فرهنگ و اندیشهٔ ریاضی
سال ۳۸، شمارهٔ ۶۴ (بهار و تابستان ۱۳۹۸)، صص. ۱۱۷ تا ۱۴۲

آشنایی با رمزنگاری خم‌های بیضوی

مجتبی بهرامیان

چکیده

بخش بزرگی از رمزنگاری در سال‌های اخیر به رمزنگاری خم‌های بیضوی اختصاص یافته است. خم‌های بیضوی دسته‌ای از خم‌های جبری با ساختار گروه هستند. رمزنگاری خم‌های بیضوی یک روش رمزنگاری کلید عمومی مبتنی بر نظریهٔ خم‌های بیضوی است که با استفاده از ویژگی‌های خم‌های بیضوی به‌جای روش‌های قبلی مانند تجزیه به حاصل‌ضرب اعداد اول، امنیت بالاتری را با طول کلید کوتاهتر فراهم می‌کند. این بخش از رمزنگاری در توافق و تبادل کلید، امضای رقمی، تجزیهٔ اعداد بزرگ، آزمون اول بودن و ... کاربرد دارد. در این مقاله، رمزنگاری بر اساس خم‌های بیضوی را مرور و کاربردهایی از آن را تشریح می‌کنیم. در پایان نیز برتریِ استفاده از خم‌های بیضوی را به‌طور خلاصه بیان می‌کنیم.

۱. سرآغاز

ایمنی اطلاعات از دیرباز مورد توجه انسان بوده است و تا پایان حیات بشر ادامه خواهد یافت. در دنیای امروزی و با پیشرفت فناوری، علاوه بر امنیت جان و مال و ...، امنیت نگهداری و تبادل اطلاعات نیز از اهمیت ویژه‌ای برخوردار است. نگهداری و تبادل ایمن اطلاعات با استفاده از رمزنگاری امکان‌پذیر شده است. رمزنگاری را می‌توان فرآیند تبدیل یك متن ساده به متنی نامفهوم و به‌عکس دانست. در رمزنگاری، ارتباط بین افراد در یک شبکهٔ ناامن با حفظ حریم خصوصی و اصالت داده‌ها به‌صورت امن امکان‌پذیر می‌گردد. متن پیام به‌کمک یك کلید و با استفاده از یك الگوریتم تغییر می‌یابد به‌طوری که تنها افراد خاصی که از کلید آگاهی دارند، می‌توانند به اطلاعات دسترسی پیدا کنند. دانش رمزنگاری بر اساس

عبارات و کلمات کلیدی. رمزنگاری؛ خم‌های بیضوی؛ پیچیدگیِ زمانی؛ لگاریتم گسسته؛ امضای دیجیتال.

زمینه‌های بسیاری از قبیل نظریهٔ اطلاعات، نظریهٔ اعداد، آمار و ... بنا شده است و امروزه در بسیاری از علوم از جمله علوم نظامی، مخابرات و ... استفاده می‌شود.

رمزنگاری از آغاز پیدایش تا به امروز، کاربردهای فراوانی داشته و همچنان در حال گسترش است. البته برخی از روش‌های رمزنگاری به‌دلیل سادگی و شکننده بودن، موارد استفادهٔ مقطعی داشته‌اند و تنها آثار اندکی از آنها بر جای مانده است. آن گونه که از اسناد باقیمانده برمی‌آید، سابقهٔ رمزنگاری به ۲۰۰۰ سال قبل از میلاد مسیح برمی‌گردد؛ زمانی که مصری‌ها با استفاده از تصاویر غیرمتداول، رمزنگاری را به کار گرفتند. یونانیان باستان، رومی‌ها و مصری‌ها را می‌توان اولین انسان‌هایی دانست که از رمز کردن برای مخفی کردن اطلاعاتشان استفاده کرده‌اند. با این حال، نخستین سامانهٔ رمز بجا مانده، مربوط به ۱۰۰ سال قبل از میلاد است که توسط ژولیوس سزار[۱]، امپراطور روم، ابداع شد. او در مکاتبات خود و برای مخفی ماندن محتوای نوشته‌هایش جای حروف الفبا را عوض می‌کرد.

پیچیدن یک نوار کاغذی بر روی استوانه‌ای با قطر مشخص و نوشتن پیام روی آن کاغذ، از ابتدایی‌ترین سامانه‌های رمزنگاری بوده است. در این سامانهٔ رمزنگاری، تنها کسانی که نسخه‌های یکسانی از استوانه را داشته‌اند، قادر به خواندن پیام بوده‌اند. در قرن بیستم میلادی از همین شیوه به همراه موتورهای الکتریکی برای رمزنگاری با سرعت بالا استفاده شد که نمونه‌های آن در ماشین رمز لورنتس[۲] و ماشین رمز انیگما[۳] دیده می‌شود.

از سال ۱۹۷۰ که رمزنگاری کلید عمومی یا رمزنگاری نامتقارن ایجاد شد، انقلابی در رمزنگاری رخ داد. در سال ۱۹۷۶ ویتفیلد دیفای[۴] و مارتین هلمن[۵] رمزنگاری کلید عمومی را ابداع کردند. در واقع آنها روشی برای تبادل کلید در کانال ناامن ارائه کردند که مبتنی بر سختیِ مسئلهٔ لگاریتم گسسته است و هم‌اکنون به نام تبادل کلید دیفای–هلمن شناخته می‌شود. پس از آن، الگوریتم‌های کلید عمومیِ زیادی ارائه گردید که از جملهٔ آنها می‌توان به RSA، سامانهٔ رمز رابین[۶]، الجمال[۷] و رمزنگاری خم‌های بیضوی اشاره کرد. سامانهٔ رمز RSA در سال ۱۹۷۷ توسط رونالد ریوست[۸]، آدی شامیر[۹] و لئونارد آدلمن[۱۰] [۲۲] ارائه شد. این سامانه که بر سختیِ تجزیهٔ اعداد بزرگ مبتنی است، موفقیت بزرگی در رمزنگاری به شمار می‌آید و در حال حاضر نیز مورد استفاده قرار می‌گیرد. در سال ۱۹۷۹ رابین سامانهٔ رمز خود را ارائه کرد. این سامانه نیز همانند رمز RSA بر سختیِ تجزیهٔ اعداد بزرگ مبتنی است. با وجود برتری‌هایی که این سامانه نسبت به رمز RSA دارد، کمتر از آن استفاده می‌شود. الجمال با ایده گرفتن از تبادل کلید دیفای–هلمن، طرح خود را در سال ۱۹۸۴ ارائه کرد. این سامانه نیز همانند رمز RSA مورد استفادهٔ زیادی قرار گرفت و پایهٔ بسیاری از سامانه‌های رمزنگاریِ امروزی است.

---

[۱] Julius Caesar [۲] Lorenz [۳] Enigma [۴] Whitfield Diffie [۵] Martin Hellman [۶] Rabin [۷] ElGamal
[۸] Ronald Rivest [۹] Adi Shamir [۱۰] Leonard Adelman

نیل کوبلیتز[۱] و ویکتور میلر[۲]، در سال ۱۹۸۵ استفاده از خم‌های بیضوی را برای استفاده در رمزنگاری پیشنهاد دادند. خم‌های بیضوی دسته‌ای از خم‌های جبری هستند که ساختار گروه دارند. به‌دلیل سختیِ حل مسئلهٔ لگاریتم گسسته در خم‌های بیضوی، استفاده از آنها در رمزنگاری بسیار کم‌هزینه و باصرفه است. برتریِ رمزنگاری خم‌های بیضوی نسبت به سایر سامانه‌ها این است که در این روش، با طول کلید کوتاه‌تر همان امنیت حاصل می‌شود.

۲. رمزنگاری چیست؟

با پدید آمدن رایانه و افزایش قدرت محاسباتی آن، دانش رمزنگاری وارد مرحله‌ای نوین شد. قدرت محاسباتی بالا، ضمن ایجاد روش‌های پیچیده‌تر و مؤثرتر در رمزنگاری، باعث شد عمل رمزگذاری روی انواع اطلاعات و بر مبنای بیت انجام شود در حالی که پیش از آن، عمدتاً روی اطلاعات متنی و با استفاده از حروف الفبا انجام می‌گرفت.

تعریف ۱.۲. یک سامانهٔ رمزنگاری[۳]، یك پنج‌تایی $(\mathcal{M}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ از مجموعه‌ها است که در آن، $\mathcal{M}$ فضای متن، $\mathcal{C}$ فضای رمز و $\mathcal{K}$ فضای کلید نام دارد. همچنین هر عضو $\mathcal{E}$ تابعی مانند $e : \mathcal{M} \to \mathcal{C}$ به‌نام تابع رمز و هر عضو $\mathcal{D}$ تابعی مانند $d : \mathcal{C} \to \mathcal{M}$ به‌نام تابع کشف رمز است که در آن، برای هر $e \in \mathcal{E}$ یك $d \in \mathcal{D}$ موجود است به‌طوری که برای هر $m \in \mathcal{M}$، $d(e(m)) = m$.

یک سامانهٔ رمز باید از امنیت کافی برخوردار باشد. از نگاه آگوست کرکهوفس[۴] برای امنیت یك سامانهٔ رمز لازم است:

(۱) رمز کردن یك پیام با استفاده از کلید رمز ساده باشد؛
(۲) باز کردن یك متن رمزی با استفاده از کلید رمزگشا ساده باشد؛
(۳) بدون داشتن کلید رمزگشا، باز کردن یك متن رمزی سخت باشد؛
(۴) با داشتن چند پیام و متن رمزیِ متناظر با آنها، باز کردن یك متن رمزی سخت باشد.

با مقایسهٔ اصول (۲) و (۳) می‌توان دید که باز کردن یك متن رمزی c با داشتن کلید رمزگشای d ساده و بدون آن، دشوار است. بنابراین امنیت کلید از اهمیت ویژه‌ای برخوردار است. برای مثال، در سامانهٔ رمز سزار با داشتن یک متن پیام و متن رمز شدهٔ نظیرش، می‌توان به کلید رمزگشا (که تعداد جابه‌جایی حروف است) پی برد. پس این سامانه فاقد شرط (۴) از اصول کرکهوفس است و از امنیت لازم برخوردار نیست.

[۱] Neal Koblitz [۲] Victor Miller [۳] Cryptosystem [۴] Auguste Kerckhoffs

۱.۲. سامانهٔ رمز متقارن و نامتقارن. اجزای اصلی در یك سامانهٔ رمزنگاری، الگوریتم و کلیدهای رمزکننده و رمزگشا هستند. در رمزنگاری نوین، الگوریتم مورد استفاده در یك سامانهٔ رمزنگاری منتشر می‌شود اما کلیدها مخفی می‌مانند. چنانچه کلید رمزگشا به‌راحتی از کلید رمزکننده به‌دست آید، سامانهٔ رمزنگاری مذکور *سامانهٔ رمز متقارن* وگرنه *سامانهٔ رمز نامتقارن* نامیده می‌شود. در هر سامانهٔ رمزنگاری، کلید رمزگشا مخفی است. با توجه به اینکه در سامانهٔ رمز متقارن، کلید رمزگشا از کلید رمزکننده به‌دست می‌آید، کلید رمزکننده نیز خصوصی است و منتشر نمی‌شود؛ در حالی که در سامانهٔ رمز نامتقارن کلید رمزکننده عمومی است. از این‌رو سامانهٔ رمز متقارن را *رمزنگاری کلید خصوصی* و سامانهٔ رمز نامتقارن را *رمزنگاری کلید عمومی* نیز می‌نامند. با توجه به اینکه در سامانهٔ رمز کلید عمومی، کلیدهای عمومی و خصوصی متفاوت هستند، از دو فضای کلید متفاوت استفاده می‌شود. در یك سامانهٔ رمز کلید عمومی، امنیت سامانه به طول کلید آن بستگی دارد و هرچه قدرت محاسبات بالاتر رود، طول کلید نیز باید بیشتر شود. برای مثال، در سامانهٔ رمز RSA، پیش از این، امنیت با طول کلید ۵۱۲ بیت حاصل می‌شد اما هم‌اکنون از کلیدهای با طول ۲۰۴۸ بیت استفاده می‌شود.

فرض کنیم فرماندهٔ یک پادگان از چند سرباز خود می‌خواهد پس از جمع‌آوری اطلاعات محرمانه، آنها را مخفیانه به او برسانند. برای این منظور، صندوقی را که در آن شکافی کوچک تعبیه شده است، در ملأ عام قرار می‌دهد. هر سرباز پس از به‌دست آوردن اطلاعات، آنها را روی کاغذ می‌نویسد و درون صندوق قرار می‌دهد. فرماندهٔ پادگان تنها کسی است که قادر به باز کردن صندوق است و این کار را با کلید خصوصی خود انجام می‌دهد. این سامانه، نمونه‌ای از یک سامانهٔ رمز کلید عمومی است که در آن، صندوق، کلید عمومی و کلید صندوق، کلید خصوصی است.

امروزه از هر دو سامانهٔ رمز متقارن و نامتقارن استفاده می‌شود و هرکدام کاربرد و ماهیت خاص خود را دارد. هر سامانه مزایا و معایبی نسبت به دیگری دارد. از برتری‌های رمز متقارن این است که الگوریتم‌ها در این سامانه نسبت به الگوریتم‌های رمز نامتقارن، سریع‌تر است. اما این سامانه معایبی نیز دارد. چون حفظ و تبادل کلید در یک سامانهٔ رمز، دارای اهمیت ویژه‌ای است، فراهم کردن کانالی امن برای تبادل کلید در کنار حفظ، توزیع و تعداد زیاد کلیدها از مشکلات استفاده از این سامانه است. مثلاً در مجموعه‌ای n نفره که هر دو نفر مایل به برقراری ارتباط محرمانه هستند، هر یک از $\binom{n}{2}$ جفت متمایز نیاز به یک کلید محرمانه دارند و به‌علاوه هر شخص باید $n-1$ کلید را حفظ کند. پس استفاده از چنین سامانه‌ای برای گروه‌های بزرگ مانند گروه کاربران اینترنت مشکل است. با وجود این، سامانهٔ رمز متقارن هنوز هم جایگاه خود را حفظ کرده و از بسیاری از سامانه‌های رمز نامتقارن مؤثرتر است.

۲.۲. پیچیدگی زمانی. پیچیدگی زمانیِ یک الگوریتم، تعداد دفعات اجرای عمل اصلی به‌عنوان تابعی از اندازهٔ ورودی است. زمان اجرای برنامه‌ها یکی از عامل‌های مقایسهٔ دو الگوریتم برای حل یک

مسئله است. هدف از محاسبهٔ پیچیدگی زمانیِ یک الگوریتم، تعیین نیاز یک الگوریتم به زمان بر اساس یک تابع رشد است. هرچه زمان اجرای یک الگوریتم سریع‌تر باشد، آن الگوریتم سودمندتر خواهد بود.

فرض کنیم $n \geq ۳$ عددی طبیعی و u و α اعداد حقیقی مثبت باشند. قرار می‌دهیم

$$L_n[u, \alpha] = e^{\alpha(\log n)^u (\log\log n)^{۱-u}}$$

که در آن، n ورودی الگوریتم است.

- اگر زمان اجرای الگوریتم به‌صورت $L_n[۰, \alpha] = (\log\ n)^\alpha$ باشد، گوییم الگوریتم دارای *پیچیدگی زمانیِ چندجمله‌ای* است؛
- چنانچه زمان اجرای الگوریتم به‌صورت $L_n[۱, \alpha] = e^{\alpha \log n} = n^\alpha$ باشد، آن را دارای *پیچیدگی زمانیِ نمایی* گوییم؛
- اگر زمان اجرای الگوریتم به‌صورت $L_n[u, \alpha]$ باشد که در آن، $۰ < u < ۱$، الگوریتم را *زیرنمایی* می‌نامیم.

الگوریتم با پیچیدگی زمانیِ چندجمله‌ای، مؤثر و الگوریتم با پیچیدگی زمانیِ نمایی، ناکارآمد تلقی می‌شود. همان گونه که مشخص است، در الگوریتم با پیچیدگی زمانیِ چندجمله‌ای، زمان اجرای الگوریتم تابعی چندجمله‌ای برحسب $\log n$ (طول رشتهٔ ورودی) و در الگوریتم با پیچیدگی زمانیِ نمایی، زمان اجرای الگوریتم تابعی نمایی برحسب $\log n$ است.

فرض کنیم $f, g : \mathbb{N} \to \mathbb{R}^+$ دو تابع باشند. تساوی $f(n) = O(g(n))$ به این معنی است که $c \in \mathbb{R}^+$ و $n_۰ \in \mathbb{N}$ وجود دارند که برای هر $n \geq n_۰$ داریم $f(n) \leq cg(n)$. نماد O رفتار حدی یک تابع را توصیف می‌کند. این نماد به کاربر اجازه می‌دهد که تابع را ساده کند و بر روی نرخ رشد آن متمرکز شود. به عبارت دیگر، توابع مختلف با نرخ رشد یکسان دارای نماد O مشابه‌اند.

تابع $f : X \to Y$ را که محاسبهٔ آن آسان ولی محاسبهٔ معکوسش دشوار باشد، یک *تابع یک‌طرفه* می‌نامیم. منظور از سادگی و دشواری محاسبه، پیچیدگی زمانیِ الگوریتم محاسبه است. الگوریتمی که دارای پیچیدگی زمانیِ چندجمله‌ای است، آسان و در غیر این صورت، دشوار تلقی می‌شود. توابع یک‌طرفه لزوماً یک‌به‌یک نیستند.

مثال ۲.۲. برای عدد اول بزرگ q و ریشهٔ اولیهٔ $g \in \mathbb{F}_q^*$، تابع $e_g : \mathbb{Z} \to \mathbb{F}_q^*$ با ضابطهٔ $e_g(k) = g^k$ را در نظر می‌گیریم. ریاضیدانان باستان، الگوریتمی برای محاسبهٔ نمای گسسته طراحی کردند. این الگوریتم دنباله‌ای از اعمالی است که هر جمله با g برابر کردن یا مربع کردنِ جملهٔ قبل به‌دست می‌آید و بنابراین محاسبهٔ g^k به حداقل $\log k$ و حداکثر $۲ \log k$ عمل نیاز دارد. پس الگوریتم محاسبهٔ $e_g(k) = g^k$ از مرتبهٔ $O(\log k)$ و بنابراین دارای پیچیدگی زمانیِ چندجمله‌ای است. از طرفی با داشتن $h \in \mathbb{F}_q^*$،

الگوریتمی با پیچیدگی زمانیِ چندجمله‌ای برای محاسبهٔ k با ویژگی $g^k = h$ وجود ندارد. بنابراین تابع e_g یک تابع یک‌طرفه است. محاسبهٔ k با داشتن g و $h = g^k$، *لگاریتم گسسته* نامیده می‌شود.

به‌طور کلی در یک گروه G اگر $g \in G$ و $h \in \langle g \rangle$، آنگاه کوچکترین عدد مثبت k را که در رابطهٔ $h = g^k$ صدق می‌کند، *لگاریتم گسستهٔ* h در پایهٔ g می‌نامیم آن را با $\log_g h$ نشان می‌دهیم. حل مسئلهٔ لگاریتم گسسته، یعنی محاسبهٔ $\log_g h$. *مسئلهٔ لگاریتم گسسته*[۱] به‌اختصار با DLP نمایش داده می‌شود.

لگاریتم گسسته، عکس نمای گسسته و محاسبهٔ آن در بعضی گروه‌ها کنترل‌ناپذیر است. پس می‌توان در چنین گروه‌هایی از تابع نمای گسسته به‌عنوان تابع یک‌طرفه استفاده کرد. بسته به نوع الگوریتم مورد استفاده، محاسبهٔ لگاریتم گسسته در یک گروه ممکن است بسیار آسان و یا خیلی سخت باشد. در رمزنگاری، معمولاً دو دسته از گروه‌ها مورد استفاده قرار می‌گیرند که عبارت‌اند از گروه ضربی میدان‌های متناهی و گروه نقاط خم‌های بیضوی. در گروه خم‌های بیضوی بجز در مواردی خاص، الگوریتمی وجود ندارد که مسئلهٔ لگاریتم گسسته در آن و در زمان چندجمله‌ای حل شود در حالی که در گروه ضربی میدان‌های متناهی، الگوریتم‌های سریع‌تری موجود است.

نخستین الگوریتم زیرنمایی برای حل مسئلهٔ لگاریتم گسسته در میدان‌های متناهی دارای پیچیدگی زمانیِ $L_n[۱/۲, \alpha]$ بود که نخست برای میدان‌های با مرتبهٔ عدد اول طراحی شده بود و سپس به سایر میدان‌ها نیز تعمیم یافت. این الگوریتم در سال ۲۰۰۶ به پیچیدگی زمانیِ $L_n[۱/۳, \alpha]$ بهبود یافت. در سال ۲۰۱۳ الگوریتم سریع‌تری برای میدان‌های با مشخصهٔ کوچک طراحی شد و این میدان‌ها را از فهرست گروه‌های مورد استفاده در سامانه‌های رمزنگاری خارج کرد. هرچند هنوز هم استفاده از گروه ضربی میدان‌های متناهی در رمزنگاری متداول است، سامانه‌های رمزنگاری خم‌های بیضوی در مقایسه با آنها و با طول کلید کوتاهتر همان امنیت را دارند.

۳. لگاریتم گسسته و اهمیت رمزنگاری

در سامانه‌های رمزنگاری مرسوم است که از اسامی آلیس و باب برای دریافت کننده و فرستندهٔ یک پیام استفاده شود. ما نیز در این مقاله از آن پیروی می‌کنیم. در سامانهٔ رمز متقارن، رمزکننده می‌تواند بازکنندهٔ رمز نیز باشد. اگر در این سامانه آلیس پیامی را برای باب ارسال کند، هر دو یک توانایی برای گشودن رمز دارند. در سامانهٔ رمز نامتقارن، چنین امکانی وجود ندارد. اگر آلیس و باب بخواهند پیامی را تبادل کنند، برای رمز کردن به یک کلید خصوصی نیاز دارند. چنانچه آنها بتوانند همدیگر را ببینند یا در یک کانال امن ارتباط برقرار کنند، می‌توانند روی یک کلید خصوصی توافق کنند اما اگر چنین امکانی وجود نداشته باشد یا هر گونه ارتباط آنها تحت نظر یک جاسوس باشد، آیا آنها می‌توانند روی یک کلید

[۱] discrete logarithm problem

مخفی توافق کنند؟ تا مدت‌ها تفکر عموم این بود که چنین توافقی حاصل نمی‌شود مگر با فرصت‌های مورد نیاز از جمله وجود یک کانال امن. اما دیفای و هلمن برای اولین بار چنین کلیدی را طراحی کردند.

۱.۳. لگاریتم گسسته و اهمیت رمزنگاری. مسئلهٔ لگاریتم گسسته به‌دلیل استفاده در پروتکل‌های رمزنگاری مانند تبادل کلید دیفای-هلمن، در سال‌های اخیر به میزان قابل توجهی مورد اقبال واقع شده است. برای ساختن یك سامانهٔ رمز کلید عمومی، از یك مسئلهٔ سخت ریاضی استفاده می‌شود که برای حل آن، الگوریتمی سریع و کارا وجود نداشته باشد. مسئلهٔ لگاریتم گسسته و مسئلهٔ تجزیهٔ اعداد صحیح، کاندیداهای اصلی مسائل سخت هستند. دیفای و هلمن برای معرفی مفهوم رمزنگاری کلید عمومی از مسئلهٔ لگاریتم گسسته استفاده کردند [۳]. یکی از روش‌های رمزنگاری، رمز کردن بر اساس گروه‌هایی است که حل مسئلهٔ لگاریتم گسسته در آنها دشوار باشد. تاکنون تعداد اندکی از گروه‌ها کاندیدای عملی برای اجرای رمزنگاری بوده‌اند که برخی از آنها عبارت‌اند از: گروه ضربی میدان‌های متناهی، عناصر وارون‌پذیر $\mathbb{Z}_n$ که در آن، n عدد طبیعی مرکب است، گروه خم‌های بیضوی، ژاکوبین خم‌های ابربیضوی، چنبره‌های جبری و گروه رده‌ای یک میدان مربعیِ موهومی. هم‌اکنون یافتن گروه‌های دیگری که حل مسئلهٔ لگاریتم گسسته در آنها سخت باشد و به موازات آن، پیدا کردن یک الگوریتم مؤثر برای حل مسئلهٔ لگاریتم گسسته در این گروه‌ها، در حال مطالعه و بررسی است.

در سال ۱۹۸۵ ایدهٔ برجسته‌ای از کوبلیتز [۹] و میلر [۱۷]، گروه خم‌های بیضوی را به خدمت رمزنگاری درآورد. کوبلیتز [۱۰] ایدهٔ خود را با در نظر گرفتن ژاکوبین خم‌های اَبَربیضوی در سال ۱۹۸۸ تعمیم داد. سپس کارل روبین[۱] و آلیس سیلوربرگ[۲] [۲۳]، خانوادهٔ دیگری از گروه‌های جبری به‌نام چنبره‌های جبری را در رمزنگاری به‌کار بردند. این گروه‌ها در رمزنگاری از جذابیت ویژه‌ای برخوردار هستند. مطالعهٔ ژاکوبین خم‌های با گونای کوچک نیز در سال‌های اخیر پیشرفت خوبی داشته است. یکی از دلایل آن، وجود خم‌هایی از این نوع است که هیچ الگوریتم زیرنمایی برای حل مسئلهٔ لگاریتم گسستهٔ متناظر با آنها موجود نیست. از سوی دیگر، چنبره‌های جبری روی میدان‌های متناهی دارای این مزیت هستند که نمایش عناصر آنها به‌صورت فشرده برای تبادل اطلاعات، به داده‌های کمتری احتیاج دارد. نمایش فشردهٔ اعضای گروه، یعنی نمایشی که بتوان آن گروه را به‌راحتی و با استفاده از یک برنامهٔ رایانه‌ای معرفی کرد. برای مثال، نمایش ماتریسی، یک نمایش فشرده است.

به‌طور خلاصه، پژوهشگران در نظریهٔ رمزنگاری، گروه ژاکوبین خم‌ها را به‌دلیل امنیت آن و چنبره‌های جبری را به‌سبب کارآمد بودنشان دوست دارند. بنابراین می‌توان گفت که این دو زیرخانواده از گروه‌های جبری به‌نوعی مکمل یکدیگر هستند.

---

[۱] Karl Rubin [۲] Alice Silverberg

۲.۳. جزئیات پروتکل دیفای-هلمن. سامانهٔ تبادل کلید دیفای-هلمن در سال ۱۹۷۶ توسط دیفای و هلمن طراحی شد. تا قبل از انتشار آن، رمزنگاری بیشتر به‌صورت رمزنگاری کلید متقارن مورد استفاده قرار می‌گرفت. بعد از آن بود که این پروتکل، پایهٔ اولیهٔ رمزنگاری کلید نامتقارن شد و سپس توسط رالف مرکل[۱] تکمیل گردید. پروتکل تبادل کلید دیفای-هلمن، یک پروتکل رمزنگاری است که با استفاده از آن، دو نفر یا دو سازمان می‌توانند بدون نیاز به آشنایی قبلی، از طریق یك مسیر ارتباطی ناامن، یک کلید رمز مشترک ایجاد نمایند. این پروتکل، اولین روش علمی مطرح‌شده برای تبادل کلید رمز در کانال‌های ارتباطی ناامن است و مشکل تبادل کلید رمز در رمزنگاری کلید متقارن را آسان می‌سازد. با طرح دیفای و هلمن، موجی از علاقه به رمزنگاری نوین آغاز شد. در کار دیفای و هلمن ایدهٔ رمزنگاری کلید عمومی و امضای رقمی[۲] ارائه شد. هرچند آنها یك سامانهٔ رمزنگاری را پیاده‌سازی نکردند اما روشی برای دستیابی به یك کلید مشترک ارائه کردند که *توافق کلید* نامیده می‌شد. در این بخش، چگونگی استفاده از مسئلهٔ لگاریتم گسسته در تبادل کلید نشان داده می‌شود.

فرض کنیم آلیس و باب قصد دارند پیامی محرمانه را مکاتبه کنند. بدین منظور، آنها نخست گروه دوری $G = \langle g \rangle$ از مرتبهٔ عدد اول p را انتخاب و آن را منتشر می‌کنند. فرض بر این است که مسئلهٔ لگاریتم گسسته در این گروه دشوار است. آنها سپس به‌شکل زیر عمل می‌کنند:

(۱) آلیس پس از انتخاب توان مخفی a، مقدار $A = g^a$ را محاسبه می‌کند و آن را برای باب می‌فرستد (a کلید خصوصی آلیس است)؛
(۲) باب نیز توان مخفی b را انتخاب می‌کند و پس از محاسبهٔ $B = g^b$، آن را برای آلیس می‌فرستد (b کلید خصوصی باب است)؛
(۳) آلیس با دریافت B و با استفاده از کلید خصوصی a، مقدار $B^a = g^{ab}$ را محاسبه می‌کند. باب نیز به‌طور مشابه $A^b = g^{ab}$ را محاسبه می‌کند.

g^{ab} کلید مشترک آلیس و باب خواهد بود.

مشخص است که برای تولید کلید مشترک، این دو نفر نیازی به کانال امن و حتی شناخت یکدیگر ندارند. حال اگر جاسوسی در کانال ناامن بخواهد به این کلید دسترسی پیدا کند، لازم است با داشتن g، g^a و g^b، کلید g^{ab} را محاسبه کند. برای این منظور، او باید نخست با استفاده از g و g^a، توان a را محاسبه و سپس با داشتن g^b و a، کلید g^{ab} را محاسبه کند. در واقع او باید مسئلهٔ لگاریتم گسسته را در گروه G حل کند که بنابر فرض، دشوار است. بنابراین امنیت این پروتکل مبتنی بر دشواری حل مسئلهٔ لگاریتم گسسته است.

[۱] Ralph Merkle [۲] digital signature

امروزه بر اساس قدرت محاسباتی رایانه‌ها، می‌توان با استفاده از عدد اول p با حدود ۳۰۰ رقم و اعداد a و b با حدود ۱۰۰ رقم، شکستن امنیت این پروتکل و یافتن کلید رمز مشترک را عملاً ناممکن ساخت. در عمل، هر عدد اول بزرگی را نمی‌توان در این پروتکل به‌کار گرفت، بلکه لازم است با توجه به نوع الگوریتم مورد استفاده، عدد p یک عدد اول امن باشد. اگرنه شکستن امنیت این پروتکل و یافتن کلید رمز مشترک با استفاده از الگوریتم‌هایی مانند الگوریتم پهلیگ–هلمن[۱]، نسبتاً آسان و در زمان کمتری قابل انجام خواهد بود.

در گروه $G = \langle g \rangle$، مسئلهٔ محاسبهٔ g^{ab} با استفاده از g^a و g^b را *مسئلهٔ محاسباتی دیفای–هلمن* می‌نامیم و به‌اختصار با CDHP نمایش می‌دهیم. همان‌طور که بیان شد، اگر شخصی قادر به محاسبهٔ a از g^a باشد، به‌راحتی می‌تواند g^{ab} را محاسبه کند، یعنی مسئلهٔ DLP، مسئلهٔ CDHP را نتیجه می‌دهد اما در اکثر موارد نمی‌دانیم که آیا DLP و CDHP با هم معادل‌اند یا خیر. در بخش بعدی، به نمونه‌ای از سامانه‌های رمزنگاری می‌پردازیم که با پروتکل تبادل کلید مذکور ارتباط مستقیم دارند.

۳.۳. سامانهٔ رمز الجمال. الجمال در سال ۱۹۸۴ با ایده گرفتن از تبادل کلید دیفای–هلمن، یك سامانهٔ رمز کلید عمومی مبتنی بر مسئلهٔ لگاریتم گسسته طراحی کرد [۴]. طرح الجمال به این صورت است: فرض کنیم باب می‌خواهد پیامی را برای آلیس بفرستد. آنها روی گروه G و عنصر $g \in G$ به توافق می‌رسند و G و g را منتشر و به‌صورت زیر عمل می‌کنند:

(۱) آلیس کلید مخفی a را انتخاب و $A = g^a$ را محاسبه می‌کند. سپس A را به‌عنوان کلید عمومی منتشر می‌کند؛

(۲) باب عدد تصادفی k را انتخاب و برای رمز کردن متن m، نخست $c_1 = g^k$ و $c_2 = mA^k$ را محاسبه و سپس (c_1, c_2) را برای آلیس ارسال می‌کند؛

(۳) آلیس پس از دریافت (c_1, c_2)، نخست $x = c_1^a$ و سپس $x^{-1}c_2$ را محاسبه می‌کند. توجه کنید که

$$x^{-1}c_2 = (c_1^a)^{-1}c_2 = (g^{ak})^{-1}mg^{ak} = m$$

و لذا m به‌راحتی توسط آلیس محاسبه می‌شود.

حال اگر کسی در شبکهٔ ناامن بخواهد به پیام m دسترسی پیدا کند، لازم است نخست از روی $c_1 = g^k$ و g و با حل یک مسئلهٔ لگاریتم گسسته، k را محاسبه کند و با استفاده از آن، A^k و $m = c_2(A^k)^{-1}$ را به‌دست آورد؛ یعنی با استفاده از مسئلهٔ لگاریتم گسسته می‌توان رمز الجمال را شکست. به عبارت دیگر، مسئلهٔ شکستن رمز الجمال حداکثر به‌سختیِ مسئلهٔ لگاریتم گسسته است. می‌توان دید که مسئلهٔ دیفای–هلمن نیز حداکثر به‌سختیِ مسئلهٔ شکستن رمز الجمال است. توجه کنید که اگر باب همواره از عدد

[۱] Stephen Pohlig-Martin Hellman

تصادفی k برای رمزکردن پیام استفاده کند، پس از مدتی، مسئلهٔ لگاریتم گسستهٔ $c_۱ = g^k$ حل شده و k به‌دست می‌آید. بنابراین عدد تصادفی k که توسط باب انتخاب شده است، تنها برای رمزکردن یک پیام استفاده می‌شود و برای رمزکردن پیام دیگر باید این عدد تغییر کند. از این‌رو k را *کلید روزانه* می‌خوانیم.

۴. سامانهٔ رمز RSA

تا قبل از انتشار تبادل کلید دیفای-هلمن، رمزنگاری بیشتر به‌صورت رمزنگاری کلید متقارن مورد استفاده قرار می‌گرفت. یك سال بعد از طراحی تبادل کلید دیفای-هلمن، الگوریتم رمز RSA توسط ریوست، شامیر و آدلمن [۲۲] مطرح گردید. رمزنگاری RSA از بزرگترین پیشرفت‌ها در زمینهٔ رمزنگاری به حساب می‌آید و همچنان به‌صورت وسیعی در تبادلات الکترونیکی به‌کار می‌رود و در صورت استفادهٔ درست با کلیدهای به‌اندازهٔ کافی طولانی، کاملاً امن به نظر می‌رسد. در سامانه تبادل کلید دیفای-هلمن، امنیت بر اساس سختیِ حل مسئلهٔ لگاریتم گسسته و در سامانهٔ رمز RSA امنیت رمز، مبتنی بر سختیِ تجزیه است.

۱.۴. پیاده‌سازی رمز RSA. اگر p و q دو عدد اول بزرگ متمایز باشند و $n = pq$، آن‌گاه $\varphi(n) = (p-۱)(q-۱)$. فرض کنیم $(e, \varphi(n)) = ۱$. در این صورت e دارای وارون حسابی d در پیمانهٔ $\varphi(n)$ است. به‌راحتی می‌توان دید که معادلهٔ همنهشتی $x^e \equiv c \pmod{n}$ دارای جواب یکتای $x \equiv c^d \pmod{n}$ است. اکنون با هدفِ ارسال پیام از سوی باب برای آلیس، مراحل تولید کلید، رمز کردن و رمزگشایی به‌صورت زیر انجام می‌شود:

(۱) **تولید کلید**: آلیس دو عدد اول بزرگ p و q را انتخاب و عدد $n = pq$ را حساب می‌کند. پس از محاسبهٔ $\varphi(n) = (p-۱)(q-۱)$، عدد تصادفی e را متباین با n و سپس d را عکس حسابیِ e در پیمانهٔ $\varphi(n)$ در نظر می‌گیرد. بنابراین $ed \equiv ۱ \pmod{\varphi(n)}$. آلیس d را به‌عنوان کلید خصوصی نزد خود حفظ و (n, e) را به‌عنوان کلید عمومی برای باب ارسال می‌کند؛

(۲) **ارسال پیام**: باب پیام m را با استفاده از کلید عمومی به‌صورت $c \equiv m^e \pmod{n}$ رمز کرده آن را برای آلیس ارسال می‌کند؛

(۳) **بازگشایی پیام**: آلیس پس از دریافت c، با استفاده از کلید خصوصی خود، d، پیام را به‌صورت $m \equiv c^d \pmod{n}$ رمزگشایی می‌کند. توجه می‌کنیم که

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv ۱ \pmod{n}.$$

اکنون اگر شخصی بخواهد در شبکهٔ ناامن به پیام m دسترسی پیدا کند، لازم است d را بیابد و $(m^e)^d$ را در پیمانهٔ n حساب کند. چون d عکس حسابیِ e در پیمانهٔ $(p-1)(q-1)$ است، پس او باید $(p-1)(q-1)$ را داشته باشد و این مستلزم دانستنِ p و q و لذا تجزیهٔ n است که نشدنی (دشوار) است.

رابین در سال ۱۹۷۹ سامانهٔ رمز خود را که بر سختیِ مسئلهٔ تجزیهٔ اعداد بزرگ پایه‌ریزی شده بود، ارائه کرد [۲۱]. به همین دلیل، این سامانه شباهت‌هایی با رمز RSA دارد و در عین حال، نسبت به آن سریع‌تر است. در اینجا نیز فرض بر این است که $n = pq$ که در آن، p و q دو عدد اول بزرگ هستند. فرض کنید باب قصد دارد پیام m را برای آلیس ارسال کند. او $c = m^2 \pmod n$ را محاسبه می‌کند و آن را برای آلیس می‌فرستد. آلیس با استفاده از کلید خصوصی خود، ریشهٔ دوم c را در پیمانهٔ m محاسبه و پیام را بازیابی می‌کند. رابین نشان داد که الگوریتم یافتن ریشهٔ دوم یك عدد در پیمانهٔ $n = pq$، با الگوریتم تجزیهٔ n هم‌ارز است و لذا امنیت سامانهٔ رمز رابین شبیه سامانهٔ رمز RSA است.

۵. امضای رقمی

مدارک الکترونیکی نیز همانند متون کاغذی نیاز به امضا دارند. این امضاها شباهت‌هایی با امضاهای دستی دارند. امضای رقمی، طرحی برای نشان دادن درستی پیام‌های ارسالی است. هنگامی که یك پیام از سوی یك نفر برای دیگری ارسال می‌شود، گیرندهٔ پیام به‌وسیلهٔ امضای رقمی، هویت فرستنده را شناسایی می‌کند و از اعتبار پیام مطمئن می‌شود. در سال ۱۹۷۶ دیفای و هلمن نخستین بار، مفهوم امضای رقمی را بیان کردند؛ هرچند آنها وجود چنین طرحی را فقط حدس زدند. سپس ریوست، شامیر و آدلمن الگوریتم امضای رقمی RSA را طراحی کردند. اولین امضای رقمی که در سال ۱۹۸۹ ارائه شد، از الگوریتم RSA استفاده می‌کرد. الگوریتم امضای رقمی DSA که در سال ۱۹۹۱ توسط مؤسسهٔ ملی استاندارد و فناوری[۱] ایالات متحده طراحی شده است، نمونه‌ای از یك امضای رقمی است. این مؤسسه با نام اختصاری NIST شناخته می‌شود.

یك امضای رقمی از سه الگوریتم تشکیل شده است: الگوریتم تولید کلید، الگوریتم تولید امضا و الگوریتم تأیید امضا. در الگوریتم تولید کلید، گیرندهٔ پیام یك کلید خصوصی تولید می‌کند، سپس با استفاده از کلید عمومی و متن ارسالی، امضا توسط فرستندهٔ پیام تولید و در پایان، صحت امضا با استفاده از کلید خصوصی توسط گیرندهٔ پیام تأیید می‌شود. فرض کنیم آلیس می‌خواهد مدرک m را امضا کند و برای باب بفرستد:

(۱) آلیس کلید خصوصی d را انتخاب می‌کند؛

[۱] National Institute of Standards and Technology

(۲) او $s = s(d, m)$ را محاسبه می‌کند؛

(۳) باب با استفاده از کلید عمومی e (متناظر با d) صحت امضای آلیس یعنی s را بررسی می‌کند.

مراحل بالا به‌ترتیب، تولید کلید، تولید امضا و تأیید امضا است.

در یک امضای رقمی امن، هیچ‌کس بدون اطلاع از کلید مخفی d نمی‌تواند $s(d, m)$ را تولید کند. تاکنون امضاهای رقمی متعددی معرفی شده‌اند که امضای RSA، امضای الجمال و طرح امضای DSA از مهم‌ترین آنها است.

۱.۵. امضای RSA. ایدهٔ امضای RSA بسیار شبیه به سامانهٔ رمز RSA است. در این روش آلیس دو عدد اول بزرگ p و q را انتخاب و $n = pq$ را محاسبه می‌کند. او e را چنان می‌یابد که $(e, (p-1)(q-1)) = 1$. در این صورت e در پیمانهٔ $(p-1)(q-1)$ دارای عکس حسابی d است، یعنی $ed \equiv 1 \pmod{(p-1)(q-1)}$. در این امضا، d کلید خصوصی آلیس و (n, e) کلید عمومی است. نخست آلیس $s = s(d, m) = m^d \pmod{n}$ و سپس باب

$$s^e = (m^e)^d = m^{ed} \equiv m \pmod{n}$$

را محاسبه می‌کند. s امضای آلیس است و اگر $s^e \equiv m \pmod{n}$، آنگاه صحت امضای آلیس توسط باب تأیید می‌شود.

امنیت این امضا دقیقاً شبیه امنیت سامانهٔ رمز RSA است. چنانچه شخصی بخواهد امضای آلیس را جعل کند، نیازمند این است که $s = m^d$ را محاسبه کند. برای این منظور، او باید به کلید خصوصی آلیس یعنی d دسترسی پیدا کند. پس لازم است عکس حسابی e را در پیمانهٔ $(p-1)(q-1)$ محاسبه کند؛ در حالی که او $(p-1)(q-1)$ را در اختیار ندارد و اطلاع از آن، نیازمند تجزیهٔ n و محاسبهٔ p و q است که کار سختی است.

در امضای RSA، تولید امضا به‌صورت توان و لذا تابعی ضربی است. یك روش حمله به امضای RSA به این شکل است که چنانچه دو متن m_1 و m_2 توسط کلید خصوصی d امضا شده باشند، آنگاه جعل امضای $m_1 m_2$ به‌راحتی صورت می‌گیرد، زیرا اگر $s_1 = m_1^d$ و $s_2 = m_2^d$ به‌ترتیب، امضاهای متون m_1 و m_2 و s امضای $m_1 m_2$ باشد، آنگاه $s = (m_1 m_2)^d = m_1^d m_2^d = s_1 s_2$.

حملهٔ دیگری که به امضای RSA و بعضی امضاهای دیگر می‌شود، جعل وجودی است که در آن، جعل‌کنندهٔ امضا، s را به گونه‌ای انتخاب می‌کند که s^e متنی بامعنی باشد. او با ارائهٔ $m' = s^e$ می‌تواند مدعی شود که امضا توسط آلیس صورت گرفته است. در واقع در صورتی که s^e متنی با معنی باشد، او در جعل امضا موفق شده است. این عمل، *جعل وجودی* نامیده می‌شود. ناگفته نماند که روش‌هایی برای جلوگیری از این حملات ارائه شده است.

۲.۵. امضای الجمال. امضای الجمال نیز شبیه سامانهٔ رمز الجمال است و سختیِ آن مبتنی بر سختیِ حل مسئلهٔ لگاریتم گسسته است. در این روش، آلیس برای تولید کلید، عدد اول بزرگ p را به‌صورت تصادفی انتخاب و g را یک ریشهٔ اولیه در پیمانهٔ p می‌گیرد. فرض بر این است که حل مسئلهٔ لگاریتم گسسته در $\mathbb{F}_p^*$ سخت است. همچنین $a \in \{۱, ۲, \ldots, p-۲\}$ را نیز به‌صورت تصادفی انتخاب می‌کند. a کلید خصوصی او است. آلیس $A \equiv g^a \pmod p$ را محاسبه و (p, g, A) را منتشر می‌کند. سپس آلیس برای امضای متن m (که می‌توان فرض کرد یك عدد است و به همراه امضا منتشر می‌شود)، عدد تصادفی $k \in \{۱, ۲, \ldots, p-۲\}$ را چنان انتخاب می‌کند که $(k, p-۱) = ۱$. اگر k' وارون حسابی k در پیمانهٔ $p-۱$ باشد، او مقادیر

$$r \equiv g^k \pmod p, \quad s \equiv k'(m - ar) \pmod{(p-۱)}$$

را محاسبه می‌کند. (r, s) امضای آلیس است. اکنون باب برای تأیید صحت امضای آلیس، با در اختیار داشتنِ متن m و با استفاده از کلید عمومی (p, g, A)، دو مقدار $A^r r^s$ و g^m را محاسبه و آنها را مقایسه می‌کند. در صورتی که همنهشتی $A^r r^s \equiv g^m \pmod p$ برقرار باشد، صحت امضای آلیس تأیید می‌شود، زیرا در چنین حالتی

$$A^r r^s \equiv g^{ar} g^{ks} \equiv g^{ar} g^{m-ar} \equiv g^m \pmod p.$$

نکات زیر دربارهٔ طرح امضای بالا قابل ذکر است:

(الف) برای جعل امضای آلیس، زوج (r, s) باید چنان ساخته شود که $A^r r^s \equiv g^m \pmod p$. چون g و m هر دو عمومی‌اند، محاسبهٔ g^m به‌سادگی انجام می‌شود. اکنون باید r و s چنان باشند که $A^r r^s \equiv g^m \pmod p$. پس از محاسبهٔ A^r با داشتن r، مقدار s از رابطهٔ $r^s \equiv (A^r)^{-۱} g^m \pmod p$ محاسبه می‌شود و این یعنی حل یک مسئلهٔ لگاریتم گسسته که بنابر فرض، نشدنی است.

(ب) در این طرح، لازم است برای هر بار امضای جدید، مقدار k متفاوت از دفعات قبل انتخاب شود. چون اگر آلیس برای امضای دو متن $m_۱$ و $m_۲$ از کلید k استفاده کند و برای $i = ۱, ۲$، (r_i, s_i) امضای m_i باشد، آنگاه $r_i \equiv g^k \pmod p$ و $s_i \equiv k'(m_i - ar_i) \pmod{p-۱}$. بنابراین $s_۱ - s_۲ \equiv k'(m_۱ - m_۲) \pmod{p-۱}$. حال اگر $(m_۱ - m_۲, p-۱) = ۱$ و $(m_۱ - m_۲)^{-۱}$ عکس حسابی $(m_۱ - m_۲)$ در پیمانهٔ $p-۱$ باشد، آنگاه از رابطهٔ

$$s_۱ - s_۲ \equiv k'(m_۱ - m_۲) \pmod{p-۱}$$

مقدار k' و در نتیجه k به‌دست می‌آید. پس لازم است برای هر بار امضای جدید، مقدار k متفاوت از دفعات قبل انتخاب شود.

۳.۵. الگوریتم امضای رقمی (DSA). الگوریتم امضای DSA نمونه‌ای از امضای الجمال و در عین حال، سریع‌تر از آن است. تفاوت این طرح با امضای الجمال این است که در امضای الجمال، سه بار عمل توان‌رسانی انجام می‌شود در حالی که در امضای DSA، این تعداد به دو عمل کاهش می‌یابد. همچنین در امضای DSA از توان‌های به‌مراتب کوچکتری در مقایسه با امضای الجمال استفاده می‌شود. این تفاوت‌ها بیانگر کاراییِ الگوریتم DSA است. در این طرح نیز جهت پیشگیری از جعل، لازم است پس از یك بار امضا، از توان جدیدی استفاده شود.

۶. تابع چکیده‌ساز

صحت پیام در ارسال آن اهمیت زیادی دارد. این مهم توسط پروتکل‌های ارتباطی از قبیل امضاهای رقمی و توابع چکیده‌ساز[۱] امکان‌پذیر است. توابع چکیده‌ساز در پروتکل‌های ارتباطی نقش اساسی دارند. برای مثال، عدم استفاده از این توابع در یك امضای رقمی باعث ناکارآمدیِ امضا خواهد شد. در این بخش، مفهوم تابع چکیده‌ساز را بیان و به چند نمونه از آنها اشاره می‌کنیم.

تابع چکیده‌ساز تابعی است مانند

$$h : \{۰,۱\}^* \to \{۰,۱\}^n$$

که در آن، n عددی ثابت است. در این تعریف، $\{۰,۱\}^*$ رشته‌ای با طول دلخواه و $\{۰,۱\}^n$ رشته‌ای با طول ثابت n است. بنابراین تابع چکیده‌ساز ورودی D را با طول دلخواه دریافت و خروجی $H = h(D)$ با طول ثابت را محاسبه می‌کند. همچنین لازم است:

(۱) محاسبهٔ تابع چکیده‌ساز ساده باشد، یعنی با داشتنِ h و $D \in \{۰,۱\}^*$، محاسبهٔ $h(D)$ به‌سادگی انجام شود؛

(۲) محاسبهٔ عکس آن سخت باشد؛ به این معنی که با داشتن h و $H \in \{۰,۱\}^n$، یافتن $D \in \{۰,۱\}^*$ با ویژگی $H = h(D)$ دشوار باشد؛

(۳) در برابر تصادم، مقاوم باشد، یعنی پیدا کردن $D_۱$ و $D_۲$ با ویژگی $h(D_۱) = h(D_۲)$ دشوار باشد.

فرض کنیم آلیس در محل کار خود از یک نرم‌افزار استفاده می‌کند. او قبل از آغاز به کار، از صحت و اعتبار نرم‌افزار خود مطمئن می‌شود و برنامه‌های خود را دنبال می‌کند. از این‌رو برای اطمینان از این

[۱] hash functions

موضوع، یک تابع چکیده‌ساز h را در نظر می‌گیرد و چکیدهٔ برنامهٔ خود را محاسبه و آن را درکارت هوشمند خود ذخیره می‌کند (چکیدهٔ این برنامه ظرفیت زیادی را اشغال نمی‌کند). آلیس صبح روز بعد و پیش از آغاز کار، مجدداً چکیدهٔ برنامهٔ خود را محاسبه و آن را با مقدار قبلی مقایسه می‌کند. با این کار، او تا حد زیادی از صحت و عدم دستکاریِ برنامهٔ خود مطمئن می‌شود، زیرا اگر شخصی که از تابع چکیده‌ساز h اطلاع دارد، بخواهد برنامهٔ آلیس را دستکاری کند، درصدد این خواهد بود که نرم‌افزار جعلی $D_۲$ را جایگزین نرم‌افزار اصلی $D_۱$ کند به‌طوری که $h(D_۱) = h(D_۲)$. با توجه به ویژگی‌های تابع چکیده‌ساز، این کار سخت است. پس دستکاریِ نرم‌افزار در مدت زمان محدود عملی نیست.

فرض کنیم D یک عدد با طول دلخواه باشد. می‌خواهیم با اعمال یک تابع چکیده‌ساز روی آن، رشته‌ای به طول n به‌دست آوریم. برای این منظور، نخست لازم است طول D مضربی از n باشد. پس احتمالاً به تعدادی صفر قبل از رشتهٔ D نیاز خواهیم داشت. با تقسیم D به رشته‌هایی به طول n به‌شکل

$$D = D_۱ \mid D_۲ \mid \cdots \mid D_k.$$

و با فرضِ $H_۰ = \underbrace{(۰, \ldots, ۰)}_{n \text{ بار}}$، $h(D)$ به‌صورت بازگشتی به‌شکل زیر تعریف می‌شود:

$$\begin{aligned} H_۱ &= H_۰ \bigoplus M(D_۱), \\ &\vdots \\ H_i &= H_{i-۱} \bigoplus M(D_i) \end{aligned}$$

که در آن، M یک جایگشت nحرفی است. $h(D) = H_k$ یك تابع چکیده ساز را تعریف می‌کند.

از توابع چکیده‌ساز معروف می‌توان به خانوادهٔ MD شامل MD۲ و MD۴ و MD۵ و همچنین خانوادهٔ SHA شامل SHA-۱ و SHA-۲ و SHA-۳ اشاره کرد. نام MD برگرفته از نام‌های رالف مرکل و ایوان دمگارد[۱] است. الگوریتم‌های چکیده‌ساز MD۲ و MD۴ و MD۵ طراحی و سپس توسط ریوست بهینه شدند. کاربرد این الگوریتم‌ها در امضای رقمی است بدین صورت که یک پیام توسط یک کلید خصوصی قبل از امضا به پیامی کوچک تبدیل می‌شود. همهٔ این الگوریتم‌ها پیامی با هر طول دلخواه را دریافت و آن را به پیامی با طول ثابت ۱۲۸ بیت تبدیل می‌کنند. هرچند ساختار این الگوریتم‌ها تا حدی شبیه به هم هستند، طراحی MD۲ متفاوت از MD۴ و MD۵ است.

الگوریتم چکیده‌ساز SHA-۱ [۲] در سال ۱۹۹۵ توسط مؤسسهٔ ملی استاندارد و فناوری ایالات متحده (NIST) منتشر شد [۱۸]. این الگوریتم پس از دریافت هر پیام، پیام فشردهٔ ۱۶۰ بیتی تولید می‌کند. الگوریتم حمله به SHA-۱ که در سال ۲۰۰۵ طراحی شد، نشان داد که این الگوریتم برای استفادهٔ مداوم، از

[۱] Ivan Damgard [۲] standard secure hash algorithm (revision 1)

امنیت کافی برخوردار نیست. به همین منظور، چند شرکت معتبر که از این الگوریتم استفاده می‌کردند، آن را متوقف و از الگوریتم‌های جایگزین SHA-۲ و SHA-۳ که در سال ۲۰۱۰ معرفی شد، استفاده کردند [۱۹]. تابع چکیده‌ساز SHA-۲ که دارای ساختار SHA-۱ است، با تغییرات گسترده‌ای از آن به‌دست آمد و دارای خانواده‌ای شامل توابع SHA-۲۲۴، SHA-۲۵۶، SHA-۳۸۴ و SHA-۵۱۲ است. با معرفی تابع SHA-۳، خانوادهٔ SHA-۲ از سوی جامعهٔ رمزنگاری کمتر مورد توجه قرار گرفت.

۷. رمزنگاری خم‌های بیضوی

در سال ۱۹۸۵ کوبلیتز و میلر رمزنگاری را روی دسته‌ای از خم‌های جبری به نام خم‌های بیضوی پیاده‌سازی کردند. به‌دلیل سختیِ حل مسئلهٔ لگاریتم گسسته در خم‌های بیضوی، امنیت در رمزنگاری خم‌های بیضوی در مقایسه با سایر سامانه‌های موجود، با طول کلید کمتر حاصل شده و استفاده از آنها در رمزنگاری بسیار مورد توجه قرار گرفته است. در این بخش، ضمن معرفی خم‌های بیضوی، خواهیم دید که چگونه از آنها در رمزنگاری کلید عمومی استفاده می‌شود. به‌علاوه ارتباط ریاضیات خم‌های بیضوی با کاربردهای آن در رمزنگاری بیان می‌شود.

نظریهٔ خم‌های بیضوی یکی از شاخه‌های ریاضیات در زمینهٔ جبر و نظریهٔ اعداد است که امروزه به‌دلیل کاربردهای فراوان آن مورد توجه قرار گرفته است. تنوع و کاراییِ این نظریه بسیار خیره‌کننده است به‌طوری که در رمزنگاری، تجزیهٔ اعداد بزرگ، آزمون‌های اول بودن اعداد صحیح، امضای رقمی، حل معادله‌های سیاله از جمله قضیهٔ آخر فرما و ... جایگاهی ویژه پیدا کرده است. هر خم بیضوی روی یک میدان تعریف می‌شود و دارای ساختار یک گروه آبلی است. دسته‌ای از این خم‌ها که روی میدان‌های متناهی تعریف می‌شوند، گروهی متناهی تعریف می‌کنند که با ایجاد شرایط مناسب، به‌دلیل سختیِ مسئلهٔ لگاریتم گسسته در این دسته از گروه‌ها نقشی ویژه در رمزنگاری و شاخه‌های مرتبط با آن ایفا می‌کنند. با پیشرفت رایانه و قدرت بالای محاسباتی آن، در حال حاضر حملات توانمندی روی الگوریتم‌های رمزنگاری فعلی در حال طراحی است. با وجود خم‌های بیضوی، آینده‌ای بهتر در طراحی سامانه‌های رمزنگاری و مقاوم در برابر این حملات تصور می‌شود.

فرض کنیم K یک میدان باشد. هر معادله به‌صورت

$$y^۲ + a_۱xy + a_۳y = x^۳ + a_۲x^۲ + a_۴x + a_۶ \qquad (۱.۷)$$

با ضرایب $a_۱, a_۲, a_۳, a_۴, a_۶ \in \overline{K}$، یک *معادلهٔ وایرشتراس* نامیده می‌شود. مبین این معادله عبارت است از

$$\Delta = -b_۲^۲b_۸ - ۸b_۴^۳ - ۲۷b_۶^۲ + ۹b_۲b_۴b_۶$$

که در آن،

$$\begin{aligned} b_۲ &= a_۱^۲ + ۴a_۲, \\ b_۴ &= ۲a_۴ + a_۱a_۳, \\ b_۶ &= a_۳^۲ + ۴a_۶, \\ b_۸ &= a_۱^۲a_۶ + ۴a_۲a_۶ - a_۱a_۳a_۴ + a_۲a_۳^۲ - a_۴^۲. \end{aligned}$$

خم با معادلهٔ وایرشتراس (۱.۷) و مبین ناصفر E، یک خم *بیضوی* نامیده می‌شود. بنابراین برای ساختن خم‌های بیضوی، می‌توان ضرایب a_i را در معادلهٔ (۱.۷) به‌طور تصادفی چنان انتخاب کرد که $\Delta \neq ۰$. چنانچه مشخصهٔ میدان K، ۲ و ۳ نباشد، با تعویض متغیر مناسبی، معادلهٔ وایرشتراس (۱.۷) قابل بازنویسی به‌شکل

$$y^۲ = x^۳ + ax + b \qquad (۲.۷)$$

با مبین $\Delta = -۱۶(۴a^۳ + ۲۷b^۲)$ است. با اضافه کردن نقطهٔ خاص $\mathcal{O}$ (به‌نام *نقطه در بینهایت*) به نقاط روی خم، مجموعهٔ

$$E = \{(x, y) \in \overline{K}^۲ \mid y^۲ = x^۳ + ax + b\} \cup \{\mathcal{O}\}$$

یک گروه آبلی خواهد شد. ایدهٔ تبدیل یک خم بیضوی به یک گروه در سال ۱۸۳۵ توسط کارل گوستاو ژاکوبی[۱] ارائه شد. سپس میلر [۱۷] و کوبلیتز [۹] به‌طور جداگانه از این گروه‌ها در رمزنگاری استفاده کردند.

فرض کنیم E یك خم بیضوی با معادلهٔ وایرشتراس (۲.۷) و نقطه در بینهایت $\mathcal{O}$ باشد. همچنین فرض کنیم $P, Q \in E$. خم E و خط $l_{P,Q}$ گذرنده از P و Q (مماس بر P هرگاه $P = Q$)، دقیقاً در سه نقطه مشترک‌اند. اگر R سومین نقطهٔ تلاقی خم E و خط $l_{P,Q}$ و همچنین S سومین نقطهٔ مشترک خم E و خط $l_{R,\mathcal{O}}$ باشد، تعریف می‌کنیم $P + Q = S$. با این عمل، E یك گروه آبلی با عنصر همانی $\mathcal{O}$ است. اگر خم E دارای معادلهٔ وایرشتراس $y^۲ = x^۳ + ax + b$ باشد و $P = (x_P, y_P)$ و $Q = (x_Q, y_Q)$ دو نقطه روی خم باشند و $S = P + Q = (x_S, y_S)$، آن‌گاه

$$\begin{cases} x_S = \lambda^۲ - x_P - x_Q, \\ y_S = \lambda(x_P - x_S) - y_P \end{cases}$$

[۱] Carl Gustav Jacob Jacobi

که در آن،

$$\begin{cases} \lambda = \dfrac{y_Q - y_P}{x_Q - x_P} & P \neq Q \\ \\ \lambda = \dfrac{۳x_P^۲ + a}{۲y_P} & P = Q \end{cases}$$

در رمزنگاری بر اساس خم‌های بیضوی، میدان زمینه یك میدان متناهی است. اگر E یک خم بیضوی روی میدان $\mathbb{F}_q$ باشد، آنگاه $E(\mathbb{F}_q)$ گروهی دوری یا ضرب مستقیم دو گروه دوری است. به عبارت دیگر، $E(\mathbb{F}_q) \cong \mathbb{Z}_{n_۱} \oplus \mathbb{Z}_{n_۲}$ که در آن، $n_۱ \mid \gcd(n_۲, q-۱)$.

در سال ۱۹۸۵ کوبلیتز و میلر استفاده از خم‌های بیضوی را در رمزنگاری پیشنهاد دادند. ساختار گروهی و سختیِ حل مسئلهٔ لگاریتم گسسته در این دسته از خم‌ها، دلیلی برای استفاده از آنها در رمزنگاری است. در رمزنگاری خم‌های بیضوی، میدان زمینه، میدان متناهی $\mathbb{F}_q$ است که در آن، q توانی از عدد اول p است. همچنین لازم است حل مسئلهٔ لگاریتم در این دسته از خم‌ها دشوار باشد. برای این منظور، لازم است $\#E(\mathbb{F}_q)$ مقداری بزرگ باشد. بنابر قضیه‌ای از هلموت هَسه[۱]، لازم است q عددی بزرگ باشد [۲۴]. هَسه نشان داد تعداد نقاط یک خم بیضوی روی یک میدان متناهی تقریباً با مرتبهٔ آن میدان برابر است. او ثابت کرد که اگر E یک خم بیضوی روی میدان متناهی $\mathbb{F}_q$ باشد، آنگاه $|q+۱-\#E(\mathbb{F}_q)| \leq ۲\sqrt{q}$. مقدار $t = q+۱-\#E(\mathbb{F}_q)$ *اثر خم* E نامیده می‌شود. ویلیام واترهوس[۲] نیز نشان داد اگر $|t| \leq ۲\sqrt{q}$ و $p \nmid t$، آنگاه یك خم بیضوی E روی میدان $\mathbb{F}_q$ با اثر t موجود است [۲۶].

۱.۷. مسئلهٔ لگاریتم گسسته در خمهای بیضوی. اگر E یک خم بیضوی روی میدان متناهی $\mathbb{F}_q$ باشد، $P \in E(\mathbb{F}_q)$ و $Q \in \langle P \rangle$، آنگاه عدد صحیح k موجود است که $Q = kP$. کوچکترین عدد صحیح مثبت k با این ویژگی، *لگاریتم گسستهٔ* Q در پایهٔ P نامیده می‌شود. حل مسئلهٔ لگاریتم گسسته در خم بیضوی، یعنی محاسبهٔ k. مسئلهٔ لگاریتم گسسته در خم‌های بیضوی[۳] به‌اختصار با ECDLP نشان داده می‌شود. هرچند الگوریتم‌های با پیچیدگی زمانیِ چندجمله‌ای برای حل مسئلهٔ لگاریتم گسسته در خم‌های بیضوی در بعضی از خم‌های خاص طراحی شده‌اند، تاکنون هیچ الگوریتم زیرنمایی برای ECDLP در حالت کلی وجود ندارد. نایجل اسمارت[۴] در سال ۱۹۹۷ نشان داد ECDLP برای خم‌های بیضوی با اثر یک، دارای الگوریتم با پیچیدگی زمانیِ چندجمله‌ای است [۲۵]. گرهارد فری[۵]، مایکل مولر[۶] و هانس جُرج راک[۷] نیز در سال ۱۹۹۸ با استفاده از زوجیت تیت[۸]، الگوریتمی ارائه کردند که در یک خم بیضوی با اثر ۲، ECDLP در زمان چندجمله‌ای به DLP روی میدان زمینه‌اش تبدیل می‌شود [۶].

[۱] Helmut Hasse [۲] William Waterhouse [۳] elliptic curve discrete logarithm problem [۴] Nigel Smart
[۵] Gerhard Frey [۶] Michael Muller [۷] Hans Georg Ruck [۸] Tate pairing

۲.۷. طرح‌های تبادل کلید با استفاده از خم‌های بیضوی. در یك طرح تبادل کلید، همهٔ اجزا در محاسبهٔ کلید نقش فعال دارند در حالی که در طرح انتقال کلید، یکی از اجزا، کلید را تولید و در کانال امن برای سایر اجزا ارسال می‌کند. در این بخش، نمونه‌هایی از طرح‌های تبادل کلید ارائه می‌گردد.

چنان‌که گفتیم، امنیت پروتکل دیفای-هلمن بر سختیِ مسئلهٔ لگاریتم گسسته پایه‌ریزی شده است و خم‌های بیضوی کاندیدایی مناسب برای اجرای آن است. فرض کنیم آلیس و باب به‌دنبال یک کلید مشترک باشند. آنها خم بیضوی $E(\mathbb{F}_q)$ و نقطهٔ $P \in E(\mathbb{F}_q)$ را انتخاب می‌کنند. این خم باید طوری انتخاب شود که مسئلهٔ لگاریتم گسسته در آن دشوار باشد. آلیس کلید مخفی a و باب کلید مخفی b را انتخاب می‌کنند. در این صورت، abP کلید مشترک آلیس و باب خواهد بود. به‌دلیل سختیِ مسئلهٔ لگاریتم گسسته در این خم، دسترسی به abP با استفاده از P، aP و bP سخت است.

مسئلهٔ پیدا کردن abP با استفاده از P، aP و bP در یک خم بیضوی، مسئلهٔ محاسباتی دیفای-هلمن در خم بیضوی[۱] نامیده و به‌اختصار با ECCDHP نشان داده می‌شود. همچنین این مسئله که با داشتن P، aP، bP و cP، آیا تساوی $cP = abP$ برقرار است، مسئلهٔ تصمیم دیفای-هلمن در خم بیضوی[۲] نامیده و با ECDDHP نشان داده می‌شود. واضح است که ECDDHP نتیجه‌ای از ECCDHP است. حال سؤالی که مطرح می‌شود این است که آیا این دو مسئله معادل‌اند؟ آنچه مشخص است این است که با استفاده از ابزاری به‌نام *زوجیت وایل*[۳]، جواب این سؤال در خم‌های بیضوی، منفی است. برای این منظور، فرض کنیم E یک خم بیضوی روی میدان متناهی $\mathbb{F}_q$، $E[n] = \{P \in E(\overline{\mathbb{F}_q}) \mid nP = \mathcal{O}\}$ مجموعهٔ نقاط nتابی خم E باشد. همچنین فرض کنیم $\mu_n = \{x \in \overline{\mathbb{F}_q}^* \mid x^n = ۱\}$ مجموعهٔ ریشه‌های واحد در میدان $\overline{\mathbb{F}_q}^*$ باشد. در این صورت، نگاشت دوخطی $e_n : E[n] \times E[n] \to \mu_n$ به‌نام زوجیت وایل وجود دارد. حال اگر P نقطه‌ای روی خم بیضوی از مرتبهٔ n باشد و a، b و c اعدادی صحیح باشند، آن‌گاه

$$e_n(aP, bP) = e_n(P, P)^{ab},$$
$$e_n(cP, P) = e_n(P, P)^{c}.$$

با مقایسهٔ طرف دوم روابط بالا، ECDDHP حل می‌شود.

طرح توافق کلید MTI برگرفته از اسامی تسوتومو ماتسوموتو[۴]، یوئیچی تاکاشیما[۵] و هیدکی ایمای[۶] است. در این طرح هر استفاده‌کننده یك جفت کلید دارد. فرض کنیم آلیس و باب می‌خواهند روی کلید مشترکی توافق کنند. آنها روی خم بیضوی E و نقطهٔ $P \in E$ به توافق می‌رسند و E و P را منتشر می‌کنند. آلیس عدد صحیح و مخفی a و باب عدد صحیح و مخفی b را انتخاب می‌کنند. آنها به‌ترتیب، $Q_A = aP$ و $Q_B = bP$ را محاسبه می‌کنند. (a, Q_A) کلید آلیس و (b, Q_B) کلید باب است.

---

[۱] elliptic curve computational Diffie-Hellman problem [۲] elliptic curve decisional Diffie-Hellman problem [۳] Weil pairing [۴] Tsutomu Matsumoto [۵] Youichi Takashima [۶] Hideki Imai

آلیس عدد صحیح و تصادفی k_A را انتخاب و $K_A = k_A Q_A$ را محاسبه و آن را برای باب ارسال می‌کند. باب نیز عدد صحیح و تصادفی k_B را انتخاب و $K_B = k_B Q_B$ را محاسبه و آن را برای آلیس ارسال می‌کند. آلیس با داشتن a، k_A و K_B مقدار

$$a^{-1} k_A K_B = a^{-1} k_A k_B a P = k_A k_B P$$

و باب نیز با داشتن b، k_B و K_A مقدار

$$b^{-1} k_B K_A = a^{-1} k_B k_A b P = k_A k_B P$$

را محاسبه می‌کند. $K = k_A k_B P$ کلید مشترک آلیس و باب است. در [۱۶] و [۱۴] طرح‌هایی مشابه با استفاده از خم‌های بیضوی ارائه شده است.

۳.۷. سامانهٔ رمز الجمال در خم‌های بیضوی. در سامانهٔ رمز الجمال، باب قصد دارد پیامی را برای آلیس ارسال کند. آنها روی خم بیضوی E روی میدان متناهی $\mathbb{F}_q$ به توافق می‌رسند و P را نقطه‌ای از مرتبهٔ n روی E در نظر می‌گیرند و سپس E و P را منتشر می‌کنند. آلیس عدد تصادفی d را انتخاب می‌کند و آن را به‌صورت مخفی نگه می‌دارد. او $Q = dP$ را منتشر می‌کند. باب برای ارسال پیام m برای آلیس، نخست آن را به نقطهٔ $M = (m_x, m_y)$ روی خم E تبدیل می‌کند و سپس عدد تصادفی k را انتخاب می‌کند و زوج $(kP, M + kQ)$ را برای آلیس می‌فرستد. آلیس پس از دریافت kP و $M + kQ$، مقدار $M = (M + kQ) - d(kP)$ را محاسبه می‌کند.

در این سامانه نکتهٔ قابل ذکر این است که هر پیام m قابل تبدیل به نقطهٔ $M = (m_x, m_y)$ روی خم E نیست و این محدودیتی برای پیام‌ها است. همچنین با داشتن هر کدام از دو مقدار m_x و m_y، دیگری و درنتیجه نقطهٔ $M = (m_x, m_y)$ قابل محاسبه است. بنابراین پیام m به‌راحتی به‌دست می‌آید. برای حل این مشکل می‌توان به‌جای خم بیضوی، از میدان متناهی استفاده کرد. در این صورت، نیازی به تبدیل پیام به نقطه روی خم نیست اما امنیت سامانهٔ جدید به نسبت قبلی کمتر است و مشکلی بزرگتر پدید می‌آید.

۴.۷. سامانهٔ رمز خم بیضوی مَسی-اُمورا. سامانهٔ رمز مَسی-اُمورا[1] را می‌توان نمونهٔ دیگری از تبادل کلید دیفای-هلمن در نظر گرفت. در این سامانه نیز پیام به نقطه‌ای روی خم تبدیل می‌شود. فرض کنیم باب قصد دارد پیامی را برای آلیس ارسال کند. آلیس و باب روی خم بیضوی E تعریف‌شده روی میدان متناهی $\mathbb{F}_q$ به توافق می‌رسند. خم E طوری انتخاب می‌شود که مسئلهٔ لگاریتم گسسته در آن سخت باشد. آلیس و باب E و $N = \#E(\mathbb{F}_q)$ را منتشر می‌کنند. آلیس عدد صحیح e_A را چنان انتخاب

[1] Massey-Omura

می‌کند که $(e_A, N) = ۱$ و وارون حسابی e_A در پیمانهٔ N را محاسبه می‌کند و آن را d_A می‌نامد. در واقع $e_A d_A \equiv ۱ \pmod N$ و بنابراین عدد صحیح k_A چنان موجود است که

$$e_A d_A = ۱ + k_A N.$$

باب نیز e_B را متباین با N انتخاب می‌کند و وارون حسابی آن را d_B می‌نامد. به‌طور مشابه عدد صحیح k_B چنان موجود است که

$$e_B d_B = ۱ + k_B N.$$

(e_A, d_A) کلید مخفی آلیس و (e_B, d_B) کلید مخفی باب است. فرض کنیم پیام m به نقطهٔ M روی خم تبدیل شده است. برای ارسال پیام m از سوی باب برای آلیس، مراحل زیر به‌ترتیب انجام می‌گیرد:

(۱) آلیس $e_A M$ را محاسبه می‌کند و آن را برای باب می‌فرستد؛
(۲) باب $e_B(e_A M)$ را محاسبه و آن را برای آلیس ارسال می‌کند؛
(۳) آلیس $d_A(e_B(e_A M))) = e_B M$ را محاسبه و آن را برای باب ارسال می‌کند؛
(۴) باب $d_B(e_B M) = M$ را محاسبه می‌کند و به نقطهٔ M و در نتیجه به پیام m دسترسی پیدا می‌کند.

در این سامانه، آلیس و باب هیچ‌کدام از کلید دیگری اطلاعی ندارد و لازم نیست آنها کلید مخفی مشترکی داشته باشند. بنابراین این سامانه نیازی به کانال امن ندارد. اکنون اگر شخصی در کانال ناامن بخواهد به پیام M دسترسی پیدا کند، باید بتواند یك مسئلهٔ لگاریتم گسسته را حل کند که بنابر انتخابمان از خم E، دشوار است. بنابراین امنیت این سامانه مبتنی بر سختیِ حل مسئلهٔ لگاریتم گسسته است [۱۵].

۵.۷. امضای رقمی. در این بخش، به معرفی چند طرح امضا مبتنی بر خم‌های بیضوی می‌پردازیم.

۱.۵.۷. *طرح‌های امضای رقمی با استفاده از خم‌های بیضوی.* فرض می‌کنیم E یك خم بیضوی روی میدان $\mathbb{F}_q$ است که در آن، q عدد اول بزرگتر از ۳ یا به‌صورت توانی از ۲ است. همچنین فرض می‌کنیم $P \in \mathbb{F}_q$ نقطه‌ای از مرتبهٔ عددی اول است. به‌علاوه فرض می‌کنیم H یك تابع چکیده‌ساز است. برای امضای پیام M مراحل زیر انجام می‌گیرد:

(۱) امضا کنندهٔ پیام عدد صحیح و تصادفی $d \in [۱, n-۱]$ را انتخاب و سپس $Q = dP$ را محاسبه می‌کند. d کلید خصوصی باب است (تولید کلید)؛
(۲) امضا کننده عدد صحیح و تصادفی $k \in [۱, n-۱]$ را انتخاب و $kP = (x_۱, y_۱)$ را محاسبه می‌کند. او بررسی می‌کند که آیا $r \equiv x_۱ \not\equiv ۰ \pmod n$ برقرار است یا نه. اگرنه مقدار دیگری برای k در نظر می‌گیرد. او همچنین $e = H(M)$ را محاسبه و بررسی می‌کند

که آیا $s = r^{-1}(e + dr) \not\equiv 0 \pmod{n}$ برقرار است یا نه. اگرنه مقدار k را تغییر می‌دهد تا هدف اخیر نیز حاصل شود. (r, s) امضای پیام M است (تولید امضا)؛

(۳) تأییدکنندهٔ امضا مقادیر $e = H(M)$،

$$u_1 = es^{-1} \pmod{n} \quad \text{و} \quad u_2 = rs^{-1} \pmod{n}$$

را در میدان و مقدار $u_1P + u_2Q = (x_2, y_2)$ را در خم E محاسبه می‌کند. چنانچه $x_2 \equiv r \pmod{n}$، امضا مورد تأیید قرار می‌گیرد (تأیید امضا).

امنیت این سامانه از جمله در [۱] و [۵] بررسی شده است.

۲.۵.۷. *طرح نایبرگ-روپل*. مراحل امضای نایبرگ-روپل[1] به‌صورت زیر است [۲۰]:

(۱) امضاکننده عدد صحیح و تصادفی $k \in [1, n-1]$ را انتخاب و سپس نقطهٔ

$$R = kP = (x_1, y_1)$$

را محاسبه می‌کند. او $e = x_1 + M$ را محاسبه و بررسی می‌کند که آیا $e \not\equiv 0 \pmod{n}$ یا نه. اگرنه مقدار k را تغییر می‌دهد تا مطلوب حاصل شود. امضاکننده سپس

$$s = k - de \pmod{n}$$

را محاسبه می‌کند. (e, s) امضا برای پیام M است؛

(۲) در این طرح چنانچه پیام به‌درستی دریافت شود، امضا مورد قبول قرار می‌گیرد. در واقع با بررسی پیام، قبل و بعد از افزونگی این نکته قابل دسترسی است؛

(۳) دریافت‌کنندهٔ پیام، نقطهٔ $sP + eQ = (z, t)$ را محاسبه می‌کند. پیام M بر اساس رابطهٔ $M = (e - z) \pmod{n}$ رمزگشایی می‌شود.

۳.۵.۷. *طرح‌های امضا-رمز*. طرح‌های امضا-رمز نخست در کارهای یولیانگ ژنگ[2] [۲۷] و ایمای [۷] معرفی شد. امضا-رمز یك روش رمزنگاری است که توابع رمز کردن و امضا همزمان و با هزینهٔ کمتر نسبت به امضا و سپس رمز کردن انجام می‌گیرد. نمونه‌ای از امضا-رمز: امضا کننده عدد صحیح و تصادفی $k \in [1, n-1]$ را انتخاب و سپس $kP = (x_1, y_1)$، $r = H(M, kP)$ و $s = k(r + D)^{-1}$ را محاسبه می‌کند. زوج (r, s) امضا است. دریافت‌کننده $K = s(rP + Q)$ را با استفاده از کلید عمومی Q محاسبه و سپس بررسی می‌کند که $H(M, K) = r$.

[1] Nyberg-Rueppel [2] Yuliang Zheng

۸. سامانه‌های رمزنگاری RSA با استفاده از خم‌های بیضوی

گفتیم که ابزار اصلی در سامانهٔ رمز RSA اعداد اول بسیار بزرگ و امنیت این سامانه مبتنی بر سختیِ تجزیه است. پس از پیاده‌سازی سامانه‌های رمز از جمله تبادل کلید دیفای-هلمن، سامانهٔ رمز الجمال و ... روی خم‌های بیضوی، تلاش بر این بوده است که سامانهٔ رمز RSA نیز با استفاده از این خم‌ها پیاده‌سازی شود. در نمونه‌ای از این تلاش‌ها، برای دو عدد اول (بزرگ) p و q، خم بیضوی روی حلقهٔ $\mathbb{Z}_n$ تعریف شده است که در آن، $n = pq$. در حالت کلی یك خم بیضوی روی حلقهٔ $\mathbb{Z}_n$ دارای معادلهٔ

$$E_{a,b} : y^2 = x^3 + ax + b$$

است که در آن، $a, b \in \mathbb{Z}_n$ و به‌علاوه $(4a^3 + 27b^2, n) = 1$. چون عناصر $\mathbb{Z}_n$ در حالت کلی وارون‌پذیر نیستند، مجموعهٔ $E_{a,b}(\mathbb{Z}_n)$ یك گروه نیست و قانون جمع نقاط خم بیضوی روی یك میدان را نمی‌توان به جمع نقاط خم بیضوی روی حلقهٔ $\mathbb{Z}_n$ گسترش داد. با محاسبهٔ همهٔ پارامترها در پیمانهٔ عامل اول p از n، خم $E_{a,b}(\mathbb{Z}_n)$ به خم بیضوی $E_{\overline{a},\overline{b}}(\mathbb{F}_p)$ تبدیل می‌شود. چنانچه $\overline{a}_p$ تبدیل‌یافتهٔ a در پیمانهٔ p باشد، نقطهٔ $P = (x, y) \in E(\mathbb{Z}_n)$ به $\overline{P}_p = (\overline{x}_p, \overline{y}_p) \in E(\mathbb{F}_p)$ و نقطه در بی‌نهایت $\mathcal{O}$ روی خم $E(\mathbb{Z}_n)$ به نقطهٔ $\mathcal{O}_p \in E(\mathbb{F}_p)$ تبدیل می‌شود. اگر p و q دو عدد اول متمایز باشند و $n = pq$، آن‌گاه با استفاده از قضیهٔ باقیماندهٔ چینی، هر $x \in \mathbb{Z}_n$ به‌طور یکتا به‌صورت $\overline{x}_p \in \mathbb{F}_p$ و $\overline{x}_q \in \mathbb{F}_q$ قابل نمایش است. با فرضِ

$$\begin{aligned}\tilde{E}(\mathbb{Z}_n) &= E(\mathbb{F}_p) \times E(\mathbb{F}_q) \\ &= \{(P_p, P_q) \ : \ P_p = (\overline{x}_p, \overline{y}_p) \in E(\mathbb{F}_p),\ P_q = (\overline{x}_q, \overline{y}_q) \in E(\mathbb{F}_q)\}\end{aligned}$$

و $\mathcal{O}_n = (\mathcal{O}_p, \mathcal{O}_q)$، خواهیم داشت $\#\tilde{E}(\mathbb{Z}_n) = \#E(\mathbb{F}_p) \times \#E(\mathbb{F}_q)$. اکنون سامانهٔ رمز RSA روی $\tilde{E}(\mathbb{Z}_n)$ به گونه‌ای پیاده‌سازی می‌شود که امنیت آن مبتنی بر تجزیهٔ n به عوامل اول آن، یعنی p و q باشد. برای مطالعهٔ جزئیات این طرح و سایر نمونه‌هایی که از خم‌های بیضوی استفاده شده است، به [۲، ۸، ۱۱، ۱۲، ۱۳] رجوع کنید.

۹. فایدهٔ خم‌های بیضوی در رمزنگاری

رمزنگاری خم‌های بیضوی در سامانه‌های تبادل کلید دیفای-هلمن، رمز و امضای الجمال، رمز مَسی-اُمورا و ... قابل پیاده‌سازی است. این سامانه‌ها در صورتی امن خواهند بود که مسئلهٔ لگاریتم گسسته در آنها دشوار باشد. بنابراین در خم بیضوی $E(\mathbb{F}_p)$ با توجه به قضیهٔ هَسه، لازم است p عدد اول بزرگی باشد. در حال حاضر سریع‌ترین الگوریتم ECDLP ، الگوریتم پهلیگ-لمن است که دارای پیچیدگی زمانیِ

نمایی است. هرچند برای خم‌های خاص، الگوریتم‌های با پیچیدگی زمانیِ چندجمله‌ای وجود دارد که به آنها اشاره شد. در سامانه‌های رمز خم بیضوی برای پیشگیری از حملهٔ الگوریتم پهلیگ–هلمن، خم‌هایی مورد استفاده قرار می‌گیرند که حداقل $۲^{۱۶۰}$ نقطه داشته باشند. پیش‌بینی می‌شود سامانه‌های رمز مبتنی بر خم‌های بیضوی، جایگزین سامانه‌هایی همچون RSA و گروه ضربی میدان‌های متناهی شوند. از دلایل آن می‌توان به کارایی و امنیت سامانهٔ رمز خم‌های بیضوی در مقایسه با RSA و گروه ضربی میدان‌های متناهی اشاره کرد. یك سامانهٔ رمز RSA با طول کلید ۱۰۲۴ دارای امنیتی است که یك سامانهٔ رمز خم بیضوی با طول کلید ۱۶۰ ایجاد می‌کند.

مراجع

[1] Brown, D. R. L., *The exact security of ECDSA*, preprint, 2000.

[2] Demytko, N., A new elliptic curve based analogue of RSA, In: Tor Helleseth (ed.), *Advances in Cryptology-Eurocrypt*, **93**, Lofthus, Norway, Springer-Verlag, 40–49, 1994.

[3] Diffie, W., Hellman, M., New directions in cryptography, *IEEE Transactions on Information Theory*, **22** (1976), no.6, 644–654.

[4] ElGamal, T., A public-key cryptosystem and a signature scheme based on the discrete logarithm, *IEEE Transactions of Information Theory*, **31** (1985), no. 4, 469–472.

[5] El Mahassni, E., Nguyen, P. Q., Shparlinski, I. E., The insecuriry of Nyberg-Rueppel and other DSA-like signature schemes with partially known nonces, *Workshop on Lattices and Cryptography*, Boston, MA., 2001.

[6] Frey, G., Muller, M., Ruck, G. H., The Tate pairing and discrete logarithm applied to elliptic curve cryptosystems, *IEEE Trans. Inform. Theory*, **45** (1998), 1717–1719.

[7] Imai, H., Zheng, Y., *Efficient signcryption schemes on elliptic curves*, IFIP/SEC 98, the 14th Interantional Information Security conference, Vienna and Budapest, 1998.

[8] Joye, M., Quisquater, J. J., Takagi, T., *How to choose secret parameters for RSA-type cryptosystems over elliptic curves*, Technical Report TI-35/97, Technische Universitat Darmstadt, 1997.

[9] Koblitz, N., Elliptic curve cryptosystems, *Mathematics of Computation*, **48** (1987), 203–209.

[10] Koblitz, N., Hyperelliptic cryptosystems, *Journal of Cryptology*, **1** (1989), 139–150.

[11] Koyama, K., Kuwakado, K., Security of RSA-type cryptosystems over elliptic curves against the Hastad attack, *Electronics Letters*, **30** (1994), no.22, 1834–1844.

[12] Koyama, K., Fast RSA-type schemes based on singular cubic curves $y^۲+axy = x^۳ \pmod n$, In: Saint-Malo, France, Louis C. Guillou & Jean-Jacques Quisquater (eds.), *Advances in Cryptology-Eurocrypt* **95**, Springer-Verlag, 329–340, 1995.

[13] Kurosawa, K., Okada, K., Tsujii, S., Low exponent attack against elliptic curve RSA, LNCS 917, *Advances in Cryptology-Asiacrypt,* **94** (1995), 376–383.

[14] Law, L., Menezes, A. J., Qu, M., Solinas, J., Vanstone, S. A., An efficient protocol for authenticated key agreement, *Technical Report CORR 98–05*, University of Waterloo, Ontario, Canada, March, 1998.

[15] Massey, J. L., Omura, J. K., Method and apparatus for maintaining the privacy of digital messages conveyed by public transmission, *U.S. Patent*, **4** (1986), 567–600.

[16] Menezes, A. J., Qu, M., Vanstone, S. A., Some new key agreement protocols providing mutual implicit authentication, *Selected Areas in Cryptology-SAC.*, **95**, 22–32.

[17] Miller, V. S., Uses of elliptic curves in cryptography, In: Hugh C. Williams (ed.), *Advances in Cryptology-CRYPTO*, 85 (218), Lecture Notes in Computer Science, Berlin, 417–426, 1986.

[18] National Institute of Standards and Technology (NIST), *Secure hash standard. Federal Information Processing Standard*, FIPS-180-1, 1995.

[19] National Institute of Standards and Technology (NIST), *Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family*. Federal Register, 27(212): 62212–62220, 2007.

[20] Nyberg, K., Rueppel, R. A., Message recovery for signature schemes based on discrete logarithm problem, *Designs, Codes and Cryptography*, **7** (1996), 61–81.

[21] Rabin, M. O., *Digitalized Signatures and Public Key Functions as Intractable as Factorisation*, Massachusetts Institute of Technology, 1979.

[22] Rivest, R. L., Shamir, A., Adleman, L., A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM.*, **21** (1978), no.2, 120–126.

[23] Rubin, K., Silverberg, A., Torus-based cryptography, In: Dan Boneh (ed.), *Advances in Cryptology-CRYPTO*, 3(2729) of Lecture Notes in Computer Science, Springer-Verlag, 349–365, 2003.

[24] Silverman, J. H., *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics **106**, Springer-Verlag, New York, 1986.

[25] Smart, N., The discrete logarithm problem on elliptic curves of trace one, *HP-LABS Technical Report* (Number HPL-97-128), preprint, 1997.

[26] Waterhouse, E., Abelian varieties over finite fields, *Ann. Sci., Ecole Normale Superieure*, **2** (1969), 521–560.

[27] Zheng, Y., Shortened digital signature, signcryption and compact and unforgeable key agreement schemes, submitted to *IEEE P1363a-Standard Specifications for Public-Key Cryptography: Additional techniques*, 1998.

تاریخ ارسال: ۹۶/۳/۲؛ تاریخ بازنگری: ۹۶/۶/۱۳؛ تاریخ پذیرش: ۹۶/۶/۱۹

مجتبی بهرامیان: دانشگاه کاشان، دانشکدهٔ علوم ریاضی
تارنما: http://bahramian.kashanu.ac.ir/
رایانامه: bahramianh@kashanu.ac.ir