هفتمین کنفرانس بین المللی انجمن رمز ایران
۲۴ و ۲۵ شهریورماه ۱۳۸۹، دانشگاه صنعتی خواجه نصیرالدین طوسی
دانشگاه صنعتی خواجه نصیرالدین طوسی
انجمن رمزایران

تحلیل امنیتی شبکه‌ی مخلوط FLASH

سید امیر مرتضوی[۱]، جواد مهاجری[۲]، محمود سلماسی‌زاده[۳]
تهران، دانشگاه صنعتی شریف، دانشکده‌ی مهندسی برق[۱]
sa_mortazavi@ee.sarif.edu
تهران، دانشگاه صنعتی شریف، پژوهشکده‌ی الکترونیک [۲و۳]
{mohajer, salmasi}@sharif.ir[2,3]

چکیده

شبکه‌ی مخلوط Flash توسط Jakobsson در سال ۱۹۹۹ در ACM معرفی شد، که یکی از کاراترین و سریعترین شبکه‌های مخلوط با قابلیت بررسی عمومی است. ما در این مقاله نشان می‌دهیم که بر خلاف ادعاهای مطرح شده این شبکه‌ی مخلوط در برابر حملات فعال آسیب پذیر است. به این منظور دو حمله‌ی جدید به این شبکه‌ی مخلوط مطرح و نشان داده می‌شود که حمله‌ی اول پایداری شبکه و حمله‌ی دوم گمنامی فرستنده‌ها را مورد هدف قرار می‌دهد. در نهایت چند راه‌کار برای بهبود طرح و افزایش امنیت شبکه‌ی مخلوط در برابر حملات ارایه شده مطرح می‌گردد .

واژه های کلیدی

شبکه مخلوط، سرور مخلوط کننده،گمنامی فرستنده‌ها، حمله فعال

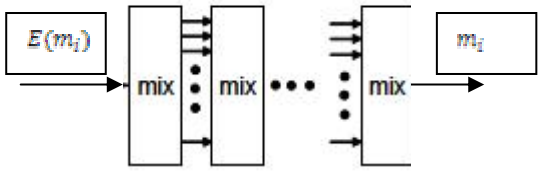
۱- مقدمه

شبکه‌ی مخلوط یک ابزار رمزنگاری است که برای ایجاد کانالی ناشناس[۱] بین گروهی از فرستنده‌ها و گروهی از گیرنده‌ها به کار می‌رود به گونه‌ای که هویت فرستنده‌ها برای گیرنده‌ها ناشناس باشد. مهمترین بعد امنیتی شبکه‌های مخلوط حفظ گمنامی فرستنده‌ها است. یک شبکه‌ی مخلوط با طراحی صحیح یک گروه از پیام‌های رمزشده را به عنوان ورودی می‌پذیرد و در خروجی مجموعه‌ای از پیام‌های رمز نشده را به گیرنده‌ها و یا گیرنده تحویل می‌دهد به نحوی که فرستنده‌ی پیام‌ها معلوم نباشد. لفظ شبکه‌ی مخلوط[۲] برای اولین بار توسط Chaum در [۱] استفاده شد و از آن زمان به عنوان یکی از ابزارهای قوی برای طراحی کانال‌های ناشناس به کار گرفته شده است (در [۲] انواع روش‌های طراحی کانال‌های ناشناس بحث شده است).

هر شبکه‌ی مخلوط از سه نهاد سرورها و فرستنده‌ها و گیرنده‌ها تشکیل می‌شود. فرستنده‌ها پیام‌های خود را رمزگذاری می‌کنند و به سرور اول تحویل می‌دهند، سرور اول عملیات مخلوط کردن را روی این پیام‌ها انجام می‌دهد که شامل عملیات رمزنگاری (رمزگذاری یا رمزگشایی) به منظور گمنام کردن پیام‌ها و یک جایگشت تصادفی روی پیام‌های ورودی به منظور حذف ارتباط بین خروجی‌ها و ورودی‌های سرورها است. این خروجی‌ها به صورت متوالی به سرور بعدی منتقل می‌شود. به این ترتیب سایر سرورها هم عملیات مخلوط کردن را انجام می‌دهند و در نهایت بعد از سرور نهایی خروجی‌ها که شامل پیام‌های رمزنشده هستند به گیرنده و یا گیرنده‌ها تحویل داده می‌شوند.

شبکه‌های مخلوط با توجه به شیوه‌ی طراحی استفاده شده در آن‌ها دارای ویژگی‌های متعددی هستند، که از آن جمله می‌توان به خاصیت‌های حفظ گمنامی[۳] فرستنده‌ها، پایدار بودن[۴] در برابر خطای سرورها، قابلیت بررسی[۵] صحت عملکرد سرورها و ... اشاره کرد. هدف مطلوب در طراحی شبکه‌های مخلوط رسیدن به خاصیت‌های ذکر شده به همراه کارایی و سرعت محاسباتی بالا است.

[1] Anonymous channel
[2] Mix-net
[3] anonymity
[4] robustness
[5] verifiability



شکل1: ساختار شبکه‌ی مخلوط

در شکل 1 ساختار کلی شبکه‌ی مخلوط و سرورهای تشکیل دهنده رسم شده است.

Chaum اولین شبکه مخلوط به نام شبکه‌ی مخلوط رمزگشا را در [1] معرفی کرد که در آن فرستنده‌ها پیام‌های ورودی خود را با استفاده از کلید عمومی تمامی سرورها با شروع از کلید عمومی سرور آخر رمزگذاری می‌کنند، در طی عملیات رمزگشایی هر سرور، یک لایه از رمزگذاری را حذف می‌کند. به این ترتیب خروجی سرور آخر مجموعه‌ای از پیام‌های رمزنشده خواهد بود. مشکل عمده این شبکه‌ی مخلوط پایدار نبودن نسبت به سرورهای خطاکار است. یعنی اگر یک سرور از ادامه عملیات رمزگشایی خودداری کند کل عملیات مخلوط کردن ناتمام خواهد ماند. برای رسیدن به خاصیت پایدار بودن در برابر سرورهای خطاکار، Park و همکاران در [3] شبکه‌ی مخلوط جدیدی معرفی کردند که به جای عملیات رمزگشایی به انجام عملیات رمزگذاری مجدد می‌پردازد. به شبکه‌ی مخلوط با رمزگذاری مجدد معرفی شده در [3] در [5 و 4] دو حمله انجام شد که یکی حمله‌ی غیر فعال و دیگری حمله‌ی فعال بود. شبکه‌ی مخلوط با رمزگذاری مجدد جدیدی در [6] معرفی شد که در برابر حملات فوق الذکر پایدار شده بود و از آن زمان به بعد هدف عمده‌ی طراحان شبکه‌ی مخلوط رسیدن به شبکه‌ی مخلوطی با قابلیت بررسی و حجم محاسباتی قابل قبولی بود تا بتواند صحت عملیات شبکه‌ی مخلوط، برای رسیدن به گمنامی را اثبات کند. اولین شبکه‌ی مخلوط با قابلیت بررسی عمومی در [7] مطرح شد که از اثبات‌های صفر دانایی مبتنی بر برش و انتخاب[1] استفاده کرده بود ولی این طرح دارای حجم محاسباتی بسیار بالایی بود. بعد از آن شبکه‌های مخلوط بسیار زیادی با قابلیت بررسی عمومی و با کارامدی بالاتر مطرح شدند، به عنوان مثال در [9 و 8] دو طرح ارایه شدند که شبیه طرح ارایه شده در [7] بودند ولی از کارایی بهتری برخوردار بودند. طرح‌های Millimix [10] و MIP-2 [11] بر اساس شبکه‌های چرخشی[2] و اثبات‌های صفر دانایی کارامدی معرفی شدند.

از شبکه‌های مخلوط با کارایی بسیار بالا طرح‌های Furukawa [12] و Neff [13] هستند که طرح اولی بر اساس ماتریس‌های چرخشی و اثبات‌های صفر دانایی مرتبط با آن بنا شده است و طرح دومی بر اساس استفاده از این ایده است که ریشه‌های چند جمله‌ای تحت جایگشت ثابت می‌مانند و تغییر نمی‌کنند.

شبکه‌ی مخلوط Flash [14] یکی از شبکه‌های مخلوط با قابلیت بررسی عمومی با کارایی بالا است که طبق ادعای نویسندگان مقاله دارای خاصیت حفظ گمنامی و پایدار بودن و قابلیت بررسی عمومی است و نسبت به طرح‌های مشابه دارای کارایی بالایی است[14]. در [15] حمله‌ای به این شبکه‌ی مخلوط مطرح شده است که تنها خاصیت پایدار بودن این شبکه‌ی مخلوط را نقض کرده است، بدون اینکه گمنامی فرستنده‌ها نقض شود. ما در این مقاله ابتدا به معرفی اجمالی شبکه‌ی مخلوط Flash خواهیم پرداخت و در ادامه دو حمله‌ی جدید به این شبکه مخلوط معرفی می‌شود که حمله‌ی اول پایداری شبکه‌ی مخلوط و حمله‌ی دوم گمنامی کلیه فرستنده‌ها را از بین می‌برد و در نهایت چند پیشنهاد برای جلوگیری از اعمال این حملات به شبکه‌ی مخلوط Flash پیشنهاد می‌شود.

۲- شبکه‌ی مخلوط با رمزگذاری مجدد[3]

شبکه‌ی مخلوط به منظور تامین گمنامی فرستنده‌ها در مقابل گیرنده‌های پیام طراحی می‌شود و معمولاً هر شبکه‌ی مخلوط شامل چندین سرور مخلوط کننده است که هر سرور یک مجموعه عملیات مخلوط کردن[4] را انجام می‌دهد، عملیات مخلوط کردن شامل یک جایگشت تصادفی و یک عمل رمزنگاری است. در این شبکه‌های مخلوط هر سرور باید روی پیام‌های ورودی خود یک عمل رمزگذاری مجدد اعمال کند، بنابراین سیستم رمزگذاری استفاده شده در این نوع شبکه‌های مخلوط باید دارای خاصیت همریختی باشد. برای این منظور به طور معمول از سیستم رمزنگاری الجمال استفاده می‌شود. در ادامه این نوع شبکه‌ی مخلوط به طور اجمالی معرفی می‌شود.

برای انجام عملیات رمزنگاری در سرورها ابتدا باید کلید عمومی و خصوصی سیستم رمزگذاری الجمال تولید شود، برای این منظور فرض کنید p,q دو عدد اول بزرگ باشند به طوری که $q|p-1$ و Z_p^* گروه ضربی تعریف شده به پیمانه‌ی p و G_q زیر گروهی از مرتبه‌ی q و g مولد این زیر گروه باشد. مقادیر g,p,q به صورت عمومی اعلام می‌شوند، برای تشکیل کلید عمومی و کلید خصوصی، تمامی سرورها با همکاری یکدیگر و با استفاده از روش‌های تسهیم راز با قابلیت بررسی[5] روی کلید عمومی $y=g^x$ توافق می‌کنند و کلید خصوصی x، بین سرورها تسهیم

[1] Cut and choose
[2] Permutation network
[3] Re-encryption
[4] mixing
[5] Verifiable secret sharing (VSS)

می‌شود. برای آشنایی بیشتر با روش تولید کلید و توزیع کلید به [16] مراجعه شود، به این ترتیب سیستم الجمال با پارامترهای عمومی (g, p, q, y) اعلام می‌شود. تمامی محاسبات به پیمانه‌ی p انجام می‌شود که در ادامه‌ی مقاله از ذکر پیمانه صرفنظر می‌شود.

بعد از تولید کلیدهای عمومی و خصوصی به معرفی عملیات مخلوط کردن و ساختار شبکه‌ی مخلوط با رمزگذاری مجدد می‌پردازیم.

فرض کنید که شبکه‌ی مخلوط از N فرستنده و k سرور تشکیل شده باشد. در این شبکه‌ی مخلوط سرور M_i ام را با i نمایش می‌دهیم. فرستنده‌ی i ام پیام خود یعنی m_i را با استفاده از سیستم رمزگذاری الجمال به صورت دوتایی

$(a_i = g^{\beta_i}, b_i = m_i y^{\beta_i})$ رمز می‌کند. به این ترتیب N فرستنده پیام‌های خود را به صورت مجموعه‌ی $\{(a_1, b_1), \dots (a_N, b_N)\}$ به سرور اول تحویل می‌دهند. سرور اول با دریافت N پیام عملیات رمزگذاری مجدد را با انتخاب تصادفی نماهای $\beta_i \in Z_q$ و تولید

$(\acute{a}_i = g^{\alpha_i} g^{\beta_i}, \acute{b}_i = m_i y^{\alpha_i} y^{\beta_i})$ برای $1 \leq i \leq N$

انجام می‌دهد. در نهایت با انتخاب π_1 به عنوان یک جایگشت تصادفی، مجموعه‌ی $\pi_1\{(\acute{a}_1, \acute{b}_1), \dots (\acute{a}_N, \acute{b}_N)\}$ را به عنوان خروجی خود تشکیل می‌دهد، و به سرور بعدی منتقل می‌کند. سایر سرورها هم به طور مشابه عملیات مخلوط کردن را انجام می‌دهند. در نهایت بعد از تشکیل خروجی سرور نهایی، با استفاده از همکاری تمامی سرورها عملیات رمزگشایی انجام می‌پذیرد و مجموعه‌ی پیام‌های خروجی تشکیل خواهد شد [16].

۳- مروری بر شبکه مخلوط Flash

ابتدا مروری مختصر به قسمت‌های مختلف شبکه‌ی مخلوط Flash [14] خواهیم داشت. سعی بر این شده است که توضیحات ارایه شده تا حد امکان به مقاله‌ی اصلی نزدیک باشد ولی برای جلوگیری از طولانی شدن مطالب بعضی از جزئیات غیر مرتبط با حملات حذف شده و بعضی از نمادهای به کار رفته در مقاله‌ی اصلی هم تغییر یافته است، برای آشنایی بیشتر با جزئیات خواننده می‌تواند به [14] مراجعه کند.

شبکه‌ی مخلوط Flash همانند سایر شبکه‌های مخلوط با رمزگذاری مجدد از K سرور مخلوط‌کننده تشکیل شده است که در این شبکه‌ی مخلوط سرور i ام را با M_i نمایش می‌دهیم. این شبکه‌ی مخلوط دارای N پیام ورودی است، که در ادامه ساختار این شبکه‌ی مخلوط را توضیح خواهیم داد:

ورودی شبکه‌ی مخلوط Flash [14] شامل N پیام رمزشده به شکل $\{(a_1, b_1), \dots, (a_N, b_N)\}$ است که در آن (a_i, b_i) پیام رمزشده‌ی فرستنده‌ی i ام (m_i) با سیستم رمزگذاری الجمال است. این شبکه‌ی مخلوط در حالت کلی شامل دو پروتکل است، که در بخش‌های بعدی توضیح داده می‌شود:

1) پروتکل کور سازی[1]
2) پروتکل حذف کورسازی[2]

3-1 پروتکل کور سازی

در شبکه‌ی مخلوط Flash ابتدا پروتکل کورسازی در 4 مرحله انجام می‌گیرد:

(1) تولید و جاسازی پیام‌های اضافی[3]:

دو پیام اضافی $(a_{N+2}, b_{N+2}), (a_{N+1}, b_{N+1})$ با همکاری تمامی اعضا ساخته می‌شوند به طوری که $a_{N+1}, b_{N+1}, a_{N+2}$ و b_{N+2} عضوهای تصادفی G_q باشند. بنابراین شبکه‌ی مخلوط دارای $N+2$ پیام ورودی است که به صورت مجموعه‌ی L_0 نمایش داده می‌شود.

$$L_0 = \{(a_1, b_1), \dots, (a_{N+2}, b_{N+2})\}$$

(2) تکثیر[4]:

به تعداد $\tau \geq 2$ روگرفت (کپی) از L_0 تولید می‌شود که به صورت $L_{1,0}, L_{2,0}, \dots, L_{\tau,0}$ نمایش داده می‌شوند، در واقع این کار برای افزایش امنیت شبکه‌ی مخلوط و جلوگیری از اعمال برخی از حملات طراحی شده است. τ مجموعه ورودی سرور jام را به طور قراردادی به صورت $L_{t,j}$ نشان می‌دهیم، که در آن $1 \leq t \leq \tau$ بیانگر مجموعه‌ها و $1 \leq j \leq K$ نشانگر سرورها است

$$L_{t,j} = \begin{Bmatrix} (a_{1,t,j}, b_{1,t,j}), \dots (a_{i,t,j}, b_{i,t,j}) \dots \\ , (a_{N+2,t,j}, b_{N+2,t,j} \end{Bmatrix}$$

(3) اولین رمزگذاری مجدد[5]

برای K سرور موجود در پروتکل، سرور jام مجموعه‌های $L_{1,(j-1)}, L_{2,(j-1)}, \dots, L_{\tau,(j-1)}$ را به عنوان ورودی دریافت و مشابه مرحله‌ی 2 (عملیات مخلوط کردن) با انجام عملیات رمزگذاری مجدد، مجموعه‌های $L_{1,j}, L_{2,j}, \dots, L_{\tau,j}$ را به عنوان خروجی تولید می‌کند که برای $1 \leq t \leq \tau$ مجموعه‌های $L_{t,j}$ رمزگذاری شده مجدد و جایگشت یافته‌ی مجموعه $L_{t,(j-1)}$ است، که با استفاده از π_{tj} جایگشت تصادفی و α_{itj} نمای تصادفی

[1] Blinding protocol
[2] Unblinding protocol
[3] dummy
[4] duplication
[5] first re-encryption

انتخابی توسط سرور jام روی $N+2$ پیام مجموعه $L_{\tau,(j-1)}$ تولید می‌شود. سایر سرورها به طور مشابه عمل می‌کنند و در نهایت خروجی سرور آخر برابر با مجموعه‌های $L_{\tau,k}$ خواهد بود که به طور قرار دادی به صورت زیر نشان می‌دهیم:

$$L'_{1,0} = L_{1,K}, L'_{2,0} = L_{2,K}, \dots, L'_{\tau,0} = L_{\tau,K}$$

(4) دومین رمزگذاری مجدد[1]:

تمامی K سرور مخلوط کننده رمزگذاری مجددی، مشابه رمزگذاری مجدد اول، برای مجموعه‌های ورودی $L'_{1,0}, L'_{2,0}, \dots, L'_{\tau,0}$ انجام می‌دهند. مجموعه‌های $L'_{1,0}, L'_{2,0}, \dots, L'_{\tau,0}$ را به عنوان ورودی سرور اول و مجموعه‌های $L'_{1,1}, L'_{2,1}, \dots, L'_{\tau,1}$ را به عنوان خروجی سرور اول در نظر می‌گیریم. خروجی سرور آخر را با مجموعه‌های $L'_{1,k}, L'_{2k}, \dots, L'_{\tau,k}$ نشان می‌دهیم.

2-3 پروتکل حذف کور سازی

در این پروتکل، سرورهای شبکه‌ی مخلوط صحت عملیات خود در قسمت کور سازی را به اثبات می‌رسانند. این پروتکل از 5 مرحله تشکیل یافته است.

(1) تایید اولین رمزگذاری مجدد[2]

هر سرور مخلوط کننده تمامی مقادیر مخفی مربوط به رمزگذاری مجدد اولیه را آشکار می‌کند برای این کار سرور مخلوط کننده‌ی iام نمای توان‌ها $\alpha_{(t,j)}$ و جایگشت تصادفی $\pi_{t,j}$ اعمال شده روی مجموعه‌های ورودی خود را برای

$1 \leq i \leq N+2, 1 \leq t \leq \tau$ آشکار می‌کند که صحت آن توسط سایر سرورهای مخلوط کننده بررسی می‌شود. به این ترتیب هر سرور می‌تواند جایگشت مرکب[3] $\pi_t = \pi_{t1} o \dots o \pi_{tk}$ و نمای رمزگذاری مجدد ترکیبی $\beta_{i,t}$ را حساب کند و به این ترتیب می‌توان رمزگذاری مجدد مجموعه‌های $L'_{1,0}, L'_{2,0}, \dots, L'_{\tau,0}$ را از روی مجموعه‌های ورودی L_0 به دست آورد.

$$L'_{1,0} = \pi_1\left((a_1 g^{\beta_{1,1}}, b_1 y^{\beta_{1,1}}), \dots, (a_{N+2} g^{\beta_{N+2,1}}, b_{N+2} y^{\beta_{N+2,1}})\right)$$

$$\vdots$$

$$L'_{\tau,0} = \pi_\tau\{(a_1 g^{\beta_{1,\tau}}, b_1 y^{\beta_{1,\tau}}), \dots, (a_{N+2} g^{\beta_{N+2,\tau}}, b_{N+2} y^{\beta_{N+2,\tau}})\}$$

(2) تایید مقادیر پیام‌های اضافی[4]

در این قسمت هر سرور مخلوط کننده به اثبات صحت عملیات خود روی پیام‌های اضافی در مرحله‌ی دومین رمزگذاری مجدد می‌پردازد.

ابتدا هر سرور جایگشت انجام شده روی دو پیام اضافی در رمزگذاری مجدد دوم را آشکار می‌کند، با توجه به مرحله‌ی تایید اولین رمزگذاری مجدد مکان پیام‌های اضافی در مجموعه‌های $L'_{1,0}, L'_{2,0}, \dots, L'_{\tau,0}$ معلوم است،

سپس هر سرور، نمای انتخابی خود برای انجام رمزگذاری مجدد را برای پیام اضافی دوم را آشکار می‌کند تا صحت انجام عملیات مخلوط کردن برای پیام اضافی دوم معلوم شود. برای عنصر اضافی اول نمای انتخابی رمزگذاری مجدد دوم آشکار نمی‌شود بلکه با استفاده از اثبات‌های صفر دانایی صحت آن تایید می‌شود[14].

(3) تایید حاصل ضرب[5]

هر سرور مخلوط کننده باید اثبات کند که در مورد حاصل ضرب تمامی عناصر مجموعه‌های ورودی خود به جز عنصر اضافی دوم در مرحله دومین رمزگذاری مجدد درست رفتار کرده است:

$(A_{t,j}, B_{t,j})$ = حاصلضرب تمام عناصر مجموعه‌های $L'_{t,j-1}$ به جز عنصر اضافی دوم

$(C_{t,j}, D_{t,j})$ = حاصلضرب تمام عناصر مجموعه‌های $L'_{t,j}$ به جز عنصر اضافی دوم

در این صورت به عنوان نمونه برای سرور اول و

$1 \leq i \leq N+2, 1 \leq t \leq \tau$ خواهیم داشت:

$$L'_{t,0} = \{(a'_{1,t,0}, b'_{1,1,j}), \dots, (a'_{N+2,t,0}, b'_{N+2,t,0})\}$$

$$L'_{t,1} = \{(a'_{1,t,1}, b'_{1,t,1}), \dots, (a'_{N+2,t,1}, b'_{N+2,t,1})\}$$

$$A_{t,j} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} a'_{i,t,0}$$

$$B_{t,j} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} b'_{i,t,0}$$

$$C_{t,j} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} a'_{i,t,1}$$

$$D_{t,j} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} b'_{i,t,1}$$

که k, k اندیس‌های نشان دهنده‌ی عناصر پیام اضافی دوم در مجموعه‌ها بوده، و اگر رمزگذاری مجدد دوم به درستی انجام شود باید داشته باشیم:

[1] Second re-encryption
[2] Verifying the first re-encryption
[3] aggregation
[4] Verification of dummy value
[5] Verification product

$$C_{t,j} = A_{t,j} g^{u_{t,j}}$$
$$D_{t,j} = B_{t,j} y^{u_{t,j}}$$

که سرور اول مقادیر $u_{t,1}$ را منتشر و سایر سرورها هم به طریق مشابه عمل می‌کنند:

(4) تایید جایگشت نسبی[1]

سرور jام ثابت می‌کند که $L'_{1,j}$ یک نمونه‌ی جایگشت یافته و رمزگذاری مجدد شده‌ی مجموعه‌های $L'_{t,j}$ برای $2 \le t \le \tau$ در مرحله‌ی دومین رمزگذاری مجدد است.

سرور j ام برای مجموعه‌های خروجی خود $L'_{t,j}$ به صورت زیر عمل می‌کند:

$$L'_{1,j} = \{(a'_{1,1,j}, b'_{1,1,j}), \dots, (a'_{N+2,1,j}, b'_{N+2,1,j})\}$$
$$L'_{t,j} = \{(a'_{1,t,j}, b'_{1,t,j}), \dots, (a'_{N+2,t,j}, b'_{N+2,t,j})\}$$

یعنی باید اثبات کند که $\phi_{t,j}$ و $\gamma_{i,t,j}$ برای $1 \le i \le N+2$ وجود دارند که در رابطه‌ی زیر صدق کنند:

$$L'_{1,j} = \phi_{t,j}\left\{\begin{pmatrix} (a'_{1,t,j} g^{\gamma_{1,t,j}}, b'_{1,t,j} y^{\gamma_{1,t,j}}), \dots, \\ (a'_{N+2,t,j} g^{\gamma_{N+2,t,j}}, b'_{N+2,t,j} y^{\gamma_{N+2,t,j}}) \end{pmatrix}\right\}$$

قابل ذکر است که سرور اول با توجه به تایید رمزگذاری اول و دانستن جایگشت مرکب و نماهای ترکیبی قادر به محاسبه‌ی مقادیر $\phi_{t,j}$ و $\gamma_{i,t,j}$ است و به همین ترتیب سایر سرورها هم باید این مقادیر را محاسبه و منتشر کنند[۱۴].

(5) تولید خروجی

در نهایت در صورت صحت تمام بررسی‌ها، پیام‌های اضافی کنار گذاشته می‌شوند و پس از رمزگشایی سایر پیام‌ها (مراجعه شود به بخش 2) خروجی شبکه‌ی مخلوط که شامل جایگشتی تصادفی از پیام‌های $(m_1, m_1, \dots, m_N)$ تولید می‌شود.

3-3 ویژگی‌های شبکه‌ی مخلوط Flash

در [14] بیان شده است که شبکه‌ی مخلوط Flash پایدار است یعنی اگر مجموعه‌ای از سرورهای نادرست سبب تولید خروجی نادرست شوند آن‌گاه با احتمال بسیار بالایی[2] این خطا توسط سرورهای درست کار تشخیص و سرورهای خطاکار شناسایی می‌شوند.

همچنین در [14] ادعا شده است که شبکه‌ی مخلوط Flash گمنامی فرستنده‌ها را تا وقتی که حداقل یک سرور درست‌کار باشد حفظ می‌کند، یعنی دشمن برای تشخیص رابطه‌ی ورودی و خروجی راه بهتری از روش حدس و خطا ندارد.

۴- حملات قبلی معرفی شده به شبکه‌ی مخلوط Flash

تا کنون دو حمله به این شبکه‌ی مخلوط مطرح شده است که در نتیجه‌ی اعمال حمله‌ی اول [15] پایدار بودن بودن این شبکه‌ی مخلوط نقض شده است و در دومین حمله [18] گمنامی پیام‌ها با کمک سرور اول و سرور آخر از بین می‌رود. ما در این مقاله دو حمله‌ی جدید به این شبکه‌ی مخلوط مطرح می‌کنیم و نشان خواهیم داد که حمله‌ی مطرح شده در [18] قابل اعمال به شبکه‌ی مخلوط Flash نیست و توسط شبکه‌ی مخلوط شناسایی می‌شود.

4-1 حمله بر اساس سرور اول خاطی

در [15] حمله‌ای روی شبکه‌ی مخلوط Flash مطرح شده است که در آن سرور مخلوط کننده‌ی اول می‌تواند مانع تولید خروجی‌های صحیح شود:

سرورها اولین مرحله‌ی رمزگذاری مجدد را به درستی انجام می‌دهند و مجموعه‌های $L'_{1,0} = L_{1,t+1}, L'_{2,0} = L_{2,t+1}, \dots, L'_{\tau,0} = L_{\tau,t+1}$ را تولید می‌کنند (اولین رمزگذاری مجدد از پروتکل کورسازی).

در مرحله‌ی دومین رمزگذاری مجدد، سرور اول خطا کار می‌تواند به طریق زیر عمل کند:

1) سرور اول اعداد تصادفی $\alpha_1, \alpha_2 \dots \alpha_N$ را طوری انتخاب می‌کند که $\alpha_1 + \alpha_2 + \dots + \alpha_N = 1 \bmod q$ برقرار باشد.

2) به ازای مجموعه‌های خروجی اولین رمزگذاری مجدد یعنی $L'_{t,0} = \{(a'_{1,t,0}, b'_{1,t,0}), \dots, (a'_{N+2,t,0}, b'_{N+2,t,0})\}$ برای $1 \le t \le \tau$ ، سرور اول می‌تواند مقادیر زیر را حساب کند:

$$\tilde{A}_i = a'_{1,t,0} \dots a'_{N+2,t,0} / a_{N+1} a_{N+2}$$
$$\tilde{B}_i = b'_{1,t,0} \dots b'_{N+2,t,0} / b_{N+1} b_{N+2}$$

3) سرور اول مجموعه‌های زیر را به عنوان خروجی خود منتشر می‌کند:

$$L'_{t,1} = \theta_{t,j}\{(\tilde{A}_i^{\alpha_1} g^{\tau_{1,t}}, \tilde{B}_i^{\alpha_1} y^{\tau_{1,t}}), \dots, (\tilde{A}_i^{\alpha_N} g^{\tau_{N,t}} \tilde{B}_i^{\alpha_N} y^{\tau_{N,t}})$$
$$(a_{N+1} g^{\tau_{N+1,t}}, b_{N+1} y^{\tau_{N+1,t}}), (a_{N+2} g^{\tau_{N+2,t}}, b_{N+2} y^{\tau_{N+2,t}})\}$$

که $\theta_{t,j}$ ، $t_{i,t}$ مقادیر انتخابی توسط سرور اول هستند،در واقع به جای عناصر اضافی از همان مقادیر معلوم در مجموعه ورودی (L_0) استفاده شده است.

اگر بقیه سرورها به شیوه‌ی صحیح ادامه بدهند این مقادیر از تمام مراحل تایید عبور خواهند کرد.

چون سرورها مرحله‌ی اول رمزگذاری را به درستی انجام داده‌اند می‌توانند مرحله‌ی تایید اولین رمزگذاری مجدد را انجام دهند و به

[1] Relative sorting

[2] overwhelming

این ترتیب مقادیر توان ها ($\beta_{i,t}$) و جایگشت‌های ترکیبی (π_t) معلوم خواهند شد. با معلوم بودن این جایگشت‌های ترکیبی سرور اول برای مرحله‌ی تایید مقادیر عناصر اضافی با محاسبه مقادیر $z_{N+2,t}, z_{N+1,t}$ به عنوان اختلاف توان پیام‌های اضافی در مجموعه‌های L' و خروجی سرور اول خواهیم داشت:

$$z_{N+1,t} = t_{N+1,t} - \beta_{N+1,t}$$

$$z_{N+2,t} = t_{N+2,t} - \beta_{N+2,t}$$

به همین ترتیب سرور اول با انتشار مقادیر $z_{N+2,t}$ و اثبات صفر دانایی برای مقادیر $z_{N+1,t}$ قادر به عبور از مرحله تایید مقادیر عناصر اضافی است. سایر سرورها هم به شیوه‌ی معمول خود ادامه می‌دهند.

سرور اول برای عبور از مرحله‌ی تایید حاصل ضرب با انتخاب مقادیر $\mu_t = t_{1,t} + \cdots + t_{N+1,t} + \beta_{N+2,t}$ از مرحله‌ی تایید حاصلضرب‌ها هم عبور می‌کند.

$$A_{i,t} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} a'_{i,t,0} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} (a_{i,t,0})\, g^{\beta_{i,t}}$$

$$C_{i,t} = \prod_{\substack{i=1 \\ i \neq k}}^{N+2} a'_{i,t,0} = \left(\prod_{\substack{i=1 \\ i \neq K}}^{N} A_i^{a_i} g^{t_{i,t}} \right) a_{N+1} g^{t_{N+1,t}}$$

$$\prod_{\substack{i=1 \\ i \neq k}}^{N+2} (a_{i,t,0})\, g^{\beta_{N+2,t}} g^{\sum_{i=1}^{N+1} t_{i,t}}$$

در نتیجه، $\mu_t = t_{1,t} + \cdots + t_{N+1,t} + \beta_{N+2,t}$ به دست خواهد آمد. و اگر سایر سرورها به روال معمول کار را دنبال کنند می‌توان از مرحله تایید حاصل ضرب عبور کرد. چون سرورها در مرحله‌ی دومین رمزگذاری مجدد مجموعه‌های خروجی خود را به صورت صحیح و رمزگذاری مجدد شده تولید کرده اند ، مرحله‌ی تایید جایگشت نسبی هم سپری می‌شود. به این ترتیب خروجی شبکه‌ی مخلوط به جای $(m_1, m_2, \dots m_N)$ برابر $((m_1 m_2 \dots m_N)^{a_1}, \dots, (m_1 m_2 \dots m_N)^{a_N})$ خواهد بود.

2-4 حمله بر اساس سرور اول و سرور نهایی خاطی

در [18] حمله‌ای مطرح شده است که با همکاری سرور اول و سرور نهایی انجام می‌شود و از این نکته استفاده می‌کند که در این پروتکل هیچ بررسی در مورد اینکه پیام‌ها به گروه G_q تعلق دارند یا نه انجام نمی‌شود (البته این مشکل در اکثر پروتکل‌های شبکه‌ی مخلوط وجود دارد که برای توضیحات بیشتر به [19] مراجعه شود) .

مراحل این حمله به طور خلاصه در ادامه آورده شده است:

مرحله‌ی اولین رمزگذاری مجدد مثل طرح Flash انجام می‌شود و در مرحله‌ی دومین رمزگذاری مجدد سرور اول خاطی دو عنصر را به دلخواه انتخاب می‌کند و با ضرب در مقادیر $\delta, \delta^{-1} \in Z_p$ آن‌ها را نشان گذاری[1] می‌کند در این حالت باید $\tilde{\delta}^q \neq 1 \bmod p$ باشد و بقیه کارها طبق روند پروتکل خواهد بود. در نهایت سرور آخر با استفاده از توان رسانی q و این نکته که برای هر عضو مانند $b \in Z_q$ باید داشته باشیم $b^q = 1 \bmod p$، می‌تواند این دو عنصر نشان دار شده را شناسایی و با حذف مقادیر $\tilde{\delta}, \tilde{\delta}^{-1}$ پیام‌ها را در خروجی قرار می‌دهد. به این ترتیب گمنامی دو پیام مذکور از بین می‌رود و در [18] اشاره شده است که چون عناصر اضافی تعداد کمی دارند احتمال موفقیت این حمله بالا است و این عناصر از تمام بررسی‌ها عبور خواهند کرد ولی این حمله ناموفق است زیرا:

پیام‌هایی که نشان‌گذاری شده‌اند باید از مرحله‌ی تایید جایگشت نسبی از پروتکل حذف کورسازی عبور کنند. به این منظور باید رابطه‌ی بین پیام‌ها در مجموعه‌های مختلف معلوم باشد. برای مثال در دومین مرحله‌ی رمزگذاری اگر سرور اول بخواهد اولین عضو مجموعه‌ی $L'_{1,0}$ را با $\tilde{\delta}$ نشان‌گذاری کند، این عضو تغییر یافته به فرم $(\tilde{\delta} a'_{1,1,0}, b'_{1,1,0})$ خواهد بود و سرور اول باید مشابه این پیام در سایر مجموعه‌های $L'_{t,0}$ را نیز به طور متناظر با $\tilde{\delta}$ نشان‌گذاری کند تا از مرحله‌ی تایید جایگشت نسبی عبور کند. در غیر این صورت اگر فرض کنیم، پیام متناظر پیام اول در مجموعه‌ی $L'_{1,0}$ در سایر مجموعه‌ها پیام x_t باشد، این پیام به فرم $(a'_{x_t,t,0}, b'_{x_t,t,0})$ خواهد بود و همانطور که مشاهده می‌شود به دلیل این که $\tilde{\delta} \in G_q$ را نمی‌توان به فرم توانی از مولد g بیان کرد و لذا امکان گذر از مرحله‌ی تایید نسبی در پروتکل حذف کور سازی از بین می‌رود. اگر سرور اول پیام‌ها را به طور تصادفی از مجموعه‌ها انتخاب کند احتمال انتخاب پیام‌های متناظر خیلی کم است و همچنین شانس انتخاب پیام‌های اضافی هم وجود دارد.

5- حملات جدید

در این بخش دو حمله‌ی جدید به شبکه‌ی مخلوط Flash معرفی می‌شود که حمله‌ی اول پایداری شبکه‌ی مخلوط را از بین می‌برد و حمله‌ی دوم گمنامی فرستنده‌ها را از بین می‌برد.

1-5 حمله بر اساس سرور اول خاطی

این یک حمله‌ی فعال جدید است که با استفاده از سرور اول خاطی انجام می‌شود و پایداری طرح را از بین می‌بردو سبب تولید خروجی نادرست در خروجی شبکه‌ی مخلوط خواهد شد.

در این حمله سرورها مرحله‌ی اول رمزگذاری مجدد را به درستی انجام می‌دهند و مجموعه‌های L' را تولید می‌کنند و در مرحله‌ی

[1] tag

دوم رمزگذاری سرور اول به جای استفاده از مجموعه‌های $L_t^{'}$ از مجموعه‌ی اولیه L_0 استفاده می‌کند و هر گونه تغییر در این مجموعه که سبب حفظ حاصل ضرب مولفه‌ها شود می‌تواند در پروتکل و این حمله به کار گرفته شود. برای نمونه سرور اول از دو عضو $(a_1,b_1),(a_2,b_2)$ متناظر با پیام‌های m_1,m_2 به شرح زیر استفاده می‌کند:

$$\{(a_1a_2,b_1b_2),(1,1),\dots,(a_N,b_N),(a_{N+1},b_{N+1}),(a_{N+2},b_{N+2})\}$$

سرور اول خطا کار فقط 2 پیام $(a_1,b_1),(a_2,b_2)$ را با پیام‌های $(a_1a_2,b_1b_2),(1,1)$ جایگزین می‌کند و بقیه‌ی پیام‌ها طبق مجموعه اولیه L_0 خواهند بود. سرور اول با انتخاب مقادیر θ_t و $t_{i,t}$ مجموعه‌های خروجی خود $L_{t,1}^{'}$ تشکیل می‌دهد.

$$L_{t,1}^{'}=\theta_t\{(a_1a_2g^{r_{1,t}},b_1b_2y^{r_{1,t}}),\dots,(1g^{r_{N,t}}1y^{r_{N,t}})(a_{N+1}g^{r_{N+1,t}},b_{N+1}y^{r_{N+1,t}}),(a_{N+2}g^{r_{N+2,t}},b_{N+2}y^{r_{N+2,t}})\}$$

سایر سرورها هم ادامه پروتکل را طبق روال معمول انجام می‌دهند.

تایید مرحله‌ی اولین رمزگذاری مجدد به درستی انجام می‌شود چون روند پروتکل به طور معمول طی می‌شود به این ترتیب جایگشت و توان‌های ترکیبی معلوم می‌شوند.

با استفاده از جایگشت و توان‌های ترکیبی و مشخص شدن عناصر اضافی در انتهای اولین رمزگذاری مجدد برای گذر از مرحله‌ی تایید مقادیر عناصر اضافی، کافی است سرور اول مقادیر زیر را محاسبه کند:

$$z_{N+1,t}=t_{N+1,t}-\beta_{N+1,t}$$
$$z_{N+2,t}=t_{N+2,t}-\beta_{N+2,t}$$

مطابق مرحله‌ی تایید مقادیر عناصر اضافی در بخش 2-3 مقدار $z_{N+2,t}$ را منتشر و برای $z_{N+1,t}$ از اثبات‌های صفر دانایی استفاده کند.

سایر سرورها هم به شیوه‌ی معمول خود تاییدها را انجام می‌دهند.

برای تایید حاصل ضرب‌ها در زیر بخش 2-3 کافی است سرور اول مقادیر μ_t را طبق روابط زیر محاسبه و منتشر کند:

$$A_{t,j}=a_{1,t}^{'}\times\dots\times a_{N+1,t}^{'}=a_1\times\dots\times a_{N+1}\times g^{(\beta_{1,t}+\cdots\beta_{N+1,t})}$$
$$B_{t,j}=(a_1a_2)\times 1\times\dots\times a_{N+1}\times g^{r_{1,t}+\cdots+r_{N+1,t}}$$
$$\mu_t=t_{1,t}+\cdots+t_{N+1,t}-(\beta_{1,t}+\cdots\beta_{N+1,t})$$

در این صورت تایید نسبی جایگشت‌ها هم به درستی انجام می‌شود چون مجموعه‌های خروجی جایگشت دوران یافته‌ی همدیگرند و سرورها می‌توانند مطابق زیر بخش 2-3 از این مرحله‌ی تایید عبور کنند.

در نهایت در مرحله‌ی تولید پیام‌های خروجی، پیام‌ها جایگشتی از پیام‌های $(m_1m_2),1,\dots m_N$ خواهد بود. لازم به توضیح است که با استفاده از سایر مجموعه‌های مشابه مجموعه اولیه L_0 که حاصل ضرب مولفه‌ها تغییر نکرده باشد می‌توان حملات مشابهی روی این شبکه‌ی مخلوط انجام داد.

2-5 حمله بر اساس سرور اول و سرور اول خاطی

این حمله‌ی در واقع ترکیب حملات بخش 4-1 و 4-2 است و مشکلات حمله بخش 4-2 را هم حل کرده است و سبب شکسته شدن گمنامی کل طرح می‌شود. این حمله با همکاری سرور اول و آخر انجام می‌شود. سرورها مرحله اولین رمزگذاری مجدد را به درستی انجام می‌دهند و برای مرحله دومین رمزگذاری مجدد، سرور اول مشابه حمله‌ی سوم، از مجموعه‌ی L_0 استفاده می‌کند و خروجی خود را به شکل زیر تولید می‌کند (اگر مقدار N زوج باشد) :

$$L_{t,1}^{'}=\theta_t\{(\delta_1a_1g^{r_{1,t}},b_1y^{r_{1,t}}),(\delta_1^{-1}a_2g^{r_{1,t}},b_1y^{r_{1,t}}),\dots$$
$$(\delta_{i/2}a_ig^{r_{1,t}},b_iy^{r_{1,t}}),(\delta_{i/2}^{-1}a_{i+1}g^{r_{1,t}},b_{i+1}y^{r_{1,t}}),\dots$$
$$,(\delta_{N/2}\,a_{N,t}g^{r_{N,t}}b_Ny^{r_{N,t}}),$$
$$(\delta_{N/2}^{-1}a_{N+1}g^{r_{N+1,t}},b_{N+1}y^{r_{N+1,t}}),$$
$$(a_{N+2}g^{r_{N+2,t}},b_{N+2}y^{r_{N+2,t}})\}$$

که در آن $\delta_i\in Z_p$ و $\delta_i^q\neq 1 \bmod p$ و $\delta_i^q\neq\delta_j^q\ mod\ p$ برای $i\neq j$ است، در واقع سرور اول مجموعه‌های خروجی خود را نشا ن‌گذاری می‌کند، سایر سرورها به کار خود به طور صحیح ادامه می‌دهند و به این ترتیب و در نهایت سرور آخر با دانستن مقادیر δ_i (با همکاری سرور اول) و با توان رسانی مجموعه‌های ورودی خود به توان q ، و محاسبه‌ی مقادیر δ_i^q می‌تواند فرستنده‌ی پیام‌ها را شناسایی کند و با حذف مقادیر δ_i، پیام‌های خروجی خود را تشکیل دهد. برای نمونه سرور آخر تمام مجموعه‌های خود را به توان q می‌رساند ،با توجه به این نکته که برای هر مولفه‌ی عضو زیر گروه G_q، توان q آن برابر یک است، حاصل برابر جایگشتی از $(\delta_1^q,\delta_2^q,\dots\delta_N^q,1,1)mod p$ خواهد شد (مقادیر یک مربوط به عناصر اضافی هست که نشان‌گذاری نشده اند) و به این ترتیب هویت فرستنده‌ی پیام‌ها برای سرور آخر معلوم می‌شود و سپس سرور آخر این نشان‌گذاری‌ها را با ضرب پیام‌ها در مقادیر معکوسشان حذف می‌کند و با یک رمزگذاری مجدد در خروجی خود قرار می‌دهد.

تایید مرحله‌ی اولین رمزگذاری مجدد به درستی انجام می‌پذیرد و به دلیل مشخص بودن مکان عناصر اضافی، سرور اول با محاسبه‌ی مقادیر

$$z_{N+1,t}=t_{N+1,t}-\beta_{N+1,t}$$
$$z_{N+2,t}=t_{N+2,t}-\beta_{N+2,t}$$

و انتشار و ارایه $Z_{N+1,t}$ اثبات صفر دانایی برای $Z_{N+2,t}$ از این تایید عبور می‌کند. برای تایید حاصل ضرب‌ها سرور اول کافی است مقادیر μ_t را به طریق زیر محاسبه و منتشر کند:

$$A_{t,j} = a'_{1,t} \times \ldots \times a'_{N+1,t} = a_1 \times \ldots \times a_{N+1} \times g^{(\beta_{1,t}+\cdots\beta_{N+1,t})}$$
$$B_{t,j} = (a_1\delta_1) \times (a_2\delta_1^{-1}) \times \ldots (\delta_{j/2}a_j).(\delta_{j/2}^{-1}a_{j+1}) \times \ldots \times a_{N+1} \times g^{r_{1,t}+\cdots+r_{N+1,t}}$$
$$\mu_t = t_{1,t} + \cdots + t_{N+1,t} - (\beta_{1,t} + \cdots \beta_{N+1,t})$$

سرور آخر به دلیل معلوم بودن مکان عناصر اضافی فقط نمای توان‌های استفاده شده در مرحله‌ی دوم رمزگذاری را فاش می‌کند و برای گذر از مرحله‌ی حاصل‌ضرب‌ها مقادیر $\mu_{t,k}$ را محاسبه و آشکار می‌کند.

$$A_{t,k} = a'_{1,k} \times \ldots \times a'_{N+1,k}$$
$$B_{t,k} = (a'_{1,k}\delta_1^{-1}) \times (a'_{2,k}\delta_1) \times \ldots (\delta_{j/2}^{-1}a_{j,k}).(\delta_{j/2}a_{j+1,k}) \times \ldots \times a_{N+1} \times g^{r_{1,k}+\cdots+r_{N+1,k}}$$
$$\mu_{t,k} = t_{1,k} + \cdots + t_{N+1,k}$$

چون تمامی مجموعه‌ها به طور متناظر با یکدیگر نشان گذاری شده‌اند مرحله‌ی تایید جایگشت نسبی مطابق با زیر بخش 3-2 انجام می‌شود. لازم به توضیح است که اگر N زوج نباشد نشان‌گذاری را روی $N-1$ پیام انجام می‌دهیم و بعد از معلوم شدن مکان عنصرهای اضافی بار دیگر گمنامی تمامی پیام‌ها شکسته خواهد شد.

در این حمله مشکلات مطرح شده در حمله‌ی بخش 4-2 وجود ندارد چون تمامی پیام‌های متناظر در مجموعه‌های متفاوت را با مقدار یکسان δ_i نشان گذاری کرده‌ایم، بنابراین مجموعه‌ها رمزگذاری مجدد شده‌ی همدیگرند و از مرحله تایید جایگشت نسبی عبور خواهد کرد و همچنین به دلیل معلوم بودن عناصر اضافی، به دلیل استفاده از مجموعه اولیه‌ی L_0، در مرحله دومین رمزگذاری مجدد مشکل حمله‌ی بخش 4-2 را ندارد و بنا براین از تمامی تاییدها عبور خواهد کرد و برای سرور نهایی گمنامی تمامی فرستنده‌ها شکسته خواهد شد.

۵- چند پیشنهاد برای بهبود شبکه‌ی مخلوط Flash

علل اصلی اعمال حملات ذکر شده در این مقاله به شبکه‌ی مخلوط Flash را می‌توان در دو مورد زیر خلاصه کرد:

1- عدم اجبار به استفاده از مجموعه‌های L_0 در مرحله‌ی شروع دومین رمزگذاری مجدد .

۲- مشخص بودن توان‌های ترکیبی و جایگشت‌های ترکیبی قبل از آغاز مرحله‌ی تایید حاصل‌ضرب و تایید مقادیر عناصر اضافی که سبب می‌شد سرور اول خاطی بتواند با استفاده از آن‌ها از مرحله‌ی تایید حاصل‌ضرب و تایید مقادیر عناصر اضافی عبور کند.

برای افزایش ایمنی شبکه‌ی مخلوط در برابر حملات اشاره شده در این مقاله بهتر است بعضی مراحل حذف کور سازی جابه‌جا و یا کمی تغییرکنند تا توان و جایگشت‌های ترکیبی در مرحله‌ی تایید حاصل ضرب‌ها معلوم نباشد. مرحله‌ی کورسازی طبق طرح Flash و بدون تغییر انجام می‌شود و در مرحله‌ی حذف کور سازی (بخش 3-2) بهتر است مراحل کار به شرح زیر تغییرکند.

1- مرحله‌ی تایید اولین رمزگذاری مجدد فقط برای عناصر اضافی و تنها مشخص کردن مکان این عناصر (بدون آشکار کردن نماها) انجام شود.
2- مرحله‌ی تایید مقادیر عناصر اضافی در مرحله‌ی تایید دومین رمزگذاری مجدد مثل طرح Flash انجام شود.
3- مرحله‌ی تایید حاصل ضرب‌ها مثل طرح Flash انجام شود.
4- تایید مرحله‌ی اولین رمزگذاری مجدد برای تمامی عناصر مثل طرح Flash انجام می‌پذیرد.
5- مرحله‌ی تایید جایگشت‌های نسبی و تولید خروجی بدون تغییر نسبت به شبکه‌ی مخلوط Flash انجام شود.

در [15] نیز بهبودی بر این طرح ارایه شده است که از ایده‌ای مشابه استفاده کرده است ولی به دلیل استفاده از اثبات‌های صفر دانایی در مرحله‌ی تایید مقادیر عناصر اضافی برای اولین رمزگذاری مجدد دارای کارایی کمتری نسبت به طرح پیشنهادی و طرح اصلی است ولی کارایی طرح پیشنهادی دقیقاً برابر با کارایی طرح اصلی است.

۶- نتیجه گیری

در این مقاله دو حمله‌ی جدید به شبکه‌ی مخلوط Flash، که یکی از کاراترین شبکه‌های مخلوط است، مطرح شد که سبب از بین رفتن خاصیت پایداری و حفظ گمنامی فرستنده‌ها شدند. همچنین نشان داده شد که با انجام چند تغییر، بدون از دست دادن کارایی و افزایش پیچیدگی، می‌توان شبکه‌ی مخلوط Flash را از لحاظ امنیتی در برابر حملات مطرح شده مقاوم کرد.

7- سپاسگزاری

این پروژه تحت قرارداد پژوهشی شماره 500/19167/ت مورخ 88/12/26 از حمایت و پشتیبانی مالی و معنوی **مرکز تحقیقات مخابرات ایران** بهره‌مند شده است.

مراجع

[1] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," Communications of the ACM, 24(2):84-88, February 1981.
[2] G. Danezis, C. Diaz, "A Survey of Anonymous Communication Channels," Technical Report MSR-TR-2008-35, Microsoft Research, January 2008.

[3] C. PARK, K. ITOH, K. KUROSAWA, "Efficient Anonymous Channel and All/Nothing Election Scheme," In EUROCRYPT ’93, vol. 765 LNCS, pages 248-259, Springer-Verlag, 1994.
[4] Birgit Pfitzmann, "Breaking efficient anonymous channel," In Lecture Notes in Computer Science Springer Berlin/Heidelberg, pages 332-340.
[5] B. Pfitzmann and A. Pfitzmann, "How to break the direct RSA-implementation of mixes," In Advances in Cryptology –Eurocrypt ’89, Volume 434 of Lecture Notes in Computer Science pages 373-381, Springer Verlag, 1990.
[6] W. OGATA, K. KUROSAWA, K. SAKO, K. TAKATANI, "fault tolerant anonymous channel," 1997.
[7] K. Sako and J. Kilian. Receipt-free mix-type voting scheme. In Proc.of Eurocrypt ’95. Springer-Verlag, 1995. LNCS 921.
[8] M. Jakobsson and A. Juels. An optimally robust hybrid mix network. In Proc. of PODC’01, pages 284-292. ACM Press. 2001
[9] M. Abe, "Universally verifiable MIX with verification work independent of the number of MIX servers, " Advances in Cryptology-EUROCRYPT’98, LNCS 1403, pages 437-447 1998.
[10] M. Jakobsson and A.Juels. "Millimix: mixing in small batches, " MIMACS Technical Report 99-33.
[11] M. Abe, "Mix-nets on permutation networks, " Asiacrypt ’99, LNCS 1716, 258-273 1999.
[12] J.Furukawa and K. Sako, "An efficient scheme for proving a shuffle," In Proc. of Crypto ’01, pp.368-387, Springer-Verlag 2001. LNCS 2139.
[13] A. Neff, "A verifiable secret shuffle and its application to E-Voting,". In Proc. of ACM CCS’0, pp.116-125 ACM Press ,2001.
[14] M. Jakobsson, "Flash mixing," In Principles of Distributed Computing PODC 99, pages 83- 89. ACM, 1999.
[15] M. Mitomo and K. Kurosawa, "Attack for flash mix," In Advances in Cryptology –ASIACRYPT In 2000.
[16] P. Golle, S. Zhong, D. Boneh, M. Jakobsson and A. Juels, "Optimistic Mixing for Exit-Polls," Asiacrypt 2002, LNCS, 2002.
[17] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "The (in)security of distributed key generation in dlog- based cryptosystem," In J.Stern, editor, Eurocrypt ’00,pages 539-556. Springer-Verlag, 2000. LNCS no.1507.
[18] D. Wikström, "Four practical attacks for optimistic mixing for exit-polls,". Technical Report T2003-04, Swedish Institute of Computer Science , SICS, Box 1263, SE-164 29 Kista, SWEDEN , 2003b.
[19] D. Wikström, "Elements in $Z_p^* \backslash G_q$ are dangerous," Technical Report T2003-05, Swedish Institute of Computer Science SICS, Box 1263, SE-164 29 Kista, SWEDEN, 2003a.