مروری بر رمزنگاری مشبکه - مبنا

امیر حسنی کرباسی [۱] و رضا ابراهیمی آتانی [۲]

[۱] دانشجوی دکتری ریاضیات رمزنگاری، گروه ریاضی، دانشگاه گیلان - رشت

[۲] استادیار دانشکده فنی، دانشگاه گیلان - رشت

reza.ebrahimi.atani@gmail.com

چکیده

هدف این مقاله، مطالعهٔ مبانی ریاضی نظریه مشبکه‌ها و کاربردهای آن در سامانه‌های رمز است. نظریهٔ مشبکه‌ها نقش مهمی در طراحی و پیاده‌سازی سامانه‌های رمز جدید و تحلیل رمز دارند. امنیت اکثر سامانه‌های رمزنگاری کلید عمومی مشبکه - مبنا برپایهٔ مسائل سخت محاسباتی یافتن کوتاه‌ترین بردار و یافتن نزدیک‌ترین بردار در مشبکه است. در این مقاله، مقدمه‌ای بر نظریهٔ مشبکه‌ها و مسائل سخت آن‌ها بیان می‌شود؛ سپس مهم‌ترین سامانه‌های رمزنگاری و امضای دیجیتال مشبکه - مبنا با تحلیل‌های امنیتی و مثال‌های کاربردی مورد مطالعه قرار می‌گیرند.

واژگان کلیدی: رمزنگاری مشبکه - مبنا، مسائل سخت مشبکه‌ها، سامانه‌های رمز GGH و NTRU، امضای دیجیتال NSS.

۱- مقدمه

مشبکه‌ها از اواخر قرن ۱۸ و اوایل قرن ۱۹ توسط ریاضی‌دانانی چون لاگرانژ و گاوس مورد بحث و بررسی قرار گرفته‌اند. در این قرن، مینکوفسکی به نتایج مهمی از کاربرد نظریه مشبکه‌ها در نظریه اعداد و هندسه اعداد دست یافت. پیدایش علوم رایانه در قرن ۲۰ منجر به کاربردهای متنوعی از مشبکه‌ها در تجزیهٔ چندجمله‌ای‌ها روی اعداد صحیح، برنامه‌ریزی صحیح و رمزنگاری کلید عمومی شد. ساختارهای رمزنگاری مشبکه - مبنا[1] جذابیت قابل توجهی را در سال‌های اخیر به‌وجود آورده‌اند و از آن‌ها به‌عنوان ساختارهای مقاوم در برابر حملات کوانتومی[2] استفاده می‌شود. امنیت آن‌ها مبتنی بر مسائل سخت ریاضی[3] است و بازدهی و کارایی مطلوبی بین سایر سامانه‌های رمز با کلید عمومی دارند[۱]. در ریاضیات انواع مختلفی از مشبکه‌ها تعریف شده‌اند که دو مورد مهم از این مشبکه‌ها به شرح زیر است:

- مجموعه‌های جزئی مرتب[4] و به‌طورکامل مرتب که تشکیل مشبکه می‌دهند.
- آرایش منظم نقاط در فضا و زیرفضاهای برداری[5] که تشکیل مشبکه می‌دهند.

درواقع رمزنگاری مشبکه - مبنا بر پایهٔ دوم استوار است. نظریهٔ مشبکه‌ها کاربردهای بسیاری در علوم ریاضی و علوم رایانه دارند که به چند مورد از آن‌ها اشاره می‌شود[۲]:

- رمزنگاری
- نظریهٔ کدگذاری و کنترل خطا
- امنیت شبکه‌ها و سیستم‌های رایانه‌ای
- نظریهٔ گروه‌ها
- ترکیبات و نظریهٔ گراف
- نظریهٔ اعداد
- جبرهای لی

1 Lattice-based Cryptography
2 Quantum Resistant
3 Hard Problems
4 POSET
5 Vector Space

- ساختارهای متریک

استفاده از مشبکه‌ها در رمزنگاری مزایای فوق‌العاده‌ای فراهم کرده است؛ درحقیقت امنیت سامانه‌های رمز مشبکه - مبنا مبتنی بر سختی مسائل ریاضی یا مسائل سخت مشبکه‌ها است. این سامانه‌ها سرعت زیادی نسبت به سامانه‌های مبتنی بر تجزیۀ اعداد صحیح[1] و سامانه‌های مبتنی بر لگاریتم گسسته[2] دارند و پیچیدگی کم‌تری نسبت به سامانه‌های رمز الجمال[3] و RSA دارند؛ یعنی از نظر نوع عملیات (جمع و ضرب) در ماتریس‌ها سریع‌اند؛ لیکن تعداد عملیات بسیار زیاد است؛ به‌طوری که در مقایسه با سامانه‌های رمز مبتنی بر منحنی‌های بیضوی[4] کارایی کمتری دارند. همچنین عملیات جبر خطی در سامانه‌های مشبکه - مبنا برای اجرا در سخت‌افزار و نرم‌افزار بسیار ساده هستند؛ یعنی برای پیاده‌سازی بسیار عملی هستند[۲].

در این مقاله همۀ تعاریف و قضایا به همراه مثال‌های عملی از سامانه‌های رمزنگاری مهم جهت ورود به حوزۀ رمزنگاری مشبکه - مبنا فراهم شده است. ادامۀ مباحث این مقاله به شرح زیر تنظیم شده است: در بخش دوم، تعاریف و مفاهیم نظریۀ مشبکه‌ها مطالعه می‌شود. در بخش سوم، سامانۀ رمزنگاری GGH با تحلیل امنیتی و مثال‌های کاربردی مطرح می‌شود. در بخش چهارم، مهم‌ترین سامانۀ رمزنگاری مشبکه - مبنا یعنی سامانۀ رمز NTRU به‌طور کامل شرح داده می‌شود. در بخش پنجم، امضای دیجیتال NSS تشریح می‌شود و درنهایت در بخش ششم، جمع‌بندی و نتیجه‌گیری را خواهیم داشت.

۲- تعاریف و مفاهیم

در این بخش تعاریف و قضایای مورد نیاز در حوزۀ رمزنگاری مشبکه - مبنا ارائه می‌شوند [۱,۲].

تعریف ۱ [۲]: ضرب داخلی[5] دو بردار به‌صورت زیر تعریف می‌شود:

$$<.,.>: R^n \times R^n \to R \qquad (۱)$$

ضرب داخلی بردارها خواص زیر را دارد:

$$\forall u,v,w \in R^n, \forall \lambda \in R:$$
$$i) <u+w,v> = <u,v> + <w,v>$$
$$<\lambda u,v> = \lambda <u,v>$$
$$<u,v+w> = <u,v> + <u,w>$$
$$<u,\lambda v> = \lambda <u,v>$$
$$ii) <u,v> = <v,u>$$
$$iii) <u,u> > 0 \quad (u \neq 0)$$

تعریف ۲ [۲]: ضرب داخلی استاندارد:

$$<(u_1,u_2,...,u_n)^T,(v_1,v_2,...,v_n)^T> := \sum_{i=1}^{n} u_i v_i \qquad (۲)$$

تعریف ۳ [۲]: فرض کنید F یک میدان[6] دلخواه است و داریم:

$$\vec{x} = (x_1,x_2,...,x_n) \in F^n \qquad (۳)$$

تابع نرم[7] به‌صورت زیر تعریف می‌شود:

$$\|.\|: R^n \to R \qquad (۴)$$

که دارای ویژگی زیر است:

$$\forall u,v \in R^n, \forall \lambda \in R:$$
$$i) \|\lambda u\| = |\lambda| . \|u\|$$
$$ii) \|u+v\| \leq \|u\| + \|v\|$$
$$iii) \|u\| > 0 \quad (u \neq 0)$$

تعریف ۴ [۲]: در حالت کلی نرم به‌صورت زیر تعریف می‌شود:

$$l_p = \|(u_1,u_2,...,u_n)^T\|_p := (\sum_{i=1}^{n} |u_i|^p)^{\frac{1}{p}} \qquad (۵)$$

و در حالت خاص داریم:

$$l_1 = \|(u_1,u_2,...,u_n)^T\|_1 := \sum_{i=1}^{n} |u_i|$$
$$l_2 = \|(u_1,u_2,...,u_n)^T\|_2 := (\sum_{i=1}^{n} u_i^2)^{\frac{1}{2}} := \sqrt{<u,u>}$$
$$l_\infty = \|(u_1,u_2,...,u_n)^T\|_\infty := \max_{i=1,2,...,n} |u_i|$$

به‌طور معمول در رمزنگاری از l_2 یا نرم اقلیدسی[8] استفاده می‌شود.

تعریف ۵ [۲]: اگر ضرب داخلی دو بردار برابر با صفر باشد، آن‌ها را متعامد[9] گویند.

۲-۱- روش متعامدسازی گرام اشمیت

فرض کنید $b_1, b_2, \ldots, b_n$ بردارهای مستقل خطی باشند، بردارهای $b^*_1, b^*_2, \ldots, b^*_n$ را بردارهای متعامد به‌دست آمده از بردارهای بالا گویند و به‌صورت زیر محاسبه می‌شوند:

1 Integer Number Factorization
2 Discrete Logarithm
3 ElGamal
4 Elliptic Curve Cryptography
5 Scalar Product
6 Field
7 Norm
8 Euclidean Norm
9 Orthogonal

$$b_1^* = b_1$$
$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j}.b_j^* \qquad (۶)$$
$$\mu_{i,j} = \frac{< b_i, b_j^* >}{< b_j^*, b_j^* >}$$

توجه شود که ترتیب بردارهای $b_1, b_2, \ldots, b_n$ مهم است و به‌عنوان یک دنباله دیده می‌شوند.

مثال ۱: فرض کنید $b_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$ و $b_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$، واضح است که $< b_1, b_2 > \neq 0$ در این‌صورت متعامدسازی بردارهای بالا به‌صورت زیر محاسبه می‌شود:

$$b_1^* = b_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$
$$b_2^* = b_2 - \mu_{2,1}.b_1^*$$
$$\mu_{2,1} = \frac{< b_2, b_1^* >}{< b_1^*, b_1^* >} = \frac{2}{4} = \frac{1}{2}$$
$$\Rightarrow b_2^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 2 \\ 0 \end{pmatrix}$$
$$\Rightarrow b_2^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow B^* = \{b_1^*, b_2^*\} = \{\begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$$

۲-۲- مشبکه‌ها

تعریف ۶ [۲]: فرض کنید V یک فضای برداری باشد و $u, v \in V$ باشند. در این صورت نامساوی کوشی شوارتز تعریف می‌شود:

$$|< u, v >|^2 \le < u, u > . < v, v > \qquad (۷)$$

زمانی تساوی برقرار است که u و v مستقل خطی باشند. واضح است که $< x, x > = \| x \|^2$، در این صورت از نامساوی رابطهٔ (۷) به‌دست می‌آید:

$$i) \; |< u, v >|^2 = | u_1 v_1 + \ldots + u_n v_n |^2 = u_1^2 v_1^2 + \ldots + u_n^2 v_n^2$$
$$= \| u \|^2 . \| v \|^2 = < u, u > . < v, v >$$
$$ii) \; |< u, v >| \le \| u \| . \| v \| \qquad (۸)$$

بنابر تعریف نرم‌های ۱ و ۲ و ∞، می‌توان نشان داد نامساوی‌های زیر بین نرم‌های ۱ و ۲ و ∞ برقرار هستند:

$$i) \; \| u \|_2 \le \| u \|_1 \le \sqrt{n}. \| u \|_2$$
$$ii) \; \| u \|_\infty \le \| u \|_2 \le \sqrt{n}. \| u \|_\infty \qquad (۹)$$
$$iii) \; \| u \|_\infty \le \| u \|_1 \le n. \| u \|_2$$

و طبق تعریف نرم‌های ۱ و ۲ و ∞ از رابطه *(iii)* داریم:

$$\max_i | u_i | \le \sum_i | u_i | \le n \sqrt{\sum_i u_i^2} = n \| u \|_2$$

تعریف ۷ [۲]: فرض کنید $b_1, \ldots, b_n \in R^n$ ستون‌های ماتریس $B \in M_{n,n}(R)$ باشند؛ در این‌صورت نامساوی هادامارد[۱] به‌صورت زیر تعریف می‌شود:

$$| \det B | \le \prod_{i=1}^{n} \| b_i \|_2 \qquad (۱۰)$$

تساوی زمانی برقرار است که بردارهای $b_1, b_2, \ldots, b_n$ متعامد باشند.

تعریف ۸ (مشبکه[۲]) [۲]: فرض کنید $B = \{b_1, \ldots, b_n\}$ یک مجموعه n تایی از بردارهای مستقل خطی در R^m باشند ($n \le m$). مشبکهٔ تولیدشده با B به‌صورت زیر تعریف می‌شود:

$$L(B) := \{\sum_{i=1}^{n} x_i \vec{b}_i \; : x_i \in Z\} \qquad (۱۱)$$

یعنی مجموعه‌ای که از ترکیب خطی صحیح بردارهای پایه، تشکیل شده است. به مجموعه B پایهٔ مشبکه[۳] گویند و به‌صورت فشرده به‌صورت ماتریسی $m \times n$ نشان داده می‌شود که ستون‌هایش (یا سطرهایش) بردارهای پایه B را شکل می‌دهند.

تعریف ۹ [۲]: رتبهٔ[۴] مشبکه را با $rank(L) := n$ و بعد[۵] مشبکه را با $\dim(L) := m$ نشان می‌دهند. هرگاه $m = n$ شود مشبکه را رتبه تمام[۶] گویند.

توجه شود درصورتی‌که بردارهای b_i مستقل خطی باشند، لزوماً *L(B)* یک مشبکه نیست؛ یعنی پایه باید بتواند کل فضای مشبکه را تولید کند.

مثال ۲: بردارهای مستقل خطی $b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ و $b_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ را در نظر بگیرید. مشبکهٔ تولیدشده توسط این دو بردار کل

1 Hadamard Inequality
2 Lattice
3 Lattice Basis
4 Rank
5 Dimension
6 Full Rank Lattice

فضای Z^2 را تولید می‌کند؛ ولی بردارهای $b_1^{'} = \begin{pmatrix}1\\0\end{pmatrix}$ و $b_2^{'} = \begin{pmatrix}0\\2\end{pmatrix}$ نمی‌توانند نقطه $\begin{pmatrix}1\\1\end{pmatrix}$ در فضای Z^2 را تولید کنند زیرا:

$$\nexists\, x, y \in Z : x.\vec{b}_1^{'} + y.\vec{b}_2^{'} = \begin{pmatrix}1\\1\end{pmatrix}$$

پس $\vec{b}_1^{'}$ و $\vec{b}_2^{'}$ پایه ای برای مشبکه نیستند.

حال نشان داده می‌شود که چه زمانی دو پایۀ B_1 و B_2 هم‌ارزند؛ یعنی این دو پایه بتوانند مشبکۀ یکسانی را تولید کنند.

تعریف ۱۰ [۲]: یک ماتریس $U \in Z^{n\times n}$ را تک‌هنگ[1] گویند اگر $\det(U) = \pm 1$ شود.

قضیۀ ۱ [۲]: دو پایه $B_1, B_2 \in R^{m\times n}$ هم ارزند اگر و تنها اگر $B_2 = B_1 U$.

تعریف ۱۱ [۲]: با فرض این‌که B یک پایۀ متشکل از بردارهای مستقل خطی $b_1,...,b_n \in R^m$ باشد، دترمینان مشبکه یا $\det(L)$ برای $L(b_1,...,b_n) \subseteq R^m$ تعریف می‌شود:

$$\det(L) := (\det[< b_i, b_j >]_{1\le i,j\le n})^{\frac{1}{2}} \qquad (۱۲)$$

درصورتی‌که تعریف ۱۱ را به ضرب داخلی استاندارد مقید کنیم، تعریف دترمینان به‌صورت زیر است:

$$\det(L) := \sqrt{\det(B^T B)} \qquad (۱۳)$$

حال اگر $m = n$ باشد، B یک ماتریس مربعی می‌شود و داریم:

$$\det(L) := |\det(B)| \qquad (۱۴)$$

تعریف ۱۲(پدید آوردن) [۲]: n بردار مستقل خطی $b_1,...,b_n \in R^m$ مفروض‌اند. فضای پدید آمده توسط این بردارها به‌صورت زیر تعریف می‌شود:

$$\text{Span}(b_1,...,b_n) = \{\sum_{i=1}^{n} x_i b_i \mid x_i \in R\}$$

با توجه به تعریف مشبکۀ تولیدشده، توسط بردارهای مستقل خطی $b_1,...,b_n$، نتیجۀ زیر حاصل می‌شود:

$$L(b_1,...,b_n) \subset \text{Span}(b_1,...,b_n)$$

قضیۀ بعدی نشان می‌دهد که دترمینان مشبکه خوش‌تعریف است. یعنی دترمینان مستقل از پایه B است.

لازم به ذکر است که در رمزنگاری به‌طورمعمول با مشبکه‌های رتبه‌تمام کار می‌کنند.

قضیه ۲ [۲]: مقدار دترمینان مشبکۀ مستقل از انتخاب پایه $b_1,...,b_n \in R^m$ است.

۲-۳- مسائل سخت مشبکه‌ها

اولین مسأله دشوار از لحاظ محاسباتی در مشبکه‌ها یافتن کوتاه‌ترین بردار[2] در مشبکه است. یعنی به‌طور عمومی یافتن پایه‌ای که بردارهای آن کوتاه‌ترین بردارهای مشبکه باشند از مسائل سخت ریاضی است و یافتن جواب مسألۀ کوتاه‌ترین بردار در مشبکۀ (SVP) از لحاظ محاسباتی بسیار دشوار است[۱].

دومین مسأله دشوار در مشبکه‌ها یافتن نزدیک‌ترین بردار[3] مشبکه به یک نقطه هدف خارج از مشبکه است که حل مسأله یافتن نزدیک‌ترین بردارها در مشبکه (CVP) از مسائل سخت ریاضی بوده و یافتن جواب از لحاظ محاسباتی بسیار دشوار است[۱].

قضایای بسیاری برای رسیدن به یک جواب تقریبی بیان شده‌اند که فقط تضمین می‌کنند کوتاه‌ترین و نزدیک‌ترین بردارها در مشبکه وجود دارند؛ ولی هیچ یک روشی عملی را ارائه نمی‌کنند. در زیربخش ۲-۴ به معرفی مهم‌ترین الگوریتم عملی جهت یافتن یک راه حل تقریبی خوب برای مسائل SVP و CVP در مشبکه‌ها می‌پردازیم که از این الگوریتم مهم در تحلیل رمز سامانه‌های رمزنگاری مشبکه - مبنا به‌وفور استفاده می‌شود.

۲-۴- الگوریتم LLL

هدف از به‌کارگیری الگوریتم‌های کاهش پایه این است که پایه‌های بد را به پایه‌های خوب تبدیل کنند. منظور از پایۀ خوب یعنی پایه‌ای که بردارهای آن بردارهای کوتاه مشبکه باشند (نه لزوماً کوتاه‌ترین بردارها). درواقع پردازشی که این الگوریتم‌ها انجام می‌دهند، کاهش پایۀ مشبکه[4] نامیده می‌شود. یکی از اولین نظریه‌های کاهش پایه، کاهش مینکوفسکی [۲]بود؛ ولی معایبی داشت؛ یعنی نمی‌توان از آن در مشبکه‌هایی با بعد بزرگ‌تر از دو استفاده کرد.

الگوریتم دوم، کاهش دوبعدی گاوسی [۲]نام دارد که یک پایۀ دلخواه از مشبکه دوبعدی را به‌عنوان ورودی گرفته و در زمان چندجمله‌ای، پایۀ کاهش‌یافتۀ تقریبی را به‌عنوان

1 Uni-modular

2 Shortest Vector Problem
3 Closest Vector Problem
4 Lattice Basis Reduction

خروجی می‌دهد. این الگوریتم به‌طورکامل شبیه به الگوریتم اقلیدسی در محاسبهٔ بزرگ‌ترین مقسومٌ علیه مشترک است.

الگوریتم سوم را Kaib و Schnorr ارائه داده‌اند [۲]که یک پایهٔ خوش‌ترتیب[1] از مشبکه دوبعدی را به‌عنوان ورودی می‌گیرد و درنهایت پایهٔ کاهش‌یافته را می‌یابد. یک پایهٔ $[\vec{b}_1, \vec{b}_2]$ خوش‌ترتیب است، اگر:

$$\|\vec{b}_1\| \le \|\vec{b}_1 - \vec{b}_2\| < \|\vec{b}_2\| \qquad (۱۵)$$

الگوریتم فوق توسط Nguyen و Stehle برای مشبکه‌های با بعد دلخواه گسترش داده شده است؛ ولی این الگوریتم اکتشافی تنها برای $m \le 4$ بهینه است[۲].

حال می‌خواهیم الگوریتمی را ارائه دهیم که در بعد دلخواه کار می‌کند و تاکنون الگوریتم بهتر از آن معرفی نشده است. این الگوریتم، LLL نام دارد که توسط Lenstra، Lenstra و Lovász پیشنهاد شده است[۲،۱]و در زمان چند جمله‌ای، بردارهای کوتاه مشبکه را برای ابعاد کوچک مشبکه با تقریب خوبی می‌یابد.

تعریف ۱۳ [۲]: تابع تصویری π_i از R^m به روی $\mathrm{Span}(b_1^*, ..., b_n^*)$ به‌صورت زیر تعریف می‌شود ($\mathrm{Span}(B)$ فضای پدید آمده توسط بردارهای B است):

$$\pi_i(\vec{x}) := \sum_{j=1}^{n} \frac{<\vec{x}, \vec{b}_j^*>}{<\vec{b}_j^*, \vec{b}_j^*>} . \vec{b}_j^* \qquad (۱۶)$$

یک پایه $B = [\vec{b}_1, ..., \vec{b}_n] \in R^{m\times n}$ را کاهش یافته LLL با پارامتر δ که $(\frac{1}{4} < \delta \le 1)$، گویند اگر:

$$\begin{aligned} &i)\, |\mu_{i,j}| \le \frac{1}{2} \;\;, \forall i > j \\ &ii)\, \delta \|\pi_i(\vec{b}_i)\|^2 \le \|\pi_i(\vec{b}_{i+1})\|^2 \end{aligned} \qquad (۱۷)$$

که $1/4 < \delta < 1$ و $\mu_{i,j}$ ضرایب گرام اشمیت بوده و $\vec{b}_i$ و $\vec{b}_{i+1}$ بردارهای متوالی هستند.

شرط اول تضمین می‌کند که پایهٔ تولیدشده کاهش یافته است؛ یعنی نزدیک به متعامد است و شرط دوم تضمین می‌کند که بردار $\vec{b}_1$ یک تقریب خوب و به اندازهٔ کوچک و نزدیک به کوتاه‌ترین بردار است.

قضیه ۳ [۲]: فرض کنید $b_1, b_2, ..., b_n$ یک پایهٔ کاهش‌یافته LLL برای مشبکه $L \in R^n$ بوده و $b_1^*, b_2^*, ..., b_n^*$ بردارهای متناظر متعامد گرام اشمیت باشند.

آنگاه:

$$\|\vec{b}_1\| \le 2^{\frac{n-1}{4}} . \det(L)^{\frac{1}{n}}$$

در نتیجه، خروجی الگوریتم LLL یک پایه کاهش یافته $\{\vec{b}_1, ..., \vec{b}_n\}$ است که نرم آن‌ها به‌صورت زیر محاسبه می‌شود:

$$\|\vec{b}_i\| \le 2^{\frac{n(n-1)}{4(n-i+1)}} . \det(L)^{\frac{1}{n-i+1}} \;, i = 1, 2, ..., n \qquad (۱۸)$$

الگوریتم LLL در زمان چندجمله‌ای اجرا می‌شود لیکن دارای ضریب تقریبی است که با افزایش بعد ماتریس پایه افزایش می‌یابد و دارای کاربردهای زیادی در یافتن راه حل‌های تقریبی قابل قبول برای برخی از مسائل سخت ریاضی مانند یافتن کوتاه‌ترین بردار مشبکه یا نزدیک‌ترین بردار مشبکه به یک بردار هدف مفروض است.

۳- سامانهٔ رمز GGH

سامانهٔ رمزنگاری GGH توسط Goldreich، Goldwasser و Halevi در سال ۱۹۹۷ ارائه شد. در این بخش، سامانهٔ GGH را مورد بحث قرار می‌دهیم[۳].

۳-۱- ساخت کلید

آلیس یک پایهٔ کاهش‌یافته یا پایهٔ خوب $v_1, v_2, ..., v_n \in Z^n$ را که دارای بردارهای متعامد و مستقل خطی است انتخاب می‌کند. برای یافتن پایهٔ خوب از تعریف زیر استفاده می‌شود:

تعریف ۱۴[۱]: نرخ هادامارد[2] برای پایهٔ $B = \{v_1, v_2, ..., v_n\}$ از رابطهٔ زیر محاسبه می‌شود:

$$H(B) = \left(\frac{\det(L)}{\|v_1\| . \|v_2\| ... \|v_n\|}\right)^{\frac{1}{n}} \qquad (۱۹)$$

که $0 < H(B) \le 1$ است و هر چه که به یک نزدیک‌تر باشد، یعنی بردارهای بیشتری از پایه متعامد هستند. آلیس خوب بودن پایه را با بررسی نرخ هادامارد تشخیص می‌دهد؛ یعنی نرخ هادامارد نباید خیلی کوچک باشد.

[1] Well Ordered

[2] Hadamard Ratio

بردارهای $v_1, v_2, \ldots, v_n \in Z^n$ کلید خصوصی آلیس هستند. برای راحتی کار، بردارها را در یک ماتریس قرار داده و آن را V می‌نامیم. همچنین مشبکهٔ تولیدشده از V را L نام‌گذاری می‌کنیم.

در ادامه آلیس ماتریس تک‌هنگ U را چنان انتخاب می‌کند که ماتریس W حاصل از رابطهٔ زیر:

$$W = UV \tag{20}$$

شامل بردارهای مستقل خطی $w_1, w_2, \ldots, w_n$، تشکیل پایهٔ جدیدی برای L دهند که آن را کلید عمومی آلیس می‌نامیم و این پایه به پایهٔ بد نیز معروف است.

۳-۲- رمزگذاری

باب بردار m را به‌عنوان متن اصلی انتخاب می‌کند. همچنین او یک بردار تصادفی r را به‌عنوان یک پارامتر نوفه (خطا) انتخاب می‌کند. متن رمزشده مانند زیر محاسبه می‌شود:

$$e = mW + r = \sum_{i=1}^{n} m_i w_i + r \tag{21}$$

توجه شود که e یک نقطه در فضای مشبکه نیست؛ زیرا بنا به تعریف مشبکه، mW یک عضو مشبکه است که با یک بردار کوچک r جمع‌شده و e را به فضای R^n منتقل می‌کند.

۳-۳- رمزگشایی

برای یافتن نزدیک‌ترین بردار در مشبکه، الگوریتم‌های متنوعی ارائه شده‌اند که مهم‌ترین آن‌ها الگوریتم Babai نام دارد. این الگوریتم یک پایهٔ خوب مشبکه و یک بردار دلخواه در فضای R^n را به‌عنوان ورودی دریافت کرده و نزدیک‌ترین بردار مشبکه به آن بردار دلخواه را در خروجی می‌دهد. به همین منظور آلیس از الگوریتم Babai و کلید خصوصی خود یعنی V برای یافتن یک بردار نزدیک (mW) به e استفاده می‌کند. درنهایت آلیس با یافتن mW می‌تواند به‌سادگی پیام m را به‌دست آورد.

مثال ۳: یک مثال عددی برای سامانهٔ رمز GGH با بعد سه را بررسی می‌کنیم.

$$v_1 = (-97, 19, 19)$$
$$v_2 = (-36, 30, 86)$$
$$v_3 = (-184, -64, 78)$$

$$\det(L) = |\det(V)| = 859516$$

$$H(V) = \left(\frac{\det(L)}{\|v_1\| \ldots \|v_3\|}\right)^{\frac{1}{3}} \approx 0.74620$$

$$U = \begin{pmatrix} 4327 & -15447 & 23454 \\ 3297 & -11770 & 17871 \\ 5464 & -19506 & 29617 \end{pmatrix}$$
$$\det(U) = -1$$

$$W = UV$$
$$w_1 = (-4179163, -1882253, 583183)$$
$$w_2 = (-3184353, -1434201, 444361)$$
$$w_3 = (-5277320, -2376852, 736426)$$
$$H(W) = 0.0000208$$

$$m = (86, -35, -32)$$
$$r = (-4, -3, 2)$$
$$e = mW + r = (-79081427, -35617462, 11035473)$$

برای رمزگشایی، e به‌عنوان یک ترکیب خطی از کلید خصوصی با ضرایب حقیقی نوشته می‌شود:

$$e \approx 81878.97v_1 - 292300v_2 + 443815.04v_3$$

حال ضرایب به نزدیک‌ترین عدد صحیح، گرد می‌شوند:

$$mW = 81879v_1 - 292300v_2 + 443815v_3 = (-79081423, -35617459, 11035471)$$

اینک می‌توان m را به‌صورت یک ترکیب خطی از mW با کلید عمومی W نوشت:

$$mW = 86w_1 - 35w_2 - 32w_3$$

۳-۴- تحلیل امنیتی GGH

با استفاده از الگوریتم LLL می‌توان کلید عمومی یا پایهٔ بد را به پایهٔ کاهش‌یافته یا پایهٔ خوب با بردارهای متعامد تبدیل کرده و سپس با به‌کارگیری این پایهٔ خوب و الگوریتم Babai، نزدیک‌ترین بردار مشبکه به بردار متن رمزشده e را یافت. درواقع اکثر حملات مشبکه بر این واقعیت استوار هستند که به‌کمک کلید عمومی یا پایهٔ بد، کلید خصوصی یا پایهٔ خوب را به دست آورند و نشان داده شده است که GGH و بسیاری از سامانه‌های رمزنگاری مشبکه - مبنا در ابعاد کوچک، ناامن هستند؛ حتی اگر درایه‌های بردارها یا نرم‌بردارها بزرگ انتخاب شوند. GGH به‌جز بعد ۴۰۰ در بعدهای ۲۰۰، ۲۵۰، ۳۰۰ و ۳۵۰ با حملات مبتنی بر مشبکه[1] شکسته شده است[۱].

[1] Lattice Attacks

۴- سامانهٔ رمز NTRU

NTRU یک سامانهٔ رمزنگاری کلید عمومی است که توسط Hoffstein، Pipher و Silverman در سال ۱۹۹۸ ارائه شد [۴]و هم‌اکنون به‌طورکامل استاندارد شده است[۱۷]. NTRU از حلقهٔ چندجمله‌ای‌ها[1] با ضرایب در اعداد صحیح Z استفاده می‌کند. NTRU یک سامانهٔ رمز مشبکه - مبنا بوده و امنیت آن مبتنی بر سختی مسائل ریاضی مشبکه از جمله یافتن کوتاه‌ترین بردار (SVP) و یافتن نزدیک‌ترین بردار (CVP) در فضای مشبکه است.

یکی از چالش‌های NTRU این است که گاهی عملیات رمزگشایی با موفقیت انجام نمی‌شود که با انتخاب صحیح پارامترها و تنظیمات اولیه مناسب می‌توان احتمال عدم موفقیت در رمزگشایی[2] را خیلی کاهش داد و به صفر رساند. از سوی دیگر NTRU مزایای بسیار زیادی دارد، برای یک پیام با طول N، عملیات رمزگذاری و رمزگشایی را با $O(N^2)$ عمل انجام می‌دهد که در سامانهٔ رمز RSA در بهترین حالت $O(N^3)$ عملیات لازم است [۴]. در جدول ۱ مقایسهٔ کارایی و سرعت بین NTRU، RSA و رمزنگاری مبتنی بر خم‌های بیضوی[3] در یک سامانهٔ رایانه‌ای هشتصد مگاهرتز پنتیوم III نشان داده شده است[۵].

همچنین NTRU برخلاف RSA و ECC در برابر حملات مبتنی بر محاسبات کوانتومی، مقاوم است [۶]. NTRU هم اکنون در JAVA و C پیاده‌سازی شده است و طول کوتاه کلیدها در آن، NTRU را به کاراترین سامانهٔ رمزنگاری عملی تبدیل کرده است.

جدول ۱. مقایسه کارایی بین NTRU، RSA و ECC [۵]

		NTRU-251	RSA-1024	ECC-163
کلید عمومی	بیت	۲۰۰۸	۱۰۲۴	۱۶۴
کلید خصوصی	بیت	۲۵۱	۱۰۲۴	۱۶۳
قالب متن اصلی	بیت	۱۶۰	۷۰۲	۱۶۳
قالب متن رمزشده	بیت	۲۰۰۸	۱۰۲۴	۱۶۳
سرعت رمزگذاری	قالب بر ثانیه	۲۲۷۲۷	۱۲۸۰	۴۵۸
	مگابیت بر ثانیه	۳.۶	۰.۹۰	۰.۰۷۵
سرعت رمزگشایی	قالب بر ثانیه	۱۰۸۶۹	۱۱۰	۷۰۲
	مگابیت بر ثانیه	۱.۷	۰.۰۷۷	۰.11

1 Polynomial Ring
2 Decryption Failure
3 Elliptic Curve Cryptography(ECC)

۴-۱- پارامترها

NTRU به چهار پارامتر (n, p, q, d) که اعداد صحیح هستند، بستگی دارد به‌طوری‌که $n > 1$ یک عدد اول است و $(n,q) = (p,q) = 1$ و q خیلی بزرگ‌تر از p انتخاب می‌شود $(q >> p)$ و d یک مقدار ثابت و کوچک‌تر از n است ($d \approx n/3$).

به حلقه خارج قسمتی $Z[x] / <x^n - 1>$ (=، مجموعه چند جمله ای‌های پیچشی[4] از درجه $n - 1$ گویند و می‌توان چند جمله ای‌های)را با بردار ضرایب آن‌ها در Z^n نشان داد. برای مثال چند جمله ای (□ $f = f_0+f_1x+\ldots+f_{n-1}x^{n-1}$ را می‌توان به شکل برداری $f = (f_0, f_1, \ldots, f_{n-1})$ □ Z^n نوشت. عمل جمع حلقه چند جمله ای‌ها جمع مؤلفه به مؤلفه بوده و مانند جمع چند جمله ای‌های معمولی است و عمل ضرب پیچشی چندجمله‌ای‌های حلقه را با نماد * در حلقه)نشان می‌دهند که در رابطه (۲۲) تعریف شده است.

(۲۲)

$$f(x) := \sum_{i=0}^{n-1} f_i x^i = [f_0, f_1, \ldots, f_{n-1}]_{1\times n} \quad , f_i \in Z$$

$$g(x) := \sum_{i=0}^{n-1} g_i x^i = [g_0, g_1, \ldots, g_{n-1}]_{1\times n} \quad , g_i \in Z$$

$$h(x) := \sum_{i=0}^{n-1} h_i x^i = [h_0, h_1, \ldots, h_{n-1}]_{1\times n} \quad , h_i \in Z$$

$$h_K := \sum_{i=0}^{K} f_i . g_{K-i} + \sum_{i=K+1}^{n-1} f_i . g_{n+K-i} = \sum_{i+j \equiv K \pmod n} f_i . g_j$$

فرض کنید $\mathcal{L}_m$، $\mathcal{L}_f$، $\mathcal{L}_g$ و $\mathcal{L}_\Phi$ زیر مجموعه‌های)باشند که در جدول ۲ تعریف شده‌اند. همچنین فرض کنید پارامتر p تعریف شده باشد در اینصورت اگر همه ضرایب (□ f به پیمانه p کاهش یابند گوییم چند جمله ای f به پیمانه p کاهش یافته است یعنی f □ $\mathcal{R}_p$ که در آن $\mathcal{R}_p = (Z/pZ)[x] / <x^n-1>$. با روش مشابه $\mathcal{R}_q$ نیز به دست می‌آید که عمل ضرب * در $\mathcal{R}_p$ و $\mathcal{R}_q$ تعریف شده است. بعلاوه، اگر همه ضرایب f در بازه $(-p/2, p/2]$ ظاهر شوند گوییم f □ $\mathcal{R}_p$ *انتقال حول مبدأ*[5] شده است یعنی f به حالت) □ fدر آمده است.

۴-۲- ساخت کلید

برای تولید کلیدهای NTRU ابتدا دو چندجمله‌ای f □ $\mathcal{L}_f$ و g □ $\mathcal{L}_g$ انتخاب می‌شوند. وارون‌های چندجمله‌ای f □

4 Convolution
5 Centered lift

$\mathcal{R}_p$ و $\mathcal{R}_q$ □ f با $\mathcal{F}_p$ و $\mathcal{F}_q$ نشان داده می‌شوند در اینصورت ∈)$\mathcal{F}_p$, $\mathcal{F}_q$ با شرایط رابطهٔ زیر محاسبه می‌شوند:

$$\mathcal{F}_p * f \equiv 1 \ (mod\ p)$$
$$\mathcal{F}_q * f \equiv 1 \ (mod\ q) \qquad (۲۳)$$

شایان ذکر است که چندجمله‌ای $\mathcal{L}_f$ □ f طوری انتخاب می‌شود که وارون‌پذیر باشد، واضح است درصورتی‌که وارون پذیر نباشد می‌توان چندجمله‌ای دیگری را که وارون پذیر است انتخاب کرد.

قرار می‌دهیم $(= \mathcal{F}_q * g\ (mod\ q)$ که کلید عمومی NTRU بوده و ثابت می‌شود که هم‌ارز عبارت $f * (\equiv g$ $(mod\ q)$ است. همچنین پارامترهای n ، p و q نیز عمومی هستند و کلید خصوصی NTRU را زوج $(f,\ g)$ تشکیل می‌دهند.

جدول ۲. تعریف پارامترهای عمومی NTRU

نماد	تعریف
$\mathcal{L}_f$	) □ f| که f به تعداد $d+1$ تا ضریب 1+ و d تا ضریب 1- دارد و بقیه ضرایب 0 هستند.
$\mathcal{L}_g$	) □ g| که g به تعداد d تا ضریب 1+ و d تا ضریب 1- دارد و بقیه ضرایب 0 هستند.
$\mathcal{L}_\Phi$	) □ Φ| که Φ به تعداد d تا ضریب 1+ و d تا ضریب1- دارد و بقیه ضرایب 0 هستند.
$\mathcal{L}_m$	) □ m| که ضرایب m به پیمانه p و بین $p/2$- و $p/2$ انتخاب می‌شوند.

۴-۳- رمزگذاری

فرض کنید آلیس می‌خواهد پیام m □ $\mathcal{L}_m$ را که ضرایب آن به پیمانهٔ p کاهش یافته است رمزگذاری کند. ابتدا Φ □ $\mathcal{L}_\Phi$ را انتخاب می‌کند که یک چندجمله‌ای تصادفی بوده و نقش یک کلید یک‌بار مصرف را ایفا می‌کند. متن رمزشده در رابطهٔ زیر نشان داده شده است:

$$(= p.\ \Phi * (+ m\ (mod\ q) \qquad (۲۴)$$

۴-۴- رمزگشایی

باب در اولین مرحله رمزگشایی با ضرب پیچشی چندجمله‌ای) در کلید خصوصی f محاسبات خود را شروع می‌کند که در رابطهٔ زیر نشان داده شده است.:

$$(:= f * ((mod\ q) = f * (p.\ (* \Phi + m)\ (mod\ q)$$
$$= p.\ f * (* \Phi + f * m\ (mod\ q)$$
$$= p.\ f * \mathcal{F}_q * g * \Phi + f * m\ (mod\ q)$$
$$= p.\ g * \Phi + f * m\ (mod\ q) \qquad (۲۵)$$

در مرحلهٔ دوم، ضرایب چندجمله‌ای $\mathcal{R}_($ □)، انتقال حول مبدأ می‌شوند پس داریم (□ $p.\ g * \Phi + f * m$، حال ضرایب این چندجمله‌ای به پیمانهٔ p کاهش می‌یابد که درنتیجه جملهٔ $p.\ g * \Phi$ حذف شده و $f * m\ (mod\ p)$ باقی می‌ماند. درنهایت $\mathcal{F}_p$ از سمت چپ در چندجمله‌ای $f *$ $m\ (mod\ p)$ ضرب شده و چندجمله‌ای حاصل‌شده انتقال حول مبدأ می‌شود که همان پیام اصلی است.

نکتهٔ قابل توجه این است که اگر $q > (6d+1).p$ انتخاب شود، مرحله رمزگشایی با موفقیت انجام می‌شود. رمزگشایی موفق بستگی به برقراری رابطهٔ نرم $| p.\ g * \Phi$ $+| f * m \|_\infty < q$ دارد. در [۷]نشان داده شده است که احتمال رمزگشایی موفق با پارامترهای مختلف از رابطهٔ زیر به‌دست می‌آید.

$$Pr(\text{رمزگشایی موفق}) = (2\Psi(q\text{-}1 / 2\sigma)\ \text{-}1)^n \qquad (۲۶)$$

که $\Psi(.)$ توزیع نرمال استاندارد است و $\sigma \approx \sqrt{\frac{36d^2}{n} + \frac{8d}{6}}$.

۴-۵- تحلیل امنیتی NTRU

در این بخش دربارهٔ حملات اصلی به NTRU و تحلیل رمز آن بحث می‌شود. یکی از روش‌های حمله به سامانهٔ رمز NTRU، یافتن کلید خصوصی f و یا کلید جعلی نزدیک به f است.

۴-۵-۱- حملهٔ جستجوی جامع

یکی از راه‌های یافتن کلید خصوصی یا کلید جعلی مشابه با آن، جستجوی جامع همهٔ چندجمله‌ای‌های $\mathcal{L}_f$ □ f' است. در زیربخش ۴-۲ بیان شد که رابطهٔ $f * (\equiv g\ (mod\ q)$ برقرار است پس در یک جستجوی جامع بررسی می‌شود که $f' *$ $((mod\ q)$ ضرایب کوچکی داشته باشد؛ در این صورت کلید خصوصی یا یک بردار نزدیک به آن به دست می‌آید. به‌طور مشابه، در این حمله می‌توان همه $\mathcal{L}_g$ □ g' ممکن را آزمایش کرد که $g' * \mathcal{H}^{-1}\ (mod\ q)$ ضرایب کوچکی داشته باشند تا کلید خصوصی یا یک بردار نزدیک به آن به دست آید. درنتیجه امنیت کلید بستگی به تعداد عناصر $\mathcal{L}_f$ یا $\mathcal{L}_g$ دارد.

درحالتی‌که $p = 2$ باشد، اندازهٔ فضای کلید $\mathcal{L}_g$ از رابطهٔ زیر به‌دست می‌آید.

$$|\mathcal{L}_g| = \binom{n}{d} = \frac{n!}{(n-d)!\,d!} \qquad (۲۷)$$

و درحالتی‌که $p = 3$ باشد، اندازهٔ فضـای کلیـد $\mathcal{L}_g$ از رابطـهٔ زیر به دست می‌آید:

$$|\mathcal{L}_g| = \binom{n}{d}\binom{n-d}{d} = \frac{n!}{(n-2d)!(d!)^2} \qquad (۲۸)$$

پس انتخاب $p = 3$ یک انتخاب مناسب است و فراهم‌کـردن امنیت قابل قبـول بـا کـاهش انـدازهٔ فضـای کلیـد از جملـه مباحث پژوهشی در حوزهٔ عملی رمزنگاری مشبکه - مبناست [۱۱].

۴-۵-۲- حملات مشبکه

Coppersmith و *Shamir* در [۸]یک حملهٔ مشبکه‌ای بـه NTRU برای یافتن کلید خصوصی اعمـال کردنـد کـه ایـن حمله براساس یافتن کوتاه‌ترین بردار مشـبکه پیشـنهاد شـد. مشبکهٔ به‌کاررفته درنسخهٔ استانداردشده NTRU، مشبکه‌ای با ابعاد $2n$ است که با نماد L^{NT} نشان داده مـی‌شـود و بـا ماتریس پایهٔ B^{NT} تولید می‌شود:

$$B^{NT} = \left[\begin{array}{cccc|cccc} \lambda & 0 & \dots & 0 & h_0 & h_1 & \dots & h_{n-1} \\ 0 & \lambda & \dots & 0 & h_{n-1} & h_0 & \dots & h_{n-2} \\ \dots & & & & \dots & & & \\ 0 & 0 & \dots & \lambda & h_1 & h_2 & \dots & h_0 \\ \hline 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \dots & & & & \dots & & & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & q \end{array}\right]_{2n\times 2n}$$

که در ماتریس B^{NT}، $\mathcal{H} = \sum h_i x^i$ کلیـد عمـومی NTRU بوده و λ یک ثابت غیر صفر است. ساده‌نویسی مـاتریس B^{NT} به شکل زیر نمایش داده می‌شود:

$$B^{NT} = \begin{bmatrix} \lambda I & \mathcal{H} \\ 0 & qI \end{bmatrix}$$

در [۱] اثبات شده است که بردار $(\lambda f, g)$ در L^{NT}قرار دارد و به‌احتمال زیاد کوتاه‌ترین بردارهای غیرصفر L^{NT}، (f, g) و چرخش‌های آن هستند. درنتیجه راه حل مسأله SVP یا apprSVP و یافتن کوتاه‌ترین بردارها در مشبکهٔ تولیدشده با ماتریس پایهٔ B^{NT} که به‌عنوان کلید رمزگشایی استفاده می‌شوند، امنیت NTRU را تهدید می‌کند. الگوریتم LLL می‌تواند در زمان چندجمله‌ای مسأله apprSVP را حل کرده و کوتاه‌ترین بردارها را بیابد؛ ولی برای n بزرگ، زمان اجرایی آن نمایی خواهد بود. همچنین الگوریتم دیگری که BKZ-LLL[۹] نام دارد، بردارهای خیلی کوتاه مشبکه را (به‌خصوص در مسأله SIS[1]) محاسبه می‌کند؛ ولی چون به یک پارامتر β بستگی دارد و با عامل $\beta^{2n/\beta}$ مسأله apprSVP را حل می‌کند، درنتیجه برای n بزرگ، زمان اجرایی آن نمایی خواهد بود.

در آزمایش‌هـای متعـددی نشـان داده شـده اسـت کـه سامانهٔ NTRU بـا انتخـاب n بـین ۲۵۱ تـا ۱۰۰۰ امنیتـی معادل بـا امنیـت پیـاده‌سـازی RSA و ElGamal و ECC دارد [۱۰] و بنـا بـه جـدول ۱ کـارایی NTRU بهتـر از RSA و ECC است. بنابراین بـا انتخـاب صـحیح پارامترهـا NTRU سبک‌وزن بوده ولـی امنیـت آن تجربـی اسـت کـه فراهم‌کردن امنیت اثبات‌پذیر از جملـه مباحـث پژوهشـی در حوزهٔ نظـری رمزنگـاری مشـبکه - مبنـا اسـت، بـرای مثـال می‌توان به[۱۸,۱۶,۱۳,۱۲]اشاره کرد.

۴-۵-۳- عامل بسط پیام

در این بخش، نسبت پیام رمزشده بـه مـتن اصـلی محاسـبه می‌شود که این نسبت، عامل مهمـی بـرای انـواع حمـلات از جمله حمله متن اصلی منتخب (CPA) و حملـه مـتن رمـز منتخب (CCA) است. همچنین به‌عنوان عامل مهمـی بـرای تعیین کارایی و سـرعت سـامانه رمزنگـاری NTRU مطـرح است. نسبت بسط به‌صورت زیر محاسبه می‌شود.

$$\log|C| / \log|P| = \log|q| / \log|p| \qquad (۲۹)$$

که C فضای متن‌های رمز شده و P فضای مـتن‌هـای اصـلی است. با انتخاب مناسب پارامترها، عامل بسط پیام بـه‌عنـوان یک مشکل جدی مطرح نمی‌شود.

۴-۵-۴- حملهٔ ارسال چندگانه

اگر فرستنده، پیام m را چندین‌بار با کلیـد عمـومی یکسـان ولی با خطاهای Φ متفاوت ارسال کند، می‌توان اطلاعـاتی را از Φ هـا بـه‌دسـت آورد. فـرض کنیـد فرسـتندهٔ مـتن‌هـای رمزشده متفاوت $\mathcal{E}_i = p.\,\Phi_i * \mathcal{H} + m \pmod q$ را ارسـال می‌کند، آنگاه در این حمله می‌تـوان $(\mathcal{E}_i - \mathcal{E}_1) \pmod q$ را محاسبه کرد. بنابراین با جمع‌آوری $\mathcal{E}_i$ ها مـی‌تـوان $\Phi_i - \Phi_1 \pmod q$ را محاسبه کرده و تعـداد بیـت کـافی بـرای تشخیص Φ_1 را بـه دسـت آورد و حملـهٔ جسـتجوی جـامع

[1] Short Integer Solution

کاراتری را ترتیب داد. درنتیجه در این سامانهٔ رمزنگاری نباید اجازه دهیم تا ارسال چندگانه شکل گیرد. درعمل برای جلوگیری از این حمله از تکنیک‌های لائی‌گذاری[1] (دنباله‌زنی) استفاده می‌شود[۱۴].

۵- امضای دیجیتال NSS

در این بخش به تشریح امضای دیجیتال مبتنی‌بر NTRU یا The NTRU Signature Scheme (NSS) می‌پردازیم [۱۹]. در جدول ۳ مقایسهٔ کارایی الگوریتم امضا و الگوریتم وارسی امضای بین NSS، امضای دیجیتال RSA و امضای دیجیتال ECC (ECDSA) مشاهده می‌شود.

در مرحله انتخاب پارامترها، آلیس اعداد صحیح $(n,\ q,\ d)$ را مطابق زیربخش ۴-۱، انتخاب می‌کند و آلیس چندجمله‌ای‌های (f, g □ را تشکیل داده و کلید عمومی برای وارسی امضا را به‌صورت رابطهٔ زیر، محاسبه می‌کند:

$$(\equiv \mathcal{F}_q * g \ (mod\ q) \qquad (۳۰)$$

شایان ذکر است که این محاسبات مشابه با NTRU است.

۵-۱- الگوریتم امضا و وارسی امضا

برای امضای یک سند $D = (D_1, D_2)$، آلیس به جفت‌های (f, g) و (F, G) نیاز دارد که F و G از رابطه زیر به دست می‌آیند و نحوه محاسبه این رابطه در [۱]بیان شده است.

$$f * G - g * F = q \qquad (۳۱)$$

آلیس دو چند جمله ای زیر را محاسبه می‌کند که در این روابط منظور از ⌊p⌉ یعنی ضرایب چندجمله‌ای p به نزدیک‌ترین عدد صحیح گرد می‌شود.

$$V_1 = \lfloor (D_1 * G - D_2 * F) / q \rceil$$
$$V_2 = \lfloor (-D_1 * g + D_2 * f) / q \rceil \qquad (۳۲)$$

در نهایت آلیس امضای خود را که به‌صورت زیر محاسبه شده است به همراه سند D منتشر می‌کند.

$$S = V_1 * f + V_2 * F \qquad (۳۳)$$

در سمت گیرنده، باب امضای آلیس (S) و سند (D) را دریافت کرده و توسط کلید عمومی $\mathcal{H}$، رابطه زیر را محاسبه می‌کند.

$$t \equiv (* S \ (mod\ q) \qquad (۳۴)$$

حال باید بردار ضرایب چندجمله‌ای t به پیمانه q تا جایی که امکان دارد به بردار ضرایب D_2 نزدیک باشد تا باب بتواند بررسی کند که بردار (S, t) به‌طور مناسب به بردار سند $D = (D_1,\ D_2)$ در فضای مشبکه نزدیک است، در این صورت امضای آلیس پس از وارسی، تأیید می‌شود.

۵-۲- تحلیل امنیتی NSS

در این بخش، دلیل وابستگی NSS به مشبکه NTRU بحث می‌شود. در این طرح، پایه خوب و پایه بد برای L^{NT} به‌صورت زیر نشان داده می‌شوند.

جدول ۳. مقایسه سرعت سامانه‌های امضای NSS، امضای دیجیتال RSA و ECDSA [۱۵]

	پنتیوم	تجهیزات سبک وزن
الگوریتم امضای NSS	۰.۳۵ میلی ثانیه	۰.۳۳ میلی ثانیه
الگوریتم امضای RSA	۶۶.۵۵ میلی ثانیه	۳۶.۱۳ میلی ثانیه
الگوریتم امضای ECDSA	۱.۱۸ میلی ثانیه	۱.۷۹ میلی ثانیه
الگوریتم تأیید اعتبار امضای NSS	۰.۲۹ میلی ثانیه	۰.۲۵ میلی ثانیه
الگوریتم تأیید اعتبار امضای RSA	۱.۲۳ میلی ثانیه	۰.۷۳ میلی ثانیه
الگوریتم تأیید اعتبار امضای ECDSA	۱.۷۰ میلی ثانیه	۳.۲۶ میلی ثانیه

$$\text{پایهٔ خوب} = \begin{pmatrix} f & g \\ F & G \end{pmatrix} \quad \text{پایه بد} = \begin{pmatrix} 1 & \mathcal{H} \\ 0 & q \end{pmatrix}$$

آلیس برای امضای سند $D = (D_1, D_2)$، بردارهای $(D_1,\ D_2)$ را برحسب پایهٔ خوب، به‌صورت زیر نمایش می‌دهد:

$$(D_1, D_2) = (u_1, u_2)\begin{pmatrix} f & g \\ F & G \end{pmatrix} \qquad (۳۵)$$

و طبق رابطهٔ زیر معادلهٔ (۳۵) را برای (u_1,u_2) محاسبه می‌کند:

$$(u_1, u_2) = (D_1, D_2)\begin{pmatrix} f & g \\ F & G \end{pmatrix}^{-1} = (D_1, D_2)\begin{pmatrix} G/q & -g/q \\ -F/q & f/q \end{pmatrix} \qquad (۳۶)$$

شایان ذکر است که وارون ماتریس پایهٔ خوب، همیشه وجود دارد؛ زیرا براساس رابطهٔ (۳۱) داریم:

$$det\begin{pmatrix} f & g \\ F & G \end{pmatrix} = q \qquad (۳۷)$$

با توجه به این که درایه‌های u_1 و u_2 لزوماً اعداد صحیح نیستند، حاصل‌ضرب سمت راست رابطهٔ (۳۵) در

[1] Padding

مشبکه L^{NT} نیست؛ پس آلیس u_1 و u_2 را بـه‌صـورت رابطـهٔ زیر به نزدیک‌ترین عدد صحیح گرد می‌کند:

$$V_1 = \lceil u_1 \rfloor ,\ V_2 = \lceil u_2 \rfloor \qquad (۳۸)$$

بنابراین منطقی است که بردار زیر

$$(S,t) = (V_1, V_2)\begin{pmatrix} f & g \\ F & G \end{pmatrix}$$

به D نزدیک باشد. درنتیجه چـون $(S, t) \in L^{NT}$ اسـت، پـس آلیس نیاز ندارد که جفت (S,t) را منتشـر و بـاب مـی‌توانـد طبق رابطهٔ (۳۴)، t را با S و کلید عمومی)محاسبه کنـد. بـا توجه به ایـن‌کـه NSS نیـز مشـابه بـا NTRU وابسـته بـه ماتریس مشبکه L^{NT} است بنابراین امنیت NSS بسـتگی بـه سختی مسـائل سـخت CVP و SVP دارد. درنتیجـه NSS امنیت مطلوبی را فراهم می‌کند. همچنین با توجه به جـدول ۳، مشاهده می‌شود که NSS کاراتر از سایر طرح‌های امضای دیجیتال عمل می‌کند. برای بهبوددادن کارایی و امنیت طرح امضای NSS پژوهش‌هایی انجام شده است کـه بـرای مثـال می‌توان بـه NTRUSign [۱۵]اشـاره کـرد کـه هـم‌اکنـون به‌طورکامل استاندارد شده است]۱۷[.

۶- نتیجه‌گیری

رمزنگاری مشـبکه - مبنـا یـک حـوزهٔ جـوان ولـی بـا رشـد چشم‌گیر است. در این مقاله تمرکز ما روی جنبه‌های نظـری و عملی از سامانه‌های رمزنگاری مشبکه - مبنـا بـوده اسـت. حوزهٔ پژوهشی در رمزنگاری مشبکه - مبنا را می‌توان بـه دو دسـتهٔ نظـری و عملـی تقسـیم کـرد. در حـوزهٔ نظـری، پژوهش‌های پایه‌ای انجام می‌گیرد کـه نقـش بسـیار مهمـی برای هرچه نزدیک‌ترکردن رمزنگاری مشبکه - مبنا به کاربرد ایفا می‌کند؛ ولـی در حـوزهٔ عملـی، هـدف ارائـهٔ طـرح‌هـای رمزنگاری کارا و در عین حال امن محاسباتی است.

مراجع

[1] J. Hoffstein, J. Pipher, and J.H. Silverman, An Introduction to Mathematical Cryptography, Springer-Verlag, 1st edition, (2008).

[2] P. Mol, Lattices and their Applications to RSA Cryptosystem, Diploma Thesis, National Technical University of Athens, Department of Electrical and Computer Engineering, (2006).

[3] O. Goldreich, Sh. Goldwasser, and Sh. Halevi, Public-key cryptosystems from lattice reduction problems, In CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, London, UK, (1997), pp. 112-131.

[4] J. Hoffstein, J. Pipher and J. H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, Algorithmic Number Theory, LNCS 142, Springer-Verlag, (1998), pp. 267-288.

[5] J. Hoffstein, J.H. Silverman, W. Whyte, Estimated Breaking Times for NTRU Lattices, NTRU Cryptosystems Technical Report 12, Version 2, updated 2006. http://www.securityinnovation.com. Accessed Nov 2014.

[6] J. Hoffstein, N. Howgrave-Graham, J. Pipher and W. Whyte, Practical lattice-Based cryptography: NTRUEncrypt and NTRUSign, The LLL Algorithm: Survey and Applications, Information Security and Cryptography Book, Springer-Verlag, (2010), pp. 349-390.

[7] R. Kouzmenko, Generalization of the NTRU cryptosystem, Master's thesis, Polytechnique Montreal, Canada, (2006).

[8] D. Coppersmith and A. Shamir, Lattice Attacks on NTRU, Advances in Cryptology, EUROCRYPT '97, LNCS 1233, Springer-Verlag, (1997), pp. 52-61.

[9] K. Jarvis, and M. Nevins, ETRU: NTRU over the Eisenstein Integers, Designs, Codes and Cryptography, (2013), DOI: 10. 1007/s10623-013-9850-3, Springer.

[10] NTRU Cryptosystems. Estimated breaking times for NTRU lattices. Technical report, 1999, Updated (2003), Tech. Note 012,www.ntru.com/cryptolab/tech_notes.htm.

[11] D. Micciancio, Improving lattice based cryptosystems using the Hermite normal form, In Cryptography and Lattices Conference (CaLC), (2001), pp. 126-145.

[12] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of ACM, Vol. 56, (2009), pp. 6-34.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, On ideal lattices and learning with errors over rings, In H. Gilbert, editor, Advances in Cryp-tology EUROCRYPT, Vol. 6110 of LNCS, (2010), pp. 1-23.

[14] T. Meskanen, On The NTRU Cryptosystem, Diploma Thesis, University of Turku, Department of Mathematics, Finland, (2005).

[15] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J.H. Silverman, and W. Whyte, NTRU Sign: digital signature using the NTRU lattice, In Topics in cryptology-CT-RSA, Vol. 2612 of LNCS, (2003), pp. 122-140, Springer, Berlin.

[16] A. Hassani Karbasi, R. Ebrahimi Atani, ILTRU: An NTRU-Like Public Key Cryptosystem Over Ideal Lattices, The 7th International IEEE Symposium on Telecommunication (IST'14), Tehran, Iran, (2014).

[17] Standard Specifications for Public-Key Cryptographic Techniques Based on Hard Problems

Over Lattices. IEEE P1363, (2008). Available at: http://grouper.ieee.org/ groups/1363/.

[18] A. Blum, A. Kalai, and H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, Journal of ACM, Vol. 50, No. 4, (2003), pp. 506-519.

[19] J. Hoffstein, J. Pipher, and J.H. Silverman, NSS: An NTRU Lattice-Based Signature Scheme, In EUROCRYPT'01, Vol. 2045 of LNCS, (2001), pp. 211-228.

امیر حسنی کرباسی مدرک‌های کارشناسی و کارشناسی ارشد خود را به‌ترتیب از دانشگاه‌های تبریز و گیلان در شهریورماه ۱۳۸۹ و اسفندماه ۱۳۹۱ دریافت کرده است. ایشان هم‌اکنون دانشجوی استعداد درخشان دکتری ریاضیات رمزنگاری در دانشگاه گیلان هستند. نامبرده عضو دانشجویی انجمن رمز ایران هستند و زمینه‌های پژوهشی مورد علاقهٔ ایشان شامل کاربرد جبر در رمزنگاری، رمزنگاری کلید عمومی، سامانه‌های رمزنگاری مشبکه – مبنا، امضاهای دیجیتال مشبکه – مبنا و امنیت اطلاعات است. از ایشان تاکنون بیش از ۲۸ عنوان مقاله در مجلات و همایش‌های ملی و بین المللی و یک عنوان کتاب به چاپ رسیده است.

رضا ابراهیمی آتانی استادیار گروه مهندسی رایانهٔ دانشکده فنی و مهندسی دانشگاه گیلان است. نامبرده دکترای خود را در سال ۱۳۸۹ در رشتهٔ مهندسی الکترونیک از دانشگاه علم و صنعت ایران دریافت کرد. ایشان عضو پیوستهٔ انجمن رمز ایران و انجمن‌های بین المللی IACR و IEEE هستند. از ایشان تاکنون دو عنوان کتاب و بیش از یکصد مقاله در مجلات و کنفرانس‌های ملی و بین‌المللی به چاپ رسیده است. زمینهٔ پژوهشی مورد علاقهٔ وی طراحی و پیاده‌سازی الگوریتم‌های رمزنگاری، امنیت شبکه و امنیت نرم‌افزار است.