می‌تواند مجموعه $\left\{c_1^{(j)}, \mu_1^{(j)}, y_1^{(j)}\right\}_{j=1}^{N}$ را تشکیل و اطلاعات لازم برای اثبات صفردانشی EQDL را به این مجموعه اضافه کند. سرورهای بعدی روال معمول را طبق الگوریتم ادامه می‌دهند.

۳-۲- بررسی رفتار سرور دوم

سرور دوم بعد از دریافت $\left\{c_1^{(j)}, \mu_1^{(j)}, y_1^{(j)}\right\}_{j=1}^{N}$ باید ابتدا صحت عملکرد سرور اول را بررسی کند. برای این منظور از اثبات صفردانشی $\mathrm{EQDl}[P_0, P_1, Y_0, Y_1]$ استفاده می‌شود.

در این حالت داریم:

$$P_0 = \prod_{j=1}^{N} y_0^{(j)} = Y_0^{\rho(N)}$$
$$P_1 = \prod_{j=1}^{N} y_1^{(j)} = \prod_{j=1}^{N} Y_0^{\rho'(j)\alpha_1} \quad (۱۰)$$

در نتیجه داریم:

$P_1 = Y_0^{\rho(N)\alpha_1}$.

همچنین $Y_0 = g, Y_1 = g^{\alpha_1}$ است و چون سرور اول مقدار $\rho(N)$ را با همکاری فرستنده همکار خود می‌داند، می‌تواند این اثبات صفردانشی را انجام دهد.

۳-۳- بررسی روند حمله

انجام این حمله به این علت امکان‌پذیر است که نیازی به اثبات آگاهی فرستنده‌ها از نمای $y_0^{(j)}$ وجود ندارد. در نتیجه با تبانی یکی از فرستنده‌ها و اولین سرور، می‌توان نقش سایر فرستنده ها را حدف و ویژگی صحت این شبکه مخلوط را نقض کرد. در نتیجه با اجرای این حمله سرور S_1 می‌تواند پیام تمامی فرستنده‌ها را با پیام‌های دلخواه خود جایگزین کند، بدون این‌که توسط سایر سرورها شناسایی شود. در نهایت سرور نهایی خروجی شبکه مخلوط را تولید می‌کند که همان پیام‌های انتخابی توسط سرور اول هستند.

۴- نتیجه‌گیری

شبکه مخلوط مرکب شبکه مخلوطی است که برای پیام‌های طولانی کارا است. شبکه مخلوط جیکوبسون شبکه مخلوط مقاومی است که در برابر تبانی چند سرور خطاکار مقاوم است. در این مقاله حمله‌ای جدید بر این شبکه مخلوط پیشنهاد شده است که در این حمله با تبانی یکی از فرستنده‌ها و اولین سرور مخلوط‌کننده می‌توان ویژگی صحت این شبکه مخلوط را نقض کرد.

۵- منابع

[1] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," Commun. ACM, vol. 24, no. 2, pp. 84–90, 1981.

[2] G. Danezis and C. Diaz, "A survey of anonymous communication channels," 2008.

[3] G. Fanti and P. Viswanath, "Algorithmic advances in anonymous communication over networks," in 2016 Annual Conference on Information Science and Systems (CISS), pp. 133–138, 2016.

[4] P. Bibiloni, A. Escala, and P. Morillo, "Vote validatability in mix-net-based eVoting," in International Conference on E-Voting and Identity, pp. 92–109, 2015.

[5] J. Furukawa, K. Mori, and K. Sako, "An implementation of a mix-net based network voting scheme and its use in a private organization," in towards trustworthy elections, Springer, pp. 141–154, 2010.

[6] C. Park, K. Itoh, and K. Kurosawa, "Efficient anonymous channel and all/nothing election scheme," in Workshop on the Theory and Application of of Cryptographic Techniques, pp. 248–259, 1993.

[7] B. Pfitzmann, "Breaking an efficient anonymous channel," in Workshop on the Theory and Application of of Cryptographic Techniques, pp. 332–340, 1994.

[8] W. Ogata, K. Kurosawa, K. Sako, and K. Takatani, "Fault tolerant anonymous channel," in International Conference on Information and Communications Security, pp. 440–444, 1997.

[9] [9] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme," in International Conference on the Theory and Applications of Cryptographic Techniques, pp. 393–403, 1995.

[10] M. Jakobsson, "Flash mixing," in PODC, vol. 99, pp. 83–89, 1999.

[11] M. Abe, "Mix-networks on permutation networks," in International Conference on the Theory and Application of Cryptology and Information Security, pp. 258–273, 1999.

[12] M. Jakobsson, "A practical mix," in International Conference on the Theory and Applications of Cryptographic Techniques, pp. 448–461, 1998.

[13] M. Abe, "Universally verifiable mix-net with verification work independent of the number of mix-servers," in International Conference on the Theory and Applications of Cryptographic Techniques, pp. 437–447, 1998.

[14] J. Furukawa and K. Sako, "An efficient scheme for proving a shuffle," in Annual International Cryptology Conference, pp. 368–387, 2001.

[15] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in Proceedings of the 8th ACM conference on Computer and Communications Security, pp. 116–125, 2001.

[16] L.-H. Li, C.-Q. Huang, and S.-F. Fu, "A CCA-secure Verifiable Mix-net," in 2017 International Conference on Networking and Network Applications (NaNA), pp. 239–245, 2017.

[17] M. Jakobsson, A. Juels, and R. L. Rivest, "Making mix nets robust for electronic voting by randomized partial checking," in USENIX security symposium, pp. 339–353, 2002.

[18] M. Ohkubo and M. Abe, "A length-invariant hybrid mix," in International Conference on the Theory and Application of Cryptology and Information Security, pp. 178–191, 2000.

[19] M. Jakobsson and A. Juels, "An optimally robust hybrid mix network," in Proceedings of the twentieth annual ACM symposium on Principles of distributed computing, pp. 284–292, 2001.

[20] M. A. Ekhtiarabadi, H. A. Yajam, J. Mohajeri, and M. Salmasizadeh, "Verifiable identity-based mix network," in 2015 23rd Iranian Conference on Electrical Engineering, pp. 406–409, 2015.

[21] S. Khazaei, T. Moran, and D. Wikström, "A mix-net from any CCA2 secure cryptosystem," in International

Conference on the Theory and Application of Cryptology and Information Security, pp. 607–625, 2012.

[22] R. Küsters, T. Truderung, and A. Vogt, "Formal analysis of chaumian mix nets with randomized partial checking," in 2014 IEEE Symposium on Security and Privacy, pp. 343–358, 2014.

[23] M. Abe and H. Imai, "Flaws in some robust optimistic mix-nets," in Australasian Conference on Information Security and Privacy, pp. 39–50, 2003.

[24] D. Wikström, "Five practical attacks for 'optimistic mixing for exit-polls,'" in International Workshop on Selected Areas in Cryptography, pp. 160–174, 2003.

[25] P. Golle, S. Zhong, D. Boneh, M. Jakobsson, and A. Juels, "Optimistic mixing for exit-polls," in International Conference on the Theory and Application of Cryptology and Information Security, pp. 451–465, 2002.

[26] [26] R. Küsters and T. Truderung, "Security Analysis of Re-Encryption RPC Mix Nets," in 2016 IEEE European Symposium on Security and Privacy (EuroS P), pp. 227–242, 2016.

[27] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in Annual International Cryptology Conference, pp. 89–105, 1992.

[28] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in Annual International Cryptology Conference, pp. 174–187, 1994.

[29] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," in International Conference on the Theory and Applications of Cryptographic Techniques, pp. 295–310, 1999.

Journal of Electronical & Cyber Defence
Vol. 7, No. 3, 2019, Serial No. 27

A New Attack on Jakobsson Hybrid Mix-Net

M. Salmasizadeh*, S. A. Mortazavi, J. Mohajeri

*Sharif University of Technology

(Received: 09/09/2018, Accepted: 05/03/2019)

ABSTRACT

The Jakobsson hybrid mix-net proposed by Jakobsson and Juels, is a very practical and efficient scheme which applies symmetrical and asymmetrical cryptography concurrently to make long input messages anonymous. In this paper a new attack on the Jakobsson hybrid mix-net is introduced. This attack infringes the faultlessness of the hybrid mix-net scheme. We will show that in this new attack, if one of the senders colludes with the first mix server, the first mix server can replace the messages of all other senders with arbitrary messages without being detected.

Keywords: mix-net, hybrid mix-net, anonymity, zero knowledge, secret sharing

* Corresponding Author Email: salmasi@sharif.ir