
بسم الله الرحمن الرحیم

دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش گراف و ترکیبیات

پایان‌نامه کارشناسی ارشد

بررسی برخی امضاهای رقمی

نگارنده: غزاله تقی‌زاده

استادان راهنما

دکتر میثم علیشاهی
دکتر فرخ‌لقا معظمی

دی ۱۳۹۵

سپاس بی کران، بر هم دلی، هم راهی و هم گامی خانواده ی عزیزم
که تکیه گاهی محکم در سختی های راه من بودند....
و به شکرانه ی این نعمت، سر تعظیم بر آستان پر عظمت الهی فرود
آورده و سجده ی شکر به جا می آورم که پشتوانه ای این چنین دارم.
تقدیم به خانواده ی خوبم...

سپاس‌گزاری...

سپاس خداوند بزرگی که نعمت بندگی‌اش را بر ما عرضه کرد...
الهی!
مرا بر آگاهی فرو مگذار، که آگاهی همه شغل است و درِ دانش بِمَبَند، که دانش همه در اوست و تا رهی بخود است جوی خشک و آهن سرد است و او که از زهد به ثنا رازست، محجوب است و نیم درم در کنف صوفی کنز است.
به مصداق «من لم یشکر المخلوق لم یشکر الخالق » بسی شایسته است از استادان فرهیخته و فرزانه جناب آقای دکتر علیشاهی و خانم دکتر معظمی که با کرامتی چون خورشید ، سرزمین دل را روشنی بخشیدند و گلشن سرای علم و دانش را با راهنمایی‌های کار ساز و سازنده بارور ساختند؛ تقدیر و تشکر نمایم.

غزاله تقی‌زاده
دی ۱۳۹۵

تعهد نامه

اینجانب **غزاله تقی‌زاده** دانشجوی کارشناسی ارشد رشته **ریاضی کاربردی علوم ریاضی** دانشگاه شاهرود، نویسنده پایان‌نامه با عنوان **بررسی برخی امضاهای رقمی** ، تحت راهنمایی **میثم علیشاهی** و **فرخ‌لقا معظمی** متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام‘‘ دانشگاه صنعتی شاهرود‘‘ یا ‘‘ Shahrood University of Technology ‘‘ به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

غزاله تقی‌زاده
دی ۱۳۹۵

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

چکیده

هدف ما در این پایان‌نامه بررسی امضاهای رقمی است. امضاهای رقمی یکی از اولیه‌های رمزنگاری هستند که سبب احراز اصالت شخص فرستنده‌ی پیام، برای شخص گیرنده می‌شوند. امضاهای گروهی، امضاهایی هستند که به یک شخص از اعضای گروه اجازه می‌دهد که پیامی را از طرف کل گروه امضا کند. اعضای گروه توسط یک رئیس/مدیر گروه مدیریت می‌شوند. امضاهای حلقوی، امضاهایی هستند که اعضای حلقه، به‌جز امضا کننده، از عضویت خود در حلقه آگاهی ندارند. شخص امضا کننده، با استفاده از کلیدعمومی اعضا، پیام خود را امضا می‌کند، به‌گونه‌ای که هویت او برای شخص گیرنده، گمنام باقی بماند، اما گیرنده مطمئن می‌شود که او یکی از اعضای حلقه است. با توجه به ظهور کامپیوترهای کوانتومی، سامانه‌های رمزنگاری موجود در معرض تهدید قرار گرفتند، ازاین‌رو عزم جوامع بین‌المللی رمزنگاری، برای دست‌یابی به سامانه‌های رمزنگاری مقاوم در برابر حملات کوانتومی، به‌کار گرفته‌شد. رمزنگاری مشبکه‌مبنا، نوعی رمزنگاری کلیدعمومی مقاوم در برابر حملات کوانتومی(پساکوانتومی)است که در سال‌های اخیر بسیار مورد توجه قرار گرفته‌است. از این‌رو در بخشی از این پایان‌نامه به بررسی یک امضای گروهی مشبکه‌مبنا پرداخته‌ایم.

کلمات کلیدی: رمزنگاری کلیدعمومی، امضای گروهی، گمنامی، ردیابی، امضای حلقوی، جعل، پاسخگوی تصادفی

فهرست مطالب

فهرست تصاویر

فصل ۱

مقدمه

۱.۱ رمزنگاری کلیدعمومی

هدف از رمزنگاری تامین امنیت اطلاعات با استفاده از شیوه‌های ریاضی است که به اهداف کلیدی زیر تقسیم می‌شود:

۱. **محرمانگی:** به معنی حفظ داده‌ها از دسترسی افراد غیرمجاز به آن است.

۲. **یکپارچگی:** به معنی قابلیت شناسایی هرگونه دست‌کاری داده‌ها توسط هستارهای[۱] غیرمجاز است.

۳. **دسترس‌پذیری:** به این معنی است که افراد مجاز امکان دسترسی به داده‌ها را در زمان مناسب داشته باشند.

۴. **احراز اصالت:** به معنی وارسی اصالت و اعتبار هستاری هویت یک فرد، منشا پیام یا یک برنامه‌ی رایانه‌ای است.

علم رمزنگاری به حوزه‌های رمزنگاری متقارن و رمزنگاری نامتقارن تقسیم می شود. معمولا هستارهایی که از روش‌های رمزنگاری متقارن جهت ارتباط امن استفاده می‌کنند باید یک کلید مخفی به اشتراک بگذارند. امنیت ارتباطات به این کلید وابسته است و مادامی که نیاز به ارتباط امن باشد، این کلید باید دور از دسترس دیگر هستارها محفوظ نگاه داشته شود.

[۱] Entity. هر دستگاه سامانه یا واحد عملکردی مستقل

یک مساله‌ی مهم در رمزنگاری کلید متقارن این است که پیش از شروع ارتباط کلیدهای مخفی باید بین هستارهای مربوطه توزیع شوند. همچنین در رمزنگاری کلید متقارن امضای رقمی بدون انکار میسر نیست. در سال ۱۹۷۶ دیفی [۲] و هلمن [۳] با فراهم کردن راه‌حلی برای این مسئله تحت عنوان رمزنگاری نامتقارن یا رمزنگاری کلید عمومی،انقلابی در رمزنگاری ایجاد کردند[25].
در محیط رمزنگاری کلید عمومی، هر هستار دو کلید مجزا دارد. یک کلید عمومی و یک کلید خصوصی که می‌تواند آنها را به صورت محلی تولید کند. کلید خصوصی باید مخفی نگه داشته‌شود اما کلید عمومی می‌تواند به صورت گسترده توزیع شود بدون آنکه محرمانگی کلید خصوصی را به خطر اندازد. لذا در این حالت مسئله تبادل امن کلیدهای مخفی به مسئله توزیع کلیدهای عمومی معتبر تغییر پیدا می‌کند.
به طور کلی سیستم های رمزنگاری کلید عمومی شامل سه الگوریتم به شرح زیر هستند[47]:

۱. الگوریتم تصادفی تولید کلید(Key-Gen): به عنوان ورودی پارامتر امنیتی k را گرفته و خروجی، کلید عمومی pkو کلید خصوصی skرا برمی‌گرداند.

۲. الگوریتم تصادفی رمزگذاری (Enc): ورودی پیام m و کلید عمومی را دریافت می‌کند و c را به عنوان خروجی می‌دهد.

۳. الگوریتم قطعی رمزگشایی (Dec): شخصی که قصد بازگشایی رمز را دارد، متن رمز شده‌ی c را به عنوان ورودی می‌گیرد و با استفاده از کلید خصوصی sk، متن m را به‌دست می‌آورد.

دو مفهوم اساسی در رابطه با امنیت سیستم‌های رمزنگاری کلیدعمومی وجود دارد:
امنیت IND-CPA: تمایزناپذیری در مقابل حمله‌ی متن انتخابی [۴]. از دو پیامی که توسط مهاجم انتخاب شده، یکی از آنها رمز می‌شود و به مهاجم داده می‌شود. امنیت زمانی بر قرار است که مهاجم نمی‌تواند تشخیص دهد که کدام متن اصلی، متناظر با متن رمز شده است. در این نوع امنیت، مهاجم به پاسخگو دسترسی ندارد.
امنیت IND-CCA: تمایزناپذیری در مقابل حمله‌ی متن رمز شده‌ی انتخابی [۵]. این نوع امنیت، بسیار قوی‌تر از نوع قبل است. مهاجم هنوز نمی‌تواند تشخیص دهد متن رمزشده، متناظر با کدام یک از دو متن اصلی است که او خود به ما داده است، در حالی‌که قبل و بعد از چالش، به یک پاسخگوی تصادفی [۶] دسترسی دارد که هر متن رمزشده‌ی دلخواه مهاجم را(به غیر از متن مورد چالش)، بازگشایی می‌کند.

۲.۱ رمزنگاری پساکوانتومی

ظهور کامپیوترهای کوانتومی در آینده نزدیک و تهدید سامانه‌های رمزنگاری موجود توسط آنها، عزم جامعه‌ی بین‌المللی رمزنگاری را برای جایگزین کردن سامانه‌های رمزنگاری موجود با سامانه‌های رمزنگاری

[۲] Diffie
[۳] Hellman
[۴] INDistinguishability against Choosen Plaintext Attcak
[۵] INDistinguishability against Choosen Chiphertext Attack.
[۶] Random Oracle

مقاوم در برابر حملات کوانتومی به کار گرفته است. با معرفی الگوریتم کوانتومی شور[۷] در سال ۱۹۹۴ برای تجزیه اعداد صحیح و حل مسئله لگاریتم گسسته، امنیت ساختارهای رمزنگاری کلید عمومی مبتنی بر نظریه اعداد از جمله RSA و الجمال در معرض تهدید قرارگرفت[46]. از آن زمان به بعد تلاش‌های زیادی در زمینه‌ی رمزنگاری پساکوانتومی برای تهیه‌ی سامانه‌های رمز مقاوم در برابر کامپیوترهای کوانتومی انجام شده است.

۳.۱ طرح های امضای رقمی

یکی از مهمترین اولیه های رمزنگاشتی که در حوزه‌ی رمزنگاری کلید عمومی جای دارد، طرح امضای رقمی است که در تامین خدمات رمزنگاشتی از قبیل احراز اصالت هستار، بنیادی هستند. امضای رقمی نوعی رمزنگاری نامتقارن است. هنگامی که پیامی از کانالی ناامن ارسال می‌شود، امضای رقمی‌ای که به شکل صحیح به انجام رسیده‌باشد می‌تواند دلیلی باشد تا شخص فرستنده برای شخص گیرنده احراز اصالت شود.

امروزه امضاها نقش مهمی را در روابط اجتماعی ایفا می‌کنند، به طوری که اعتبار بسیاری از مدارک اداری، تجاری یا حقوقی با امضای آنها تحقق می‌یابد. با پیشرفت ارتباطات و مخابرات، شبکه‌های کامپیوتری پدید آمدند و روش‌های ارسال و ذخیره کاغذی اطلاعات، جای خود را به روش‌های رقمی و کامپیوتری دادند. این امضاها به عنوان جایگزینی برای امضای دست نوشته است. این روش مبتنی بر رمزنگاری باعث به رسمیت شناختن اطلاعات الکترونیکی شده است، به‌طوری‌که هویت پدید آورنده سند و جامعیت اطلاعات آن، وارسی‌پذیر است.

امضاهای رقمی همزمان با سیستم‌های رمزنگاری کلیدعمومی به عنوان ابزاری برای تشخیص اینکه پیام داده شده، حقیقتا توسط فردی که ادعا می‌کند آن را نوشته است، نگاشته شده، معرفی شد [55, 53]

طرح‌های امضای رقمی با الگوریتم‌های زیر تعریف می‌شوند:

۱. الگوریتم تصادفی تولید کلید (Key-Gen): به منظور تولید کلید عمومی pk و کلید خصوصی sk برای هر عضو، این الگوریتم اجرا می‌شود. کلیدهای عمومی، بعدا در لیستی تحت عنوان لیست کلید عمومی، پخش خواهند شد.

۲. الگوریتم تصادفی تولید امضا (Sign): این الگوریتم به ازای ورودی پیام m و کلید خصوصی sk امضای σ را تولید می‌کند.

۳. الگوریتم قطعی وارسی امضا (Verify): این الگوریتم،به ازای ورودی پیام m کلید عمومی pk، و امضای σ اقدام به وارسی امضا می‌کند. اگر خروجی ۱ بود، امضا متعلق به شخص امضاکننده است، اگر ۰ بود، امضا جعلی بوده‌است(و یا به‌طور خوش‌بینانه می‌توان گفت امضا ناقص می‌باشد).

امنیت در مورد امضاهای دیجیتال توسط میکالی [۸]، رایوست [۹] و گلدویزر [۱۰] معرفی شد [34]. امضا

[۷] Shor
[۸] Micali
[۹] Rivest
[۱۰] Goldwasser

غیر قابل جعل[۱۱] است به این معنا است که کسی نمی‌تواند امضای معتبری را از طرف شخصی دیگر جعل کند. امنیت قوی‌تر که توسط گلدویزر معرفی شد، EUF-CMA است، که معادل با عبارت ”جعل‌نشدنی در برابر حمله‌ی متن انتخابی“ [۱۲] است. به این معنا که، حتی بعد از دیدن چندین امضای مختلف، روی هر پیام دلخواه مهاجم، باز هم مهاجم نمی‌تواند امضای معتبری جعل کند.

۱.۳.۱ امنیت قابل اثبات

توابع چکیده‌ساز [۱۳]در رمزنگاری کاربرد بسیاری دارد. این توابع به ازای ورودی با طول بزرگ، خروجی با طول ثابت تولید می کنند که از آن به عنوان چکیده پیام یاد می‌شود. در رمزنگاری، پاسخگوی تصادفی پاسخگویی است که به هر پرسش، با پاسخ تصادفی صحیح که به طور یکنواخت از دامنه خروجی آن انتخاب شده‌است جواب می‌دهد، مگر در حالتی که یک سوال خاص، مجددا پرسیده‌شود که در این حالت همان پاسخ قبلی ارائه می‌شود. پاسخگوهای تصادفی به عنوان ابزار ریاضی در اثبات‌های رمزنگاری به کار گرفته‌می‌شوند. از آنها معمولا زمانی استفاده می‌شود که هیچ تابع قابل پیاده‌سازی شناخته شده‌ای قادر نباشد ویژگی‌های ریاضی مورد نیاز اثبات را تامین کند. سامانه‌ای که دارای امنیت قابل اثبات تحت این شرایط باشد، به عنوان سامانه‌ی امن در مدل پاسخگوی تصادفی [۱۴]توصیف می‌شود که در مقابل سامانه‌ی امن در مدل استاندارد قرار دارد. مدل استاندارد در رمزنگاری مدلی محاسباتی است که در آن مهاجم فقط از نظر مقدار زمان و توان محاسباتی در دسترس، محدود می‌شود. طرح‌هایی که دارای امنیت قابل اثبات صرفا با فرض‌های پیچیدگی هستند، طرح‌های امن در مدل استاندارد نامیده می‌شود. اثبات‌های امنیتی در مدل استاندارد به نحو بارزی مشکل هستند، البته همه موارد استفاده از توابع چکیده‌ساز در طرح‌های رمزنگاری نیازمند استفاده از مدل پاسخگوی تصادفی نیست. طرح‌هایی که نیازمند فقط برخی ویژگی‌ها مثل یک‌طرفه بودن و مقاومت در برابر تصادم(برخورد) [۱۵] هستند، دارای تعریف در مدل استاندارد بوده و معمولا دارای امنیت قابل اثبات در مدل استاندارد هستند.

طرح‌های با امنیت قابل اثبات در مدل استاندارد دارای کارایی کمتری، نسبت به طرح‌های متناظر با امنیت قابل اثبات در مدل پاسخگوی تصادفی هستند. با این وجود، تا کنون هیچ تابع حقیقی قادر به پیاده‌سازی پاسخگوی تصادفی صحیح نبوده‌است. یک تفاوت کلیدی بین حالت مدل پاسخگوی تصادفی و حالت استفاده تصادفی از خروجی تابع چکیده‌ساز به وسیله‌ی تابع تصادفی این است که در حالت دوم مهاجم، به توصیفی از ماشین تورینگی [۱۶] که تابع چکیده‌ساز را محاسبه می‌کند، دسترسی دارد اما در حالت اول چنین نیست و گویی با جعبه‌ی سیاه روبروست. همین امر سبب شد تا کنتی[۱۷] و همکاران

---

[۱۱] Unforgeable

[۱۲] Unforgeability against Choosen Message Attack

[۱۳] Hash Function

[۱۴] Random Oracle Model

[۱۵] Collision Resistance. تابع چکیده‌ساز H را مقاوم در برابر تصادم گوییم، اگر یافتن a و b که $H(a) = H(b)$ اما $a \neq b$ سخت باشد.

[۱۶] turing machine. ماشین محاسباتی فرضی با تعداد حالات محدود که دارای توان خواندن و نوشتن نامحدود است.

[۱۷] Canetti

در سال ۲۰۰۴ اثبات کنند که ممکن است طرحی در مدل پاسخگوی تصادفی دارای امنیت قابل اثبات باشد ولی در عمل که پاسخگوی تصادفی با هر تابع چکیده‌سازی جایگزین شود،طرح ناامن شود.

۲.۳.۱ امضای رقمی مشبکه مبنا

یک مسئله مهم در زمینه مشبکه‌ها، ارائه طرح‌های امضای رقمی مشبکه‌مبنایی است که امنیت آنها مبتنی بر مسائل سخت مشبکه باشد.

در این رابطه، دسته‌ای از امضاها با رویکرد کاربردی و بدون اثبات امنیتی ارائه شدند. طرح امضای GGH به عنوان اولین طرح امضای رقمی مشبکه‌مبنا در سال ۱۹۹۷ ارائه‌شد که فاقد اثبات امنیتی بوده و در سال ۲۰۰۶ شکسته شد[33]. امضای NTRUSign به عنوان حالت خاصی از طرح امضای GGH نیز شکسته شد. از سال ۲۰۱۳ این رویکرد از سرگرفته‌شد[25]. دسته دیگری از امضاهای مشبکه‌مبنا، در سال ۲۰۰۹ با تبدیل پروتکل احراز اصالت به امضای رقمی با استفاده از تبدیل فیات شامیر توسط لوباشوفسکی [۱۸]مطرح‌شد و تاکنون در حال پیشرفت است[49].

اولین طرح امضای رقمی کارا همراه باامنیت اثبات پذیر توسط میشیانشیو [۱۹] و ودهم[۲۰] در سال ۲۰۰۳ ارائه‌شد که ابتدا بااستفاده از اثبات‌های هیچ‌آگاهی [۲۱] برای مسائل مشبکه، یک طرح احرازاصالت معرفی کرده [51] و سپس از روی آن یک طرح امضای رقمی امن در مدل پاسخگوی تصادفی ساختند.

کار مستقل دیگر که در سال ۲۰۰۸ ارائه‌شد، طرح لوباشوفسکی و میشیانشیو بود که دارای امنیت قابل اثبات در مدل استاندارد است. در ادامه‌ی این مسیر امضای کاراتری ارائه‌شد که در آن از روش‌های حدس زدن محدود [۲۲] و روش محو دریچه [۲۳] استفاده شده‌است[48]. در سال ۲۰۱۱، بونه[۲۴] و فریمن[۲۵] یک طرح امضای هم‌ریخت خطی [۲۶] مبتنی بر مشبکه ارائه دادند[16]. در سال ۲۰۱۴، گربنوو[۲۷] و همکاران طرح‌های امضای تمام هم‌ریخت مبتنی بر مشبکه باامنیت اثبات‌پذیر در مدل استاندارد و مدل پاسخگوی تصادفی ارائه‌دادند.

در این پروژه به بررسی یک امضای گروهی مشبکه مبنا با امنیت اثبات پذیر در مدل پاسخگوی تصادفی می‌پردازیم. لذا در ابتدا تعاریف اولیه را ارائه می‌دهیم و سپس به بررسی امضاهای گروهی، و امنیت این امضاها، در دو نوع استفاده از تعاریف پایه‌ای و کلی، و استفاده از مشبکه‌ها ارائه می‌دهیم و در فصل آخر، به بررسی امضای حلقوی و امنیت آن، در دو نوع مبتنی بر سیستم RSA و استفاده از سیستم رابین[۲۸] خواهیم پرداخت.

---

[۱۸] Lyubashevsky
[۱۹] Mocciancio
[۲۰] Vadham
[۲۱] Zero-Knowledge Proofs
[۲۲] confined guessing
[۲۳] vanishing trapdoor
[۲۴] Boneh
[۲۵] Freeman
[۲۶] linearly homomorphic signature
[۲۷] Gorbnnov
[۲۸] Rabin

۴.۱ امضای گروهی

امضاهای گروهی [۲۹] نخستین بار توسط چام [۳۰] وهیست [۳۱] در سال ۱۹۹۱ معرفی شده‌اند [21] که به افراد یک گروه اجازه می‌دهد که بدون اینکه شناسه‌ی فرد مشخص شود، این فرد از اعضای گروه، پیامی را از طرف کل گروه امضا کند.

در سال ۱۹۹۷، کامنیش[۳۲] و استدلر[۳۳] پیشنهاد اولین طرح امضای گروهی را دادند که در آن اندازه‌ی کلید عمومی و امضا بستگی به اندازه‌ی گروه نداشت، یعنی اندازه‌ی آنها ثابت بود[19]. در این طرح، از اولیه‌های رمزنگاری مانند تک‌ـپیام [۳۴] و امضاـپاسخ[۳۵] استفاده می‌شد[38]. در ۲۰۰۳، بلیر[۳۶] و میشیانیو و وارینشی[۳۷] تعاریف جدید و خواص امنیتی برای امضاهای گروهی معرفی کردند که ما اکنون استفاده می‌کنیم، و یک طرح کلی از ساختار طرحی که تمامی خواص در آن هست، ارائه دادند[8]. در این طرح از امضای دیجیتال و امنیت IND-CCA سیستم‌های رمزنگاری کلیدعمومی و اثبات‌های هیچ‌آگاهی غیرتعاملی [۳۸]برای گروه‌های ثابت استفاده شده است که دارای یک رئیس گروه است.

امضاهای گروهی مشبکه‌مبنا در سال ۲۰۱۰، توسط گاردن [۳۹] کتز [۴۰]، و وکونتاناتان [۴۱] معرفی شد که در آن طول امضا با تعداد اعضا نسبت خطی دارد[31]. در سال ۲۰۱۳ ، لگویلامای [۴۲] و همکاران اولین طرح امضای گروهی مشبکه مبنا با طول امضای لگاریتمی و امنیت اثبات‌پذیر در مدل پاسخگوی تصادفی را ارائه کردند[39]. در سال ۲۰۱۵ دو طرح مستقل در راستای کاهش طول امضای گروهی و کلید عمومی با امنیت اثبات‌پذیر در مدل پاسخگوی تصادفی ارائه شدند.

از کاربردهای امضای گروهی می‌توان به ”تصدیق گمنام“[۴۳] اشاره کرد به این معنی که فرض کنید یک سرور، قصد دارد یک کاربر را احراز اصالت کند، اما کاربر تمایل به محافظت از اطلاعات شخصی خود، مانند کلمه‌عبور دارد، به شکلی که کاربر برای سرور احراز اصالت گردد، اما فقط مشخص شود که او یک کاربر مجاز است، و هویت شخص آشکار نگردد. برای این منظور امضای گروهی ابزار بسیار مناسبی است. از دیگر کاربردها می‌توان به سیستم ارتباط امن وسایل نقلیه[۴۴] اشاره کرد. در این سیستم‌ها، ماشین‌ها مجهز به فرستنده‌ی کوتاه برد هستند که به ماشین اجازه می‌دهد با ماشین‌های اطراف خود، در یک شعاع مشخصی، ارتباط برقرار کند. برای مثال زمانی که ماشینی که مجهز به این سیستم است

---

[۲۹] Group Signature
[۳۰] Chaum
[۳۱] Heyst
[۳۲] Camenisch
[۳۳] Stadler
[۳۴] Single-Message
[۳۵] Signature-Response
[۳۶] Bellare
[۳۷] Warinshi
[۳۸] Non-interactive Zero-Knowledge Proofs
[۳۹] Gordon
[۴۰] Katz
[۴۱] Vaikuntanathan
[۴۲] Laguillamumie
[۴۳] Anonymous Attestation
[۴۴] Vehicle Safety Communication

قصد دارد به‌طور ناگهانی ترمز کند، از امضای گروهی استفاده می‌کند، تا برای اعضای گروه، مشخص شود که او فردی مجاز است و اعضای گروه، ماشین‌های مجهز به این سیستم هستند. در این صورت از اطلاعات شخصی راننده، همانند سرعت دقیق او، و یا موقعیت مکانی او، حفاظت می‌گردد.

۱.۴.۱ برخی مفاهیم امضای گروهی

رئیس گروه کسی است که اعضای گروه را مشخص می‌کند، کلیدهای خصوصی را به افراد می‌فرستد، و گاهی ممکن است مسئولیت ردیابی را نیز به عهده داشته باشد. در برخی موارد این مسئولیت‌ها بین دونفر که مدیر گروه و رئیس گروه هستند، تقسیم می شود، که این خاصیت خوبی است که شخص ردیاب، به کلید خصوصی افراد دسترسی نداشته باشد. گاهی مدیر گروه به جای رئیس گروه استفاده می‌شود، دراین صورت مدیر گروه کلیدهای خصوصی را برای افراد می‌فرستد اما آنها را نمی‌شناسد و فقط خود افراد به کلیدهای خودشان دسترسی دارند.

اعضای گروه ممکن است ثابت [۴۵] (از ابتدای تشکیل گروه ثابت باشند) یا پویا [۴۶] (اعضایی بتوانند در حین کار گروه اضافه و یا حذف شوند) باشند.

امضای گروهی ممکن است دارای قابلیتی به نام ''لغوشدنی''[۴۷] باشد، که این قدرت را به مدیر/رئیس گروه می‌دهد که شخص خاطی را لغو امضا کند. به این معنی که شخص دیگر نمی‌تواند امضایی از سمت کل گروه انجام دهد.

حتی با وجود امضایی که خاصیت لغوشدنی و پویایی دارد، یک نقص باقی است:

در زمانی که پیامی امضا می‌شود، اعضای گروه ثابت هستند، به این معنی که حتی اگر گروه دارای پویایی باشد، این نقص باقی است که تمام اعضایی که از قبل تعیین شده‌اند، در امضا نقش دارند.

برای رفع این نقص، شامیر [۴۸]، رایوست و تاومان [۴۹] امضای حلقوی را ارائه دادند که امضا کننده‌ی پیام هنگام امضا می‌تواند تشکیل حلقه بدهد.

۵.۱ امضای حلقوی

امضای حلقوی [۵۰]برای اولین بار توسط رایوست، شامیر و تاومان [56] معرفی شد. در این طرح فرض برپایی بسیار کم است: افراد جفت کلیدهای امضای خود را برای هر طرح امضا، تولید می‌کنند که امنیت این طرح‌ها بستگی به وجود جایگشت‌های دریچه‌دار [۵۱] دارد. در این پروژه به بررسی این نوع از امضاها می‌پردازیم.

---

[۴۵] Static

[۴۶] Dynamic

[۴۷] Revocation

[۴۸] Shamir

[۴۹] Tauman

[۵۰] Ring Signature

[۵۱] Permutation Trapdoor جایگشت‌هایی که محاسبه‌ی آن آسان ولی محاسبه‌ی معکوس آنها بدون داشتن تعدادی اطلاعات اضافه، بسیار سخت باشد. به این اطلاعات اضافه که محاسبه‌ی معکوس را برای دارنده‌ی آن بسیار راحت می‌کند، دریچه گفته‌می‌شود.

در امضاهای حلقوی مجموعه‌ای از امضا کنندگان تشکیل حلقه می‌دهند تا گمنامی برای امضاکنندگان مطرح باشد. گمنامی بدین معنی است که وارسی کننده‌ی امضا متقاعد می‌شود امضاکننده یکی از اعضای حلقه است، اما نمی‌تواند امضاکننده‌ی اصلی را تشخیص دهد و هیچ راهی برای از بین بردن گمنامی امضا کننده وجود ندارد. امضاهای حلقوی ابزار مناسبی جهت تامین گمنامی هستند. برای مثال اگر عضوی از گروه بخواهد اطلاعات خرابکارانه در مورد فعالیت‌های گروه منتشر کند، می‌تواند ازامضای حلقوی استفاده کند و دیگران را متقاعد کند که اطلاعات از خود گروه نشئت گرفته‌است. ساختار بعدی مربوط به کتز، بندر و مرسلی [12] است. این ساختار شبیه به ساختار کلی امضای گروهی است که توسط میشیانشیو ارائه شد. این طرح ترکیبی از رمزنگاری کلیدعمومی، امضاها، و اولیه‌هایی نظیر اثبات‌های هیچ‌آگاهی است. امضاهای حلقوی دیگری هستند که بر فرض RSA یا لگاریتم گسسته، یا تلفیقی از این دو هستند. در سال ۲۰۰۳، بونه، جنترای [۵۲]، لین [۵۳] و شاچام [۵۴] یک طرح امضای موثر با امنیت در مدل پاسخگوی تصادفی ارائه کردند. در سال ۲۰۰۷، شاچام و واترز [۵۵] یک طرح امضای حلقوی پیشنهاد کردند که دارای امنیت در مدل استاندارد بود [58]. در سال ۲۰۰۸، لی [۵۶] و کیم [۵۷] اولین طرح امضای حلقوی شناسه مبنا [۵۸] را ارائه دادند [41] و در سال ۲۰۱۲، لی، فن[۵۹] و جی[۶۰] یک طرح امضای حلقوی شناسه مبنا مبتنی برمشبکه، ارائه دادند [42].

از تفاوت‌های امضای گروهی و امضای حلقوی می‌توان به نحوه تشکیل و برپایی اشاره کرد. امضاهای حلقوی عموما بدون برپایی هستند. یعنی نیازی به الگوریتم تولید کلید ندارند. همچنین در امضای گروهی، شخص سوم مورد اعتماد[۶۱] یا مدیر گروه وجود دارد که شکل‌گیری اعضا را مدیریت می‌کند، درحالی که، در امضای حلقوی چنین مرجع مورد اعتمادی وجود ندارد. تفاوت دیگر در امنیت این امضاهاست. امضاهای حلقوی دارای گمنامی بی‌قیدوشرط هستند، به‌طوری‌که مرجع مجازشناس ردیابی [۶۲]، ندارند. به علاوه اعضا کنترل بیشتری بر گمنامی خود دارند به این معنا که می‌توانند در زمان امضای پیام، اعضای دیگری از حلقه را انتخاب کنند، در صورتی‌که در امضا گروهی، اعضا بر افراد دیگر گروه، کنترل ندارند و گروهی که از ابتدا شکل گرفته‌است، همه در امضای پیام نقش دارند. از طرف دیگر در امضای گروهی، هدف محافظت از حریم خصوصی افراد است در ساختاری که در آن، نیاز یا فایده‌ای برای آشکارسازی شناسه‌ی افراد نیست در صورتی‌که، در امضای حلقوی تمایل به حفاظت از حریم خصوصی شخص داریم در ساختاری که در آن، آشکار شدن شناسه‌ی فرد، نامطلوب است. برای مثال، همان‌طور که در بررسی امضای حلقوی فصل دوم خواهیم دید، امضای حلقوی برای نشت یک راز مورد استفاده قرار می‌گیرد، که در آن فردی که امضا را تولید کرده‌است، در واقع عضو هیچ تشکیلاتی

---

[۵۲] Gentry
[۵۳] Lynn
[۵۴] Shacham
[۵۵] Waters
[۵۶] Li
[۵۷] Kim
[۵۸] Attribute-based Ring Signature
[۵۹] Fan
[۶۰] Jia
[۶۱] Trusted Third Party
[۶۲] Tracer Authority

نیست، اما همچنان تمایل دارد که گمنام باشد. از کاربردهای دیگر امضای حلقوی می‌توان به ارتباط امن و احراز اصالت شده بین کاربر و سرور اشاره کرد. فرض کنیم کاربر امضا کننده تمایل دارد که فقط سرور قابلیت وارسی پیام را داشته‌باشد. لذا حلقه‌ای (کاربر و سرور) ایجاد می‌کند و امضای حلقوی ایجاد می‌کند. در نتیجه فقط سرور مطمئن است که امضا از سوی کاربر وارسی گردد و برای سایرین ابهام میان کاربر و سرور وجود دارد.

۶.۱ مفاهیم پایه‌ای

در این بخش به بررسی دو مفهوم پایه‌ای(تعهد و اثبات‌های هیچ‌آگاهی که در ادامه‌ی پایان‌نامه از آن استفاده می‌شود) [24]، و تعاریفی از مشبکه‌ها خواهیم پرداخت.

۱.۶.۱ طرح تعهد

نظریه‌ی تعهد[۶۳] قلب بسیاری از ساختارهای رمزنگاری مدرن است. در این مفهوم ''تعهد'' به این معنی است که یک بازیکن در یک پروتکل قادر به انتخاب یک مقدار از یک مجموعه‌ی متناهی است و به انتخابش تعهد می‌کند به طوری‌که نمی‌تواند انتخابش را تغییر دهد، حتی اگر مجبور به افشای انتخاب خود گردد.

به عنوان یک مثال بازی بین دو شخص P و V را درنظر بگیرید:

شخص P به بیت b تعهد می‌کند. برای این‌کار، b را بر روی کاغذ می‌نویسد، و درون جعبه‌ای می‌گذارد. جعبه را قفل می‌کند.

سپس P جعبه را برای V می‌فرستد.

اگر P بخواهد می‌تواند با دادن کلید به V جعبه را باز کند.

در اینجا دو خاصیت پایه‌ای برای این بازی وجود دارد که برای هر طرح تعهدی پایه‌ای است:

زمانی که جعبه فرستاده می‌شود، P قادر به تغییر محتوای درون جعبه نیست. این خاصیت ''انقیاد'' [۶۴] نامیده می‌شود.

زمانی که V جعبه را دریافت می‌کند، نمی‌تواند به محتوای درون جعبه پی ببرد، مگر اینکه P تصمیم بگیرد کلید را به او بدهد. این خاصیت ''پنهانی''[۶۵] است.

تعریف طرح تعهد با استفاده از RSA:

ابتدا به توضیح سیستم رمزنگاری RSA می‌پردازیم:

الگوریتم RSA شامل چهار الگوریتم است:

تولید کلید: تولید کلید، توزیع کلید، رمزگذاری و رمزگشایی. این الگوریتم شامل کلیدعمومی و کلید خصوصی است. کلید عمومی را می‌توان در معرض عموم قرار داد و برای رمزگذاری مورد استفاده قرار می‌گیرد. پیامی که توسط کلیدعمومی رمزگذاری شده‌است، در یک زمان منطقی، توسط کلید خصوصی

[۶۳] Commitment
[۶۴] binding
[۶۵] hiding

رمزگشایی می‌شود. نکته‌ی اساسی در RSA پیدا کردن سه عدد بسیار بزرگ e، d و n است، به‌طوری‌که:

$$(m^e)^d = m \ (mod\, n)$$

و برای هر شخص دیگری به غیر از رمزگشا، حتی با داشتن e و n و حتی m، یافتن d بسیار سخت است.
توزیع کلید: برای انتقال پیام رمزشده، از باب[۶۶] به آلیس[۶۷]، آلیس کلید عمومی (n, e) را از طریق یک کانال(نه لزوما امن)، به باب می‌فرستد. کلید خصوصی d، توزیع نمی‌شود.
رمزگذاری: فرض کنید باب پیام M را به منظور فرستادن به آلیس انتخاب کرده‌است. ابتدا پیام M را به عدد صحیح m تبدیل می‌کند، که $0 \leq m < n$ و $gcd(m, n) = 1$. سپس متن رمزشده‌ی c را محاسبه می‌کند:

$$c = m^e \ (mod\ n)$$

و c را برای آلیس می‌فرستد.
رمزگشایی: برای یافتن متن اصلی m، آلیس مقدار زیر را با استفاده از کلید خصوصی محاسبه می‌کند:

$$c^d = (m^e)^d = m \ (mod\ n)$$

تولیدکلید: کلیدهای الگوریتم RSA به شکل زیر تولید می‌شوند:

- دو عدد اول و مجزای p و q انتخاب می‌کنیم. برای امنیت بیشتر این دو عدد باید تصادفی و بزرگ باشند.
- مقدار $n = pq$ را محاسبه می‌کنیم.
- مقدار $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1) = n - (p+q-1)$ را محاسبه می‌کنیم(φ تابع اویلر است). این مقدار خصوصی است.
- یک عدد صحیح e انتخاب می‌کنیم که $1 < e < \varphi(n)$ و $gcd(e, \varphi(n)) = 1$.
- مقدار d را طوری مشخص می‌کنیم که $d = e^{-1} \ (mod\ \varphi(n))$. واضح است که $de = 1 \ (mod\ \varphi(n))$.

اکنون به ادامه‌ی مبحث تعهد می‌پردازیم:
همان‌طور که گفته‌شد، در سیستم RSA کلیدها توسط یک الگوریتم تولید می‌شوند. این الگوریتم طول پیمانه را به عنوان ورودی دریافت می‌کند و یک n و e تصادفی، برای استفاده به عنوان کلید عمومی RSA انتخاب می‌کند. دراین مثال الگوریتم باید توسط P اجرا شود و P باید اطمینان داشته باشد که کلید تولیدی توسط الگوریتم به سادگی توسط V نمی‌شکند.
زمانی که P تعهد می‌کند، نکتهِ اساسی این است که، P باید یک انتخاب تصادفی داشته باشد. طرح تعهدی که این انتخاب تصادفی را ندارد، دارای امنیت نیست.

[۶۶] Bob
[۶۷] Alice

تعهد برای V ارسال می‌شود که دارای دو بیت ''تعهد'' و ''تصادفی انتخابی'' است.
طرح تعهدی را در نظر می‌گیریم که با یک الگوریتم زمان چندجمله‌ای احتمالاتی (PPT)[۶۸] G (تولید کننده) تعریف شده‌است. ورودی 1^l را می‌گیرد که l پارامتر امنیتی و متناظر با طول پیمانه‌ی RSA (در این مثال)، است. خروجی یک رشته‌ی pk که کلیدعمومی طرح تعهد است، می‌باشد. طرح تعهد، برای هر کلیدعمومی pk به صورت زیر تعریف می‌شود:

$$Commit_{pk} : \{0,1\}^l * \{0,1\} \to \{0,1\}^l$$

ابتدا مرحله‌ی برپایی، (اجرای G) یک‌بار برای همیشه توسط P یا V اجرا می‌شود و اجرا کننده کلیدعمومی pk را برای دیگری می‌فرستد. در برخی طرح‌ها لازم است که اجرا کننده شخص گیرنده را قانع کند که pk به درستی انتخاب شده‌است. بنابراین یکی از دو طرف ممکن است pk را رد کند.
فرض کنید کلیدعمومی پذیرفته شده‌است. برای تعهد بیت b، شخص P یک r تصادفی از $\{0,l\}^l$ انتخاب می‌کند و تعهد $C \leftarrow Commit_{Pk}(r,b)$ را محاسبه می‌کند. برای بازگشایی تعهد، r و b آشکار می‌شوند و V بررسی می‌کند که $C = Commit_{Pk}(r,b)$ بر قرار باشد.

تعریف ۱.۶.۱. مقدار ناچیز[۶۹]: $\varepsilon(l)$ در l ناچیز است اگر برای هر چندجمله‌ای p و برای هرl به اندازه‌ی کافی بزرگ داشته باشیم: $\varepsilon(l) \leq \frac{1}{p(l)}$

تعریف ۲.۶.۱. تابع ناچیز: تابع $f : \mathbb{N} \longrightarrow \mathbb{R}$ چندجمله‌ای درl است اگر ثابت c و l_0 باشند که برای هر $l > l_0$ داشته‌باشیم: $f(l) \leq l^c$. تابع $\varepsilon : \mathbb{N} \longrightarrow \mathbb{R}$ در l ناچیز است، اگر برای هر ثابت c یک ثابت l_c باشد که برای هر $l \geq l_c$ داشته‌باشیم: $\varepsilon(l) \leq l^{-c}$

تعریف ۳.۶.۱. فاصله‌ی آماری میان دو توزیع احتمال U و V روی مجموعه‌ی X به صورت زیر تعریف می‌شود:

$$SD(U,V) = \sum_{x \in X} |U(x) - V(x)|$$

برای هر دو خانواده توزیع احتمال $U = \{u_l\}_{l \in \mathbb{N}}$ و $V = \{v_l\}_{l \in \mathbb{N}}$ روی مجموعه‌ی X فاصله‌ی آماری را با تابع $SD : \mathbb{N} \longrightarrow \mathbb{R}; l \longrightarrow (u_l, v_l)$ نشان می‌دهیم و توزیع‌ها را ''به‌طور آماری نزدیک''[۷۰] گوئیم اگر $SD(U,V)$ ناچیز باشد و می‌نویسیم: $U \overset{s}{\sim} V$

اکنون به تعریف دو خاصیت اساسی طرح تعهد می‌پردازیم:

انقیاد بی‌قیدوشرط:[۷۱] به این معنا که حتی با قدرت محاسباتی بی‌نهایت، P نمی‌تواند انتخابش را بعد از تعهد، تغییر دهد.

انقیاد محاسباتی:[۷۲] بدون داشتن منابع خیلی بزرگ محاسباتی، شانس تغییر تعهد بسیار ناچیز است.

---

[۶۸] Probabilistic Polynomial Time
[۶۹] Negligible
[۷۰] Statistilally Close
[۷۱] Unconditional Binding
[۷۲] Computational Binding

پنهانی بی‌قیدوشرط:[۷۳] تعهد P به b هیچ اطلاعاتی در مورد b به V نمی‌دهد. حتی اگر V قدرت محاسباتی بی نهایت داشته‌باشد.

پنهانی محاسباتی:[۷۴] به این معنی که یک V که به طور چندجمله‌ای کراندار است برای حدس زدن تعهد، کار بسیار سختی دارد. احتمال موفقیت او، ناچیز است.

نتیجه‌ای که از طرح تعهد به‌دست می‌آید: به سادگی می‌توان دید که اگر هر طرح تعهد در مدل دو بازیکنی، وجود داشته باشد، یک تابع یک‌طرفه نیز وجود دارد. برای مثال در تعریفی که ارائه دادیم، واضح است که تابع $Commit_{Pk}$ برای امن بودن طرح تعهد، باید یک‌طرفه باشد.

قضیه ۴.۶.۱. اگر یک تابع یک‌طرفه وجود داشته‌باشد، طرح تعهدی با انقیاد بی‌قیدوشرط و پنهانی محاسباتی، داریم.

۲.۶.۱ پروتکل‌های هیچ‌آگاهی

برای یک شبکه‌ی کامپیوتری مدرن، برای حفظ سرویس‌های مربوط به امنیت ضروری است اعضا به اطلاعات شخصی مانند کلمه عبور دسترسی داشته‌باشند. مساله‌ی پایه‌ای این است که به محض استفاده‌ی شخص از اطلاعات شخصی خود، به عنوان ورودی، مانند کلمه عبور، به منظور ارسال پیام از طریق شبکه‌ی اینترنتی، خطر نشت اطلاعاتی بالا می‌رود. این مساله با قرار دادن پروتکل‌هایی روی شبکه، حل می‌شود. مفهوم هیچ‌آگاهی برای اولین بار توسط میکالی، گلدویزر و راکوف [۷۵] معرفی شد که راهی برای طراحی چنین پروتکل‌هایی است. به عنوان مثال فرض کنید یک کامپیوتر میزبان داریم که می‌خواهد شناسه‌ی شخصی که تلاش می‌کند وارد سیستم شود، را شناسایی کند. راه حل کلاسیک طراحی یک کلمه عبور خصوصی برای عضو است. زمانی که فرد بخواهد وارد سیستم شود، نام و کلمه عبور خصوصی را وارد می‌کند. این خصوصیات برای کامپیوتر میزبان فرستاده می‌شود و بررسی می‌کند که نام، مخالف لیست ذخیره شده نباشد. فرض کنید مهاجم خط را کنترل می‌کند و می‌تواند کلمه عبور را بردارد، و جعل هویت کند. از این‌رو از پروتکل‌های هیچ‌آگاهی استفاده می‌کنیم که در آن هدف ما، شناسایی شخص است.

اگر پروتکل درست طراحی شده باشد تنها با وارد کردن نام و کلمه عبور، شخص حقیقی وارد سیستم می‌شود، و شخص جعلی حق دست‌یابی به سیستم را ندارد و مهاجم که خط را کنترل می‌کند، قبل از دیدن مکالمات و بعد از دیدن مکالمات به یک میزان از خصوصیات فرد اطلاع دارد. به عنوان مثال ساده فرض کنید :

شخص A کلید خصوصی S_A را دارد. اگر سیستم رمزنگاری امن باشد، رمزگشایی متن رمز شده‌ی $C = P_A(M)$ بدون داشتن کلید خصوصی مساله‌ای سخت است. بنابراین اگر متن رمز شده‌ای برای شخصی می‌فرستیم و او بتواند رمزگشایی کند، به این معنی است که او کلید خصوصی را دارد. با این توضیحات به سراغ پروتکل می‌رویم: فرض کنید شخصی که می‌خواهد وارد سیستم شود(اثبات‌کننده)،

[۷۳] Unconditional Hiding
[۷۴] Computational Hiding
[۷۵] Rackoff

P باشد و میزبان(وارسی‌کننده)، V باشد.

- اگر اثبات‌کننده ادعا کند که شخص A است، وارسی‌کننده یک پیام تصادفیM انتخاب می‌کند و متن رمزشده‌ی $C = P_A(M)$ را به او می‌فرستد.
- اثبات‌کننده متن رمزشده را با استفاده از S_A رمزگشایی می‌کند و نتیجه را در قالب M' به وارسی‌کننده می‌فرستد.
- وارسی‌کننده شناسه‌ی اثبات‌کننده را می‌پذیرد اگروتنهااگر $M' = M$

فرض کنید که وارسی‌کننده جعلی باشد به این معنی که یک مهاجم کنترل وارسی‌کننده را بر عهده دارد و به جای C درست، برای اثبات‌کننده، C' را می‌فرستد. طبق پروتکل، اثبات‌کننده C' را برای مهاجم رمزگشایی می‌کند. این پروتکل، یک پروتکل هیچ‌آگاهی نیست. به این منظور، برای تبدیل پروتکل به پروتکل هیچ‌آگاهی، طرح را با استفاده از ''تعهد'' بازسازی می‌کنیم.
فرض کنید تعهدی داریم که به اثبات‌کننده اجازه می‌دهد به هر پیامی که می‌تواند بازگشایی کند، تعهد کند. فرض کنید $Commit_{Pk}(r, M)$ تعهد پیام M با استفاده از بیت تصادفی r باشد(می‌توان بیت به بیت تعهد کرد):

- اگر اثبات‌کننده ادعا کند شخص A است، وارسی‌کننده پیام تصادفی M را انتخاب کرده و $C = P_A(M)$ را برای اثبات‌کننده می‌فرستد.
- اثبات‌کننده C را با استفاده از S_A رمزگشایی می‌کند و به حاصل $Commit_{Pk}(r, M')$ تعهد می‌کند و به وارسی‌کننده می‌فرستد.
- وارسی‌کننده M را به اثبات‌کننده می‌فرستد.
- اثبات‌کننده بررسی می‌کند اگر $M = M'$ تعهد را باز می‌کند، یعنی r و M' را برای وارسی‌کننده می‌فرستد. در غیر این‌صورت، متوقف می‌شود.
- وارسی‌کننده، شناسه‌ی اثبات‌کننده را می‌پذیرد، اگروتنهااگر $M = M'$ و جفت r و M' به درستی تعهد را باز می‌کند.

سیستم اثبات تعاملی و اثبات‌های هیچ‌آگاهی

این پروتکل بین دو ماشین تورینگ احتمالاتی معمولی که مجهز به نوارهای ارتباطاتی به منظور اجازه‌ی یک ماشین به فرستادن یا گرفتن پیام از ماشین دیگر است، می‌باشد.
برای تعریف سیستم‌های اثبات تعاملی[۷۶]، فرض می‌کنیم یک ماشین به نام ماشین اثبات‌کننده (P) قدرت محاسباتی بی‌نهایت دارد، و ماشین دیگر وارسی‌کننده (V) کراندار زمان چندجمله‌ای است. ماشین یک رشته‌ی ورودی رایج، معمولا x، دریافت می‌کند، خروجی در V ''قبول'' یا ''رد'' است. در این‌صورت

[۷۶] Interactive Proof Systems

می‌گوییم (P, V) عبارت x را ''قبول کرده'' یا ''رد کرده'' است. لازم به ذکر است که یک زبان دودوئی $L \subset \{۰, ۱\}^*$داده شده‌است. در بخش قبل مدلی ارائه کردیم که اثبات کننده ادعا می‌کرد ''یک عبارت مشخص، درست است''. اکنون به این مورد توجه می‌کنیم: ''اثبات کننده ادعا می‌کند یک عبارت منطقی درست است''. یعنی $x \in L$. در دنیای حقیقی معادل با این است که بگوییم یک قضیه درست است.

تعریف ۵.۶.۱. زوج (P, V) یک سیستم اثبات تعاملی، برای L است، اگر دو شرط زیر را داشته‌باشیم:
تمامیت [۷۷]: اگر $x \in L$، احتمال اینکه (P, V) رد شود، ناچیز است.
درستی [۷۸]: اگر $x \notin L$ برای هر اثبات‌کننده‌ی P^* احتمال این‌که (P^*, V) عبارت x را قبول کند، ناچیز است.
خاصیت اول بیان‌گر این است که اثبات‌کنندهِ صادق همیشه می‌تواند وارسی‌کننده را متقاعد کند، و خاصیت دوم بیان‌گر این است که راهی برای قانع کردن وارسی‌کننده، در مورد عبارت نادرست، نیست.

در توضیح اثبات‌های هیچ‌آگاهی می‌توان بدین گونه گفت که، اثبات‌کننده ادعا می‌کند بخشی از اطلاعات را دارد، (به عنوان مثال کلید خصوصی متناظر با یک کلید عمومی). این نوع سیستم‌های اثبات می‌تواند در مدلی مشابه تعریف گردد با این تفاوت که ''تمامیت'' با ''تمامیت آگاهی'' و ''درستی'' ، با ''درستی آگاهی'' جایگزین می‌شود. خاصیت اول بیان‌گر این است که اگر اثبات‌کننده اطلاعاتی را که ادعا کرده‌است، داشته باشد و پروتکل را دنبال کند می‌تواند وارسی‌کننده را قانع کند، و دومی بیان‌گر این است که اگر اثبات‌کننده با هر تکنیکی بتواند وارسی‌کننده را قانع کند، با احتمال قوی می‌توان گفت که اثبات‌کننده اطلاعاتی را که ادعا کرده‌است، داراست.

استدلال‌های تعاملی

نوع دیگری از سیستم‌های اثبات تعاملی، ''استدلال‌های تعاملی''[۷۹] هستند که پروتکل‌هایی کاربردی هستند. در این نوع پروتکل، اثبات‌کننده زمان چندجمله‌ای است. اثبات کننده و وارسی کننده، دارای قدرت محاسباتی واقعی هستند.
یک استدلال تعاملی، برای زبانL، یک سیستم اثبات تعاملی است، با دو تفاوت:
اثبات‌کننده‌ی صادق باید چندجمله‌ای زمان احتمالاتی باشد و تنها مزیت او بر وارسی‌کننده این است که یک ورودی کمکی خصوصی دارد. شرط ''تمامیت'' در این نوع، به این معنی است که یک ورودی کمکی، به ازای هر $x \in L$ هست که به اثبات‌کننده اجازه می‌دهد همیشه وارسی‌کننده را قانع کند.
خاصیت درستی در این نوع به ''برای هر اثبات‌کننده‌ی زمان چندجمله‌ای احتمالاتی''، به جای عبارت ''برای هر اثبات‌کننده''، تغییر پیدا می‌کند.

---
[۷۷] Compeletness
[۷۸] Soundness
[۷۹] Interactive Arguments

هیچ‌آگاهی

هیچ‌آگاهی به عنوان یک خاصیت مازاد است که یک سیستم اثبات تعاملی، یک اثبات آگاهی، یا یک استدلال تعاملی، می‌تواند داشته باشد. فرض کنید وارسی‌کننده، تلاش می‌کند اثبات‌کننده را با استفاده از یک ماشین زمان چندجمله‌ای احتمالاتی دلخواه V^* فریب دهد.

تعریف ۶.۶.۱. سیستم اثبات تعاملی یا استدلال (P, V) برای زبان L ''هیچ‌آگاهی'' است، اگر برای هر وارسی‌کننده‌ی زمان چندجمله‌ای احتمالاتی V^* یک شبیه‌ساز M_{V^*} وجود داشته‌باشد به‌طوری‌که برای هر $x \in L$ و هر ورودی کمکی H، توزیع مکالمات خروجی توسط M_{V^*} روی ورودی x، و H، از توزیع مکالمه‌ی تولیدی توسط (P, V^*) روی ورودی x و H تمایزناپذیر محاسباتی باشد.
چندجمله‌ای زمان احتمالاتی D که $x \in L$ و H را (همانند بالا) به عنوان ورودی می‌گیرد، درنظر بگیرید. در مورد ۰، D علاوه بر ورودی‌های گفته شده، یک مکالمه‌ی تولیدی (P, V^*) روی این ورودی را نیز می‌گیرد. در مورد ۱، D علاوه بر ورودی‌های گفته‌شده، مکالمه‌ی شبیه‌سازی شده‌ی تولید شده، روی همان ورودی را دریافت می‌کند. D بیتی را به عنوان خروجی می‌دهد. فرض کنیم $p_i(x, H)$ برای $i = ۰, ۱$ احتمال خروجی i توسط ماشین D باشد. ''تمایزناپذیری محاسباتی'' [۸۰] به این معناست که حاصل عبارت $|p_۰(x, H) - p_۱(x, H)|$ ناچیز است.

برای برخی پروتکل‌ها مطالعات واقعی و شبیه‌سازی شده دقیقا توزیع یکسانی دارند. این مفهوم ''هیچ‌آگاهی کامل'' [۸۱] است.
در برخی پروتکل‌ها توزیع‌ها، متفاوت هستند اما فاصله‌ی آماری آنها است. این مفهوم ''هیچ‌آگاهی آماری'' [۸۲] است.

اثبات‌های هیچ‌آگاهی غیرتعاملی

نوع خاصی از پروتکل‌های هیچ‌آگاهی، اثبات‌های هیچ‌آگاهی غیرتعاملی هستند. به این معنا که تعاملی بین شخص اثبات‌کننده (P) و شخص وارسی‌کننده (V) وجود ندارد و فقط بر اساس یک پیام (اثبات) از P به سمت V است که هر دو به یک رشته‌ی تصادفی رایج R [۸۳]، دسترسی دارند. قبل از ارسال پیام، الگوریتم برپایی، توسط یک شخص سوم قابل اعتماد، و یا توسط P و V به طور پی‌درپی برای بدست آوردن R اجرا می‌شود. P به ازای عبارت x و مدرک w [۸۴] مقدار $P(R, x, w)$ را محاسبه می‌کند و π را به عنوان خروجی برمی‌گرداند. V به ازای R و برای عبارت x مقدار $V(R, x, \pi)$ را محاسبه می‌کند و خروجی ۰ یا ۱ را می‌دهد.

---
[۸۰] Computationlly indistinguishable
[۸۱] Perfect Zero-Knowledge
[۸۲] Statistical Zero-Knowledge
[۸۳] Common Random String
[۸۴] Witness

۳.۶.۱ مشبکه

تعریف ۷.۶.۱. مشبکه: بردارهای مستقل خطی $b_1, \ldots, b_n$ را در فضای R^m در نظر بگیرید. ترکیبات خطی صحیح این بردارها، تشکیل مجموعه‌ی منظمی از نقاط را می‌دهد، که آن را مشبکه می‌نامیم. مشبکه‌ی L، تولید شده توسط بردارهای مفروض را به صورت زیر نشان می‌دهیم:

$$L(b_1, \ldots, b_n) = \{\sum_{i=0}^{n} x_i b_i : x_i \in \mathbb{Z};\ 1 \leq i \leq n\}$$

m را بعد مشبکه و n را رتبه مشبکه گویند و در صورتی که $m = n$ مشبکه را رتبه‌تمام نامند. مجموعه بردارهای $b_1, \ldots, b_n$ را پایه‌ی مشبکه گویند. اگر B ماتریس پایه‌ی مشبکه باشد که b_i ها ستون‌های B هستند، می‌توان مشبکه L را به صورت $L(B) = \{Bx \mid x \in \mathbb{Z}^n\}$ تعریف کرد.

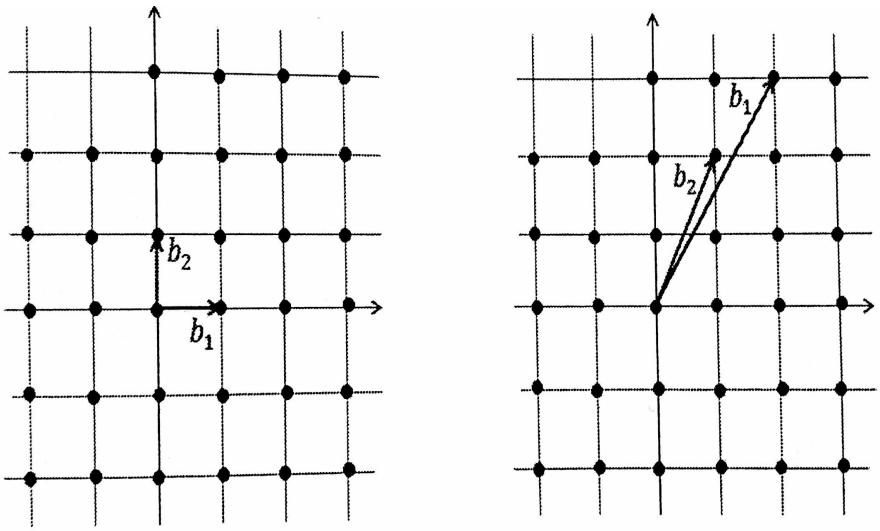
برای تعریف مشبکه، روش دیگری نیز وجود دارد:

تعریف ۸.۶.۱. (مشبکه): یک مشبکه L زیر گروه جمعی گسسته از $\mathrm{R^m}$ است.
گسسته بودن بدین معنی که:

$$\exists \epsilon > 0 : \forall x \neq y \in L,\ \| x - y \| \geq \epsilon$$

زیرگروه جمعی بودن بدین معنی که:

$$\forall x, y \in L, x - y \in L$$



شکل ۱.۱: دو پایه ی متفاوت برای $\mathbb{Z}^2$

تعریف ۹.۶.۱. (ماتریس تک هنگ): ماتریس $U \in \mathbb{Z}^{n\times n}$ یک ماتریس تک هنگ است، اگر :

$$| \, det(U) \, | = ۱$$

لم: اگر ماتریس U تک هنگ باشد، آنگاه $U^{-۱}$ نیز تک هنگ است.

قضیه ۱۰.۶.۱. اگر $B \in \mathbb{R}^{\kappa\times\kappa}$ و $B' \in \mathbb{R}^{\kappa\times\kappa}$ ماتریس‌های پایه باشند، آنگاه دو گزاره زیر معادلند:

۱. $L(B) = L(B')$

۲. ماتریس تک هنگ U وجود دارد به‌طوری‌که: $B' = BU$

از قضیه فوق نتیجه می‌شود که پایه‌ی یک مشبکه یکتا نیست.
اگر B و B' دو پایه‌ی مشبکه باشند، آنگاه ماتریس تک هنگی مانند U وجود دارد، به‌طوری‌که $B' = BU$
بنابراین داریم: $|detB'| = |detB|$

۴.۶.۱ مسائل سخت مشبکه

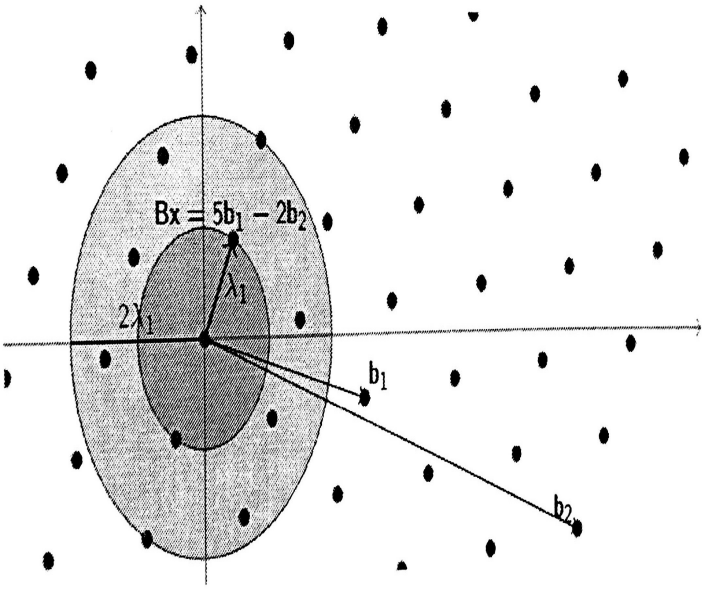
تعریف ۱۱.۶.۱. (مساله کوتاه‌ترین بردار[۸۵]): در مسئله‌ی کوتاه‌ترین بردار (SVP) مشبکه تولید شده توسط پایه ی B، ($L(B)$)، داده شده‌است و هدف یافتن کوتاهترین بردار ناصفر x از مشبکه ی $L(B)$ است. اندازه‌ی کوتاهترین بردار مشبکه را با $\lambda_۱(L)$ نشان می‌دهیم. حل این مساله برای مشبکه‌ای با بردارهای پایه‌ی کوچک و تقریبا متعامد ساده است ولی وقتی طول بردارها بزرگ و خیلی نامتعامدند، بسیار دشوار است[27].

تعریف ۱۲.۶.۱. (مساله تقریب کوتاه‌ترین بردار): در مسئله تقریب SVP هدف یافتن یک بردار ناصفر y از مشبکه است به‌طوری‌که، $\| \, y \, \| \leq \gamma\lambda_۱(L)$ (که γ ضریب تقریب است.)[27].

تعریف ۱۳.۶.۱. (مساله‌ی تصمیم[۸۶] کوتاه‌ترین بردار): در این مساله(GapSVP)، به ازای تابع گپ γ، یک نمونه از مساله‌ی $GapSVP^p_\gamma$ جفت (B, d) می‌باشد، که در آن B پایه‌ی مشبکه‌ی L و d عدد گویا است. برای این مساله، این نمونه قابل پذیرش است هرگاه $v \in L/۰$ موجود باشد که $\|v\|_p \leq d$ و این نمونه قابل پذیرش نیست هرگاه به ازای هر $v \in L/۰$، $\|v\|_p \geq \gamma.d$.

الگوریتم کاهش پایه‌ی مشبکه LLL در سال ۱۹۸۲ توسط لنستر[۸۷] و همکاران با زمان اجرای چند جمله‌ای به ازای هر پایه‌ی دلخواه ورودی ارائه‌شد[40]. الگوریتم LLL مساله SVP را با ضریب تقریب نمایی در زمان چند جمله‌ای حل می‌کند.

تعریف ۱۴.۶.۱. (مساله نزدیک‌ترین بردار[۸۸]): در مساله‌ی نزدیک‌ترین بردار (CVP) مشبکه تولید شده توسط پایه‌ی B، و یک نقطه خارج از مشبکه داده شده‌است و هدف یافتن نزدیکترین نقطه x از مشبکه $L(B)$ به t است[25].



شکل ۲.۱: مسئله تقریب SVP

تعریف ۱.۶.۱۵. (مساله تقریب نزدیک‌ترین بردار): در مسئله تقریب CVP هدف یافتن یک بردار ناصفر y از مشبکه است، به‌طوری‌که $\| y - t \| \leq \gamma.dist(t, L(B))$ باشد.

تعریف ۱.۶.۱۶. (مساله‌ی تصمیم نزدیک‌ترین بردار): در این مساله(GapCVP)، به ازای تابع گپ γ، هر نمونه‌ی $GapCVP_\gamma^p$ سه تایی (B, t, d) است که در آن، B پایه‌ی مشبکه‌ی Λ، $t \in \mathbb{Q}^n$ بردار هدف، و d عددی گویاست. $GapCVP_\gamma^p$ نمونه را می‌پذیرد هرگاه $v \in \Lambda$ موجود باشد که $\|v - t\|_p \leq d$ و پاسخ مساله برای این نمونه منفی است هرگاه به ازای هر $v \in \Lambda$، $\|v - t\|_p > \gamma.d$.

بابای[۸۹] در سال۱۹۸۶ برای حل مساله CVP یک الگوریتم بازگشتی تقریبی تحت عنوان نزدیکترین صفحه، ارائه‌داد که در دو مرحله خلاصه می‌شود[5]. ابتدا کاهش LLL را روی پایه ورودی انجام می دهد سپس یک ترکیب خطی صحیح از بردارهای پایه را جستجو می‌کند که نزدیکترین نقطه به نقطه‌ی هدف t باشد. روش دیگر برای حل تقریبی CVP، الگوریتم گردکردن است که توسط بابای پیشنهاد شده است.

قضیه ۱.۶.۱۷. SVP سختتر از CVP نیست و یک کاهش از SVP به CVP وجود دارد.

---

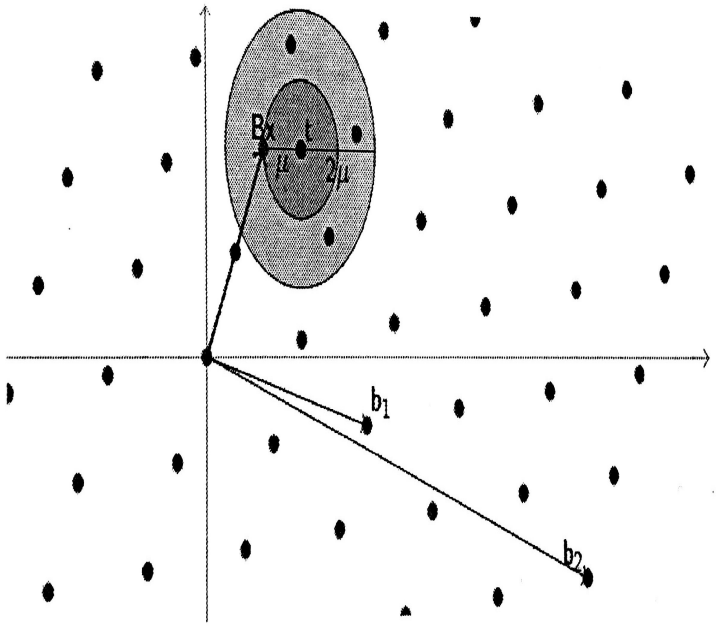
[۸۵] Shortest Vector Problem

[۸۶] Decision Problemمساله‌ای که خروجی آن ''بله'' یا ''خیر'' است، وابسته به ورودی‌هایی که دریافت می‌کند.

[۸۷] Lenstrs

[۸۸] Closest Vector Problem

[۸۹] Babai



شکل ۳.۱: مسئله تقریب CVP

قضیه‌ی فوق بیان می‌کند که اگر CVP حل شود می‌توان SVP را بااستفاده از آن حل کرد.

الگوریتم‌های کاهش مشبکه‌ی شناخته شده، دارای تقریب در حدود δ^n هستند، که در آنها n رتبه مشبکه و δ یک عدد ثابت وابسته به الگوریتم است.

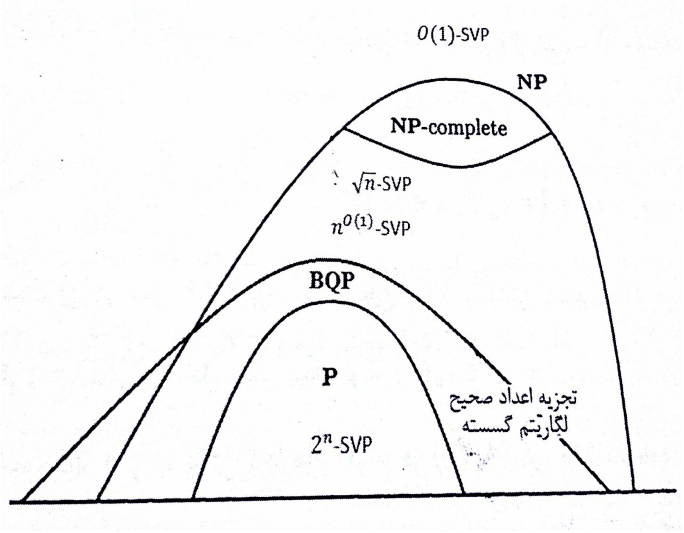
حدس ۱: الگوریتم زمان چندجمله‌ای که بتواند مسائل مشبکه را با ضریب چند جمله‌ای تقریب بزند، وجود ندارد.

تاکنون الگوریتمی برای حل مسائل مشبکه وجود نداشته‌است که بهتر از الگوریتم‌های غیرکوانتومی عمل کند.

حدس۲: الگوریتم کوانتومی در زمان چندجمله‌ای که بتواند مسائل مشبکه را با ضریب چندجمله‌ای تقریب بزند، وجود ندارد.

در شکل ۴.۱ مسائل CVP و SVP و حالت تقریبی آنها در کلاس‌های پیچیدگی قابل مشاهده است. مشاهده می‌شود که مسائل CVP و SVP بدون تقریب در کلاس NP-hard قرار دارند، ولی در حالت تقریب با ضرایب $\sqrt{n}$ و $n^{o(n)}$ دیگر NP-hard نیستند. مسائل تجزیه اعدادصحیح و لگاریتم گسسته درون کلاس BQP[۹۰] قرار دارند و بنابراین با الگوریتم‌های کوانتومی در زمان چندجمله‌ای قابل حل هستند. مساله CVP و SVP باضریب تقریب نمایی در کلاس P قرار دارد و در زمان چندجمله‌ای

[۹۰] Bounded error quantum polynomial time



شکل ۴.۱: کلاس‌های پیچیدگی مسایل مشبکه

قابل حل است.

فصل ۲

امضای گروهی

در این بخش ابتدا به توضیحاتی اجمالی در مورد امضای گروهی می‌پردازیم [47]. پس از آن، به بررسی یک طرح امضای گروهی خواهیم پرداخت که با استفاده از آن نشان می‌دهیم هسته‌ی اصلی تمام ملزومات امنیتی یک طرح امضای گروهی، وابسته به دو خاصیت گمنامی کامل[۱] و ردیابی کامل[۲] است [8]. پس از آن با توجه به نیاز روز افزون جوامع رمزنگاری به سیستم‌های پساکوانتومی، به بررسی یک امضای گروهی مشبکه‌مبنا و بررسی امنیت این امضا خواهیم پرداخت [31].

امضای گروهی نخستین بار توسط چام و هیست معرفی شد که به افراد یک گروه اجازه می‌دهد بدون اینکه شناسه‌ی خود را آشکار کنند، پیامی را از طرف کل گروه امضا کنند.

هر امضای گروهی شامل چهار الگوریتم به شرح زیر است:

- الگوریتم تولید کلید: یک الگوریتم تصادفی است که پارامتر امنیتی 1^k و 1^n (که n اندازه‌ی گروه است) را به عنوان ورودی گرفته و کلیدعمومی pk ، کلید خصوصی رئیس گروه msk و مجموعه $\{sk_i\}_{i=1}^n$ که کلید sk_i کلید خصوصی شخص i است، خروجی می‌دهد.

- الگوریتم امضا (Sign): یک الگوریتم تصادفی است. ورودی این الگوریتم، sk_i، کلید خصوصی شخص i و پیام m است، و خروجی الگوریتم امضای σ است.

- الگوریتم وارسی کردن (Verify): یک الگوریتم قطعی است. کلید pk و امضای σ و پیام m ورودی این الگوریتم است و مقدار ۱ (اگر شخصی از اعضای گروه پیام m را امضا کرده باشد) یا

[۱] Full Anonymity
[۲] Full Traceability

◦ (اگر امضا از طرف عضوی از گروه نباشد) را به عنوان خروجی برمی‌گرداند.

- الگوریتم ردیابی/ بازگشایی (Trace/ Open): یک الگوریتم قطعی است که ورودی این الگوریتم کلید msk و یک امضای σ است . i (شناسه‌ی عضو امضا کننده) یا $\perp$ (امضا معتبر نبوده است یا اینکه الگوریتم قادر به تشخیص شخص امضاکننده نبوده‌است) خروجی این الگوریتم است.

دو خاصیت اصلی امضاهای گروهی، گمنامی و قابلیت ردیابی است.

۱.۲ گروه‌های پویا و مدیر گروه

مدیر گروه کسی است که کمک می‌کند کلیدها به اعضای گروه فرستاده‌شود، اما از کلیدها خبر ندارد.
یادآوری: در امضای گروهی معرفی شده توسط چام و هیست[21] الگوریتم تولید کلید برای بدست آوردن کلید عمومی و کلید خصوصی اشخاص، توسط رئیس گروه اجرا می‌شود. این کلیدها باید توسط کانالی امن به شخص مربوطه می‌رسید. در این روش دو نقص وجود دارد:
رئیس گروه تمام کلیدهای خصوصی افراد را می‌داند.
افراد گروه باید از ابتدا ثابت باشند و نمی‌توانند تغییر کنند.
این دو مورد می‌توانند همزمان اتفاق نیفتند. یعنی می‌توانیم طرحی داشته‌باشیم که افراد بتوانند بعد از تشکیل گروه وارد گروه شوند اما به محض عضویت بتوانند کلیدها را از رئیس گروه بگیرند. به این منظور در حین اجرای الگوریتم تولید کلید رئیس گروه، n را بیشترین تعداد اعضای ممکن که حدس می‌زند وارد گروه شوند، در نظر می‌گیرد. دراین‌صورت کلید کافی برای اعضایی که بعدا وارد گروه می‌شوند، دارد. بدین ترتیب، نقص دوم حل می‌شود.
برای حل کردن هر دو نقص به صورت همزمان، در امضای گروهی پایه‌ای، الگوریتم تولید کلید با الگوریتم برپایی[۳] جایگزین می‌شود. خروجی، کلیدعمومی و احتمالا برخی پارامترهای رایج[۴] ، کلید خصوصی رئیس گروه و کلید ردیابی tk[۵] است. همچنین یک پروتکل جدید $Join \longleftrightarrow Enroll(msk)$ میان اعضا و مدیر گروه قرار می‌دهیم. در آخر این پروتکل، هر عضو فقط sk_i خودش را خواهد داشت و چیزی در مورد msk نخواهد دانست. مدیر گروه نیز، چیزی به غیر از اینکه فرد گیرنده‌ی کلید، عضوی از گروه است، نمی‌داند. الگوریتم ردیابی، به جای کلید msk کلید tk را دریافت می‌کند و بدین ترتیب، مدیر گروه قدرت ردیابی نخواهدداشت.
این ساختار که توسط بلیر، شی [۶] و ژنگ [۷] معرفی شد[11]، نظریه‌ای تازه در مورد امنیت مطرح کرد:
غیر قابل قاب‌بندی [۸]: (یا همان قابلیت تبرئه کردن قوی [۹] [13]) این مفهوم را بیان می‌کند که: هیچ ائتلافی از اعضای مخرب گروه، که حتی شامل مدیر گروه باشد، نمی‌تواند امضای معتبری از طرف

[۳] Setup
[۴] Params
[۵] Tracing Key
[۶] Shi
[۷] Zhang
[۸] Non-frameability
[۹] Exculpability

گروه ارائه دهد.
در بسیاری از گروه‌های پویا، الگوریتم جدیدی علاوه بر الگوریتم‌های گفته شده وجود دارد به نام الگوریتم قضاوت [۱۰]. الگوریتم ردیابی، علاوه بر این‌که شناسه‌ی شخص امضاکننده را مشخص می‌کند، اثبات می‌کند که شخص، امضاکننده‌ی اصلی است. الگوریتم قضاوت، مشخص می‌کند که این اثبات درست است یا نه.

۲.۲ الغاسازی

یکی از خواص امضای گروهی قابلیت لغو امضای اعضای بد رفتار گروه است که توسط رئیس گروه انجام می‌شود [3, 14]. در این مواقع رئیس(مدیر) گروه، کلیدهای عمومی جدید و برای افراد باقی‌مانده‌ی گروه، کلید خصوصی جدید تولید می‌کند. (اگر گروه مدیر گروه داشت، باید الگوریتم $Join \longleftrightarrow Enroll$ را دوباره به کار گیرد.) عضوی که کلید جدید دریافت نمی‌کند از گروه اخراج شده‌است.
این راه‌حل پرهزینه است. راه بهتر که در سال ۲۰۰۴ توسط بونه و شاچام تدوین شد [14]، الغاسازی وارسی‌کننده‌ی محلی [۱۱] است. به این معنی که بررسی اینکه شخصی که لغو امضا شده‌است، نمی‌تواند امضایی تولید کند که تایید شود، بر عهده‌ی وارسی‌کننده است. در بسیاری از امضاهای گروهی وارسی‌کننده علاوه بر m، σ و pk به لیستی به نام ''لیست الغاسازی'' [۱۲] (RL) دسترسی دارد. در این لیست شناسه‌ی اعضایی که لغو امضا شده‌اند، وجود دارد. به عبارت دیگر در این روش فقط نیاز به این است که وارسی‌کننده، و نه دیگر امضاکنندگان، اطلاعات به روز شده‌ی فسخ را داشته‌باشد. شخصی که لغو امضا شده‌است، حریم خصوصی خود را از دست می‌دهد، زیرا امضاهای این شخص، توسط هر وارسی‌کننده که الگوریتم وارسی را یک بار با استفاده از لیست الغا و یک بار بدون استفاده از آن اجرا می‌کند، وارسی می‌شود، اگر برخی امضاها، بدون استفاده از لیست الغا تایید شوند و در زمانی که از لیست استفاده می‌شود، تایید نشوند، وارسی‌کننده پی می‌برد شخص لغو امضا شده، در تولید امضا دخیل بوده‌است. به همین دلیل بونه و شاچام تعریف جدیدی از گمنامی ارائه دادند:
گمنامی فارغ از خود[۱۳]: از آنجاکه یک نفر توانایی دارد خودش را لغو امضا کند، با استفاده از روش گفته‌شده، می‌تواند پی ببرد که امضایی با استفاده از کلید او، ساخته‌شده‌است یا نه.
بونه، بوین وشاچام [13]، روشی پیشنهاد کردند که لیست الغا، علاوه بر وارسی‌کننده به دست اعضای باقی‌مانده‌ی گروه نیز می‌رسد. از این لیست برای به‌روزرسانی کلید عمومی استفاده می‌کنند. به علاوه باقی امضاکنندگان می‌توانند کلیدهای خصوصی خود را با استفاده از این کلیدعمومی به‌روزرسانی کنند. بوین و واترز، طرح خود را به نحوی بسط دادند که توانایی لغو امضا را داشته‌باشد [18]. در ساختار آنها، الگوریتم‌های امضا و وارسی، باید با یک ''اثبات'' که توسط امضا کننده شکل می‌گیرد، تکمیل شود. اثبات این‌که امضاکننده از اعضای لغو امضا شده، نیست. وارسی‌کننده وظیفه دارد که به بررسی درستی

[۱۰] Judge
[۱۱] Verifier-Local Revocation
[۱۲] Revocation List
[۱۳] Selfless-Anonymity

این اثبات بپردازد.
در این بخش به بررسی یک نوع از امضاهای گروهی می‌پردازیم، که در مورد زیرساخت‌های امضای گروهی است [8].

۳.۲ پایه‌های امضای گروهی: تعاریف رسمی و ساختاری مبنی بر مفروضات کلی

یکی از کارهای اصلی در رمزنگاری، تهیه‌ی تعاریف رسمی و قوی از امنیت اولیه‌های رمزنگاری است، و سپس تهیه‌ی ساختاری مبتنی بر مفروضاتی که پیچیدگی محاسباتی دارند (مانند وجود توابع دریچه‌دار یا یک‌طرفه)، به‌طوری‌که در تعاریف اصلی، صدق کنند.
در ساختار امضای گروهی که توسط چام و هیست معرفی شد، یک گروه با تعداد شمارش‌پذیری عضو و یک مدیرگروه وجود دارد. این طرح شامل کلید وارسی‌ـامضای pk است که کلیدعمومی گروه نامیده می‌شود. هر عضو i یک کلید خصوصی امضا دارد که می‌تواند یک امضای مرتبط با pk تولید کند. هسته‌ی اصلی ملزومات گروه، msk است، که توسط آن هر امضای σ داده شده، می‌تواند به شناسه‌ی شخص امضاکننده برگردد(قابلیت ردیابی). اگر کسی این msk را نداشته‌باشد، قادر به یافتن شناسه‌ی تولیدکننده‌ی σ نیست(گمنامی).
ملزومات امنیتی غیررسمی دیگری هستند که تکمیل‌کننده‌ی ملزومات هسته‌ای گفته‌شده هستند. مانند پیوند ناپذیری[۱۴]، جعل ناپذیری، مقاومت در برابر تبانی [۱۵] [4]، قابلیت تبرئه کردن [4]، و مقاومت در برابر قاب‌بندی [22].
در این بخش به مدل جدیدی از حمله توجه می‌کنیم و سپس به تعریف گمنامی کامل و ردیابی کامل که نوع بسیار قوی از ملزومات امنیتی اصلی هستند، به‌طور رسمی می‌پردازیم. به‌علاوه نشان می‌دهیم که این دو خاصیت، برای برقرار بودن تمام ملزومات دیگر کافی هستند. طرحی که بررسی می‌کنیم، براساس تعاریف رمزنگاری و امضای دیجیتال است.

در ادامه نشان می‌دهیم که تمام ملزومات امنیتی غیر رسمی امضاهای گروهی که مطرح کردیم، از طریق این دو خاصیت بالا نتیجه می‌شوند.
طرحی که بررسی می‌کنیم، به‌طور قابل اثبات دارای گمنامی کامل و ردیابی کامل است، پس طبق آنچه گفته شد تمام ملزومات امنیتی دیگر را نیز دارد و نشان می‌دهیم که برای وجود چنین طرحی وجود جایگشت‌های دریچه‌دار، کفایت می‌کند.
توجه به چند نکته ضروری است:

- ساختاری که مطرح می‌کنیم، ”غیربدیهی“[۱۶] است. یعنی اندازه‌ی تمام کلیدها به‌طور لگاریتمی به تعداد اعضای گروه وابسته است.

[۱۴] Unlinkability
[۱۵] Collusion Resistance
[۱۶] Non-Trivial

- این طرح می‌تواند به قابلیت پویایی مجهز شود.
- می‌تواند به امضای گروهی با قابلیت ''امنیت پیش به‌سو'' [۱۷] گسترش پیدا کند[59].

در این طرح از امنیت IND-CCA طرح رمزنگاری کلیدعمومی[26]، خاصیت اثبات‌های هیچ‌آگاهی غیرتعاملی انطباقی شبه‌ـدرستی[۱۸] [28, 57] و یک امضای رقمی با امنیت در برابر حمله‌ی متن انتخابی [6] که با استفاده از وجود جایگشت‌های دریچه‌دار ساخته شده‌اند، استفاده می‌کنیم.
چهارچوب مبنایی طرحی که مطرح می‌کنیم، بر اساس ایده‌های طرح‌های قبلی است. کلید امضای خصوصی عضو گروه شامل یک جفت کلید برای طرح امضای رقمی استاندارد است که توسط مدیر گروه تایید شده‌است. امضای عضو گروه، یک رمزگذاری، تحت کلید رمزگذاری عمومی است.
برای توضیح طرح ابتدا تعاریفی در مورد گروه‌های ساده‌ی ''ثابت''، مطرح می‌کنیم:

۱.۳.۲ تعاریف امنیتی طرح امضاهای گروهی

نمادها و اصطلاحات: اگر x یک رشته باشد، $|x|$ برابر طول x است.
اگر S یک مجموعه باشد $|S|$ اندازه‌ی مجموعه است.
رشته‌ی خالی را با نماد ε نشان می‌دهیم.
فرض کنیم $k \in \mathbb{N}$ در این صورت 1^k به معنی k تا ۱ است.
اگر n عدد صحیح باشد، داریم: $[n] = \{1, \ldots, n\}$ اگر A الگوریتم تصادفی باشد، $[A(x, y, \ldots)]$ مجموعه‌ی تمام نقاطی است که می‌توانند خروجی الگوریتم A برای ورودی‌های $x, y, \ldots$ باشند. $z \xleftarrow{\$}$ $A(x, y, \ldots)$ نتیجه‌ی اجرای A و انتخاب تصادفی، روی همان ورودی است.
تابع $f : \mathbb{N} \longrightarrow \mathbb{N}$ را تابع خوب[۱۹] گوییم اگر به‌طور چندجمله‌ای کراندار باشد (یعنی یک چندجمله‌ای $p(.)$ باشد که برای هر $k \in \mathbb{N}$ ؛ $f(k) \leq p(k)$) و قابل محاسبه در زمان چندجمله‌ای باشد.
پیش از این در مورد ''توابع ناچیز'' بحث کرده‌بودیم. در این طرح به ناچیز بودن تابع دومتغیره نیاز داریم. تابع $\mu : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$ را ناچیز گوییم اگر برای هر تابع خوب مانند $n : \mathbb{N} \longrightarrow \mathbb{N}$ تابع $\mu_n : \mathbb{N} \longrightarrow \mathbb{N}$ که برای $k \in \mathbb{N}$ ؛ $\mu_n(k) = \mu(k, n(k))$ ناچیز باشد.

تعریف ۱.۳.۲. امضای σ را امضای ''درست'' [۲۰] روی متن m گوییم اگر یک $i \in [n]$ وجود داشته‌باشد که $\sigma \in [Sign(sk[i], m)]$.

تعریف ۲.۳.۲. امضای σ را امضای ''معتبر''[۲۱] روی متن m گوییم، اگر $Vf(pk, m, \sigma) = 1$.

درستی: امضای گروهی، باید در شرایط درستی زیر صدق کند:
برای هر $k, n \in \mathbb{N}$ ، هر $(pk, msk, sk) \in [kg(1^k, 1^n)]$ ، هر $i \in [n]$ و هر $m \in \{0, 1\}^*$:

[۱۷] Forward Security. خاصیتی است که در مقابل تغییر کنونی کلیدها، از بخش‌هایی که قبل از تغییر کلیدها وجود داشتند، محافظت می‌کند و بخش‌هایی که از قبل رمزگذاری شده بودند، با کلیدهای جدید رمزگشایی نمی‌شوند.
[۱۸] Simulation-Sound Adaptive Non-interactive Zero-Knowledge proofs
[۱۹] nice
[۲۰] True
[۲۱] Valid

$Vf(pk, Sign(sk[i], m)) = ۱$ و $Open(msk, m, Sign(sk[i], m)) = i$ تساوی اول بیان‌گر این است که یک امضای درست، همیشه معتبر است. تساوی دوم به این حقیقت اشاره دارد که الگوریتم بازگشایی به‌درستی شناسه‌ی امضاکننده‌ی یک امضای درست را کشف می‌کند.
تعاریفی که ارائه دادیم، در مورد گروه‌های ''ثابت'' است. گسترش به گروه‌های ''پویا'' به سادگی امکان‌پذیر است.
در کاربرد ترجیح می‌دهیم که اندازه‌ی کلید و امضا، در طرح امضای گروهی، به تناسب اعضای گروه رشد نکند. طبیعتا بستگی چندجمله‌ای این اندازه ها، به $\log(n)$ اجتناب‌ناپذیر است.

تعریف ۳.۳.۲. طرح امضای گروهی $GS = (Kg, Sign, Vf, Open)$ را ''فشرده''[۲۲] گوییم اگر چند جمله‌ای‌های $p_۱(.,.)$ و $p_۲(.,.,.)$ وجود داشته‌باشند که :
برای هر $k, n \in \mathbb{N}$ ، هر $(pk, msk, sk) \in [Kg(۱^k, ۱^n)]$ ، هر $i \in [n]$ ، هر $m \in \{۰, ۱\}^*$ و هر $\sigma \in [Sign(sk[i], m)]$ داشته باشیم:

$$|pk|, |msk|, |sk[i]| \leq p_۱(k, \log(n))$$

و

$$|\sigma| \leq p_۲(k, \log(n), |m|)$$

برای بررسی طرح و امنیت آن، ابتدا به تعریف های ارائه شده در زیر می‌پردازیم:

تعریف ۴.۳.۲. گمنامی کامل: یک امضای هدف به مهاجم داده می‌شود که به طور تصادفی توسط یکی از دو شناسه‌ی داده شده به مهاجم، تولید شده‌است. گمنامی کامل به این معناست که تعیین این‌که کدام یک از دو شناسه امضا را تولید کرده‌است، برای مهاجم، تمایز ناپذیر است. این در شرایطی است که مهاجم این قدرت را دارد که بتواند تمام افراد گروه، حتی شخص امضاکننده را فاسد[۲۳] کند(به این معنا که او را از گروه خارج کند و یا لغو امضا کند و یا تشکیل ائتلاف دهند). او هم‌چنین به پاسخگوی بازگشایی $Open(msk, ., .)$ که به پیام‌های دلخواه مهاجم مانند m، و امضای σ پاسخ می‌دهد، دسترسی دارد(البته به جز امضای مورد چالش).

برای طرح امضای گروهی $GS = (Kg, Sign, Vf, Open)$ مهاجم A و بیت b آزمایش زیر را در نظربگیرید. A مهاجمی است که در دو مرحله عمل می‌کند. مرحله‌ی انتخاب و مرحله‌ی حدس زدن. در مرحله‌ی انتخاب، A کلیدهای خصوصی افراد گروه به همراه کلید عمومی گروه را دریافت می‌کند. در طول این مرحله، او به پاسخگوی بازگشایی دسترسی دارد. در انتهای این مرحله A دو شناسه‌ی معتبر $۱ \leq i_۰, i_۱ \leq n$ و یک پیام m به همراه برخی ''اطلاعات حالتی''[۲۴] را به‌عنوان خروجی می‌دهد.

[۲۲] Compact
[۲۳] corrupt
[۲۴] State Information

در مرحله‌ی دوم، A یک امضای تولید شده روی متن m که توسط یکی از دو عضو i_0 یا i_1 به‌طور تصادفی تولید شده‌است، دارد. هدف حدس زدن این است که کدام یک از دو کلید خصوصی برای تولید امضا استفاده شده‌است. مهاجم هم‌چنان به پاسخگوی بازگشایی، دسترسی دارد(اما نه در مورد امضای مورد چالش).

آزمایش ۱:

$Exp^{anon-b}_{GS,A}(k,n)$

$(pk, msk, sk) \xleftarrow{\$} Kg(1^k, 1^n)$

$(St, i_0, i_1, m) \xleftarrow{\$} A^{Open(msk,..)}(Choose, pk, sk); \sigma \xleftarrow{\$} Sign(sk[i_b], m)$

$d \xleftarrow{\$} A^{Open(msk,..)}(guess, St, \sigma)$ if A did not query its oracle with m , σ in the guess stage then return d EndIf

return 0

تابع زیر را تعریف می‌کنیم:

$$Adv^{anon}_{GS,A}(k,n) = Pr\left[Exp^{anon-1}_{GS,A}(k,n) = 1\right] - Pr\left[Exp^{anon-0}_{GS,A}(k,n) = 1\right]$$

طرح امضای گروهی GS را دارای '' گمنامی کامل '' گوئیم، اگر برای هر دشمن زمان چندجمله‌ای A تابع دو متغیره‌ی $Adv^{anon}_{GS,A}(.,.)$ ناچیز باشد. به عبارت دیگر، در آزمایش ارائه شده، دشمن موفق است اگر بتواند حدس بزند کدام‌یک از دو شناسه‌ی i_0 یا i_1 پیام m را امضا کرده‌است.

تعریف ۵.۳.۲. ردیابی کامل: فرض کنید گروهی داریم که در آن اعضای گروه با هم تبانی کرده‌اند(ممکن است گروه شامل تمام اعضا باشد). در این‌صورت می‌گوییم گروه خاصیت ردیابی کامل دارد هرگاه هر امضای معتبری که توسط اعضای تبانی کرده، تولید شده را بتوان با کمک الگوریتم بازگشایی به یکی از اعضایی که امضا را تولید کرده‌اند(یعنی افراد تبانی کرده)، نسبت داد. حتی اگر افراد تبانی کرده کلید خصوصی مدیر گروه را بدانند، یا مدیر گروه یکی از افراد تبانی کرده باشد.

برای امضای گروهی $GS = (Kg, Sign, Vf, Open)$ مهاجم A و بیت b آزمایش زیر را در نظر بگیرید. A آزمایش را در دو مرحله اجرا می‌کند. مرحله‌ی انتخاب و مرحله حدس زدن. مهاجم با داشتن دو مقدار pk و msk حمله‌ی خود را به وسیله‌ی فساد یک مجموعه‌ی C از اعضای گروه آغاز می‌کند. در انتهای مرحله‌ی انتخاب، مجموعه‌ی C شامل شناسه‌ی افراد فاسد شده‌است. در مرحله‌ی حدس زدن مهاجم تلاش می‌کند یک امضای جعلی (m, σ) تولید کند.

گوییم A موفق است (آزمایش خروجی ۱ را برمی‌گرداند)، اگر σ یک امضای گروهی معتبر روی پیام m باشد، اما الگوریتم بازگشایی $\perp$ را خروجی دهد، یا اینکه الگوریتم شناسه‌ی شخص i را به عنوان خروجی می‌دهد که $i \notin C$. در غیر این‌صورت، مهاجم A شکست خورده، آزمایش $\circ$ را به عنوان خروجی می‌دهد.

آزمایش ۲:

$Exp^{trace}_{GS,A}(k,n)$

$(pk, msk, sk) \xleftarrow{\$} Kg(1^k, 1^n)$

$St \longleftarrow (msk, pk); C \longleftarrow \emptyset; K \longleftarrow \varepsilon; Cont \longleftarrow true$

$While(Cont = true)do$

$(Cont, St, j) \xleftarrow{\$} A^{Sign(sk[.],.)}(Choose, St, K)$

If $Cont = true$ then $C \longleftarrow C \cup j; K \longleftarrow sk[j]$ EndIf

Endwhile

$(m, \sigma) \xleftarrow{\$} A^{Sign(sk[.],.)}(guess, St)$

If $Vf(pk, m, \sigma) = 0$ then return 0 ; If $Open(msk, m, \sigma) = \perp$ return 1

If there exists $i \in [n]$ such that following are true then return 1 else return 0:

1. $Open(msk, m, \sigma) = i \notin C$
2. i, m was not queried by$A$ to its oracle

تابع زیر را تعریف می‌کنیم:

$$Adv^{trace}_{GS,A}(k, n) = Pr\,[\,Exp^{trace}_{GS,A}(k, n) = ۱]$$

امضای گروهی GS را دارای '' ردیابی کامل'' گوییم، اگر برای هر مهاجم زمان چندجمله‌ای A تابع دومتغیره‌ی $Adv^{trace}_{GS,A}(.,.)$ ناچیز باشد.

۲.۳.۲ رابطه‌ی بین نظریه‌های امنیتی موجود

در این بخش نشان می‌دهیم که گمنامی کامل و ردیابی کامل، تمام ملزومات امنیتی دیگر را نتیجه می‌دهند.

جعل ناپذیری: یک نیاز پایه‌ای برای امضاهای دیجیتال، قابلیت '' جعل ناپذیری'' است. به این معنی که از لحاظ محاسباتی تولید (m, σ) که توسط الگوریتم وارسی، پذیرفته شود، بدون داشتن کلید(های) خصوصی، امکان‌پذیر نیست. این خاصیت به سرعت از ''ردیابی کامل'' نتیجه می‌شود.

قابلیت تبرئه کردن: این خاصیت بیان‌گر این است که هیچ‌یک از اعضای گروه و حتی مدیر گروه، نمی‌توانند از طرف بقیه‌ی اعضا امضای معتبری تولید کند. به‌طور واضح، طرح امضای گروهی که قابلیت ردیابی کامل را دارد، خاصیت تبرئه کردن را نیز دارد. اگر مدیر گروه یا یکی از اعضا، این خاصیت را بشکند، مهاجم به سادگی می‌تواند در مقابل ردیابی کامل، بایستد. اگر مدیر گروه این خاصیت رابشکند، مهاجم به سادگی تکنیک مدیر گروه را دنبال می‌کند و یک امضا تولید می‌کند که در مفهوم قابلیت ردیابی کامل، جعلی است: به این معنی که نمی‌توان به شخص امضاکننده برگشت. اگر شخصی که خاصیت را شکسته است، یکی از اعضای گروه، به‌غیر از مدیر گروه باشد، برای مثال شخص i، مهاجم مانند مرحله‌ی اول آزمایش شماره‌ی ۲، یک درخواست برای $sk[i]$ می‌فرستد، سپس تکنیک شخص را برای تولید امضای جعلی، دنبال می‌کند. با این کار مهاجم در مقابل قابلیت ردیابی کامل، موفق است.

قابلیت ردیابی: ردیابی، به خاصیتی کارآمد گفته‌می‌شود که اگر پیامی با کلید $sk[i]$ امضا شده‌باشد و

الگوریتم بازگشایی برای امضای نتیجه شده استفاده شود، خروجی این الگوریتم باید شناسه‌ی i باشد. این نیاز، بعدها به یک نیاز امنیتی حقیقی گسترش پیدا کرد. به این معنی که: ممکن نیست امضایی تولید شود که الگوریتم بازگشایی نتواند به یکی از اعضای گروه که امضا را تولید کرده‌است، برگردد. این دو نیاز در ابتدا جداگانه به نظر می‌رسید، اما بعدها با عنوان '' مقاومت در برابر تبانی'' شناخته شد.

مقاومت در برابر تبانی: احتمال اینکه گروهی از امضاکنندگان که تبانی کرده‌اند، امضایی تولید کنند که نتواند به هر یک از اشخاص این گروه برگردد، در ابتدا به‌عنوان یکی از مسائل امنیتی امضاهای گروهی، مورد توجه نبود. نیاز به قابلیت ردیابی، حتی در مواجهه با حمله‌ی یک ائتلاف از اعضای گروه، بعدها مورد توجه قرارگرفت، و مقاومت در برابر تبانی نامیده شد. در توضیح این خاصیت جزئیاتی نظیر این‌که '' آیا این خاصیت، پویا هست یا نه''، نامشخص است. یک تعریف از این طرح، می‌تواند با استفاده از آزمایش شمارهِ ۲، توضیح داده‌شود با این تفاوت که مهاجم کلید خصوصی مدیر گروه را ندارد. به این طریق مشخص است که طرح امضای گروهی که قابلیت ردیابی کامل را دارد، در برابر تبانی مقاوم است.

قاب‌بندی: قاب‌بندی، نوعی از مقاومت در برابر تبانی است که اولین بار در [22] مطرح شد. مجموعه‌ای از اعضای گروه را داریم که کلیدهای خود را ترکیب کرده‌اند تا توسط آنها امضاهای معتبری تولید کنند که الگوریتم بازگشایی، این امضا را به اعضای دیگر گروه، نسبت دهد. همانند مقاومت در برابر تبانی، برای قاب‌بندی تعاریف رسمی و تدوین شده نداریم. قوی‌ترین تعریف قاب‌بندی به این صورت است که: فرض کنید آزمایشی داریم که شناسه‌ی شخص U به طور تصادفی از مجموعه‌ی شناسه‌ی همه‌ی اعضا انتخاب شده‌است. کلیدهای خصوصی اعضا به غیر از کلید خصوصی شخص U به همراه کلید خصوصی مدیر گروه، به مهاجم داده‌شده‌است. اگر مهاجم امضایی تولید کند که به شخص U برگردد دراین‌صورت موفق شده‌است. طرح امضای گروهی را مقاوم در برابر قاب‌بندی گوییم اگر روش کارایی وجود نداشته‌باشد که مهاجم با احتمالی بیشتر از ناچیز، بتواند چنین امضایی ایجاد کند.

طرحی که ردیابی کامل دارد، در برابر قاب‌بندی مقاوم است. در واقع مهاجم B در مقابل قاب‌بندی می‌نواند به مهاجم A در مقابل ردیابی کامل تبدیل گردد. او شناسه‌ی یک شخص را به‌طور تصادفی انتخاب و کلیدهای خصوصی بقیه‌ی اعضا را درخواست می‌کند. سپس B را با ورودی کلیدهای خصوصی و کلید خصوصی مدیرگروه اجرا می‌کند، و یک خروجی جعلی می‌دهد. اگر B در برابر قاب‌بندی پیروز باشد، A می‌تواند در مقابل ردیابی کامل، بایستد.

گمنامی: این خاصیت نوع بسیار کم قدرتی از گمنامی کامل است، که مهاجم به پاسخگو دسترسی ندارد و همچنین اطلاعی در مورد کلیدهای خصوصی ندارد. به وضوح طرحی که گمنامی کامل دارد، خاصیت گمنامی را دارد.

قابلیت پیوندناپذیری: این خاصیت، بیشتر از ردیابی کامل، به خاصیت گمنامی کامل مربوط است. به این معنی که اگر شخصی به لیستی از امضاها دسترسی داشته‌باشد، نتواند میان دو امضایی که توسط شخص یکسانی تولید شده‌اند، ارتباط برقرار کند. در بررسی این مورد، باید مهاجم داخلی و خارجی را به‌طور مجزا بررسی کنیم. در هر مورد می‌توان نشان داد که این خاصیت از گمنامی نتیجه می‌شود. به صورت تکنیکی، قابلیت پیوند ناپذیری و گمنامی یکسان هستند.

۳.۳.۲ ساختار امضای گروهی

ابتدا نیاز است اولیه‌هایی را تعریف کنیم:

سیستم‌های اثبات هیچ‌آگاهی غیرتعاملی شبه‌ـ درستی:

در این بخش از سیستم‌های اثبات های NIZK شبه‌ـ درستی در زبان NP استفاده می‌کنیم. سیستم‌های اثبات NIZK ساده، در [28] ارائه شده‌است، و در [57] نشان داده‌شده‌است که چگونه می‌توان هر سیستم اثباتی را به یک سیستم شبه‌ـ درستی تبدیل کرد. این سیستم‌های اثبات، وابسته به رابطه‌های مدرک[۲۵] است. یک رابطه‌ی NP تحت دامنه‌ی $DOM \subseteq \{۰,۱\}^* * \{۰,۱\}^*$ یک زیرمجموعه‌ی ρ از $\{۰,۱\}^*$ است که عضویت $(x,w) \in \rho$ برای هر x در دامنه‌ی Dom قابل تصمیم‌گیری در زمان چندجمله‌ای است. زبان مربوط به ρ مجموعه‌ی تمام $x \in \{۰,۱\}^*$ است، به‌طوری‌که برای هر $(x,w) \in \rho$ یک w وجود داشته‌باشد. اگر $(x,w) \in \rho$ گوییم x یک قضیه و w اثبات است.
فرض کنید الگوریتم‌های زمان چندجمله‌ای P و V که P تصادفی و V قطعی است، و یک رابطه‌ی NP مانند ρ روی دامنه‌ی Dom داریم. P و V به رشته‌ی مرجع رایج[۲۶] دسترسی دارند. گوییم (P,V) تشکیل یک سیستم اثبات غیرتعاملی برای ρ تحت دامنه‌ی Dom می‌دهد، اگر چندجمله‌ای p وجود داشته‌باشد که در دو خاصیت زیر صدق کند:

1) Completness: $\forall k \in \mathbb{N}, \forall (x,w) \in \rho \; with \; |x| \leq k \; and \; x \in Dom :$

$$Pr\,[\, R \xleftarrow{\$} \{0,1\}^{p(k)} ; \pi \xleftarrow{\$} P(k,x,w,R) \; : \; V(k,x,\pi,R) = 1] = 1$$

2) Soundness: $\forall k \in \mathbb{N}, \forall \hat{P}, \forall x \in Dom \; such \; that \; x \notin L_\rho :$

$$Pr\,[\, R \xleftarrow{\$} \{0,1\}^{p(k)} , \pi \longleftarrow \hat{P}(k,x,R) \; : \; V(k,x,\pi,R) = 1] \leq 2^{-k}$$

مشخصات ρ: یک سیستم اثبات (P,V) برای ρ در نظرمی‌گیریم. رابطه‌ی ρ به صورت زیر تعریف می‌شود:
می‌گوییم $((pk_e, pk_s, M, C), (i, pk', cert, s, r)) \in \rho$ اگر و تنها اگر:

$$Vf(pk_s, < i, pk' >, cert) = ۱ \; ; \; Vf(pk', M, s) = ۱ \; ; \; Enc(pk_e, < i, pk', cert, s >, r) = C$$

در تساوی‌های بالا، M پیامی k بیتی، C متن رمزشده و s امضا است و $Enc(pk_e, m; r)$ رمزگذاری پیام M تحت کلید pk_e و استفاده از r تصادفی است که $|r| = k$. دامنه‌ی Dom متناظر با ρ مجموعه‌ی تمام (pk_e, pk_s, M, C) هایی است که pk_e (و همین‌طور pk_s) یک کلیدعمومی است که با احتمال غیرصفر توسط K_e (و همین‌طور Ks) با ورودی m تولید شده‌است و M رشته‌ای k بیتی است. از این‌رو می‌توان گفت ρ رابطه‌ای NP روی Dom است، و می‌توانیم یک سیستم اثبات هیچ‌آگاهی غیرتعاملی (P,V)، برای آن درنظربگیریم. بر این اساس، الگوریتم‌های تعریف امضای گروهی را به صورت زیر داریم:

[۲۵] Witness Relation
[۲۶] Common Reference String

$Algorithm\ Kg(k,n)$

$R \xleftarrow{\$} \{0,1\}^{p(k)}$; $(pk_e, sk_e) \xleftarrow{\$} K_e(k)$; $(pk_s, sk_s) \xleftarrow{\$} K_s(k)$

For $i \leftarrow 1$ to n do

$(pk_i, sk_i) \xleftarrow{\$} K_s(k)$; $cert_i \longleftarrow Sign(sk_s, < i, pk_i >)$

$sk[i] \longleftarrow (k, R, i, pk_i, sk_i, cert_i, pk_e, pk_s)$

Endfor

$pk \longleftarrow (R, pk_e, pk_s)$; $msk \longleftarrow (n, pk_e, sk_e, pk_s)$; $Return(pk, msk, sk)$

$Algorithm\ Sign(sk[i], m)$

Parse $sk[i]$ as $(k, R, i, pk_i, sk_i, cert_i, pk_e, pk_s)$

$s \longleftarrow Sign(sk_i, m)$; $r \xleftarrow{\$} \{0,1\}^k$; $C \longleftarrow Enc(pk_e, < i, pk_i, cert_i, s >, r)$

$\pi \xleftarrow{\$} P(k, (pk_e, pk_s, m, C), (i, pk_i, cert_i, s, r), R)$; $\sigma \longleftarrow (C, \pi)$; Return σ

$Algorithm\ Vf(pk, (m, \sigma))$

Parse pk as (R, pk_e, pk_s) ; Parse σ as (C, π)

Return$V(k, (pk_e, pk_s, M, C), \pi, R)$

$Algorithm\ \ Open(msk, pk, m, \sigma)$

Parse msk as (n, pk_e, sk_e, pk_s) ; Parse σ as (C, π)

If $V(k, (m, C), \pi, R) = 0$ then return $\perp$

Parse $Dec(sk_e, C)$ as $< i, pk, cert, s >$

If $(n < i$ OR $Vf(pk, m, s) = 0$ OR $Vf(pk_s < i, pk >, cert) = 0)$ return $\perp$

Else return i.

نتایج امنیتی: امضای دیجیتال $DS(K_s, Sign, Vf)$، طرح رمزنگاری کلیدعمومی $A\varepsilon = (K_e, Enc, Dec)$، رابطه‌ی NP تحت دامنه‌ی Dom و سیستم اثبات غیرتعاملی (P, V) را درنظرمی‌گیریم. طرح امضای $GS = (Kg, Sign, Vf, Open)$ را، طرح امضای متناظر با آنچه گفته‌شد و بر طبق ساختاری که بیان کردیم، درنظرمی‌گیریم. دو ادعای زیر در [9] اثبات شده‌است:

۱: اگر طرح رمزنگاری $A\varepsilon$ دارای امنیت IND-CCA باشد و (P, V) یک سیستم اثبات هیچ‌آگاهی شبه‌ـدرستی محاسباتی، برای ρ روی دامنه‌ی Dom، باشد، طرح امضای گروهی GS گمنامی کامل دارد.

۲: اگر طرح امضای دیجیتال DS در مقابل جعل شدن، تحت حمله‌ی متن انتخابی مقاوم باشد و (P, V) سیستم اثبات غیرتعاملی برای ρ روی دامنه‌ی Dom باشد، طرح امضای GS ردیابی کامل دارد.

طرح امضای گفته‌شد، فشرده است. یعنی کلیدها مانند امضاهای پیام‌های kـبیتی از اندازه‌ی $poly(k, log(n))$ هستند. هم‌چنین می‌دانیم که وجود خانواده‌ی جایگشت‌های دریچه‌دار، وجود اولیه‌های مورد نیاز ما را نتیجه می‌دهد [6, 26, 28, 57] . بنابراین داریم:

قضیه ۶.۳.۲. اگر خانواده‌ای از جایگشت‌های دریچه‌دار وجود داشته‌باشد، یک طرح امضای گروهی فشرده وجود دارد که گمنامی کامل و ردیابی کامل دارد.

گروه‌های پویا و دیگر بسط‌ها

در بخش قبل در مورد گروه‌های ثابت مواردی را بیان کردیم. در این بخش در مورد گروه‌های پویا صحبت می‌کنیم. گروه‌های پویا دارای دوقسم هستند: گروه‌های پویای جزئی و گروه‌های پویای کامل.
گروه‌های پویای جزئی: طرح‌های امضای گروهی پویا، خاصیت‌های الحاق(افزایشی) یا انفصال(کاهشی) را دارند. در اینجا به گروه‌های افزایشی توجه می‌کنیم و بحث در مورد عملگرهای انفصال را به بخش بعد می‌سپاریم. در طرح گروهی افزایشی، الگوریتم تولید کلید(در کنار تولید کلیدعمومی) دو کلید خصوصی تولید می‌کند: کلید ''صدور'' isk و کلید ''بازگشایی'' msk. هیچ کلید امضای sk تولید نمی‌شود. کلید isk و msk به دو مدیر گروه متفاوت داده می‌شود. که یکی مرجع مجازشناس عضویت در گروه[۲۷] است و دیگری مرجع مجاز شناس ردیابی. با استفاده از isk صادرکننده‌ی کلید می‌تواند کلیدهای امضای $sk[i]$ را (در یک مسیر تعاملی)، تولید کند و بین اعضای گروه، که آنها را با اندیس i می‌شناسیم، پخش کند. تعاریف پایه‌ای امنیت، به همان شکلی است که در بخش قبل توضیح دادیم. کلید isk به مهاجم داده‌می‌شود. با داشتن این کلیدها، او اجازه دارد اعضای گروه ''مصنوعی'' تولید کند و از کلیدهای آنها برای امضای پیام‌هایی که قابل ردیابی نیستند، استفاده کند. ساختاری که در بخش پیش ارائه دادیم، به راحتی می‌تواند به ساختاری با خاصیت‌های گروه‌های افزایشی تبدیل شود. با این تفاوت که کلید تشکیل گواهی[۲۸] sk_s به جای isk استفاده می‌شود. اکنون فرض کنیم امضاکننده‌ی i در زمان t به گروه متصل می‌شود. این شخص می‌تواند از $sk[i]$ جدیدی که به‌دست آورده‌است، برای امضای پرونده‌ای مربوط به قبل از زمان t استفاده کند. این مشکل می‌تواند به راحتی با تجهیز امضا به ''شمارنده‌ی زمان''، و تکنیک‌های امنیت پیش به‌سو حل شود. ابتدا در مورد امنیت پیش به‌سو صحبت می‌کنیم. همانند امضای رقمی [7]، امنیت پیش به‌سو برای امضای گروهی با استفاده از یک تغییر برای کلید، تعریف می‌شود. زمان عمر کلیدعمومی، به دوره‌های زمانی تقسیم می‌گردد و کلید امضای اعضای گروه، در هر زمان تغییر می‌کند. کلید شخص i در زمان j، $gsk_j[i]$ است. در پایان هر دوره، هر عضو کلید خود را با استفاده از الگوریتم به‌روزرسانی $sk_{j+1}[i] = GUpd(sk_j[i])$ به‌روزرسانی می‌کند. امنیت پیش به‌سو که در [59] تعریف شده‌بود، نشت امنیتی دارد [59]: در این طرح فقط بر این نکته تاکید شده‌است که هیچ مهاجمی با داشتن $sk_j[i]$، نتواند از $sk_j[i]$ برای $j < t$ آگاه شود. در اینجا ما امنیت پیش به‌سو را این‌گونه تکمیل می‌کنیم: حمله‌کننده نمی‌تواند امضای معتبری برای پیام‌های قبلی، تولید کند. ساختاری که در این فصل ارائه دادیم، به راحتی می‌تواند به‌گونه‌ای تغییر کند که امنیت پیش‌به‌سو را داشته‌باشد. برای این کار امضایی را که اعضای گروه استفاده می‌کنند، با امضایی که دارای امنیت پیش به‌سو است، جایگزین می‌کنیم. این امضا به شکل $DS = (K_s, Vf, Sign, Upd)$ است.

---

[۲۷] Authority of the Group Membership

[۲۸] Certificate Creation Key. کلیدی که برای ایجاد یک گواهی به‌کار گرفته می‌شود. این گواهی نمایشی رقمی از اطلاعات است که حداقل شامل نام مرجع صدور گواه، نام صاحب گواهی نامه، کلید عمومی او، دوره‌ی اعتبار و تاریخ انقضای گواهی است و توسط مرجع صدور گواهی امضای رقمی شده‌است.

به‌طور کلی اگر گروه دارای امنیت پیش به‌سو باشد، اعضا نمی‌توانند از کلیدهای قانونی خود، برای امضای پرونده‌هایی که مربوط به قبل از زمان ورود آنهاست، استفاده کنند. طرح امضای گروهی پویای جزئی دارای امنیت پیش به‌سو، که عملگر الحاق را پوشش می‌دهد، به صادرکننده‌ی کلید اجازه می‌دهد کلید $sk_t[i]$ را، زمانی که عضو i در زمان t وارد گروه می‌شود، تولید کند.

گروه‌های پویای کامل: در این بخش به گروه‌هایی توجه می‌کنیم که اعضای گروه هم می‌توانند وارد گروه شوند و هم از گروه خارج شوند. همانند گروه‌های پویای جزئی، کلید امضای $sk_j[i]$ ممکن است در طول زمان تغییر کند، اما در اینجا کلید عمومی pk_j نیز اجازه‌ی تغییر دارد. به امضای σ که توسط شخص i در زمان a (با استفاده از کلید $sk_a[i]$)، تولید شده‌است، توجه کنید. فرض کنید این شخص در زمان ۱ به گروه تعلق داشت اما در زمان ۰ و ۲ عضو گروه نبود. در این صورت گوییم σ در زمان b توسط کلید pk_b وارسی شده‌است. اما چه زمانی σ توسط Vf پذیرفته می‌شود؟ دو پاسخ وجود دارد: ۱) اگر i در هنگام تولید امضا عضو گروه بوده و یا ۲) اگر در هنگام فراخوانی الگوریتم وارسی‌کننده، i عضو گروه بوده باشد. (در مورد اول، امضا باید معتبر باقی بماند، (و گمنامی امضاکننده)، حتی بعد از این‌که امضاکننده گروه را ترک کرد، درحالی‌که در مورد دوم، با حذف امضاکننده از گروه تمامی امضاهایی که او داشته‌است، نامعتبر می‌شود). به وضوح جواب درستی وجود ندارد و اینکه کدام تعریف را باید استفاده کرد، بستگی به کاربرد آن دارد. در مورد دوم، σ پذیرفته می‌شود اگر $b = ۱$ ،اما اگر $b = ۰$ یا $b = ۲$ پذیرفته‌نمی‌شود. در حالت خاص، کلیدهای عمومی pk_j باید متفاوت باشند. این خاصیتی نامطلوب است زیرا مستلزم این است که وارسی‌کننده به‌طور مداوم با مدیر گروه، برای به‌روزرسانی کلیدعمومی، ارتباط داشته‌باشد. فرض کنید کلیدعمومی در طول عمر گروه، تغییر نکند اما تابع به‌روزرسان $sk_j[i] = Upd(sk_{j-۱}[i])$ نباید توسط اعضای گروه قابل محاسبه باشد. در نتیجه اعضای گروه نیازمند این هستند که برای به‌روزرسانی کلید امضای خود، از یک زمان به دوره‌ی بعدی، با صادرکننده‌ی کلید در ارتباط باشند.

ساختاری که ارائه دادیم، می‌تواند به ساختاری تبدیل گردد که هر دو تعریف امضای گروهی پویای کامل را، می‌پذیرد. به این منظور در پایان هر دوره‌ی زمانی به تمام اعضا کلید تازه ارسال می‌کنیم. با توجه به هزینه‌ی بالایی که عملگرهای کلیدسازی مجدد، دارند، امضاهای گروهی پویا فقط باید در زمان نیاز در کاربرد، استفاده شوند. در تمام شرایط دیگر، گروه‌های افزایشی، ترجیح داده می‌شوند.

مدیرگروه‌های غیرصادق : فرض کنیم مدیر گروه که msk را دریافت می‌کند، صادق نیست. آزمایش‌هایی که در مورد گمنامی کامل و ردیابی کامل ارائه دادیم، در شرایطی است که مهاجم کلید خصوصی مدیر گروه را به‌دست آورده بود. اگر مدیر گروه صادق نباشد، انتظار می‌رود که در زمان اعمال الگوریتم بازگشایی، صادقانه رفتار نکند. برای مثال زمانی که از او خواسته شود امضایی را باز کند، او شخص دیگری را متهم می‌کند و یا اینکه ادعا می‌کند که امضا نمی‌تواند باز شود. به شرایطی توجه می‌کنیم که در هنگام بازگشایی امضا، خروجی مدیر گروه تنها شناسه‌ی شخص i نیست، بلکه یک اثبات τ نیز هست. این اثبات می‌تواند توسط یک الگوریتم ''قضاوت''، وارسی گردد. به این صورت که اگر $Open(msk, m, \sigma) = (i, \pi)$ دراین صورت $Judge(m, \sigma, i, \tau) = true$. به این شکل امنیت امضا در برابر مدیر گروه غیرصادق، حفظ می‌شود.

طرحی که ارائه دادیم، می‌تواند بر این اساس اصلاح شود. می‌توانیم از یک طرح رمزنگاری قدرتمند $A\varepsilon$

استفاده کنیم به گونه‌ای که در هنگام بازگشایی متن رمز شده، مقدار تصادفی که استفاده شده بود نیز، بازیابی گردد. باز کردن امضای (C, π) به این صورت است: مدیر گروه متن اصلی را تحت C به صورت $< i, pk_i, cert_i, s >$ و همین‌طور r تصادفی را که در ایجاد C استفاده شده‌بود، می‌یابد. سپس (i, τ) را به‌عنوان خروجی می‌دهد که $\tau = (< i, pk_i, cert_i, s >, r)$ یک ''اثبات'' است که امضا توسط i تولید شده‌است. الگوریتم قضاوت، بررسی می‌کند که آیا C نتیجه‌ی رمزنگاری $< i, pk_i, cert_i, s >$ با استفاده از pk_e و r تصادفی هست یا نه.

طرحی که ارائه دادیم مبتنی بر اولیه‌هایی(نظیر امنیت CCA سیستم‌های رمزنگاری کلید عمومی و اثبات‌های هیچ‌آگاهی و امضاهای رقمی مقاوم در برابر حمله‌ی متن انتخابی) بود، که این اولیه‌ها بر اساس وجود توابع دریچه‌دار شکل گرفته‌اند.

از آنجا که ظهور کامپیوترهای کوانتومی سبب شکستن بسیاری از توابع دریچه‌دار شدند، سیستم‌های رمزنگاری و از جمله امضاهای دیجیتال، به سیستم‌های پساکوانتومی روی آوردند. از این جهت، در ادامه به بررسی یک امضای گروهی مشبکه‌مبنا می‌پردازیم.

۴.۲ امضای گروهی مشبکه‌مبنا

در این بخش به بررسی اولین امضای گروهی مشبکه‌مبنا می‌پردازیم[31]. ساختار ما مبتنی بر ترکیب چندین ایده است که با استفاده از تکنیک جدیدی که توضیح خواهیم داد، به هم متصل می‌شوند. در این طرح کلید عمومی رئیس شامل یک کلیدعمومی pk_E برای یک طرح رمزنگاری کلیدعمومی با N کلید وارسی امضا به شکل $pk_1, \ldots, pk_N$ است. کلید خصوصی برای شخص iـام sk_i است که کلید امضای متناظر با pk_i است. برای امضای پیام M عضو گروه: ۱) پیام M را با استفاده از sk_i امضا می‌کند. ۲) امضای منتج را با استفاده از pk_E رمزگذاری می‌کند. و سپس ۳) یک اثبات $NIZK$(به این معنی که متن رمزشده‌ی داده شده، یک امضا روی M، مرتبط با یکی از pk_iها است.) ایجاد می‌کند. این مطلب گمنامی را نتیجه می‌دهد (زیرا کسی به جز مدیر گروه کلید رمزگشایی sk_E متناظر با pk_E را نمی‌داند)، و ردیابی را نتیجه می‌دهد زیرا مدیر گروه می‌تواند متن رمزشده را رمزگشایی کند که شامل هر امضای معتبری است.

برای طرح امضایی که مطرح خواهیم کرد، از طرح امضای GPV [30] استفاده می‌کنیم که به شکل زیر کار می‌کند. کلید عمومی یک پایه‌ی $A \in \mathbb{Z}_q^{n \times m}$ برای یک مشبکه‌ی تصادفی است. برای امضای پیام M امضاکننده از یک دریچه‌ی T به منظور پیدا کردن یک بردار کوتاه $e \in \mathbb{Z}^m$ به همراه $Ae = H(M)$ (که H تابع چکیده‌ساز مدل شده به عنوان یک پاسخگوی تصادفی است)، استفاده می‌کند. در شرایط مناسب، پیدا کردن چنین بردار کوتاهی بدون دریچه، سخت است.

امضای منتج را تحت طرحی تغییر یافته و استاندارد از طرح Regev رمزگذاری می‌کنیم: ماتریس $B \in \mathbb{Z}_q^{n \times m}$ (به عنوان کلیدعمومی) داده شده‌است. با انتخاب یک بردار تصادفی $s \in \mathbb{Z}_q^n$، $e \in \mathbb{Z}^m$ را رمزگذاری می‌کنیم و خروجی متن رمزشده‌ی $z = B^T s + e$ است. در اینجا e به‌عنوان اغتشاش [۲۹] در

[۲۹] Noise

مساله‌ی ''یادگیری با خطا''[۳۰] (LWE) [54] است. قابل ذکر است تمام آنچه که نیاز داریم این است که رمزگذاری یک $e \in \mathbb{Z}_q^m$ تصادفی و یکنواخت، از رمزگذاری یک بردار e انتخاب شده از یک توزیع گوسی تمایزناپذیر گسسته محاسباتی است.

همان‌طور که گفته شد، طرح امضای گروهی پیشنهادی، یک کلیدعمومی رئیس، شامل کلیدهای وارسی $A_1, \ldots, A_N$ به همراه یک کلید رمزگذاری B، یک امضا شامل $z = B^T s + e$ که e به صورت $A_i e = H(M)$ برای برخی (i)، به همراه یک اثبات برای متن رمزشده‌ی z است. ساخت اثبات یک بخش سخت از کار ما است. (برای طراحی چنین اثباتی، به خواص مهم طرح امضای GPV تکیه می‌کنیم.).

به این منظور طرح امضای خود را تغییر می‌دهیم: کلید عمومی رئیس شامل N کلید وارسی به شکل $A_1, \ldots, A_N$ (مانند قبل) و همچنین کلیدهای رمزگذاری $B_1, \ldots, B_N$ است. برای امضای پیام M، شخص i یک امضای حقیقی e_i (با استفاده از دریچه‌ی مرتبط با A_i) و ''شبه‌ـامضای'' e_j برای هر $j \neq i$ محاسبه می‌کند. هر شبه‌ـامضای e_j این خاصیت را دارد که $A_j e_j = H(M)$ گرچه e_j ''کوتاه نیست'' (و بنابراین معتبر نیست). تمام $\{e_j\}_{j=1}^N$ها مانند قبل رمزگذاری شده‌اند، که هر e_j با استفاده از B_j برای ایجاد متن رمزشده‌ی z_j رمز شده‌است. سپس امضاکننده اثباتی تولید می‌کند که ۱) هر z_j یک شبه امضا با توجه به A_j رمزگذاری می‌کند و ۲) حداقل یکی از این شبه امضاها در حقیقت ''کوتاه'' است(و یک امضای معتبر). به‌عنوان راهی برای امضاکننده که بتواند اثبات کند هر متن رمزشده‌ی z_j یک شبه امضا را رمزگذاری می‌کند، یک ابزار تکنیکی جدید راارائه می‌دهیم: یک راه برای یک نمونه‌گیری یک پایه برای ''مشبکه‌ی متعامد'' با دریچه‌ی مربوطه‌ی آن. تکنیکی ارائه می‌دهیم که در آن ماتریس B داده‌شده‌است. این تکنیک یک (A, T) تولید می‌کند که $AB^T = 0 ; mod\, q$ و T یک ''دریچه‌ی خوب'' برای A است (همان‌گونه که در GPV خواسته شده‌است). اگر از ماتریس‌های $\{A_i\}$ تولیده شده توسط همین روش به‌عنوان کلیدهای وارسی در طرح امضای گروهی توضیح داده‌شده، استفاده کنیم، وارسی این‌که یک متن رمزشده‌ی داده شده‌ی z_j یک شبه امضا را رمزگذاری می‌کند، با بررسی $A_j e_j \stackrel{?}{=} H(M)$ میسر می‌شود. این روش به درستی کار می‌کند زیرا $A_j z_j = A_j.(B_j^T s_j + e_j) = A_j e_j = H(M)$.

تنها چیزی که باقی می‌ماند اثبات این است که حداقل یکی از z_jها یک بردار e_j را که ''کوتاه'' است، رمزگذاری می‌کند. به عبارت دیگر حداقل یکی از بردارهای $z_j = B_j^T s_j + e_j$ ''نزدیک به'' مشبکه‌ی تولید شده توسط ستون‌های B_j^T، است. این امر با استفاده از پروتکل هیچ‌آگاهی(آماری)، به همراه تکنیکی استاندارد [23, 29] برای ساخت اثبات مدرک‌ـ تمایزناپذیر [۳۱] و غیرتعاملی در مدل پاسخگوی تصادفی ، که توسط میشیانشیو و ودهم [51] اثبات شده‌است، امکان‌پذیر است. در اصل از یک سیستم اثبات NIZK برای یک زبان ساده استفاده می‌کنیم.

---

[۳۰] Leatning With Errors

[۳۱] Witness-Indistinguishable Proof. نوع ضعیفی از اثبات‌های هیچ‌آگاهی است و تنها ضمانت می‌کند که وارسی کننده نمی‌تواند بین اثبات کننده‌هایی که از مدارک متفاوت استفاده می‌کنند، تمایز قائل شود. در شرایط خاص ممکن است اطلاعات را در مورد مجموعه‌ی مدارک نشت دهد و یا حتی مدرک استفاده شده را نشت دهد، اگر فقط یک مدرک ممکن باشد.

۵.۲ اولیه‌هایی از مشبکه

پارامتر امنیتی را n در نظرمی‌گیریم. بردارهای استفاده شده معمولا ستونی هستند. $\|x\|$ را $l_۲$ (نرم اقلیدسی) درنظرمی‌گیریم و $\|B\|$ ماکزیمم نرم‌های اقلیدسی ستون‌های B است. متعامدسازی گرام‌ـ اشمیت ماتریس B را با $\tilde{B} = (\tilde{b_۱}|\dots|\tilde{b_n})$ در نظرمی‌گیریم که به این شکل به‌دست می‌آید: $\tilde{b_۱} = b_۱$ و برای هر $i = ۲, \dots, n$، $\tilde{s_i}$ مولفه‌ی s_i متعامد به $span(s_۱, \dots, s_{i-۱})$ است. اگر $x \in \mathbb{R}$ در این‌صورت $\lfloor x \rceil$ نزدیک‌ترین عدد صحیح به x است.

برای عدد صحیح q گروه استاندارد از اعداد صحیح به پیمانه‌ی q را با $\mathbb{Z}_q$ نمایش می‌دهیم. برای $q \in \mathbb{Z}^+$ و $x \in \mathbb{R}$، x به پیمانه‌ی q را به عنوان عددی حقیقی و یکتا در $y \in [۰, q)$ در نظرمی‌گیریم به‌طوری که $x - y$ یک مضرب صحیح q است. فاصله‌ی بین دو عضو در $\mathbb{Z}_q$ را به این شکل تعریف می‌کنیم: فرض کنیم $x, y \in \mathbb{Z}_q$ داده شده‌اند. فاصله‌ی بین آنها با نگاشت $(x - y)$ به پیمانه‌ی q به مجموعه‌ی اعداد صحیح $\{-\lfloor \frac{q}{۲} \rfloor, \dots, \lfloor \frac{q}{۲} \rfloor\}$ و سپس با در نظر گرفتن قدرمطلق، به‌دست می‌آید.
یک q ثابت و ماتریس $B \in \mathbb{Z}_q^{n \times m}$ داده شده‌است. مشبکه‌ی m بعدی $\mathcal{L}(B^T)$ را به شکل $\mathcal{L}(B^T) \stackrel{def}{=} \{y \in \mathbb{Z}^m | y \equiv B^T \ mod\ q \ for\ some\ s \in \mathbb{Z}^n\}$. و ''مشبکه‌ی متعامد'' $\Lambda^\perp(B)$ را به شکل $\Lambda^\perp(B) \stackrel{def}{=} \{w \in \mathbb{Z}^m | B.w = ۰ \ mod\ q\}$ تعریف می‌کنیم. برای یک بردار $z \in \mathbb{Z}_q^m$ تعریف می‌کنیم:

$$dist\left(\mathcal{L}\left(B^T\right), z\right) \stackrel{def}{=} min_{s \in \mathbb{Z}_q^n} \|B^T s - z\|.$$

$dist\left(\mathcal{L}\left(B^T\right), z\right)$ فاصله‌ی z از مشبکه‌ی پوشیده شده توسط ستون‌های B^T است.

۱.۵.۲ توزیع‌های خطای گاوسی

توزیع گاوسی(پیوسته[۳۲]) یک بعدی روی $\mathbb{R}$ توسط $s \in \mathbb{R}^+$ نشان داده می‌شود و توسط تابع چگالی زیر تعریف می‌شود:

$$\forall x \in \mathbb{R}: \quad D_s(x) = \frac{۱}{s}.exp(-\pi(\frac{x}{s})^۲).$$

در این طرح بیشتر از گاوسی ''کوتاه شده''[۳۳] استفاده می‌کنیم، که همان توزیع گاوسی D_s محدود به اعداد $x \in \mathbb{R}$ است به‌طوری‌که $|x| < s.\omega(\sqrt{logn})$. توزیع‌های کوتاه شده و غیر کوتاه شده به‌طور آماری نزدیک هستند. به همین خاطر از این به بعد از واژه‌ی کوتاه شده استفاده نمی‌کنیم. توزیع گاوسی m بعدی پیوسته به شکل مشابه به وسیله‌ی تابع چگالی $D_s(x) = \frac{۱}{s^m}.exp(-\pi(\frac{\|x\|}{s})^۲))$ تعریف می‌شود. توزیع گاوسی پیوسته‌ی m بعدی مرکزی در نقطه‌ی $c \in \mathbb{R}^m$ را توسط $D_{s,c}$ نمایش می‌دهیم: $D_{s,c}(x) = \frac{۱}{s^m}.exp(-\pi(\frac{\|x\|}{s})^۲)$.
فرض کنیم $\Lambda \subseteq \mathbb{Z}^m$ یک مشبکه باشد. ''توزیع گاوسی گسسته'' $D_{\Lambda,s,c}$ توزیع گاوسی m بعدی مرکزی در نقطه‌ی c است، اما محدود به مشبکه‌ی Λ. (برای اختصار به‌جای $D_{\Lambda,s,۰}$ از $D_{\Lambda,s}$ استفاده می‌کنیم).

[۳۲] Continuous
[۳۳] Truncated

تابع چگالی توزیع گاوسی گسسته به شکل زیر است:

$$\forall x \in \Lambda : \quad D_{\Lambda,s,c}(x) = \frac{D_{s,c(x)}}{\sum_{y\in\Lambda} D_{s,c}(y)}$$

جنترای نشان داد[30] اگر پایه‌ی B برای مشبکه‌ی Λ داده شده‌باشد، این توزیع می‌تواند نمونه‌ای باشد (در فاصله‌ی آماری ناچیز) برای $s \geq \|B\|.\omega(\sqrt{logn})$.

۲.۵.۲ مساله‌ی یادگیری با خطا

این مساله توسط رگو [54] به‌عنوان کلیتی از مساله‌ی ''یادگیری برابر با خطا''[۳۴] معرفی شد. عدد مثبت n اعداد $m \geq n$ و $q \geq ۲$ یک بردار $s \in \mathbb{Z}_q^n$ و توزیع احتمال χ روی فاصله‌ی $[۰, q)^m$ را در نظر بگیرید. دو توزیع زیر را روی $\mathbb{Z}_q^{n\times m} \times [۰, q)^m$ در نظر بگیرید:

- $LWE_{m,q,\chi}(s)$ توزیع به‌دست آمده توسط انتخاب یکنواخت $A \in \mathbb{Z}_q^{n\times m}$، نمونه‌ی $e \longleftarrow \chi$ و خروجی $(A, A^T s + e \ mod \ q)$.
- $U_{m,q}$ توزیع به‌دست آمده با انتخاب یکنواخت $A \in \mathbb{Z}_q^{n\times m}$ و انتخاب یکنواخت $y \in [۰, q)^m$ و با خروجی (A, y).

مساله‌ی تصمیم LWE (مرتبط با توزیع χ) می‌تواند به عنوان مساله‌ی تمایز بین $U_{m,q}$ و $LWE_{m,q,\chi}$ برای یک s یکنواخت بیان شود. در این‌صورت برای m و q و χ که ممکن است بستگی به پارامتر امنیتی n داشته‌باشند، مساله‌ی $LWE_{m,q,\chi}$ ''سخت'' است، اگر برای هر الگوریتم زمان چندجمله‌ای D مقدار زیر ناچیز باشد:

$$|Pr[s \leftarrow \mathbb{Z}_q^n; (A, y) \leftarrow LWE_{m,q,\chi}(s) : D(A, y) = ۱] - Pr[(A, y) \leftarrow U_{m,q} : D(A, y) = ۱]|$$

در ساختار استاندارد، مساله‌ی LWE با خطای مربوط توزیع Ψ_α^m در $[۰, q)^m$ به این شکل تعریف می‌شود: m عدد به شکل $\eta_۱, \ldots, \eta_m \longleftarrow D_\alpha$ در نظر می‌گیریم. قرار می‌دهیم $e_i := q.\eta_i \ (mod\, q)$ و مقدار $e := (e_۱, \ldots, e_m)^T$ را محاسبه می‌کنیم. برای راحتی از $LWE_{m,q,\alpha}(s)$ به‌جای $LWE_{m,q,\Psi_\alpha^m(s)}$ استفاده می‌کنیم. دلیل سختی مساله‌ی $LWE_{m,q,\alpha}$ از یک نتیجه در [54] نتیجه می‌گردد.
توزیع خطای دوم در مساله‌ی LWE (و گونه‌ای که در این طرح استفاده خواهیم کرد) توزیع گاوسی گسسته‌ی $D_{\mathbb{Z}^m,s} \ (mod\, q)$ است. اگرچه این توزیع شبیه به نوع گسسته‌ی مرتبط با توزیع Ψ_α^m است، این توزیع‌ها به‌طور آماری ''دور'' از یکدیگر هستند و بنابراین نمی‌توانیم به سادگی در مورد سختی مساله‌ی LWE با توجه به یک توزیع، در مورد سختی مساله‌ی LWE با توجه به یک توزیع دیگر نتیجه بگیریم. با استفاده از نتیجه‌ی آخر از [52] می‌توان نشان داد که سختی مساله‌ی LWE مرتبط با خطای توزیع $D_{\mathbb{Z}^m,\alpha.q.\sqrt{۲}}$، از سختی مساله‌ی LWE مرتبط با خطای توزیع Ψ_α^m نتیجه می‌شود. برای راحتی از $\widehat{LWE}_{m,q,\alpha.q\sqrt{۲}}$ به‌جای $LWE_{m,q,D_{\mathbb{Z}^m,\alpha q\sqrt{۲}}}$ استفاده می‌کنیم.

[۳۴] Learning Parity With Noise

لم ۱.۵.۲. برای هر α سختی مساله‌ی $LWE_{m,q,\alpha}$ از سختی مساله‌ی $\widehat{LWE}_{m,q,\alpha q\sqrt{۲}}$ نتیجه می‌شود.

برهان. یک تبدیل T را در نظر می‌گیریم که $(A,y)\in\mathbb{Z}_q^{n\times m}\times[۰,q)^m$ و دو خاصیت زیر را دارد:

- اگر (A,y) روی $\mathbb{Z}_q^{n\times m}\times[۰,q)^m$ یکنواخت باشد، در این‌صورت خروجی $T(A,y)$ روی $\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^m$ یکنواخت است.
- اگر (A,y) دارای توزیعی مرتبط با $LWE_{m,q,\alpha}(s)$ باشد، در این‌صورت $T(A,y)$ دارای توزیعی متناسب $\widehat{LWE}_{m,q,\alpha q\sqrt{۲}}(s)$ است.

لم به سادگی از این دو خاصیت نتیجه می‌شود. □

فرض کنیم تبدیلی داریم به نام T که به این شکل کار می‌کند: (A,y) داده شده‌است. این تبدیل یک بردار $w\longleftarrow D_{\mathbb{Z}^m-y,\alpha q}$ را به‌عنوان نمونه می‌گیرد و خروجی این تبدیل $(A,y+w\,(mod\,q))$ است. (A,y) روی $\mathbb{Z}_q^{n\times m}\times[۰,q)^m$ به‌طور یکنواخت توزیع شده‌است. توجه کنید که $y+w$ همیشه صحیح است، و توزیع $w\longleftarrow D_{\mathbb{Z}^m-y,\alpha q}$ فقط بستگی به قسمت کسری در هر y دارد. این نشان می‌دهد که $(A,y+w\,(mod\,q))$ به‌طور یکنواخت روی $\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^m$ توزیع شده‌است.
از سوی دیگر داریم $y=A^Ts+e\,(mod\,q)$ که $e\sim\Psi_\alpha^m$. از آنجا که داریم $A^Ts\in\mathbb{Z}^m$، نمونه‌ی $w\sim D_{\mathbb{Z}^m-y,\alpha q}\,(mod\,q)$ معادل با نمونه‌ی $w\sim D_{\mathbb{Z}^m-e,\alpha q}\,(mod\,q)$ است. می‌توان نشان داد [52] که دو توزیع زیر به‌طور آماری نزدیک به هم هستند:

- نمونه‌ی $e\sim\Psi_\alpha^m$. و ساختار $e'=e+D_{\mathbb{Z}^m-e,\alpha}q\,(mod\,q)$
- نمونه‌ی $e'\sim D_{\mathbb{Z}^m,\alpha q\sqrt{۲}}\,(mod\,q)$.

می‌توان نتیجه گرفت خروجی $T(A,y)=\big(A,A^Ts+(e+w)\,(mod\,q)\big)$ بر طبق $\widehat{LWE}_{m,q,\alpha q\sqrt{۲}}(s)$ توزیع شده‌است.

۳.۵.۲ توابع دریچه‌دار و طرح امضای GPV

اجتای [1]، و الون و پیکرت [2] یک الگوریتم ارائه دادند که یک ماتریس یکنواخت $A\in\mathbb{Z}_q^{n\times m}$ به همراه یک ماتریس دریچه‌ی $T\in\mathbb{Z}^{m\times m}$ تولید می‌کند که در شرایط زیر صدق می‌کنند:

لم ۲.۵.۲. ([2]) یک الگوریتم زمان چندجمله‌ای TrapSamp وجود دارد به‌طوری‌که با ورودی ۱^n و ۱^m و q که $q\geq۲$ و $m\geq۸nlogq$، ماتریس‌های $A\in\mathbb{Z}_q^{n\times m}$ و $T\in\mathbb{Z}^{m\times m}$ را تولید می‌کند، که:

- توزیع روی A به عنوان خروجی TrapSamp به‌طور آماری نزدیک به توزیع یکنواخت روی $\mathbb{Z}_q^{n\times m}$ است.
- ستون‌های T یک پایه برای مشبکه‌ی $\Lambda^\perp(A)$ تشکیل می‌دهند. به‌عنوان نتیجه می‌توان گفت $A.T=۰\,(mod\,q)$

- برای برخی $C < ۴۰$ داریم $\|T\| = O(nlogq)$ و $\|\tilde{T}\| \leq C.\sqrt{nlogq}$.

طرح امضای GPV: جنترای، پیکرت و واکونتاناتان در [30] نشان دادند که چگونه دریچه برای ساخت یک تابع نمونه‌بردارپذیر پیش‌تصویر[۳۵] یک‌طرفه، استفاده می‌شود[30]. اکنون توضیح می‌دهیم که عملکرد یک تابع نمونه‌بردارپذیر پیش‌تصویر چگونه است.
قرار می‌دهیم $q = poly(n)$ و $m \geq ۸nlogq$ و $s \geq C.\sqrt{nlogq}.$ (که ثابت C از لم ۲.۵.۲است). تابع نمونه‌بردارپذیر پیش‌تصویر یک‌طرفه توسط الگوریتم زیر تعریف می‌شود:

- **GPVGen**(۱^n)**:** الگوریتم $TrapSample(۱^n, ۱^m, q)$ را برای به‌دست آوردن (A, T) اجرا می‌کند. تابع $f_A(e) = Ae \ (mod\, q)$ را با دامنه‌ی $\{e \in \mathbb{Z}^m : \|e\| \leq s\sqrt{m}\}$ و برد $\mathbb{Z}_q^n$ تعریف می‌کنیم. سختی وارونگی این تابع مرتبط با توزیع $D_{\mathbb{Z}^m,s}$ روی دامنه است.

- الگوریتم وارونگی دریچه $GPVInvert(A, T, s, u)$: این الگوریتم از $f_A^{-۱}(u)$ نمونه می‌گیرد: با استفاده از جبرخطی، $t \in \mathbb{Z}^m$ به طوری محاسبه می‌کند که $At = u \ (mod\, q)$ (به‌غیر از یک کسر ناچیز A، این چنین t همیشه وجود دارد). در این مرحله یک $e \longleftarrow D_{\Lambda^\perp(A)+t,s}$ نمونه می‌گیرد و به عنوان خروجی می‌دهد.

تابع بالا یک‌طرفه است اگر $GapSVP_\gamma$ در بدترین حالت برای تخمین چندجمله‌ای γ، سخت باشد [1].

۴.۵.۲ نمونه‌گیری یک مشبکه‌ی متعامد با دریچه

در این بخش نوعی از الگوریتم نمونه‌گیری شرح داده‌شده در ۲.۵.۲ را ارائه می‌دهیم. این نوع، الگوریتم یک ماتریس $B \in \mathbb{Z}_q^{n\times m}$ دریافت می‌کند و (لزوما) یک ماتریس $A \in \mathbb{Z}_q^{n\times m}$ به همراه یک دریچه‌ی $T \in \mathbb{Z}^{m\times m}$ مرتبط با این خاصیت اضافی که سطرهای A روی $\mathbb{Z}$ عمود بر سطرهای B هستند، خروجی می‌دهد. به عبارت دیگر باید $AB^T = ۰ \ (mod\, q)$.
بررسی اجمالی طرح: ایده‌ی پایه‌ای به شکل زیر است. قرارمی‌دهیم:

$$B^T = \begin{pmatrix} B_۱ \\ B_۲ \end{pmatrix}$$

که $B_۲$ ماتریس مربعی وارون‌پذیر از بعد $n \times n$ است. سپس یک ماتریس متعامد $A = [A_۱|A_۲]$ در دو مرحله تولید می‌کنیم. $A_۱$ با استفاده از پروتکل TrapSample تولید می‌شود. یادآوری می‌کنیم که یک ماتریس که به‌طور آماری نزدیک به یکنواخت است، به همراه یک دریچه‌ی مرتبط $T_۱$ برگشت داده‌می‌شود. $A_۱$ را انتخاب می‌کنیم و مولفه‌ی دوم، $A_۲$ مقداری ثابت است به‌طوری‌که داشته‌باشیم $AB^T = ۰ \ (mod\, q)$، و به این شکل $A_۲$ را با استفاد از حل معادله‌ای جبری، تولید می‌کنیم.
اکنون، پیدا کردن یک دریچه‌ی T به‌طوری‌که ستون‌های T کوتاه باشند و $A.T = ۰$ هدف بعدی است. برای این‌کار از تکنیکی استفاده می‌کنیم که در [20] شرح داده شده‌است: می‌توان پایه‌ی $T_۱$ را به پایه‌ی بزرگ‌تر T برای $\Lambda^\perp(A)$ گسترش داد. لم زیر را در نظر می‌گیریم:

[۳۵] Preimage-Sampleable Function

لم ۳.۵.۲. یک الگوریتم زمان چندجمله‌ای احتمالاتی مانند OrthoSamp وجود دارد که با استفاده از ورودی‌های 1^n و 1^m و q، (که $q \geq 2$ و $m \geq n + 8nlogq$) و یک ماتریس $B \in \mathbb{Z}_q^{n\times m}$ که ستون‌های آن، $\mathbb{Z}_q^n$ را می‌پوشانند، ماتریس‌های $A \in \mathbb{Z}_q^{n\times m}$ و $T \in \mathbb{Z}^{m\times m}$ را به‌عنوان خروجی می‌دهد، به‌طوری‌که:

- $AB^T = 0 \pmod q$. به‌علاوه توزیع روی A به‌طور آماری نزدیک به یکنواخت روی $\mathbb{Z}_q^{n\times m}$ است.
- ستون‌های T پایه‌ای برای مشبکه‌ی $\Lambda^\perp(A)$ تشکیل می‌دهند. در حالت خاص می‌توان نتیجه گرفت $A.T = 0 \pmod q$.
- هر ستون t_i از T، بر طبق $D_{\Lambda^\perp(A),s}$ توزیع شده‌است، که $s = C.\sqrt{nlogq}.\omega(\sqrt{logm})$ و C ثابت لم۲.۵.۲ است.

الگوریتم OrthoSamp به شکل زیر کار می‌کند:

- مقدار $(A_1, T_1) \longleftarrow TrapSample(1^n, 1^{m_1}, q)$ را محاسبه می‌کنیم(که $m_1 = 8nlogq$ و $m_2 = n$). قرار می‌دهیم $A_2 \in \mathbb{Z}_q^{n\times m_2}$ یک ماتریس تصادفی یکنواخت باشد که در شرط زیر صدق می‌کند:

$$A_2B_2 = -A_1B_1 \pmod q$$

از آنجاکه B_2 روی $\mathbb{Z}_q$ وارون‌پذیر است، A_2 می‌تواند به شکل $A_1B_1B_2^{-1} \pmod q$ محاسبه گردد. اگر ستون‌های A_1، $\mathbb{Z}_q^n$ را نپوشانند، خروجی $\perp$ است. که این احتمال، ناچیز است.

- پایه‌ی T_1 را با استفاده از تکنیک [20] به پایه‌ی $T' \in \mathbb{Z}_q^{m\times m}$ برای $\Lambda^\perp(A)$ گسترش می‌دهیم. برای توضیح بیشتر تکنیک را شرح می‌دهیم: بگذار T' به شکل زیر باشد:

$$T' = \begin{pmatrix} T_1 & W \\ 0 & I \end{pmatrix}$$

که $W \in \mathbb{Z}_q^{m_1\times m_2}$ ماتریسی دلخواه است که $A_1W = -A_2$ و $I \in \mathbb{Z}_q^{m_2\times m_2}$ ماتریس همانی است. (توجه کنید که W بااین فرض وجود دارد که ستون‌های A_1، $\mathbb{Z}_q^n$ را می‌پوشانند.)

- پایه‌ی T' را به "پایه‌ی تصادفی" T تبدیل می‌کنیم. برای این‌کار از الگوریتم RandBasis [20] استفاده می‌کنیم. این الگوریتم با ورودی T' و با استفاده از پارامتر $s = \|\tilde{T'}\|.\omega(\sqrt{logm})$، خروجی $A = [A_1|A_2]$ و T را می‌دهد.

اکنون بررسی می‌کنیم که این الگوریتم در شرایط مورد نیاز صدق کند. ابتدا توجه می‌کنیم که

$$AB^T = A_1B_1 + A_2B_2 = A_1B_1 - A_1B_1 = 0 \pmod q$$

همچنین داریم:

$$A.T' = [A_1|A_2].\begin{pmatrix} T_1 & W \\ 0 & I \end{pmatrix} = [A_1T_1 + A_20|A_1W + A_2] = 0 \pmod q$$

برای تساوی آخر می‌توان گفت از آنجاکه $A_1T_1 = 0$ (با توجه به خاصیت TrapSample) تساوی اتفاق می‌افتد. با توجه به ساختار داریم: $A_1W = -A_2$. بنابراین، T' یک پایه برای $\Lambda^{\perp}(A)$ است. سرانجام از آنجاکه T نتیجه‌ی اجرای الگوریتم RandBasis روی T' است، T نیز یک پایه برای $\Lambda^{\perp}(A)$ است. از [20] می‌دانیم که $\|\tilde{T'}\| \leq \|\tilde{T_1}\| = O(\sqrt{nlogq})$. بنابراین، از خاصیت الگوریتم RandBasis از [20] هر ستون T بر طبق $D_{\Lambda^{\perp}(A),s}$ که $s = C.\sqrt{nlogq}.\omega(\sqrt{logm})$ به‌طور وابسته، توزیع شده‌است. نتیجه‌ی زیر از ساختار بالا به‌دست می‌آید و ما از آن در اثبات امنیت طرح امضا استفاده خواهیم کرد.

نتیجه ۴.۵.۲. توزیع‌های

$$\{B \longleftarrow \mathbb{Z}_q^{n\times m}; (A,T) \longleftarrow OrthoSamp(1^n, 1^m, q, B) \ : \ (A,T,B)\}$$

و

$$\{(A,T') \longleftarrow TrapSamp(1^n, 1^m, q); T \longleftarrow RandBasis(T');$$

$$(B,S) \longleftarrow OrthoSamp(1^n, 1^m, q, A) \ : \ (A,T,B)\}$$

به‌طور آماری نزدیک هستند.

۵.۵.۲ اثبات‌های NIWI برای مسائل مشبکه

قرارمی‌دهیم $B_1, \ldots, B_N \in \mathbb{Z}_q^{n\times m}$ و $z_1, \ldots, z_N \in \mathbb{Z}_q^m$. در این بخش به‌طور خلاصه توضیح می‌دهیم که چگونه می‌توان یک اثبات تمایزناپذیر_مدرک غیرتعاملی [۳۶] (NIWI) (در مدل پاسخگوی تصادفی) برای یک زبان $L_{s,\gamma} = (L_{YES}, L_{NO})$ ساخت. این زبان به شکل زیر تعریف می‌شود:

$$L_{YES} = \left\{ \begin{pmatrix} B_1 & \ldots & B_N \\ z_1 & \ldots & z_N \end{pmatrix} \middle| \exists s \in \mathbb{Z}_q^n \ and \ i \in [N] \ : \ \|z_i - B_i^T s\| \leq s\sqrt{m} \right\}$$

$$L_{NO} = \left\{ \begin{pmatrix} B_1 & \ldots & B_N \\ z_1 & \ldots & z_N \end{pmatrix} \middle| \forall s \in \mathbb{Z}_q^n \ and \ i \in [N] \ : \ \|z_i - B_i^T s\| > \gamma.s\sqrt{m} \right\}$$

در اینجا L_{YES} مجموعه‌ی N عضوی از نقاطی است که حداقل یکی از آنها نزدیک به مشبکه‌ی متناظر است، و L_{NO} مجموعه‌ی N عضوی از نقاطی است که همه‌ی آنها از مشبکه‌ی متناظر دور هستند. نقطه‌ی شروع ما، یک سیستم اثبات تمایزناپذیر_مدرک (WI) (تعاملی) برای نوع مساله‌ی تصمیم نزدیک‌ترین بردار است، به این معنی که برای زبان $L'_\gamma = \{L'_{YES}, L'_{NO}\}$ [32, 51] داریم:

$$L'_{YES} = \left\{(B,z,t) | \exists s \ : \ \|z - B^T s\| \leq t\right\}.$$

$$L'_{NO} = \left\{(B,z,t) | \forall \ : \ \|z - B^T s\| > \gamma.t\right\}.$$

[۳۶] Nonitteractive Witness-Indistinguishable

می‌توانیم زبان $L_{s,\gamma}$ را با کمک L'_{γ} تعریف کنیم:

$$\begin{pmatrix} B_1 & \dots & B_N \\ z_1 & \dots & z_N \end{pmatrix} \in L_{YES} \Leftrightarrow \vee_i((B_i, z_i, s\sqrt{m}) \in L'_{YES}).$$

$$\begin{pmatrix} B_1 & \dots & B_N \\ z_1 & \dots & z_N \end{pmatrix} \in L_{NO} \Leftrightarrow \wedge_i((B_i, z_i, s\sqrt{m}) \in L'_{NO}).$$

بنابراین می‌توانیم با استفاده از روش دامگارد، مرکل و شوماخر [23]، یک اثبات WI تعاملی برای $L_{s,\gamma}$ با خطای درستی ناچیز به‌دست آوریم. با استفاده از تبدیل فیات ـ شامیر[۳۷] [29] پروتکل منتج می تواند در مدل پاسخگوی تصادفی به شکل غیر تعاملی ساخته شود.
یادآوری می‌کنیم در این ساختار ما تنها نیاز به خاصیت درستی داریم و (الزامی وجود ندارد که سیستم اثبات، یک سیستم اثبات آگاهی باشد) و همچنین به تمایزناپذیری مدرک احتیاج داریم(بیشتر از هیچ‌آگاهی). مشاهدات این بخش در لم زیر خلاصه شده است.

لم ۵.۵.۲. بگذار $\gamma \geq O(\sqrt{\frac{m}{logm}})$. در این صورت یک سیستم اثبات تمایزناپذیر ـ مدرک غیر تعاملی برای زبان $L_{s,\gamma}$ در مدل پاسخگوی تصادفی وجود دارد که طول اثبات $O(mnNlogq)$ است.

۶.۲ یک طرح امضای گروهی مشبکه‌مبنا

۱.۶.۲ تعاریف

در این بخش از تعریف طرح امضا گروهی [8] و پیشنهاد [13] استفاده می‌کنیم.
نیاز اصلی یک طرح امضای گروهی برای داشتن ثبات، این است که یک امضای حقیقی که توسط یکی از اعضای گروه تولید شده‌است، به درستی پذیرفته شود، و بتوان امضا را به همان شخص ردیابی کرد. به‌علاوه، یک طرح امضای گروهی باید دو خاصیت ردیابی و گمنامی را داشته باشد. با استفاده از مدل [13] ما در طرح خود از گمنامی CPA استفاده می‌کنیم که مهاجم به پاسخگو دسترسی ندارد.

۲.۶.۲ طرح امضای گروهی پیشنهادی

قرار می‌دهیم $H : \{0,1\}^* \longrightarrow \mathbb{Z}_q^n$ یک تابع چکیده‌ساز باشد که به‌عنوان پاسخگوی تصادفی مدل شده است. طرح امضای گروهی به شرح زیر است:

- الگوریتم تولید کلید $(1^n, 1^N)$: ابتدا مقدار $(B_1, S_1), \dots, (B_N, S_N) \longleftarrow TrapSamp(1^n, 1^m, q)$ و برای هر $1 \leq i \leq N$ مقدار $(A_i, T_i) \longleftarrow OrthoSamp(1^n, 1^m, q, B_i)$ را محاسبه می‌کنیم. خروجی، کلید عمومی $pk = \left((A_i, B_i)_{i=1}^N\right)$، کلید ردیابی $tk = (S_i)_{i=1}^N$ و کلیدهای $gsk = (T_i)_{i=1}^N$ به عنوان کلیدهای امضای اعضا است.

[۳۷] تکنیکی برای تبدیل طرح شناسه‌مبنا، به طرح امضا

- الگوریتم امضای $(gsk[j], M)$: برای امضای پیام M با استفاده از کلید خصوصی $gsk[j] = T_j$، $r \longleftarrow \{0,1\}^n$ را به صورت تصادفی انتخاب می‌کنیم، قرار می‌دهیم $\bar{M} = M\|r$ و سپس برای $1 \le i \le N$ مقدار $h_i = H(\bar{M}\|i)$ را محاسبه می‌کنیم. سپس:
  - مقدار $e_j \longleftarrow GPVInvert(A_j, T_j, s, h_j)$ را محاسبه می‌کنیم.
  - برای $i \neq j$، $e_j \in \mathbb{Z}_q^m$ را به‌طور یکنواخت انتخاب می‌کنیم به‌طوری که $A_i e_i = h_i \pmod q$.

  برای هر i، $s_i \longleftarrow \mathbb{Z}_q^n$ را نمونه می‌گیریم و مقدار $z_i = B_i^T s_i + e_i \pmod q \in \mathbb{Z}_q^m$ را به‌دست می‌آوریم. سرانجام یک اثبات NIWI π برای زبان $L_{s,\gamma}$ همان‌طور که در ۵.۵.۲ بیان شد، (با استفاده از $((s_i, i))$ تولید می‌کنیم. خروجی امضای $(r, z_1, \ldots, z_N, \pi)$ است.

- الگوریتم وارسی (pk, M, σ): امضا را به شکل $(r, z_1, \ldots, z_N, \pi)$ تجزیه می‌کنیم، و قرار می‌دهیم $\bar{M} = M\|r$. خروجی ۱ است اگرو تنهااگر اثبات π صحیح باشد، و برای هر i داشته‌باشیم $A_i z_i = H(\bar{M}\|i) \pmod q$.

- الگوریتم بازگشایی (tk, M, σ): امضارا به‌صورت $(r, z_1, \ldots, z_N, \pi)$ تجزیه می‌کنیم. با استفاده از $\{S_i\}$ خروجی کوچکترین i است که $dist(\mathcal{L}(B_i^T), z_i) \le s\sqrt{m}$.

ابتدا درستی را بررسی می‌کنیم. قرار می‌دهیم $(r, z_1, \ldots, z_N, \pi)$ امضای تولیدی توسط یک امضا کننده‌ی حقیقی باشد. واضح است که π یک اثبات معتبر است. به‌علاوه، برای هر i داریم:

$$A_i z_i = A_i(B_i^T s_i + e_i) = A_i e_i = H(\bar{M}\|i) \pmod q,$$

و بنابراین، وارسی موفق است. درستی الگوریتم بازگشایی به سادگی نتیجه می‌شود.

قضیه ۱.۶.۲. بگذار m و q و s به شکل توضیح داده‌شده در بالا باشند. اگر $LWE_{m,q,\alpha}$ برای $\alpha = \frac{s}{(q\sqrt{2})}$ سخت باشد، و سیستم اثبات استفاده شده، تمایزناپذیر مدرک باشد، طرح امضای توضیح داده‌شده در بالا، گمنام است. اگر $GapSVP_\gamma$ برای $\gamma = O(nlog^4 n)$ سخت باشد، طرح امضای گروهی توضیح داده‌شده در بالا دارای ردیابی است.

توجه می‌کنیم برای مقادیر s توضیح داده‌شده، سختی $LWE_{m,q,\alpha}$ به‌وسیله‌ی سختی پیدا کردن یک الگوریتم کوانتومی برای تقریب $GapSVP_{\hat{\gamma}}$ برای $\hat{\gamma} = \tilde{O}(\frac{n}{\alpha})$ [54]نتیجه‌می‌شود، بنابراین طرح پیشنهادی، می‌تواند مبتنی بر سختی یافتن یک الگوریتم کوانتومی، برای $GapSVP$ باشد. در بخش‌های زیر، گمنامی و ردیابی را اثبات می‌کنیم.

۳.۶.۲ گمنامی

عدد $N = poly(n)$ را ثابت در نظر می‌گیریم و فرض می‌کنیم $\mathcal{A}$ یک مهاجم PPT باشد. بگذار G_0 آزمایش مشخص شده در ۱.۳.۲ باشد با $b = 0$ و G_1 همان آزمایش با $b = 1$ باشد. یک سری از آزمایش‌های G_0 و G'_0 و G'_1 و G_1 را درنظر می‌گیریم. نشان می‌دهیم هر آزمایش، از آزمایش قبلی

تمایزناپذیر است. این گمنامی را نتیجه می‌دهد.
ابتدا در مورد G_0: الگوریتم تولید کلید $(1^n, 1^N)$ اجرا می‌شود و $\mathcal{A}$ کلید عمومی $pk = \left((A_i, B_i)_{i=1}^N\right)$ و کلیدهای خصوصی اعضا که به شکل $sk = (T_i)_{i=1}^N$ هستند، را دارد که هر B_i به طور آماری نزدیک به یکنواخت است و $\left((A_i, B_i)_{i=1}^N\right) \longleftarrow OrthoSamp\,(1^n, 1^m, q, B_i)$. (کلید ردیابیtk، به نوع گمنامی ـ CPA مربوط نمی‌باشد.)

در این‌صورت، $\mathcal{A}$، خروجی i_1 و i_n و M و یک امضای عضو i_0 روی M، را می‌دهد، که به شکل زیر محاسبه می‌گردد: قرار می‌دهیم $h_i = H\,(M\|r\|i)$، برای یک r تصادفی که $r \in \{0,1\}^n$. سپس e_{i_0}، به عنوان $e_{i_0} \longleftarrow GPVInvert\,(A_{i_0}, T_{i_0}, s, h_{i_0})$، محاسبه می گردد، در حالی‌که e_i برای $(i \neq i_0)$، با شرط $A_i e_i = h_i \,(mod\, q)$ به طور یکنواخت انتخاب شده است. سپس برای هر $i \in [N]$ مقدار تصادفی $s_i \longleftarrow \mathbb{Z}_q^n$ را انتخاب می‌کنیم، و مقدار $z_i = B_i^T + s_i$ را محاسبه می‌کنیم. اکنون یک اثبات π تولید شده‌است، و $\mathcal{A}$ امضای $(r, z_1, \ldots, z_N)$ را دارد.
در آزمایش مرتبط با G'_0 با توجه به G_0 یک تغییر داریم:
در اینجا، مقادیر $e_{i_0} \leftarrow GPVInvert(A_{i_0}, T_{i_0}, s, h_{i_0})$ و $e_{i_1} \leftarrow GPVInvert(A_{i_1}, T_{i_1}, s, h_{i_1})$ را محاسبه می‌کنیم. (برای $j \notin \{i_0, i_1\}$ مقدار e_j به شکل قبل محاسبه می‌گردد).

لم ۲.۶.۲. [31] اگر مساله‌ی $LWE_{m,q,\alpha}$ سخت باشد، دراین‌صورت، G_0 و G'_0 تمایزناپذیر محاسباتی هستند.

مابقی اثبات گمنامی، مشابه است. آزمایش G'_1 همان G'_0 است، با این تفاوت که اثبات π در اینجا با مدرک (s_{i_1}, i_1) محاسبه می‌گردد(به‌جای (s_{i_0}, i_0)). تمایزناپذیری مدرک سیستم اثبات نتیجه می‌دهد که، G'_1 و G'_0 تمایزناپذیر محاسباتی هستند.
تمایزناپذیری محاسباتی G'_1 و G_1 (همان آزمایش با $b = 1$)، دقیقا همانند اثبات لم قبل است.

۴.۶.۲ ردیابی

بگذار $N = poly(n)$ و $\mathcal{A}$ یک مهاجم PPT باشد(آزمایش ۱.۳.۲). یک جاعل PPT مانند $\mathcal{F}$ برای طرح امضای GPV در نظر می‌گیریم [30]، (در مدل پاسخگوی تصادفی) که احتمال موفقیت، به‌طور چندجمله‌ای مرتبط با $\mathcal{A}$ است. با توجه به اینکه طرح امضای GPV تحت فرض سختی مساله‌ی $GapSVP_\gamma$ امن است، می‌توانیم ردیابی را نتیجه بگیریم.
بدون از دست دادن کلیت فرض می‌کنیم A هرگز تمام افراد درون $[N]$ را فاسد نمی‌کند زیرا $\mathcal{A}$ در این مورد با احتمال ناچیز موفق می‌شود. (یک امضای معتبر $(r, z_1, \ldots, z_N)$ داده‌شده‌است. درستی سیستم اثبات نتیجه می‌دهد G.Open، $i \in [N]$ را خروجی می‌دهد(جز با احتمال ناچیز)). این فرض را در زیر داریم:
$\mathcal{F}$ یک کلید عمومی A برای طرح امضای GPV دارد، و با یک شاخص تصادفی $i^* \in [N]$ و ساختار $A_{i^*} = A$ آغاز می‌کند. مقدار $(B_{i^*}, S_{i^*}) \longleftarrow OrthoSamp(1^n, 1^m, q, A_{i^*})$ را محاسبه می‌کند.
برای تمام اندیس‌های باقی‌مانده‌ی $i \neq i^*$ جاعل مقادیر $(B_i, S_i) \longleftarrow TrapSamp(1^n, 1^m, q)$

و $(A_i, T_i) \longleftarrow OrthoSamp(1^n, 1^m, q, B_i)$ را با الگوریتم قانونی تولید کلید محاسبه می‌کند. بعد ازاین، $\mathcal{F}$، $pk = (A_i, B_i)_{i=1}^N$ و $tk = (S_i)_{i=1}^N$ را به $\mathcal{A}$ می‌دهد. با توجه به ۴.۵.۲ توزیع این کلیدها به توزیعی که مهاجم انتظار دارد، به‌طور آماری، نزدیک است.
$\mathcal{F}$ سوال‌های پاسخگوی تصادفی $\mathcal{A}$ را با استفاده از پاسخگوی تصادفی خود پاسخ می‌دهد. $\mathcal{F}$ به سوال‌های دیگر $\mathcal{A}$ به شکل زیر پاسخ می‌دهد:

- فاسد کردن(i): اگر $i \neq i^*$، $\mathcal{F}$ T_i را به $\mathcal{A}$ می‌دهد. اگر $i = i^*$ در این‌صورت، $\mathcal{F}$ بی‌نتیجه می‌ماند.
- امضای (i, M): اگر $i \neq i^*$، $\mathcal{F}$ با استفاده از T_i و الگوریتم امضا، امضا را محاسبه می‌کند. اگر $i = i^*$:
  - $\mathcal{F}$ r تصادفی را که $r \in \{0, 1\}^n$، انتخاب می‌کند و روی پیام $M\|r\|i^*$ از پاسخگوی امضای خود، سوال می‌کند.
  - باقی‌مانده‌ی امضا، توسط الگوریتم امضا، محاسبه می‌شود. (توجه کنید محاسبه‌ی e_{i^*} تنها بخش امضاست که بستگی به کلید خصوصی شخص i^* دارد.)

بگذار $\mathcal{C}$ مجموعه‌ی تمام شناسه‌هایی است که $\mathcal{A}$ مایل به تخریب آنها بوده‌است. (یادآوری کنیم که اگر $\mathcal{F}$ به نتیجه برسد، $i^* \notin \mathcal{C}$.) در برخی نقاط $\mathcal{A}$ یک پیام M و امضای $\sigma = (r, z_1, \ldots, z_N, \pi)$ را به‌عنوان خروجی می‌دهد. فرض کنیم $G.Vrfy(pk, M, \sigma) = 1$ و $Sign(i, M)$ هرگز برای $i \notin \mathcal{C}$ سوال نشده‌است. از آنجاکه $\mathcal{F}$ کلید ردیابی tk را دارد، مقدار $j \longleftarrow G.Open(tk, M, \sigma)$ را محاسبه می‌کند. اگر $j \neq i^*$ در این صورت $\mathcal{F}$ به نتیجه نمی‌رسد. در غیراین‌صورت، به شکل زیر انجام می‌دهد:

- S_{i^*} را برای بازیابی e_{i^*} به‌کار می‌گیرد، که:
  - $\|e_{i^*}\|_\infty \leq s\sqrt{m}$ و
  - $z_{i^*} - e_{i^*} \in \mathcal{L}(B_{i^*}^T)$.
- خروجی، عبارت جعلی $(M\|r\|i^*, e_{i^*})$ است.

ϵ را احتمال موفقیت $\mathcal{A}$ در آزمایش ۱.۳.۲ در نظر می‌گیریم. می‌توان دید که $\mathcal{F}$ با احتمال حداکثر $\frac{(N-1)}{N}$ به نتیجه نمی‌رسد، و اگر به نتیجه برسد، نتیجه‌ی $\mathcal{A}$ زمانی‌که به‌عنوان یک زیرـدست توسط $\mathcal{F}$ اجرا می‌شود، به‌طور آماری نزدیک به نتیجه‌ی او، در آزمایش شرح داده شده در ۱.۳.۲ است. بنابراین، با احتمال حداقل $\frac{\epsilon}{N}$، $\mathcal{A}$، (M, σ) را به‌عنوان خروجی می‌دهد، که $G.Vrfy(pk, M, \sigma) = 1$ و $G.Open(tk, M, \sigma) = i^*$ و $\mathcal{A}$ هرگز در مورد $Sign(i^*, M)$ سوال نکرده‌است. نشان می‌دهیم اگر شرایط فوق برقرار گردد، $\mathcal{F}$ یک جعلی معتبر را به‌عنوان خروجی می‌دهد(مگر با احتمال ناچیز).
(M, σ) را به‌گونه‌ای درنظر می‌گیریم که شرایط گفته‌شده، برقرار شود و قرار می‌دهیم $\sigma = (r, z_1, \ldots, z_N, \pi)$. از آنجاکه $G.Open(tk, M, \sigma) = i^*$، نتیجه می‌گیریم که $\mathcal{F}$ قادر به بازیابی e_{i^*} است، به‌طوری‌که

۱) $\|e_{i^*}\|_\infty \leq s\sqrt{m}$ و ۲) $z_{i^*} - e_{i^*} \in \mathcal{L}(B_{i^*}^T)$. به‌علاوه از آنجاکه ۱ $= G.Vrfy(pk, M, \sigma)$ داریم $A_{i^*} z_{i^*} = H(M\|r\|i^*)$، و از آنجاکه ۰ $= A_{i^*}(z_{i^*} - e_{i^*})$ داریم $A_{i^*} e_{i^*} = H(M\|r\|i^*)$. بنابراین e_{i^*} یک طرح امضای معتبر GPV روی پیام $M\|r\|i^*$ است. از آنجاکه $\mathcal{A}$ هرگز در مورد $Sign(i^*, M)$ سوال نکرده‌است، می‌دانیم که $\mathcal{F}$ هرگز از پاسخگوی خود برای یک امضا روی $M\|r\|i^*$ سوال نکرده‌است. این نتیجه می‌دهد که خروجی $\mathcal{F}$ یک جعلی معتبر است.

آنچه در این فصل گفته‌شد، بررسی امضاهای گروهی ساده و مشبکه‌مبنا بود که هر کدام به تفکیک از جهت اولیه‌های استفاده شده در آنها، ملزومات امنیتی، برقراری امنیت و درستی طرح‌ها، مورد بررسی قرار گرفتند.

فصل ۳

امضای حلقوی

امضای حلقوی نخستین بار توسط رایوست، شامیر و تاومان معرفی شد[56]. به‌طور کلی ساختار این امضا از دو الگوریتم شامل الگوریتم امضا و الگوریتم وارسی تشکیل شده‌است. این امضاها بدون برپایی هستند به این معنی که نیازی به الگوریتم تولید کلید ندارند. اخیرا امضاهایی پیشنهاد شده‌اند که به‌عنوان راهی برای تضمین این‌که تمام افراد یک ’’نوع’’ کلید دارند، الگوریتم تولید کلید به آنها اضافه شده‌است. در این فصل، یک نظریه‌ی امضای حلقوی ارائه شده‌است، به شکلی که شخص امضاکننده این امکان را دارد که یک مجموعه از امضاکنندگان ممکن را انتخاب می‌کند، بدون اینکه مشخص شود دقیقا کدام عضو امضا را ایجاد کرده است[56].

این امضاها برخلاف امضای گروهی، مدیر گروه ندارند، و بدون روش الغاسازی هستند. شخص می‌تواند بدون هماهنگی گروهی از امضا کنندگان ممکن را انتخاب کند که شامل خودش و امضای هر پیامی، به وسیله‌ی کلید خصوصی خودش و سایر کلیدهای عمومی است، بدون آنکه نیاز به موافقت و کمک سایرین باشد. امضای حلقوی ،راهی خوب برای نشت مخفیات مرجع در یک مسیر گمنام است به‌طوری‌که فقط توسط گیرنده‌ی نامه می‌تواند وارسی شود. هسته اصلی این پروژه ساختار جدیدی شامل امضاهایی است که به طور بی‌قیدوشرط، امضاکننده‌ـ مبهم هستند. این طرح دارای امنیت قابل اثبات در مدل پاسخگوی تصادفی است. به علاوه اضافه شدن هر فرد ارزش امضا را افزایش می دهد.

نظریه کلی طرح امضای گروهی، توسط چام و هیست در سال ۱۹۹۱ معرفی شده است[21]. در چنین طرحی مدیر گروه قابل اعتماد، از قبل اعضای گروه را مشخص می کند و کلیدهای طراحی شده را به اعضا می فرستد. اعضا می توانند از این کلیدها به‌طور گمنام، برای امضای پیام از طرف گروه استفاده کنند. امضای تولید شده به‌وسیله‌ی اعضای مختلف گروه، برای وارسی‌کننده غیرقابل تمایز به نظر می‌رسد اما

نه برای مدیر گروه که می‌تواند گمنامی اعضای بد رفتار را لغو کند.
در این بخش نظریه‌ی طرح امضای حلقوی را بررسی می‌کنیم. این طرح ساده شده‌ی طرح‌های امضای گروهی است که مدیر گروه ندارد. چنین طرحی امضای حلقوی نامیده شد زیرا حلقه‌ها ناحیه‌های هندسی با محیط یکتا و بدون مرکز هستند.

امضای گروهی زمانی مفید است که اعضا با هم همیاری دارند در حالی‌که امضاهای حلقوی زمانی مفیدند که اعضا نمی خواهند همیاری کنند. هردو امضا کننده‌ـمبهم هستند، اما در طرح امضای حلقوی، گروهی از قبل ترتیب داده شده از اعضا، روشی برای تنظیمات (کارگذاری)، تغییر یا پاک کردن گروه، راهی برای توزیع کلیدهای تخصیص یافته و راهی برای لغو گمنامی امضا کننده‌ی حقیقی وجود ندارد (جز اینکه خودش بخواهد خود را افشا کند). تنها فرض ما این است که هر عضو مرتبط با کلید عمومی یک طرح امضای استاندارد مانند RSA است. برای تولید یک امضای حلقوی، ''امضاکننده حقیقی'' یک مجموعه از امضاکنندگان ممکن را احضار می کند که شامل خودش نیز هست و امضاهایی را محاسبه می کند که کاملا توسط خودش با استفاده از کلید خصوصی خود و کلید عمومی سایر اعضا تولید شده‌است. ممکن است بقیه امضاکنندگان کلیدهای RSA خود را تنها به منظور هدایت تجارت الکترونیکی اینترنت انتخاب نمایند و کاملا بی خبر باشند که کلیدهای عمومی آنها توسط شخصی، برای تولید چنین امضای حلقوی، روی پیامی که آنها هرگز نمی‌بینند و نمی‌خواستند آن را امضا کنند، استفاده شده‌است.

ساختار مستقیم امضاهای حلقوی که در این پروژه پیشنهاد شده است بر مبنای ایده‌ای کاملا متفاوت نسبت به کارهای گذشته است، و به‌طور چشم‌گیری برای حلقه‌های بزرگ (با اضافه کردن یک ضرب پیمانه‌ای و یک رمزنگاری متقارن برای هر عضو حلقه، هم برای تولید و هم برای وارسی چنین امضاهایی) کارا است. امضاهای منتج، به‌طور بی قیدوشرط امضا کننده مبهم و دارای امنیت قابل اثبات در مدل پاسخگوی تصادفی هستند.

۱.۳ تعاریف و کاربردها

۱.۱.۳ مفهوم امضای حلقوی

اصطلاحات: یک مجموعه از امضاکنندگان ممکن را یک حلقه می‌خوانیم. عضو حلقه را که امضای حقیقی تولید کرده‌است، امضاکننده و اعضای دیگر غیر امضاکننده نامیده می‌شوند. فرض می‌کنیم هر امضاکننده‌ی ممکن، مرتبط با یک کلید عمومی است که امضا و کلید وارسی شخص را تعیین می‌کند. کلید خصوصی متناظر (که برای تولید امضاهااستفاده می‌شود) با S_k نشان داده‌می‌شود.
امضای حلقوی در دو مرحله‌ی زیر تعریف می‌شود :

- **حلقه‌ـامضا** $(m, P_۱, P_۲, \ldots, P_r, s, S_s)$ که با فرض کلیدهای عمومی $P_۱, P_۲, \ldots, P_r$ برای هر r عضو حلقه، به همراه کلید خصوصی S_s برای عضو sـام که امضاکننده‌ی حقیقی است، تولید امضا می‌کند.

- **حلقه‌ـ‌وارسی** (m, σ) که پیام m و امضای σ (که شامل کلیدهای عمومی تمام امضاکننده‌های ممکن است)، وارسی می‌کند و خروجی ''درست'' یا ''غلط'' می‌دهد.

امضای حلقوی بدون برپایی است. امضاکننده نیازی به داشتن رضایت و یا کمک سایر اعضای حلقه برای قراردادن آنها در حلقه ندارد. تنها چیزی که نیاز دارد، دانستن کلیدهای عمومی آنهاست. اعضای متفاوت می‌توانند از طرح‌های امضای کلید عمومی مستقل مختلف، با کلیدهای متفاوت و اندازه‌های مختلف امضا استفاده کنند. لگوریتم وارسی باید شرایط ''تمامیت'' و ''درستی'' را داشته‌باشد، به علاوه تمایل داریم که طرح امضا، امضا کننده‌ـ‌مبهم باشد، به این معنی که وارسی‌کننده نتواند شناسه‌ی امضاکننده‌ی اصلی در حلقه‌ای با r عضو را با احتمالی بهتر از $\frac{۱}{r}$ پیدا کند. این گمنامی می‌تواند به صورت محاسباتی یا بی‌قیدوشرط باشد. ساختاری که ارائه می‌دهیم، دارای گمنامی بی‌قیدوشرط است. به این معنی که مهاجم، با توان محاسباتی بی‌نهایت، با دسترسی به بی‌نهایت امضای متن انتخابی، که توسط اعضای همان حلقه تولید شده‌اند، نمی‌تواند شناسه‌ی امضاکننده را حدس بزند، به علاوه نمی‌تواند امضاهایی که بعدا اضافه می‌شود را به همان امضاکننده ارتباط دهد.

۲.۱.۳ نشت راز

فرض کنید ''باب'' یکی از اعضای کابینه‌ی دولتی است. او می‌خواهد اطلاعات مهمی را در مورد فرار شخصی به نام ''پرایم'' [۱] به روزنامه‌نگار نشت دهد، به طوری‌که گمنام بماند اما روزنامه‌نگار مطمئن باشد که خبر از طرف یکی از اعضای کابینه است. باب نمی‌تواند به روزنامه‌نگار پیامی از طریق یک گمنام‌ساز [۲] بفرستد، زیرا گمنام‌ساز تمام منابع تشخیص هویت و احراز اصالت را از بین خواهد برد: روزنامه‌نگار دلیلی ندارد تا باور کند که پیام حقیقتا از سوی اعضای کابینه فرستاده شده‌است.
یک طرح امضای گروهی استاندارد این مشکل را حل نمی‌کند، زیرا نیاز به هم‌یاری از پیش، بین اعضای گروه برای برپایی دارد و باب را در معرض خطر شناسایی توسط مدیر گروه، قرار می‌دهد. رویکرد درست برای باب، فرستادن اطلاعات از طریق یک گمنام‌ساز است که از طریق یک امضای حلقوی که هر عضو کابینه (شامل خودش) به عنوان عضوی از حلقه هستند، امضا شده‌است. روزنامه‌نگار می‌تواند امضای حلقوی روی پیام را وارسی کند و مطمئن شود که از سوی یکی از اعضای کابینه است. روزنامه‌نگار می‌تواند حتی امضای حلقوی را روی شبکه‌ی خود بگذارد تا خوانندگان اطمینان حاصل نمایند که اطلاعات از منبع موثقی است. اگرچه خواننده نمی‌تواند تشخیص دهد که منبع اصلی رسوخ کجاست و گوینده‌ی خبر به‌طور کامل محافظت می‌شود.

---

[۱] Prime Minister

[۲] Anonymizer. ابزارهایی که به یک کاربر امکان می‌دهد تا به صورت گمنام به شبکه وارد شود و به انجام عملیات بپردازد.

۳.۱.۳ طرح امضای وارسی‌کننده‌ی مشخص

طرح امضای وارسی‌کننده‌ی مشخص[۳]، طرح امضایی است که امضاها فقط توسط یک ''وارسی‌کننده‌ی مشخص'' وارسی می‌شوند، که این شخص توسط امضاکننده انتخاب می‌شود. این مفهوم نخستین بار توسط ژاکوبسون و ایمپگلازو مطرح شد[36]. نوعی از کاربرد این طرح به‌عنوان مثال، در شرایطی است که دو کمپانی قصد دارند که طرح قراردادهای پیشنهاد شده را مبادله کنند. آنها می‌خواهند به هر ایمیل یک احرازگر اصالت[۴] اضافه کنند، اما نه به‌عنوان یک امضای حقیقی که بتوان به یک شخص سوم نمایش داد، بلکه به عنوان اثبات که طرح مذکور توسط کمپانی دیگر پیشنهاد شده‌است. طرح وارسی‌کننده‌ی مشخص، می‌تواند به عنوان یک طرح امضای ساده در نظر گرفته شود که می‌تواند پیام را برای گیرنده‌های مقصد، احراز اصالت کند بدون اینکه خاصیت انکارناپذیری[۵] را داشته‌باشد.

یک راه استفاده از اثبات‌های هیچ‌آگاهی تعاملی است که می‌تواند وارسی‌کننده را قانع کند. از آنجاکه اثبات‌های تعاملی نیازمند تعامل بین اثبات‌کننده و وارسی‌کننده است، (که این با توجه به خاصیت ایمیل استاندارد و گمنام‌سازها دشوار است)، از اثبات‌های هیچ‌آگاهی غیرتعاملی می‌توان استفاده کرد. اما در این‌صورت احرازگرهای اصالت تبدیل به امضاهایی می‌شوند که می‌توان به شخص سوم، نمایش داد. راه دیگر، موافقت بر سر تسهیم کلید متقارن خصوصی k است، که برای احراز اصالت هر طرح قرارداد، با اضافه کردن یک کد احراز اصالت (MAC)[۶] به پیش‌نویسی که با کلید k محاسبه شده‌است، مشکل را حل می‌کنیم. اگر قرار باشد به یک شخص سوم نشان دهیم، باید کلید خصوصی را برای تأیید اعتبار یک MAC، به او بدهیم، در حالی‌که، او نمی‌داند کدام کمپانی MAC را محاسبه کرده‌است. اما این روش به یک روش برپایی آغازین نیاز دارد که ما هنوز هم در آن با مشکل احراز اصالت ایمیلی که کلید k، را بدون استفاده از امضا برای شخص مقابل می‌فرستد، مواجهیم.

طرح وارسی‌کننده‌ی مشخص، یک راه حل برای حل این مشکل فراهم کرده‌است:

کمپانی A، می‌تواند هر پیش‌نویسی را که می‌فرستد امضا کند. وارسی‌کننده مشخص را کمپانی B، درنظر می‌گیریم. این به آسانی به‌وسیله امضای حلقوی با درنظر گرفتن کمپانی A و B به عنوان اعضای حلقه، به‌دست می‌آید. فقط با یک MAC، کمپانی B پی می‌برد که پیام از سمت کمپانی A آمده است (از آنجا که هیچ شخص سومی نمی‌تواند چنین امضای حلقوی تولید کند)، اما کمپانی B نمی‌تواند به فرد دیگری ثابت کند که پیش‌نویس قرارداد، توسط کمپانی A امضا شده بود، چون کمپانی B می توانست خودش این پیش‌نویس را تولید کند(زیرا هم کمپانی A و هم کمپانی B به کلید k دسترسی دارند). برخلاف مورد MAC، این طرح از سیستم رمزنگاری کلید عمومی استفاده کرده‌است و بنابراین A، می‌تواند در صورت لزوم و بدون درخواست B، برای او، ایمیل بفرستد که با امضای حلقوی بدون هرگونه مقدمات، تعامل یا تعویض کلید خصوصی، امضا شده‌است. با استفاده از طرح امضای حلقوی پیشنهاد ی ما، می‌توانیم طرح امضای استاندارد را تبدیل به طرح‌های وارسی‌کننده‌ی مشخص کنیم که می‌تواند به هر سیستم ایمیلی اضافه شود، بدون تقریبا هیچ هزینه اضافی.

---

[۳] Designed Verifier Signature Schemes
[۴] Authenticator
[۵] Nonrepudation
[۶] Message Authentication Code

۲.۳ طرح امضای حلقوی پیشنهاد شده نسخه‌ی RSA

فرض کنید آلیس می‌خواهد پیامی را با امضای حلقوی که شامل r عضو $A_r, \dots A_1$ است، امضا کند که امضاکننده، آلیس، را A_s درنظرمی‌گیریم. برای برخی مقادیر s که $1 \leq s \leq r$. ابتدا یک طرح امضای حلقوی را توضیح می‌دهیم که تمام اعضای حلقه، به‌عنوان طرح امضای شخصی، از RSA استفاده می کنند. همین ساختار می‌تواند برای هر جایگشت یک طرفه‌ی دریچه‌دار، استفاده شود، اما مجبور به کمی تغییر به منظور استفاده از توابع یک طرفه‌ی دریچه‌دار (برای مثال، طرح امضای رابین[53])، هستیم.

۱.۲.۳ جایگشت‌های دریچه‌داررRSA

هر عضو حلقه A_i، یک کلید عمومی RSA، $p_i = (n_i, e_i)$، را دارد که جایگشت یک طرفه‌ی f_i از Z_{n_i} را معین می‌کند:

$$f_i(x) = x^{e_i} \quad (mod\ n_i)$$

فرض می‌کنیم که فقط A_i می‌تواند وارون جایگشت f_i^{-1} را با استفاده از اطلاعات دریچه، محاسبه کند، این مدل اصلی دیفی‌ـ هلمن[25] برای سیستم های رمزنگاری کلید عمومی است.

بسط جایگشت‌های دریچه‌دار به یک دامنه‌ی رایج

پارامترهای جایگشت‌های دریچه‌دار RSA برای اعضای متفاوت حلقه، دامنه‌های اندازه های مختلف دارند (حتی اگر همه‌ی n_i ها تعداد بیت‌های یکسان داشته‌باشند). بنابراین ما تمام جایگشت‌های دریچه‌دار را به داشتن یک دامنه‌ی رایج از مجموعه‌ی یکسان $\{0,1\}^b$ بسط می‌دهیم که 2^b توانی از ۲ است که از تمام n_i ها بزرگ‌تر است. برای هر جایگشت دریچه‌دار f_i روی Z_{n_i} جایگشت دریچه‌دار بسط داده شده‌ی g_i روی $\{0,1\}^b$ را به روش زیر تعریف می‌کنیم:
برای هر بیت b، ورودی m را برای اعداد صحیح غیر منفی q_i و r_i به شکل زیر تعریف می‌کنیم:

$$m = q_i n_i + r_i \quad 0 \leq r_i \leq n_i$$

سپس داریم:

$$g_i(m) = \begin{cases} q_i n_i + f_i(r_i); & (q_i + 1)n_i \leq 2^b \\ m & else \end{cases}$$

به‌طور خلاصه، برای یک اندازه کردن تمام کلیدهای عمومی، از تابع g، استفاده کردیم. تابع g_i یک جایگشت روی $\{0,1\}^b$ است و یک جایگشت یک‌طرفه‌ی دریچه‌دار است زیرا تنها کسی وارون f_i را می‌داند که، وارون $g_i(m)$ را با احتمالی بیشتر از یک کسر ناچیز ورودی‌های ممکن، می‌داند.

۲.۲.۳ رمزگذاری متقارن

E الگوریتم رمزگذاری متقارن است اگر برای هر کلید k از طول l، تابع E_k، یک جایگشت روی رشته‌های bـ بیتی است. از پاسخگوی (جایگشت) تصادفی استفاده می‌کنیم که فرض آن بر این است که همه‌ی

بخش‌ها دسترسی به پاسخگویی دارند و پاسخ‌های تصادفی درست، برای سؤال‌هایی به شکل $E_k(x)$ و $E_k^{-1}(y)$ ارائه می‌دهد که با توجه به جواب‌های قبلی منطقی هستند و با نیاز به اینکه $E_k(x)$ یک جایگشت باشد [45].

۳.۲.۳ توابع چکیده‌ساز

تابع چکیده‌ساز مقاوم در برابر تصادم h، ورودی‌های دلخواه را به رشته‌هایی از طول l، نگاشت می‌کندکه به عنوان کلیدهای E استفاده می‌شوند. h را به‌عنوان پاسخگوی تصادفی مدل می‌کنیم(h لزومی ندارد که جایگشت باشد، سؤال‌های مختلف ممکن است پاسخ‌های یکسان داشته باشند و به سؤال در مورد h^{-1} اجازه پاسخ نمی‌دهیم).

۴.۲.۳ توابع ترکیبی

یک خانواده از توابع ترکیبی کلیددار مانند $C_{k,v}(y_1,\ldots,y_r)$ تعریف می‌کنیم که کلید k، یک مقدار آغازین v و مقادیر دلخواه $y_1,\ldots,y_r$ از $\{0,1\}^b$ را به‌عنوان ورودی دریافت می‌کند. چنین تابع ترکیبی، از E_k به عنوان یک زیرـروش استفاده می‌کند و به عنوان خروجی مقدار z از $\{0,1\}^b$ را تولید می‌کند به‌طوری که برای هر مقدار ثابت k و v خواص زیر را داریم:

۱. جایگشت روی هر ورودی: برای هر s، $1 \le s \le r$، و برای هر مقدار ثابت ورودی‌های دیگر y_i که $i \neq s$، تابع $C_{k,v}$ یک نگاشت یک به یک از y_s به خروجی z است.

۲. کارایی قابل حل برای هر ورودی تنها: برای هر s، که $1 \le s \le r$، مقدار یک z ـ bـبیتی و مقادیر برای تمام ورودهای y_i به جز y_s داده شده‌است، در این صورت، یافتن یک مقدار bـبیتی برای y_s که $C_{k,v}(y_1,\ldots,y_r)=z$، به‌طور کارایی، ممکن است.

۳. حل نشدن معادله‌ی وارسی برای همه ورودی ها به جز دریچه‌ها: k و r و z داده شده اند. برای یک مهاجم، حل کردن معادله‌ی

$$C_{k,v}(g_1(x_1),g_2(x_2)\ldots,g_r(x_r))=z \qquad (۱.۳)$$

برای $x_1,\ldots,x_r$ نشدنی است، (با وجود دسترسی به هر g_i و E_k)، اگر مهاجم نتواند هر تابع دریچه‌دار $g_1,\ldots,g_r$ را وارون کند.

برای مثال تابع $C_{k,v}(y_1,\ldots,y_r)=y_1\oplus\cdots\oplus y_r$ (که $\oplus$ عملگر یای‌ـانحصاری روی کلمات bـبیتی است)، در دو خاصیت اول صدق می‌کند و می‌تواند به عنوان نماینده‌ی تابع ترکیبی درنظر گرفته شود. اما در شرط سوم شکست می‌خورد زیرا برای هر انتخاب جایگشت‌های یک‌طرفه‌ی دریچه‌دار g_i ممکن است با استفاده از جبرخطی برای r به اندازه کافی بزرگ، یک جواب برای $x_1,\ldots,x_r$ بدون وارون هر g_i بیابد. ایده‌ی پایه‌ای حمله، انتخاب یک مقدار تصادفی برای هر x_i و محاسبه هر $y_i=g_i(x_i)$ در

یک مسیر رو به جلو است. اگر تعداد مقادیر r از تعداد b، تجاوز کرد ما با احتمال زیادی می‌توانیم زیرمجموعه‌ای از رشته‌های y_i پیدا کنیم که XOR [۷] هدف z ـ bـبیتی مطلوب، است. اگرچه هدف ما نشان داد z به عنوان XOR تمام مقادیر $y_1, \ldots, y_r$ است، نه اینکه نشان دهیم z به عنوان XOR یک زیرمجموعه تصادفی این مقادیر است.

برای حل این مشکل برای هر i، دو مقدار تصادفی x'_i و x''_i را انتخاب می‌کنیم و متناظر آنها $y'_i = g_i(x'_i)$ و $y''_i = g_i(x''_i)$ را محاسبه می‌کنیم. برای هر i، $y'''_i = y'_i \oplus y''_i$ را تعریف می‌کنیم و مقدار هدف را به $z' = z + y'_1 \oplus \cdots \oplus y'_r$ اصلاح می‌کنیم. از الگوریتم قبلی برای نشان دادن اینکه z' به عنوان XOR یک زیر مجموعه تصادفی مقادیر y'''_i است، استفاده می‌کنیم. سپس، یک z به‌عنوان نماینده‌ی اصلی می‌گیریم، به منظور XOR یک مجموعه r مقداری که دقیقا از هر جفت (y'_i, y''_i) یک مقدار انتخاب شده است. با انتخاب مقدار متناظر x'_i یا x''_i می‌توانیم معادله‌ی وارسی را بدون وارون هر جایگشت یک‌طرفه‌ی دریچه‌دار g_i حل کنیم. (یک مقابله با این حمله که در اینجا توضیح نمی‌دهیم این است که اجازه دهیم b با r رشد کند).

در بدترین مسائل می‌توان چهره‌ی دیگری از وجود توابع ترکیبی مانند اضافه کردن پیمانه‌ی 2^b را نشان داد. فرض کنید از توابع دریچه‌دار RSA $g_i(x_i) = x_i^3 \pmod{n_i}$، که تمام n_iها سایز b دارند استفاده می‌کنیم. مشخص است[35] که هر عدد صحیح نامنفی z را می‌توان به عنوان مجموع دقیقا ۹ مکعب صحیح نامنفی $x_1^3 + \cdots + x_9^3$ نوشت. اگر z هدف، b ـ بیتی باشد، انتظار داریم هر x_i^3 کمی کوتاه‌تر از z باشد و بنابراین به احتمال زیاد کاهش هر x_i^3 به پیمانه‌ی n_i متناظر b ـ بیتی، موثر نیست. در نتیجه می‌توانیم معادله‌ی وارسی

$$(x_1^3 mod n_1) + (x_2^3 mod n_2) + \cdots + (x_9^3 mod n_9) = z (mod 2^b)$$

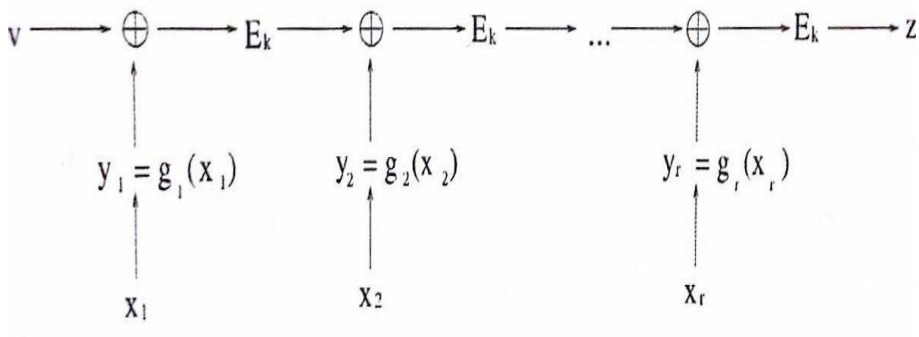
را با ۹ جایگشت RSA، بدون وارون هر یک از آنها حل کنیم.

تابع ترکیبی پیشنهاد شده ما، تابع رمزنگاری متقارن E_k را به صورت زیر به‌کار گرفته‌است:

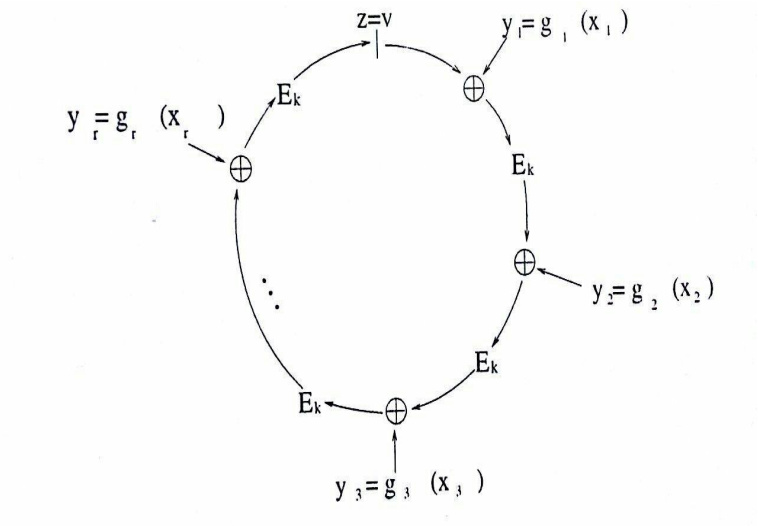
$$C_{k,v}(y_1, \ldots, y_r) = E_k(y_r \oplus E_k(y_{r-1} \oplus E_k(y_{r-2} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v) \ldots)))).$$

این تابع برای دنباله‌ی $(y_1, \ldots, y_r)$ که $y_i = g_i(x_i)$ به‌کار گرفته شده‌است. همان‌طور که در شکل ۱.۳ نشان داده شده‌است تابع نتیجه شده دارای امنیت قابل اثبات در مدل پاسخگوی تصادفی است. به طور واضح این یک جایگشت روی هر ورودی است زیرا XOR، g_i و E_k توابع جایگشتی هستند. به علاوه برای هر ورودی تنها قابل حل است، زیرا آگاهی از k، این امکان را فراهم می‌آورد، در یک حرکت رو به جلو با شروع از v آغازین و برگشت به عقب از z نهایی به منظور محاسبه‌ی هر مقدار از دست رفته‌ی y_i، عمل کنیم. این تابع می‌تواند برای وارسی امضاها با استفاده از یک نوع چکیده‌شده‌ی m برای انتخاب کلید متقارن k استفاده شود و ملزم می‌کند که خروجی z، مساوی ورودی v باشد. این شرط ثبات $C_{k,v}(y_1, \ldots, y_r) = v$، سبب خمیده شدن خطوط به شکل حلقه شده است که در شکل ۲.۳ نشان داده شده است.

[۷] Exclusive Or



شکل ۱.۳: یک شکل از تابع ترکیبی پیشنهادی



شکل ۲.۳: امضاهای حلقوی

یک امضای حلقوی فشرده‌تر می‌تواند با انتخاب همیشه‌ی ۰ به عنوان ”مقدار چسب“ [۸] v بدست آید. این تغییر امن است، اما حلقه‌ی مجموع متقارن پیشنهادی خود را ترجیح می‌دهیم.
هم اکنون تولید امضا و روش وارسی را به شکل رسمی توضیح می‌دهیم.

۵.۲.۳ تولید یک امضای حلقوی

پیام m مفروض است و توسط کلید خصوصی S_s امضا شده است و دنباله‌ی $p_1, \ldots, p_r$ کلیدهای عمومی تمام اعضا است. امضاکننده به شکل زیر امضای حلقوی را تولید می‌کند:

۱. انتخاب یک کلید: ابتدا امضاکننده کلید k را به عنوان تابع چکیده‌ساز پیام m محاسبه می‌کند:

$$k = h(m)$$

[۸] glue value

(محاسبه‌ی پیچیده‌تر k محاسبه‌ی $h(m, p_1, \ldots, p_r)$ است، گرچه ساختار ساده‌تر هم امن است.)

۲. انتخاب یک مقدار چسب تصادفی: امضاکننده یک مقدار آغازین(چسب) به‌طور یکنواخت و به تصادف از $\{0,1\}^b$ انتخاب می‌کند.

۳. انتخاب تصادفی x_i: امضاکننده، برای تمام اعضای دیگر گروه $(1 \leq i \leq r \; i \neq s)$ به‌طور یکنواخت و مستقل از $\{0,1\}^b$، x_i تصادفی انتخاب می‌کند و $y_i = g_i(x_i)$ را محاسبه می‌کند.

۴. حل برای y_s: امضاکننده معادله‌ی حلقوی زیر را برای y_s حل می‌کند:

$$C_{k,v}(y_1, \ldots, y_r).$$

با توجه به فرض، مقادیر دلخواه برای ورودی‌های دیگر داده‌شده‌است. یک مقدار یکتا برای y_s وجود دارد که در معادله صدق می‌کند و می‌تواند به صورت عملی، محاسبه شود.

۵. وارون جایگشت دریچه‌دار امضاکننده: امضاکننده از دانش خود در مورد دریچه به منظور وارون g_s روی y_s برای مشخص کردن x_s استفاده می‌کند:

$$x_s = g_s^{-1}(y_s)$$

۶. خروجی امضا: امضا روی پیام m، $(2r+1)$ـتایی است:

$$(P_1, P_2, \ldots, P_r; v; x_1, \ldots, x_r)$$

وارسی امضای حلقوی

وارسی کننده می‌تواند یک امضای ارائه‌شده‌ی $(P_1, P_2, \ldots, P_r; v; x_1, \ldots, x_r)$ را روی پیام m، به شکل ذیل وارسی کند:

- به‌کارگیری جایگشت‌های دریچه‌دار: برای $i = 1, \ldots, r$ مقدار زیر را محاسبه می‌کند:

$$y_i = g_i(x_i)$$

- به‌دست آوردن k: وارسی کننده پیام m را برای محاسبه‌ی کلید رمزگذاری k، تحت چکیده‌ساز h به‌کار می‌گیرد:

$$k = h(m)$$

- وارسی کردن معادله‌ی حلقوی: وارسی کننده بررسی می‌کند آیا y_s در معادله‌ی بنیادی صدق می‌کند:

$$C_{k,v}(y_1, y_2, \ldots, y_r) = y_s \qquad (۲.۳)$$

اگر در معادله‌ی حلقوی ۲.۳ صدق کرد، وارسی کننده قبول می‌کند امضا معتبر است، در غیر این‌صورت، رد می‌کند.

۶.۲.۳ امنیت

با طرح امضای حلقوی پیشنهادی، شناسه‌ی امضاکننده، به‌طور بی‌قیدوشرط، حفظ می‌شود. برای بررسی این گفته، توجه کنید برای هر k و v، معادله‌ی حلقوی دقیقا $(۲^b)^{(r-۱)}$ جواب دارد، و تمام جواب‌ها می‌توانند توسط روش تولید امضا، با احتمال مساوی، صرف‌نظر از شناسه‌ی امضاکننده، انتخاب شوند.

اکنون هدف ما نشان دادن این است که در مدل پاسخگوی تصادفی، هر الگوریتم جعل A، که می‌تواند با احتمال غیرناچیزی یک امضای حلقوی جدید برای m، (با بررسی برخی امضاهای حلقوی دیگر برای پیام‌های $m_j \neq m$) تولید کند، می‌تواند به یک الگوریتم B تبدیل شود که یکی از توابع یک‌طرفه‌ی دریچه‌دار g_i، روی ورودی تصادفی y با احتمال غیرناچیز، را وارون کند.

الگوریتم A کلیدهای عمومی $P_۱, \ldots, P_r$ را قبول می‌کند(اما نه هر کلید خصوصی متناظر را)، و دسترسی پاسخگویی به h، E و $E^{-۱}$ و به یک پاسخگوی امضای حلقوی دارد. A می‌تواند به صورت وفقی[۹] کار کند، به این معنی که سوال کردن از پاسخگو، بستگی به سوال‌های قبلی دارد. در نهایت، ممکن است یک امضای حلقوی معتبر روی یک پیام جدید که توسط پاسخگوی امضا، ارائه شده‌بود، با یک احتمال غیرناچیز(روی پاسخ‌های تصادفی پاسخگوها) ایجاد کند.

الگوریتم B از الگوریتم A به‌عنوان یک جعبه‌ی سیاه استفاده می‌کند، اما کنترل کامل روی پاسخگوی آن دارد. A از پاسخگو در مورد تمام رمزگذاری‌های متقارن، برای امضای حلقوی جعلی روی m، سوال می‌کند(در غیر این‌صورت احتمال صدق کردن در معادله‌ی حلقوی ناچیز است). بدون از دست دادن کلیت می‌توان فرض کرد هر کدام از r رمزگذاری‌های متقارن سوال شده هر کدام یک‌بار "در جهت عقربه‌های ساعت" E_k یا "در جهت عکس عقربه‌های ساعت" $E_k^{-۱}$ (اما نه در هر دو جهت)، سوال شده‌اند. زمانی که A سوال‌های زیادی از E_k و $E_k^{-۱}$ با کلیدهای مختلف $k = h(m)$ به‌دست آورده‌است، B می‌تواند با احتمال غیرناچیز حدس بزند کدام k گرفتار جعل شده‌است، اما نمی‌تواند حدس بزند کدام زیر مجموعه از r سوال، در جعل نهایی استفاده خواهد شد و در معادله‌ی حلقوی صدق خواهدکرد، زیرا احتمال‌های زیادی وجود دارد.

الگوریتم B می‌تواند پاسخگوی امضای حلقوی را برای همه‌ی m_jهای دیگر، به‌وسیله‌ی تهیه‌ی بردارهای تصادفی $(v, x_۱, \ldots, x_r)$ به‌عنوان امضاهای حلقوی آنها، شبیه‌سازی کند . هم‌چنین می‌تواند پاسخ‌های تصادفی برای سوال‌های به فرم $E_{h(m_j)}$ و $E_{h(m_j)}^{-۱}$ تنظیم کند به‌طوری‌که در معادله‌ی حلقوی این پیام‌ها، صدق کند. توجه کنید A نمی‌تواند سوال‌هایی از پاسخگو بپرسد که آزادی انتخاب B را در رابطه با m_j که می‌خواهد از پاسخگوی امضا سوال کند، محدود کند، زیرا تمام مقادیر در طول امضای حلقوی(شامل v)، توسط B به‌طور تصادفی، انتخاب شده‌اند، و نباید از قبل توسط A حدس زده شود. به‌علاوه از آنجا که h مقاوم در برابر تصادم است، سوال‌های E و $E^{-۱}$ با کلید $k_j = h(m_j)$ تاثیری در پاسخ‌های سوال‌های E و $E^{-۱}$ با کلید $k = h(m)$ که در جعل نهایی استفاده خواهد شد، ندارد، زیرا از کلیدهای متفاوت استفاده کرده‌اند.

هدف الگوریتم B محاسبه‌ی $x_i = g_i^{-۱}(y)$ (برای برخی i)، برای ورودی‌های y تصادفی با احتمال

[۹] Adaptively

غیرناچیز است. این، امنیت امضای حلقوی را به امنیت امضای شخصی کاهش می‌دهد. ایده‌ی پایه‌ای برای کاهش‌سازی، از قلم انداختن این y تصادفی، به‌عنوان یک ''گسست''[۱۰]، بین ورودی و خروجی دو E متوالی گردشی در میان معادله‌ی حلقوی جعل نهایی است، که A را مجبور به بستن شکاف به‌وسیله‌ی تهیه‌ی x_i متناظر، در امضای تولید شده می‌کند. توجه کنید y یک مقدار تصادفی است که B می‌داند، اما A نه، و بنابراین A نمی‌تواند ''دریچه را تشخیص دهد''، و امضای پیام متناظر را رد کند.
مشکل اصلی این است که A می‌تواند شکاف بین مقادیر E را نه تنها با وارون توابع یک‌طرفه‌ی دریچه‌دار، بلکه با ارزیابی این توابع در جهت رو به جلو، پر کند(همان‌طور که توسط امضاکننده‌ی اصلی در تولید امضای حلقوی انجام می‌شود). برای حل این مشکل، در هر امضای حلقوی تولیدی A، باید یک گسست، حدودا جایی بین دو برخورد پی‌درپی گردشی E، که سوال‌ها محاسبه شده‌اند، ایجاد گردد، در یکی از سه روش زیر:

- پاسخگو برای iـامین E، در جهت عقربه‌های ساعت و برای $i+۱$ـامین E، در جهت عکس عقربه‌های ساعت سوال خواهدشد.
- هر دو E در جهت عقربه‌های ساعت سوال شوند، اما iـامین E بعد از $i+۱$ـامین E سوال شده‌است.
- هر دو E، در جهت عکس عقربه‌های ساعت سوال شده‌اند، اما iـامین E قبل از $i+۱$ـامین E سوال شده‌است.

در هر سه مورد، B می‌تواند یک جواب تصادفی برای سوال بعدی آماده کند که بر مبنای دانش او از ورودی و خروجی سوال قبلی است، به‌طوری‌که XOR مقادیر گسست، y خواسته‌شده‌است. این A را مجبور به محاسبه‌ی $g_i^{-۱}(y)$ می‌کند، که این گسست را در آخرین امضای حلقوی پر کند.
B نمی‌داند کدام سوال‌ها، سوال‌های پی‌درپی گردشی در امضای حلقوی جعلی خواهند شد، بنابراین او باید شناسه‌ها را حدس بزند، و از آنجاکه باید ''دو'' حدس بزند، پس احتمال درستی حدس $\frac{۱}{Q^۲}$ است که Q تمام سوال‌های پرسیده شده توسط جاعل A است. به همین ترتیب B، $g_i^{-۱}(y)$ را برای یک y تصادفی و برخی i، با احتمال غیرناچیز، محاسبه می‌کند.
زمانی که توابع یک‌طرفه‌ی دریچه‌دار g_i توابع RSA هستند، می‌توانیم کمی نتیجه را قوی‌تر کنیم: از آنجا که RSA هم‌ریخت[۱۱] است، می‌توانیم y را به‌وسیله‌ی محاسبه‌ی $y' = y * t^e \,(mod\, n_i)$ برای یک t انتخاب شده‌ی تصادفی، تصادفی کنیم. با استفاده از y' به‌جای y می‌توان نشان داد جعل‌های فوق امضاهای حلقوی، می‌توانند برای بیرون کشیدن ریشه‌های پیمانه‌ای از اعداد خاص، مانند $y = ۲$ و نه فقط از ورودی‌های تصادفی y، استفاده شوند. این ضرورتا برای توابع دریچه‌دار دیگر درست نیست، زیرا جاعل A، می‌تواند به‌طور عمدی تصمیم بگیرد هیچ جعلی تولید نکند، که یکی از گسست‌ها بین توابع E پی‌درپی گردشی ۲ شود.

[۱۰] Gap
[۱۱] Homomorphic

۳.۳ طرح امضای حلقوی پیشنهاد شده نسخه‌ی رابین

سیستم رمزنگاری رابین، یک سیستم رمزنگاری کلیدعمومی است که مانند RSA، امنیت این سیستم بستگی به تجزیه‌ی اعداد دارد. این سیستم مانند هر سیستم رمزنگاری دارای الگوریتم‌های تولید کلید، رمزگذاری و رمزگشایی است. به شکل زیر:

- تولید کلید: در این سیستم، کلید عمومی، برای رمزگذاری کاربرد دارد و به وضوح، می‌تواند منتشر شود. کلید خصوصی، تنها نزد گیرنده‌ی پیام، می‌ماند. الگوریتم تولید کلید به شکل زیر کار می‌کند:
  - دو عدد اول بزرگ p و q انتخاب کنید. برای سادگی استفاده، ممکن است کسی دوعدد را به‌گونه‌ای انتخاب کند که $p = q = ۳ \ (mod\, ۴)$. این الگوریتم با هر عدد اولی کار می‌کند.
  - بگذار $n = pq$ که n کلید عمومی و p و q کلیدهای خصوصی هستند.
- رمزگذاری: برای رمز گذاری فقط کلید عمومی کاربرد دارد:
بگذار $P = \{۰, \ldots, n-۱\}$ فضای متن‌های اصلی باشد، و $m \in P$ متن اصلی ما باشد. در این صورت متن رمزشده، به شکل زیر به‌دست می‌آید:

$$c = m^۲ \ (mod\, n)$$

- برای رمزگشایی متن رمزشده، وجود کلید خصوصی ضروری است:
با داشتن c و n، متن اصلی $m \in \{۰, \ldots, n\}$ است که $c = m^۲ \ (mod\, n)$. اگر n مرکب باشد(مانند همین سیستم)، روشی برای یافتن m وجود ندارد. اما در شرایطی که n عددی اول باشد(مانند p و q)، قضیه‌ی باقی‌مانده‌ی چینی، m را پیدا می‌کند. پس داریم:

$$m_p = \sqrt{c} \ (mod\, p)$$

و

$$m_q = \sqrt{c} \ (mod\, q).$$

سیستم رمزنگاری کلیدعمومی رابین[53]، نسبت به RSA وارسی موثرتری دارد، زیرا سرعت بالاتری دارد. گرچه ما با این حقیقت که نگاشت رابین، $f_i(x_i) = x_i^۲ \ (mod\, n_i)$، یک جایگشت روی $\mathbb{Z}_n^*$ به دلیل یک‌به‌یک و پوشا نبودن، نیست، سروکار داریم و بنابراین فقط یک چهارم پیام‌ها می‌توانند امضا شوند، و پیام‌هایی که امضا می‌شوند امضاهای چندگانه دارند.
در زمان امضا اگر $g_s^{-۱}(y_s)$ تعریف شده‌باشد، آخرین انتخاب تصادفی $x_{s-۱}$ را تغییر می‌دهیم. از آنجاکه تنها یک تابع یک‌طرفه‌ی دریچه‌دار باید وارون شود، امضا کننده باید قبول کند، به‌طور میانگین چهار بار قبل از موفقیت در تولید یک امضای حلقوی، تلاش کند.
یک تفاوت مهم این طرح و طرحی که براساس سیستم RSA وجود دارد، در اثبات گمنامی بی‌قیدوشرط

است، که بستگی به این حقیقت دارد که در RSA، تمام نگاشت‌ها، جایگشت هستند اما در اینجا اینطور نیست. زمانی که g_i جایگشت نیست، تفاوت‌های قابل توجهی بین توزیع انتخاب تصادفی و مقادیر محاسبه شده‌ی x_i در امضای حلقوی مفروض، وجود دارد. این می‌تواند منجر به شناسایی امضا کننده‌ی واقعی از میان امضاکنندگان ممکن شود و می‌تواند به یک مساله‌ی اصلی در بسیاری از انواع به‌هم پیوسته‌ی توابع یک‌طرفه شرح داده‌شود.
می‌توانیم این مشکل را در مورد امضاهای رابین، به شکل زیر حل کنیم:

قضیه ۱.۳.۳. بگذار S مجموعه‌ی متناهی از تیله‌ها باشد. قرار می‌دهیم $B_1, \ldots, B_n$ زیرمجموعه‌های مجزای S باشند(سطل‌ها)، به‌طوری‌که تمام سطل‌های غیرخالی، تعداد یکسانی تیله دارند و هر تیله در S، دقیقا در یک سطل است. به مثال زیر توجه کنید: یک سطل تصادفی انتخاب می‌کنیم و سپس یک تیله به تصادف از آن سطل انتخاب می‌کنیم. این روش، تیله‌ها را از S با توزیع احتمال یکنواخت انتخاب می‌کند.

توابع رابین $f_i(x_i) = x_i^2 \ (mod\, n_i)$ به توابع $g_i(x_i)$ روی $\{0,1\}^b$ بسط داده‌شده‌اند. هر دو(سطل و تیله‌ها)، همگی اعداد bـبیتی $u = q_i n_i + r_i \,;\, r_i \in \mathbb{Z}_n^*$ و $(q_i + 1)n_i \leq 2^b$ هستند، هر تیله در یک سطل جای گرفته‌است، که این توسط نگاشت بسط داده‌شده‌ی رابین، g_i نگاشته‌شده‌است.
اکنون می‌دانیم هرسطل دارای ۰ یا ۴ تیله است. با توجه به قضیه، می‌توان گفت توزیع نمونه شده‌ی تیله‌های x_i دقیقا یکسان است، بدون در نظر گرفتن اینکه تصادفی انتخاب شده‌اند، یا به‌طور تصادفی در میان وارون‌های محاسبه شده در یک سطل انتخاب شده‌ی تصادفی برگزیده‌شده‌اند. در نتیجه حتی دشمن با قدرت بی‌نهایت نیز نمی‌تواند بین امضاکننده و غیرامضاکننده، با بررسی امضاهای حلقوی تولید شده توسط یکی از امضاکننده‌های ممکن، تمایز قائل شود.

۴.۳ کلیت و موارد خاص

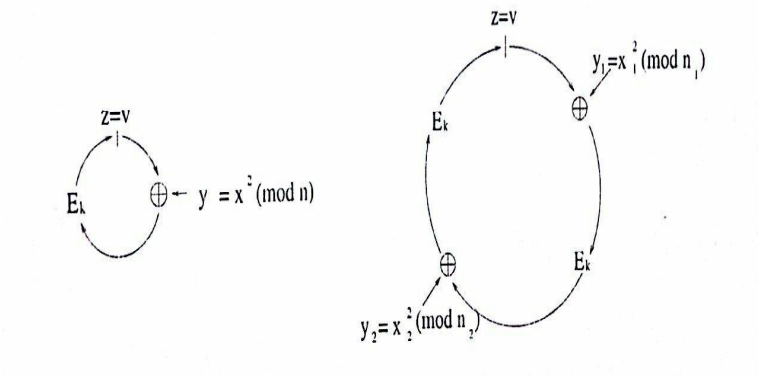
مفهوم امضای حلقوی بسط‌های جالب و موارد خاص زیادی دارد. به‌طور خاص، امضاهای حلقوی با $r = 1$ می‌تواند از منظر نسخه‌ای تصادفی شده از طرح امضای رابین درنظر گرفته‌شود. همان‌طور که در شکل ۳.۳ می‌بینید، شرط وارسی می‌تواند به صورت $(x^2 \ mod\, n) = v \oplus E_{h(m)}^{-1}(v)$ در نظرگرفته‌شود. سمت راست یک چکیده‌ساز از پیام m است، که با استفاده از انتخاب v تصادفی شده‌است.
امضاهای حلقوی $r = 2$ معادله‌ی حلقوی زیر را دارند:

$$E_{h(m)}(x_2^2 \oplus E_{h(m)}(x_1^2 \oplus v)) = v$$

(شکل ۳.۳ را ببینید.)
معادله‌ی حلقوی ساده‌تر (که معادل با قبلی نیست، اما خواص امنیتی یکسان دارد):

$$(x_1^2 \, mod\, n_1) = E_{h(m)}(x_2^2 \, mod\, n_2)$$

که مربعات پیمانه‌ای به $\{0,1\}^b$ بسط داده‌شده‌اند. این روش برای استفاده در امضاهای وارسی کننده‌ی مشخص، در سیستم‌های ایمیل است که n_1 کلیدعمومی فرستنده و n_2 کلیدعمومی گیرنده است.



شکل ۳.۳: امضاهای حلقوی رابین با $r = ۱, ۲$

در امضاهای حلقوی منظم، به‌طور قابل اثبات برای یک مهاجم، افشای شناسه‌ی فرد امضاکننده، غیرممکن است، گرچه ممکن است مواردی وجود داشته‌باشد، که امضاکننده خودش بخواهد گزینه‌ای داشته‌باشد به این منظور که بعدها اثبات کند که خودش مرجع ایمیل گمنام شده، است. (برای مثال زمانی که موفق شود پرایم رسوا شده را از قدرت بیندازد).

به‌طور کلی آنچه گفته‌شد، مفهومی از امضاهای حلقوی بود و برخی کاربردهای این امضاها. در طرحی که مطرح کردیم، از طرح‌های رمزنگاری RSA و رابین، بهره گرفتیم. با وجود کامپیوترهای کوانتومی، امنیت این طرح‌ها به قدرت قبل باقی نماند. از این‌جهت امضاهای حلقوی مشبکه‌مبنا مطرح گردیدند. بحث در مورد این امضاها در این طرح مطرح نکردیم.

۵.۳ نتیجه‌گیری

آنچه در این طرح گفته‌شد، نظریه‌هایی در مورد امضاهای رقمی، که بخشی از رمزنگاری کلید عمومی هستند، بود. امضاهای گروهی امضاهایی هستند که در شرایطی که گروهی با هم‌یاری هم، تمایل به ارسال پیامی به شکل گمنام دارند، استفاده می‌شود. امضاهای گروهی مشبکه‌مبنا دارای امنیت بسیار بالایی هستند که تاکنون روشی برای از شکستن این امضاها، یافت نشده‌است.
امضاهای حلقوی در شرایطی کاربرد دارند که یک شخص، تمایل به ارسال پیامی دارد و تمایل به هم‌یاری با فرد دیگری ندارد. در این شرایط برای حفظ گمنامی خود، تشکیل حلقه‌ای از اعضا می‌دهد. امضای حلقوی بررسی شده در این طرح، به سبب استفاده از سیستم RSA و رابین، نسبت به طرح‌های قبلی، پرسرعت‌تر است.

مراجع

[1] Ajtai, M. (1996). *Generating Hard Instances of Lattices Problems(Extended Abstract).* In STOC, pages 99-1008. ACM.

[2] Alwen, J., Peikert, C. (2011). *Generating Shorter Bases for Hard Random Lattices.* Theory Comput Syst., 48(3):535-553.

[3] Ateniese, G., Tsudik, G., Song, D. (2003). *Quasi-efficient revocation of group signatures.* In M. Blaze, editor, Proceedings of Financial Cryptography 2002, volume 2357 of LNCS, pages 183-97. Springer-Verlage.

[4] Ateniese, G., Tsudik, G. (1999). *Some open issues and directions in group signatures.* In: Financial Crypto'99, volume 1648 of LNCS, Pages 196-211. Springer-Verlage.

[5] Babai, L. (1986). *On Lovász'lattice reduction and the nearest lattice point problem.* In: Combinatorica. Volume 6, Issue 1, pp 1–13

[6] Bellare, M., Micali, S. (1992). *How to sign given any trapdoor permutation.* Journal of ACM, 39(1):214-233, January.

[7] Bellare, M., Miner, S. (1999). *A forward-secure digital signature scheme.* In: M. Wiedner, editor, CRYPTO'99, volume 1666 of LNCS, Pages 431-448. Springer-Verlage.

[8] Bellare, M., Micciancio, D., Warinschi, B. (2003). *Foundation of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions.* In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614-629.Springer, Heidelberg.

[9] Bellere, M., Micciancio, D, Warnisch, B. Full version of this paper. Available at http://www.cs.ucsd.edu/users/bogdan.

[10] Bellare, M., Rogaway, P. (1993). *Random oracles are practical: A paradigm for designing efficient protocols.* In: 1st ACM Conference on Computer and Communications Security, pp. 62-73. ACM Press, New York.

[11] Bellare, M., Shi, H., Zhang, C. *Foundation of group signatures: The case of dynamic groups.* In A. J. Menezes, editor, Proceedings of CT-RSA 2005, volume 3376 of LNCS, pages 136-53. Springer-Verlage.

[12] Bender, A., Katz, J., Morselli, R. (2006). em Ring signature: Stronger definitions, and constructions without random oracles. In S. Halevi and T. Rabin, editors, Proceedings of TCC 2006, volume 3876 of LNCS, pages 60-79. Springer-Verlag, Mar.

[13] Boneh, D., Boyen, X., Shacham, H. (2004). *Short group signatures.* In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41-55. Springer, Heidelberg.

[14] Boneh, D., Shacham, H. (2004). *Group signatures with verifier-local revocation.* In: B. Pfitzmann and P. Liu, editors, Proceedings of CCS 2004, pages 168-77. ACM Press, Oct.

[15] Boneh, D., Canetti, R., Halevi, S., Katz, J. (2007). *Chosen-Ciphertext Security from Identity-Based Encryption.* SIAM J. Comput, 36(5):1301-1328.

[16] Boneh, D., freeman, D. (2011). *Homomorphic signatures for polynomial functions.* In proceedings of Eurocrypt 2011, LNCS 6632, pp. 149-168.

[17] Boyen, X. (2010). *Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signature and More.* In public Key Cryptography, volume 6056 of Lecture Notes in Computer Science, pages 499-517. Springer.

[18] Boyen, X., Waters, B. (2006). *Compact group signatures without random oracles.* In: S. Vaudenay, editor, Proceedings of Eurocrypt 2006, volume 2004 of LNCS, Pages 427-44. Springer-Verlage, May.

[19] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S. (2001). *Resettable zero-knowledge.* In Proceedings of the 32st Symposium on te Theory of Computing (STOC), Pages 235-224.

[20] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C. (2011). *Bonsai Tree, or How to Delegate a Lattice Basis.* In EUROCRYPT, volume 6110 of Lecture Notes in Computer Science, pages 523-552. Springer.

[21] Chaum, D., and heyst, E. (1991). *Group Signatures.* In D. W. Davies, editor, Proceedings of Euroucrypt1991, volume 547 of LNCS, Pages 257-65. Springer-Verlag.

[22] Chen, L., Pedersen, T.P. (1994). *New group signature schemes.* In A. DeSantis, editor, EUROCRYPT'94, volume 950 of LNCS, Pages 171-181. Springer-Verlag,

[23] Cramer, R., Damgrd, I., Schoenmarkers, B. (1994). *Proofs of partial knowledge and simplified design of witness hiding protocols.* In: Demedt, Y.G. (ed.) CRYPTO 1994.LNCS, vol. 839,pp. 174-187. Springer, Heidelberg.

[24] Damgard, I. *Commitment schemes and zero-knowledge protocols.* In: Computer Science, Center of the National Research Foundation.

[25] Diffi, Whitfield, and Martin E. Hellman. (1976). *New directions in cryptography.* Information Theory, IEEE Transactions on 22, no. 6.

[26] Dolev, D., Dwork, C., Naor, M. (2000). *Nonmalleable cryptography.* SIAM Journal of Computing, 30(2):391-437.

[27] Elgemal, Taher. (1985). *A public key cryptosystem and a signature scheme based on discrete logarithm.* In Advances in cryptology , pp. 10-18. Springer Berlin Heidelberg .

[28] Feige, U., Lapidot, D., Shamir, A. (1999). *Multiple non-interactive zero-knowledge. proofs under general assumptions.* SIAM journal on Computing, 29(1):1-28.

[29] Fiat, A., Shamir, A. (1987). *How to prove yourself: Practical solution to identification and signature problems.* In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186-194. Springer, Heidelberg.

[30] Gentry, C., Peikert, C., Vaikuntanathan, V. (2008). *Trapdoors for Hard Lattices and New Cryptographic Constructions.* In STOC, pages 197-206. ACM.

[31] Gorden, S., Katz, J., Vaikuntanathan, V. (2010). *A Group Signature from Lattice Assumptions.* In: M. Abe (ed.): ASIACRYPT 2010, LNCS 6477, pp. 395-412.

[32] Goldreich, O., Goldwasser, S. (2000). *On the limits of nonapproximability of lattice problems.* Journal of Computer and System Science 60(3), 540-563.

[33] Goldreich, O., Goldwasser, S., Halevi, S. (1997). *Public key cryptosystems from lattice reduction problems.* MIT, Laboratory for Computer Science.

[34] Goldwasser, S., Micali, S., Rivest,R. (1988). *A digital signature scheme secure against adaptive chosen-message attacks.* SIAM Journal of Computing, 17(2):281-308.

[35] Hardy, G.H., Wright, E.M. (1979). *An Introduction to the Theory of Numbers.* Oxford, fifth edition.

[36] Jakobsson, M., Sako, K., Impagliazzo, R. (1996). *Designed verifier proofs and their applications.* In: Ueli Mauer , editor, Advances in Cryptology - EuroCrypt ’96. Pages 143-154, Berlin. Springer-Verlage. Lecture Notes in Computer Science Volume 1070.

[37] Kawachi, A., Tanaka, K., Xagawa, K. (2008). *Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems.* In ASIACRYPT, volume 5350 of Lecture Notes in Computer Science, pages 372-389. Springer.

[38] Kiayias, A., and Yung, M. (2005). *Group signatures whit efficient concurrent join.* In R. Cramer, editor, Proceedings of Eurocrypt 2005, volume 3494 of LNCS, pages 198-214. Springer-Verlag, May.

[39] Laguillaumie, F., Langlois, A., Libert, B.,Stehle, D. (2013). *Lattice-Based Group Signature with Logarithmic Signature Size.* In ASIACRYPT, volume 8270 of Lecture Notes in Computer Science, pages 41-61. Springer.

[40] Lenstra, A. K., Lenstra, H. W., Lovász, L. (1982). *Factoring polynomials with rational coefficient*. Mathematische Annalen. 261 (4): 515–534.

[41] Li, J., Kim, K. *Attribute-based ring signatures.* Cryptography ePrint Archive, Report 2007/159, http://eprint .iacr.org/2007/159.

[42] Li, W., Fan, M., Jia, Zh. (2012). *An attribute-based ring signature scheme in lattice.* Wuhan University Journal of Natural Sciences. August , Volume 17, Issue 4, pp 297–301.

[43] Ling, S., Nguyen, KH., Stehle, D., Wang, H. (2013). *Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications.* In Public Key Cryptography, volume 7778 of Lecture Notes in Computer Science, pages 107-128. Springer.

[44] Ling, S., Nguyen, KH., Wang, H. *Group Signature from Lattices: Simpler, Tighter, Shorter.* Division of Mathematical Science.

[45] Luby, M., Rackoff, C. (1998). *How to construct pseudorandom permutations from pseudorandom functions.* SIAM J. Computing, 17(2):373-386, April.

[46] McEliece, Robert J. (1978). *A public-key cryptosystem based on algebraic coding theory*.DSN progress report 42, no.44: 114-116.

[47] Meiklejohn, S. (2011). *An exploration of group and ring signatures.* Available online at http://cseweb.ucsd.edu/ smeiklejohn/, February.

[48] Menezes, Alfred J., Paul C., Oorschot, V., and Scatt A. Vanstone. (1997). Handbook of applied cryptography. CRCpress.

[49] Mercle, Charles, R. (1979). *Security, authentication, and public key systems.* Stanford University, Stanford, CA.

[50] Micciancio, D., Peikert, C. (2012). *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller.* In EUROCRYPT, volume 7237 of Lecture Notes in Computer Science, pages, 700-718. Springer.

[51] Micciancio, D., Vadhan, S. (2003). *Statistical zero-knowledge proofs with efficient provers: Lattice problems and more.* In: Boneh, D. (ed). CRYPTO 2003. LNCS, vol. 2729, pp. 282-298. Springer, Heidelberg.

[52] Peikert, C. (2010). *An efficient and parallel gaussian sampler for lattice.* In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 80-97. Springer, Heidelberg.

[53] Rabin, M. (1979). *Digitalized signatures and public key functions as intractable as factorization.* MIT Technical Report MIT-LCS-TR-212.

[54] Regev, O. (2005). *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography.* In STOC, pages 84-93. ACM.

[55] Rivest, R., Ronald L., Shamir, A., Adleman, L. (1978). *A method for obtaining digital signature and public-key cryptosystems.* Communication of the ACM 21,no. 2: 120-126

[56] Rivest, R., Shamir, A., Tauman, Y. (2001). *How to leak a secret*. In C. oyd, editor, Proceedings of asiacrypt 2001, volume 2248 of LNCS, pages 552-65. Springer-Verlag, Dec.

[57] Sahai, A. (1999). *Non-malleable non-interactive zero knowledge and adaptive chosen ciphertext security.* In FOCG'99, pages 543-553.

[58] Shacham, H., Waters, B. (2007). *Efficient ring signatures without random oracles.* In proceedings of PKC 2007, volume 4450 of LNCS, pages 166-180. Springer-Verlag.

[59] Song, D. (2001). *Practical forward-secure group signature schemes.* In ACM Symposium on Computer Communication Security, pages 225-234, November.

Aabstract

Our purpose in this study is investigation of some digital signatures. Digital signatures are primary of cryptography, that leading to authentification of the message sender for the recipient. Group signatures are the signatures that, allow one group member to sign a message on behalf of the whole group. The group members manage by group master/manager. Ring signatures are the signatures that, members of the ring, except signer, are not aware of their membership in the ring. The signer sign his message by using public key of other members, as his identity remain anonymous for the recipient, but the recipient ensures that he is one of the ring member. Available encryption systems has been threatened due to the development of quantum computers, thus international encryption communities applied all efforts to obtaining encryption systems that are resistant against quantum attacks. Lattice-based encryption, is kind of resistant against quantum attacks cryptography, that in resent decays has been taken into consideration too much. Therefore in this study, we are investigating one type of lattice-based group signature.

keywords: Public key cryptography, Group signature, Anonymity, Traceability, Ring signature,

Forge, Random oracle

Shahrood University of Technology

Faculty Of Mathematical Sciences

MSc Thesis in: Graph And Combinatory

On Digital Signatures

By: Ghazale Taghizade

Supervisors

Dr. Meysam Alishahi
Dr. Farrokhlagha Moazemi

Junary 2017