مجموعه مقالات پانزدهمین کنفرانس بین‌المللی انجمن رمز ایران
دانشکده مهندسی برق، دانشگاه تربیت دبیر شهید رجایی، تهران، ۶-۷ شهریور ۱۳۹۷

ارتقای حمله به سیستم رمز RSA با استفاده از تکنیک مشبکه

سید مهدی سجادیه، روح الله جلائی

استادیار دانشگاه خوراسگان

m.sajadieh@khuisf.ac.ir

دانشجوی دانشگاه صنعتی مالک اشتر اصفهان

r.crypto.js@gmail.com

چکیده

سیستم رمز کلید عمومی RSA مبتنی بر مسأله سخت تجزیه اعداد است و به دلیل سرعت نسبتاً بالای رمزنگاری در بسیاری از بخش‌های الکترونیکی کاربرد عملی پیدا کرده است و هم‌چنان نیز از آن استفاده می‌شود. با توجه به اهمیت مفهوم امنیت در سیستم‌های رمز، حملات بسیاری به سیستم رمز RSA شده است که از جمله مهم‌ترین آنها، حمله غربال میدان اعداد، حمله غربال مربعی، و حملات مبتنی بر مشبکه‌ها از جمله حمله Winner می‌باشد.

اساس روش پیشنهادی در این مقاله، استفاده از تکنیک کپراسمیت مبتنی بر مشبکه‌ها است. برای ارتقای حمله به سیستم رمز RSA و یافتن کلید خصوصی، کران بالای نمای خصوصی d افزایش داده می‌شود. در روش جدید، ابتدا عدد مرکب N را به شکل $N = x^2 - y^2$ در نظر گرفته و سپس با استفاده از روش کپراسمیت، یافتن تابع $\varphi(n)$ در RSA را معادل با پیدا کردن ریشه یک چند جمله‌ای در نظر می‌گیریم و در نتیجه با ساختن پایه مشبکه با بعد کوتاه، کوتاهترین بردار را به وسیله الگوریتم LLL یافته و کران بالای نمای خصوصی را برای $d < N^{\delta}$ ارتقا می‌دهیم.

کلمات کلیدی

RSA، روش کپراسمیت، مشبکه‌ها، الگوریتم LLL

۱-مقدمه

RSA مبتنی بر دو تابع یک طرفه‌ی تجزیه اعداد و لگاریتم گسسته است که در صورت شکسته شدن حتی یکی از این دو ، این رمز قابلیت خود را از دست می‌دهد. کلید خصوصی در سیستم RSA عبارت است از دو عدد اول بزرگ p و q و یک نمای خصوصی d که به پیمانه $(p-1)(q-1)$ معکوس پذیر است. کلید عمومی شامل عدد مرکب $N = pq$ و نمای عمومی $e = d^{-1} \bmod (p-1)(q-1)$ است. پیام m بوسیله اعدادی در حلقه Z/NZ نمایش داده می‌شود و بوسیله $c = m^e \bmod N$ رمزگذاری می‌شود. توجه داریم که $\phi(n) = (p-1)(q-1)$ که در آن ϕ تابع اویلر است. در حلقه Z/NZ نماها می‌توانند به پیمانه هم نهشتی $\phi(n)$ محاسبه شوند [۱]، چونکه $a^{\phi(n)} \equiv a \bmod N$ برای تمام a های معکوس پذیر در Z/NZ

بنابراین، دریافت کننده متن رمز شده‌ی c می‌تواند با استفاده از کلید خصوصی عبارت زیر را محاسبه کند:

$$c^d = m^{ed} = m \ (mod\ N) \qquad (۱)$$

چون $e = d^{-1} \bmod \phi(n)$، پیام m بدست می‌آید.

اگر دشمن قادر باشد N را تجزیه کند، می‌تواند به اعداد اول راز p و q دست یابد و با استفاده از $\phi(n)$، معکوس نمای عمومی را بدست آورده و در نتیجه نمای خصوصی را بیابد.

حملات مختلفی تا کنون روی سیستم رمز RSA صورت گرفته است و در اغلب موارد دارای پیچیدگی نمایی است. این حملات توانسته‌اند مقدار N با طول کمتر از ۱۰۲۴ بیت را بشکنند [۱]. با پیدایش مشبکه‌ها و الگوریتم LLL، حمله به RSA در زمان چندجمله‌ای صورت گرفت که از جمله مهم‌ترین آنها، در ابتدا حمله Winner بوده که کران نمای خصوصی را به اندازه $d \approx N^{0.250}$ ارتقا داده و نشان داد برای کلید عمومی با اندازه ۳، سریعترین حمله، حمله مشبکه است [۳]. در سال ۲۰۱۵، Boneh و Durfee با تشریح کامل حمله مشبکه، کران را به $d \approx N^{0.292}$ ارتقا دادند [۲].

باید دقت داشت در صورت وجود کامپیوترهای کوانتومی، رمز RSA شکسته می‌شود ولی با کامپیوترهای فعلی امنیت این سیتم رمز تاکنون وجود دارد [۲]. در این مقاله با استفاده از رابطه $N = x^2 - y^2$ و ترکیب آن با تکنیک کپراسمیت و ساخت یک چندجمله‌ای جدید به عنوان پایه برای مشبکه، حمله روی RSA به‌طور جدی تری ارتقا یافته است. در ادامه، ابزار اصلی مشبکه و روش کپراسمیت بیان خواهد شد و سپس تعمیم روش کپراسمیت ارائه خواهد شد.

۲ - مشبکه و ابزارهای آن

مشبکه‌ها از ابرازهای مهم رمزنگاری هستند که هم در طراحی و هم در تحلیل رمزهای کلید عمومی کاربرد دارند. مشبکه‌ها دارای خاصیت‌های فراوانی هستند که تقریباً در تمام زمینه‌های رمزنگاری ورود پیدا کرده و ساختارهای مدرن آن هم چون SIS و LWE در امنیت توابع چکیده ساز وسیستم‌های رمز تأثیر بسزایی دارد [۳].

در ادامه تعاریف مورد نیاز از آنها آورده می‌شود.

تعریف۲-۱- (مشبکه). مشبکه $\mathcal{L}$ برابر مجموعه تمام ترکیبات خطی از بردارهای مستقل خطی روی $\mathbb{R}$ به شکل زیر است :

$$\mathcal{L} = \{a_1 v_1 + a_2 v_2 + \cdots + a_n v_n \mid a_i \in \mathbb{Z}\} \qquad (۲)$$

که در آن بردار $B = \{v_1, v_2, \dots, v_n\}$ یک پایه برای مشبکه $\mathcal{L}$ است.

تعریف ۲-۲-(دترمینان مشبکه). دترمینان مشبکه برابر با دترمینان ماتریس پایه B از مشبکه محاسبه می‌شود [۴].

تعریف ۲-۳-(رتبه مشبکه). رتبه مشبکه برابر است با بیشترین تعداد بردارهای مستقل خطی در مشبکه و معادل است با بیشترین بردارهای مستقل خطی از پایه [۴].

تعریف ۲-۴-(بردارهای گرام‌اشمیت). برای یک دنباله از n بردار مستقل خطی $b_1, b_2, \dots, b_n \in \mathbb{R}^n$، بردارهای گرام اشمیت آنها به شکل زیر تعریف می‌شوند [۴]:

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^* \qquad (۳)$$

که درآن $\mu_{i,j} = \frac{<b_i, b_j^*>}{<b_j^*, b_j^*>}$ ضرایب گرام اشمیت نام دارند.

نکات زیر در رابطه با بردارهای گرام اشمیت دارای اهمیت است:

۱- بردارهای گرام اشمیت لزوماً عضو مشبکه نیستند.

۲- طول اقلیدسی بردارهای گرام اشمیت b_i^* حداکثر برابر طول بردارهای پایه b_i است، یعنی $\|b_i^*\| \le \|b_i\|$.

قضیه ۲-۵ . فرض کنید $b_1^*, b_2^*, \dots, b_n^*$ بردار گرام اشمیت از بردارهای مستقل خطی $b_1, b_2, \dots, b_n$ باشد. در این صورت [۴]

$$\det(\mathcal{L}(B)) = \prod_{i=1}^{n} \|b_i^*\| \qquad (۴)$$

تعریف ۲-۶- (مسأله کوتاهترین بردار SVP)( ). این مسأله عبارت است از یافتن کوتاهترین بردار غیر صفر از مشبکه $\mathcal{L}$، به عبارت دیگر پیدا کردن کوتاهترین بردار غیر صفر v به‌طوریکه نرم اقلیدسی $\|v\|$ کوچکترین شود [۵].

قضیه ۲-۷-(کران هرمیتی). هر مشبکه $\mathcal{L}$ با بعد n ، شامل بردار غیر صفر $v \in \mathcal{L}$ است که در رابطه زیر صدق می‌کند [۵]:

$$\|v\| \le \sqrt{n}\, det(\mathcal{L})^{\frac{1}{n}} \qquad (۵)$$

۲-۸-الگوریتم LLL

در سال ۱۹۸۲، $Lenstra, Lenstra, Lovasz$ الگوریتم LLL را ارائه دادند که مبتنی بر نامساوی هرمیتی است و الگوریتمی برای کاهش پایه مشبکه است. این الگوریتم برای ابعاد مشخص، بردار کوتاهی را که نزدیک به کوتاهترین بردار غیر صفر است، در زمان چندجمله‌ای می‌یابد [۶]. این الگوریتم دارای مراحل زیر است:

۱. مرحله کاهش: بردار b_i به بردار $b_i^* = b_i - \sum_{j=1}^{i-1} \lfloor \mu_{i,j} \rceil b_j^*$ تبدیل می‌شود که $\mu_{i,j}$ ضرایب گرام اشمیت و علامت $\lfloor \ \rceil$ به معنای گرد شده به نزدیک‌ترین عدد صحیح می‌باشد.

۲. مرحله تعویض: اگربردارهای متوالی b_i و b_{i-1} در شرط δLLL-کاهش یافته $\delta \|b_{i-1}^*\|^2 \le \left\|b_i^* + \mu_{i,i-1} b_{i-1}^*\right\|^2$ صدق نکند (که در آن $\frac{1}{2} \le \delta < 1$)، آنگاه آن دو بردار باهم تعویض می‌شوند و مقدار $i = max(i-1, 2)$ در نظر گرفته شده و ضرایب متناظر گرام اشمیت و پایه، دوباره خوانی می‌شوند، در غیر این صورت به i یک مرحله اضافه می‌شود.

۳. تکرار: تکرار از مرحله ۱ شروع می‌شود. وقتی‌که $i = n$ باشد الگوریتم به پایان می‌رسد. کاهش اندازه بردار b_k تأثیری در اندازه بردارهای دیگر ندارد.

قضیه ۲-۹-(پایه کاهش یافته LLL)

پایه کاهش‌یافته $\{v_1, v_2, \dots, v_n\}$ از الگوریتم LLL دارای خواص زیر است [۶]:

$$\prod_{i=1}^{n} \|v_i\| \le 2^{n(n-1)/4}\, det\, \mathcal{L} \qquad (۶)$$

و برای $1 \le j \le i \le n$: $\|v_j\| \le 2^{(i-1)/2} \|v_i^*\|$ و برای بردار اولیه داریم:

$$\|v_1\| \le 2^{(n-1)/4} |det\, L|^{1/n} \qquad (۷)$$

و

$$\|v_1\| \le 2^{(n-1)/2} \min_{0 \ne v \in \mathcal{L}} \|v\| \qquad (۸)$$

۳- روش کپراسمیت

یکی از کاربردهای الگوریتم LLL ، تکنیک کپراسمیت برای یافتن ریشه‌های کوچک یک چندجمله‌ای به پیمانه عدد داده شده N می‌باشد (حتی زمانیکه تجزیه N ناشناخته باشد). این ابزار، یک تکنیک قوی در آنالیز رمز است. هم‌چنین قوی‌ترین الگوریتم حمله به RSA با نمای عمومی ضعیف، الگوریتم مبتنی بر روش کپر اسمیت است [۷].

کپراسمیت نشان داد که می‌توان حداقل یک ریشه کوچک از چند جمله‌ای را در زمان چندجمله‌ای پیدا کرد هرگاه کران بالایی از ریشه مفروض، موجود باشد [۷].

در ایده این روش معادله چندجمله‌ای را برای رمزنگاری RSA در نظر می‌گیرد و ریشه‌های کوچک این معادله را با اعمال الگوریتم LLL، و ساختن بردارهای کوتاه می‌یابد.

قضیه ۳-۱- (طرح کپراسمیت). فرض کنید که N عدد صحیح مثبت و $f(x) \in Z[x]$ چند جمله‌ای تکین از درجه d باشد. الگوریتمی بازمان چندجمله‌ای وجود دارد که با داشتن N و f مناسب، x_0 را طوری می‌یابد که

$$f(x_0) \equiv 0 \ (mod\ N)\ ,\ |x_0| \le B \approx N^{\frac{1}{d}} \qquad (۹)$$

وقتی‌که N اول باشد، به‌عبارت‌دیگر Z_N میدان متناهی باشد، الگوریتمی مشخص برای یافتن تمام ریشه‌های f به پیمانه داده‌شده N که تعدادشان حداکثر d تاست، وجود دارد [۸].

برای عدد مرکب N، تعداد ریشه‌های $f(x)$ به پیمانه N می‌تواند نزدیک به نمائی باشد حتی برای مربع $f(x)$. برای مثال اگر N برابر حاصل‌ضرب k عدد اول متمایز باشد، آنگاه مربع تابع $f(x)$ به پیمانه N دقیقاً 2^k ریشه مربعی متمایز دارد (این از قضیه باقیمانده چینی نتیجه می‌شود). بطور کلی الگوریتم کافی برای یافتن تمام ریشه‌های f به پیمانه N وجود ندارد و شرط "کوچک بودن ریشه‌ها" برای همین است.

شرط اندازه، محدودیت دیگر برای ریشه‌هاست. داشتن دو ریشه مربعی r_1, r_2 بطوریکه $r_1 \neq \pm r_2$ ، از پیمانه عدد مرکب N فاکتور غیر بدیهی از N یعنی $GCD(r_1 - r_2, N)$ را نتیجه می‌دهد. لذا اگر تعداد ریشه‌ها کم باشد، یافتن همه آنها حداقل به‌اندازه تجزیه کردن مشکل است.

اثبات (قضیه کپراسمیت). قضیه برای کران ضعیف‌تر B به‌اندازه $B \approx N^{2/d(d+1)}$ اثبات می‌شود:

استراتژی حل مسأله یافتن چندجمله‌ای غیر صفر دیگر مثل $h(x) = \sum h_i x^i \in Z[x]$ است بطوریکه:

۱. هر ریشه f به پیمانه N ، ریشه h نیز باشد.

۲. چندجمله‌ای $h(Bx)$ کوچک باشد یعنی $|h_i B^i| < N/(degh+1)$ برای هر i .

برای این نوع از $h(x)$ و برای هر x_0 بطوریکه $|x_0| \le B$ داریم:

$$|h_i x_0^i| \le |h_i B^i| < N/(degh+1) \qquad (۱۰)$$

که نتیجه می‌دهد $|h(x_0)| < N$. برای هر ریشه کوچک x_0 بطوریکه $|x_0| \le B$ روی اعداد صحیح : $h(x_0) = 0$. برای یافتن ریشه‌های کوچک $f(x)$ به پیمانه N می‌توان $h(x)$ را روی اعداد صحیح تجزیه کرد و با آزمایش مکرر می‌توان فهمید که کدام‌یک از ریشه‌های کوچک، ریشه $f(x)$ به پیمانه N است. حال الگوریتم کافی برای یافتن چنین $h(x)$ ای ارائه داده می‌شود.

تعریف ۳-۲-(چندجمله‌ای‌های دو متغیره با ضرایب صحیح). مسأله عبارت است از یافتن ریشه‌های کوچک برای چندجمله‌ای‌های دومتغیره با احتساب عمل پیمانه گیری.

در حقیقت، برای تسریع در انجام محاسبات و دخیل کردن تمام پارامترهای RSA، چند جمله‌ای‌های دو متغیره بهترین ابزار برای پیاده سازی روش کپراسمیت هستند [۹].

ابتدا چند جمله‌ای دو متغیره $h(k,x) \in Z[k,x]$ از در جه n در نظر گرفته می‌شود؛ سپس ریشه های تابع $h(k,x)$ به پیمانه هم نهشتی بدست می‌آید.

قضیه ۳-۳-(کپراسمیت). فرض کنید $h(x) \in R[x]$ چند جمله‌ای از درجه w باشد و $X \in R$ داده شده باشد. فرض کنید $|x_0| < X$ طوری وجود داشته باشد که :

۱- $h(x_0) \in Z$ باشد

۲- $\|h(xX)\| < \frac{1}{\sqrt{w}}$ باشد

آنگاه $h(x_0) = 0$.

اثبات.

$$|h(x_0)| = \left|\sum a_i x_0^i\right| = \left|\sum a_i X^i \left(\frac{x_0}{X}\right)^i\right| \le$$

$$\sum \left|a_i X^i \left(\frac{x_0}{X}\right)^i\right| \le \sum |a_i X^i| \le \sqrt{w}\|h(xX)\| < 1$$

از طرفی می‌دانیم که $h(x_0) \in Z$ ، بنابراین $h(x_0) = 0$.

قضیه ۳-۴- فرض کنید $h(k,x) \in Z[k,x]$ چند جمله ای دو متغیره باشد و برای عدد مثبت n،

$$\|h(kK, xX)\| < \frac{e^n}{\sqrt{w}} \text{ و } h(k,x) \equiv 0\ mod\ e^n \qquad (۱۱)$$

که در آن $|k| < K$ و $|x| < X$، آنگاه $h(k,x) = 0$ [۸].

نتیجه ۳-۵- اگر $b_1^*, \dots, b_w^*$ بردارهای کاهش یافته از الگوریتم LLL برای مشبکه L باشند، آنگاه

$$|b_1^*| \le 2^{\frac{w-1}{4}} det(L)^{\frac{1}{w}} \qquad (۱۲)$$

از طرفی طبق کران هرمیتی، کوتاهترین طول بردار در مشبکه با پایه های $b_1^*, \dots, b_w^*$ برابر است با:

$$|b_1^*| \le \sqrt{w}\, det(L)^{\frac{1}{w}} \qquad (۱۳)$$

و همچنین طبق فرض، b_1^* برابر کوتاهترین بردار مورد انتظار است:

$$|b_1^*| = \|h(xX)\| < \frac{1}{\sqrt{w}} \qquad (۱۴)$$

با توجه به اینکه بردار کاهش یافته رابطه (۱۲) کوتاه است اما لزوماً کوتاهترین نیست، از دو رابطه (۱۲) و (۱۴) نتیجه می‌شود:

$$det(L) < 2^{-w(\frac{w-1}{4})} w^{\frac{-w}{2}} < 1 \qquad (۱۵)$$

از طرف دیگر، با توجه به اینکه $h(kK, xX)$ برابر اولین بردار کاهش یافته LLL (b_1^*) است و با توجه به کران b_1^*، شرط اینکه رابطه (۱۱) در قضیه (۳-۴) برقرار باشد این است که $det(L) \ll 1$ باشد.

۴- روش جدید

در این بخش، روش جدید خود را معرفی می‌کنیم که از روش کپراسمیت در حالت دو متغیره استفاده کرده و تابع جدیدی را می‌سازیم. این تابع در شرط قضیه (۶) صدق می‌کند و در نتیجه پایه‌های مشبکه را از طریق توابع معرفی

شده در روش (H-G) [۲,۱۰] می‌سازیم. مزیت این روش آن است که نیازی به تعیین هر دو ریشه تابع نمی باشد. در واقع کافی است که ریشه x پیدا شود که در این صورت، $\varphi(n)$ بدست می آید.

فرض کنید عدد مرکب N را بتوان به شکل $N = x^2 - y^2$ نوشت. بنابراین $p = x + y$ و $q = x - y$ و $\varphi(N) = N - (p + q) + 1 =$ $N - 2x + 1$. از طرفی $ed \equiv 1\ mod(\varphi(N))$ لذا :

$$\exists\ k \in Z\ s.t \quad ed = k\varphi(N) + 1 \qquad (۱۶)$$

$$ed = k(N - 2x + 1) + 1 \qquad (۱۷)$$

$$k(N - 2x + 1) + 1 \equiv 0 \quad mod\ e \qquad (۱۸)$$

۱-۴- رویکرد اول

تابع $f(k,x)$ را به شکل زیر تعریف می‌کنیم:

$$f(k,x) \coloneqq k(N - 2x + 1) + 1 \qquad (۱۹)$$

فرض کنید کران‌های کلید عمومی و خصوصی به شکل $d < N^{\delta}$ و $e \leq N^{\alpha}$ باشند که در آنها α عددی نزدیک به 1 است و δ در ادامه مشخص می‌شود. فرض کنید p و q با کران $p, q < 2\sqrt{N}$ و به طور تقریبی برابر $\sqrt{N}$ باشند. می‌خواهیم کران های $x < X$ و $k < K$ را بیابیم:

$$|x| = \left|\frac{p+q}{2}\right| \cong |\sqrt{N}| \qquad (۲۰)$$

از طرفی $\varphi(N) = N - 2x + 1 > 2x$ در نتیجه $\frac{1}{|\varphi(N)|} < \frac{1}{|2x|} \approx$ $\frac{1}{2|\sqrt{N}|}$ بنابراین خواهیم داشت:

$$|k| < \left|\frac{ed}{\varphi(n)}\right| < \left|\frac{e.e^{\delta}}{2\sqrt{N}}\right| < \left|\frac{e^{\delta+1}}{2e^{\frac{1}{2\alpha}}}\right| \qquad (۲۱)$$

با درنظر گرفتن $\alpha \approx 1$، داریم:

$$|k| < \frac{1}{2}|e^{\delta+1/2}| \qquad (۲۲)$$

حال به دنبال ریشه‌هایی مثل (k_0, x_0) هستیم بطوریکه

$$f(k_0, x_0) = 0 \quad mod\ e\ \ , |k_0| < K\ , |x_0| < X \qquad (۲۳)$$

مسأله بر می‌گردد به یافتن ریشه تابع $f(k,x)$. اگر بتوانیم x را بیابیم، آنگاه $\varphi(N)$ بدست آمده و در نتیجه مسأله RSA حل می‌شود.

در ابتدا تابع $f_1(k,x)$ را به شکل زیر تعریف می‌کنیم:

$$f_1(k,x) \coloneqq \frac{k(N-2x+1)+1}{e} \qquad (۲۴)$$

حال اگر (k_0, x_0) ریشه های تابع f باشند، آنگاه $f_1(k_0, x_0)$ مقداری صحیح است و بنابراین تابع $f_1(k,x)$ در شرط قضیه (۳-۴) صدق می‌کند. بنا بر قضیه کپراسمیت، توابع $g_{i,l}(k,x)$ و $g_{i,l}(k,x)$ را طوری می سازیم که ریشه‌ای همانند ریشه تابع $f(k,x)$ داشته باشند و عمل پیمانه هم نهشتی در آن حذف شود. توابع پایه زیر را برای ساختن مشبکه تشکیل می‌دهیم:

$$g_{i,l}(k,x) \coloneqq k^i f_1^l(k,x) \text{ و } h_{j,l}(k,x) \coloneqq x^j f_1^l(k,x) \qquad (۲۵)$$

که در آن $0 \leq l \leq m$ و $0 \leq i \leq m - l$ و $0 \leq j \leq t$. توجه داریم که

$$f_1^l(k,x) = \left[\frac{k(N - 2x + 1) + 1}{e}\right]^l \equiv 0\ \ mod\ e^l$$

و هم چنین کران های $|x_0| < X$, $|k_0| < K$ در بخش قبل تعیین گشتند. بنابراین انتخاب توابع (۲۵) معقول است وانتخاب ضرایب این توابع، برای ایجاد استقلال خطی در پایه‌ها است.

حال مشبکه حاصل از ترکیب‌های صحیح با نرم پایین چندجمله‌ای‌های $g_{i,l}(kK, xX)$ و $h_{j,l}(kK, xX)$ را می‌سازیم. به عنوان مثال، با در نظر گرفتن $m = 2$ و $t = 1$، ماتریس پایه زیر را برای مشبکه $\mathcal{L}_1$ می‌سازیم:

جدول (۱) : ماتریس پایه مشبکه $\mathcal{L}_1$

$\mathcal{L}_1$	1	k^2x	k^3x^2	x^2	x^3
f_1	e^{-1}	0	0	0	0
${f_1}^2$	0	$-4Ne^{-2}K^2X$	0	0	0
$k{f_1}^2$	0	$-4e^{-2}K^2X$	$4e^{-2}K^3X^2$	0	0
$x{f_1}^2$	0	$N^2e^{-2}K^2X$	0	$-4e^{-2}X^2$	0
$x^2{f_1}^2$	0	0	0	$(2N+2)X^2$	$-4e^{-2}X^3$

با محاسبه دترمینان $\mathcal{L}_1$ و با توجه به کران‌های k و x خواهیم داشت:

$$det(L_1) = -2^8K^5X^8e^{-9}N < 1 \qquad (۲۶)$$

با محاسبه مقدار δ از رابطه قبل و با توجه به اینکه $e \approx N$ خواهیم داشت:

$$\delta \cong 3/5\ log_N 2 + 0.3 \qquad (۲۷)$$

توجه داریم که روش بونه[۲] کران را به $\delta \cong 0.292$ ارتقا داده است و روش جدید، در مقایسه با روش های دیگر بهبود زیادی یافته است.

جدول (۲) مقادیر δ را برای حالت‌های مختلف N نشان می‌دهد.

جدول (۲): مقادیر δ بر اساس اندازه بیت

مقادیر N (بیت)	۱۰۰	۵۰۰	۸۰۰	۱۰۲۴	۲۰۰۰	۲۵۰۰	۳۰۰۰
مقادیر $\boldsymbol{\delta}$	۰.۳۰۶	۰.۳۰۱	۰.۳۰۰	۰.۳۰۰	۰.۳۰۰	۰.۳۰۰	۰.۳۰۰

۴-۲- رویکرد دوم

در این بخش تابع $f_2(k,x)$ را به شکل زیر تعریف می‌کنیم:

$$f_2(k,x) := \frac{(k(N-2x+1)+1)^2}{e^2} \qquad (۲۸)$$

توجه داریم که برای ریشه های (k_0,x_0)، $f_2(k_0,x_0)$ مقداری صحیح است و لذا تابع $f_2(k,x)$ در شرط قضایای (۳-۳) و (۳-۴) صدق می‌کند. توابع پایه زیر را برای ماتریس پایه مشبکه $\mathcal{L}_2$ می‌سازیم:

$$h_{j,l}(k,x) := x^j f_2^l(k,x) \quad \text{و} \quad g_{i,l}(k,x) := k^i f_2^l(k,x) \qquad (۲۹)$$

با در نظر گرفتن $m=3$ و $t=1$، ماتریس پایه (۳) را برای مشبکه $\mathcal{L}_2$ تشکیل می‌دهیم که در آن $A = N+1$ است و ماتریس دوم ادامه ماتریس نخست است.

جدول (۳): ماتریس پایه مشبکه $\mathcal{L}_2$

	1	k	kx	k^2	k^2x	k^2x^2
1	1	0	0	0	0	0
k	0	k	0	0	0	0
f	e^{-2}	$\frac{AK}{e^2}$	$\frac{2KX}{e^2}$	0	0	0
K^2	0	0	0	K^2	0	0
kf	0	$\frac{K}{e^2}$	0	$\frac{AK^2}{e^2}$	$\frac{2K^2X}{e^2}$	0
f^2	e^{-4}	$\frac{2Ak}{e^4}$	$\frac{4KX}{e^4}$	$\frac{A^2K^2}{e^4}$	$\frac{4AK^2X}{e^4}$	$\frac{4K^2X}{e^4}$
k^3	0	0	0	0	0	0
k^2f	0	0	0	$\frac{K^2}{e^2}$	0	0
kf^2	0	$\frac{K}{e^4}$	0	$\frac{2AK^2}{e^4}$	$\frac{4K^2X}{e^4}$	0
f^3	e^{-6}	$\frac{3AK}{e^6}$	$\frac{6KX}{e^6}$	$\frac{3A^2K^2}{e^6}$	$\frac{12AK^2X}{e^6}$	$\frac{12K^2X^2}{e^6}$
x	0	0	0	0	0	0
xf	0	0	$\frac{AKX}{e^2}$	0	$\frac{2K^2X}{e^2}$	0
xf^2	0	0	$\frac{2AKX}{e^4}$	0	$\frac{A^2K^2X}{e^4}$	$\frac{4AK^2X^2}{e^4}$
xf^3	0	0	$\frac{3AKX}{e^6}$	0	$\frac{3A^2K^2X}{e^6}$	$\frac{12AK^2X^2}{e^6}$

k^3	k^3x	k^3x^2	k^3x^3	x	kx^2	k^2x^3	k^3x^4
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
K^3	0	0	0	0	0	0	0
$\frac{AK^3}{e^2}$	$\frac{2K^3X}{e^2}$	0	0	0	0	0	0
$\frac{A^2K^3}{e^4}$	$\frac{4AK^3X}{e^4}$	$\frac{4K^3X^2}{e^4}$	0	0	0	0	0
$\frac{A^3K^3}{e^6}$	$-$	$\frac{6A^2K^3X^2}{e^6}$	$\frac{8K^3X^3}{e^6}$	0	0	0	0
0	0	0	0	x	0	0	0
0	0	0	0	$\frac{K}{e^2}$	$\frac{2KX^2}{e^2}$	0	0
0	0	0	0	$\frac{K}{e^4}$	$\frac{4KX^2}{e^4}$	$\frac{4K^2X^3}{e^4}$	0
0	$\frac{A^3K^3X}{e^6}$	$-$	$-$	$\frac{X}{e^6}$	$\frac{6KX^2}{e^6}$	$-$	$\frac{8K^3X^4}{e^6}$

که در آن علامت‌های – همان ضرایب پایه‌های داده شده هستند و به دلیل حجم زیاد از نوشتن آنان خودداری نمودیم.

با محاسبه دترمینان $\mathcal{L}_2$ و با توجه به کران‌های k و x خواهیم داشت:

$$det(L_2) = 2^{16}e^{-30}K^{26}X^{19} < 2^{16}e^{-30}(\frac{1}{2}e^{\delta-\frac{1}{t}+1})^{26}e^{\lfloor\frac{19}{2}\rfloor}$$

$$= \frac{1}{2^{10}}e^{-30+26\delta-13+26+9} = \frac{1}{2^{10}}e^{-30+26\delta+22}$$

در نتیجه کران بالای دترمینان $\mathcal{L}_2$ برابر است با:

$$det(L_2) < \frac{1}{2^{10}}e^{26\delta-8} \qquad (۳۰)$$

بنا بر رابطه (۱۵)، باید دترمینان ماتریس $\mathcal{L}_2$کمتر از 1 باشد:

$$det(L_2) < \frac{1}{2^{10}}e^{26\delta-8} < 1 \qquad (۳۱)$$

با محاسبه مقدار δ از رابطه (۳۱) و با توجه به اینکه $e \approx N$ خواهیم داشت:

$$\delta \cong 0.385\, log_N\, 2 + 0.308 \qquad (۳۲)$$

جدول ۴ مقادیر δ را برای حالتهای مختلف N نشان می‌دهد:

جدول (۱): مقادیر δ

مقادیر $\boldsymbol{\delta}$	مقادیر $\boldsymbol{N}$ بر حسب بیت
۰,۳۱۲	۱۰۰
۰,۳۱۰	۲۰۰
۰,۳۰۹	۴۰۰
۰,۳۰۹	۵۱۲
۰,۳۰۹	۶۰۰
۰,۳۰۹	۷۰۰
۰,۳۰۸	۸۰۰

۰,۳۰۸	۱۰۲۴
۰,۳۰۸	۲۰۰۰

حال با اعمال الگوریتم LLL روی ماتریس $\mathcal{L}_2$ و در زمان چندجمله ای، کوتاهترین بردارهای $\mathrm{H}_2(k,x)$ و $H_1(k,\mathrm{x})$ بدست می‌آیند که دارای ریشه مشترک x هستند. در نتیجه وقتی که ریشه x از دستگاه معادلات $\mathrm{H}_2(k,x)$ و $H_1(k,\mathrm{x})$ بدست آمد، مقدار $\varphi(N)$ بدست می‌آید و سپس پارامترهای p و q تجزیه می‌شوند و کلید خصوصی از رابطه (۱۷) کشف می‌شود.

۵- نتیجه گیری

در این مقاله با ارائه روشی جدید که مبتنی بر روش کپراسمیت است، ساختار خاصی از RSA را درنظر گرفته و با تکنیک‌های مشبکه، حمله جدیدی علیه RSA ارائه شد. با توجه به حملات مختلفی که علیه سیستم رمز RSA صورت گرفته است، حمله مشبکه مبتنی بر یافتن کوتاهترین بردار (SVP) یکی از سریعترین حملات است. این حمله بر پایه الگوریتم LLL بنا شده است که بهترین الگویتم برای یافتن کوتاهترین بردار در زمان چند جمله‌ای می‌باشد. در روش ارائه شده با استفاده از یک پایه ۱۴ بعدی، ارتقا حملات قبلی را نمایش داده‌ایم و با توجه به پایین بودن بعد مشبکه، هم‌چنان این ایده وجود دارد که با ابعاد دیگری، حملات وسیع تری به سیستم رمز RSA صورت گیرد.

در پایان، روش پیشنهادی خود را ارائه داده‌ایم که در آن توانسته‌ایم برای حمله به RSA ، نمای خصوصی (d) را از 0.292 به $0.308 + 2\ log_N\ 0.385$ ارتقا دهیم و این به این معنی است که با در نظر گرفتن مفروضات لازم، این روش می‌تواند تهدیدهای جدی را برای RSA داشته باشد.

مراجع

[1] D. Boneh, *Twenty years of attacks on the RSA cryptosystem*, Notices of the AMS, Feb 1999.

[2] G. Durfee, *Cryptanalysis of RSA using algebraic and lattice methods* (Doctoral dissertation, Stanford University), 2002.

[3] C. Peikert, *Decade of Lattice Cryptography*. World Scientific; Feb 2016.

[4] J. van de Pol, *Lattice-based cryptography*, Master's thesis, Eindhoven University of Technology, Jul 2011.

[5] J. Hoffstein, J. Pipher, JH. Silverman, *An introduction to mathematical cryptography*, Vol. 1. New York: springer, 2008.

[6] A. May, *Using LLL-reduction for solving RSA and factorization problems. In The LLL algorithm*, pp. 315-348. Springer, Berlin, Heidelberg, 2009.

[7] D. Coppersmith, *Finding small solutions to small degree polynomials*, In Cryptography and lattices, Springer Berlin Heidelberg (pp. 20-31) 2001.

[8] I. Sletta, *Finding Small Roots of Polynomial Equations Using Lattice Basis Reduction*, brage.bibsys.no 2009.

[9] M. Herrmann, A. May. *Maximizing small root bounds by linearization and applications to small secret exponent RSA*, In International Workshop on Public Key Cryptography, Springer Berlin Heidelberg, (pp. 53-69), May 2010.

[10] N. Howgrave-Graham, *Finding small roots of univariate modular equations revisited*, In IMA International Conference on Cryptography and Coding, Springer Berlin Heidelberg, (pp. 131-142) Dec 1997.