امضای دیجیتال مقاوم کوانتومی بر اساس همسانی های بین خم های سوپرسینگولار

مصطفی قربانی

استاد راهنما: دکتر حسن دقیق

۱ اثبات دانش صفر هویت

[۱] ما از عدد اولی به فرم $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ استفاده می‌کنیم که ℓ_A و ℓ_B اعداد اول کوچک (معمولا ۲ و ۳) می‌باشند با این خاصیت که $\ell_A^{e_A} \approx \ell_B^{e_B}$ و f یک عامل کوچک است که باعث می‌شود p یک عدد اول شود. پارامترهای عمومی شامل عدد اول $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ ، خم سوپرسینگولار $E(\mathbb{F}_{p^2})$ از مرتبه‌ی $(\ell_A^{e_A}\ell_B^{e_B})^2$ و همچنین نقاط P_B و Q_B که مولدهای زیرگروه $E[\ell_B^{e_B}]$ می‌باشند.

اثبات دانش صفر مطابق با طرح شکل ۱ می‌باشد. پگی(اثبات کننده) نقطه مخفی S که تولیدکننده هسته همسانی $\phi : E \to E/\langle S\rangle$ است را به عنوان یک راز در اختیار دارد. بنابریان کلید خصوصی پگی S (یا هر مولدی از $\langle S\rangle$) و کلید عمومی آن شامل خم بیضوی $E/\langle S\rangle$ و تصویر نقاط مولد عمومی یعنی $\phi(P_B)$ و $\phi(Q_B)$ می‌باشد.

پگی برای آن که به ویکتور (تاییدکننده) اثبات کند که دانش $\langle S\rangle$ را می‌داند ، یک نقطه تصادفی R از مرتبه‌ی $\ell_B^{e_B}$ انتخاب و همسانی $\psi = E \to E/\langle R\rangle$ را تعریف می‌کند.توجه به این نکته لازم است که تساوی زیر برقرار است :

$$(E/\langle S\rangle)/\langle\phi(R)\rangle = E/\langle R, S\rangle = (E/\langle R\rangle)/\langle\psi(S)\rangle$$

[1] Zero-Knowledge Proof of Identity

پگی همسانی‌های طرح ارائه شده در شکل ۱ را محاسبه و آنها را برای ویکتور ارسال می‌کند. در ادامه ویکتور یک بیت چالشی $b \in \{۰, ۱\}$ خود را برای پگی ارسال می‌کندو متعاقبا پگی همسانی‌ای بر اساس چالش انتخابی ویکتور برای وی ارسال می‌کند تا ویکتور آنها را تایید کند.

$$\begin{tikzcd} E \arrow[r, "\phi"] \arrow[d, "\psi"] & E/\langle S\rangle \arrow[d, "\psi'"] \\ E/\langle R\rangle \arrow[r, "\phi'"] & E/\langle R, S\rangle \end{tikzcd}$$

شکل ۱: هر فلش با همسانی و و هسته‌اش نشانه گذاری شده است

برای فهم و تشریح بیشتر این پروتکل آن را به صورت الگوریتمیک نشان می‌دهیم :

۱.
- پگی یک نقطه تصادفی R از مرتبه‌ی $\ell_B^{e_B}$ انتخاب می‌کند.
- او همسانی $\psi : E \to E/\langle R\rangle$ را محاسبه می‌کند.
- پگی در ادامه همسانی $\hat{\phi} : E/\langle R\rangle \to E/\langle R, S\rangle$ را به همراه هسته $\langle \psi(S)\rangle$ (از سوی دیگر $\hat{\psi} : E/\langle S\rangle \to E/\langle R, S\rangle$ را به همراه هسته $\langle \phi(R)\rangle$) محاسبه می‌کند.
- پس از محاسبات بالا ، پگی تعهد [2] $com = (E_۱, E_۲)$ که $E_۱ = E/\langle R\rangle$ و $E_۲ = E/\langle R, S\rangle$ را برای ویکتور ارسال می‌کند.

۲. ویکتور به طور تصادفی یک بیت چالشی $ch \in \{۰, ۱\}$ را انتحاب و برای پگی ارسال می‌کند.

۳. پگی پاسخ $resp$ را برای ویکتور ارسال می‌کند چنانکه :
- اگر $ch = ۰$ آنگاه $resp = (R, \phi(R))$
- اگر $ch = ۱$ آنگاه $resp = \psi(R)$

[2]commitment

(آ) b = ۰

$$\begin{tikzcd} E \arrow[r, dashed, "\phi"] \arrow[d, "\psi"] & E/\langle S\rangle \arrow[d, "\psi'"] \\ E/\langle R\rangle \arrow[r, dashed, "\phi'"] & E/\langle R,S\rangle \end{tikzcd}$$

(ب) b = ۱

$$\begin{tikzcd} E \arrow[r, dashed, "\phi"] \arrow[d, dashed, "\psi"] & E/\langle S\rangle \arrow[d, dashed, "\psi'"] \\ E/\langle R\rangle \arrow[r, "\phi'"] & E/\langle R,S\rangle \end{tikzcd}$$

شکل ۲: همسانی‌های مخفی با خط‌های مقطع نماش داده شده است. خط های توپر نمایش دهنده همسانی‌هایی می‌باشد که پگی نسبت به چالش انجام شده ظاهر می‌کند. با این حال همسانی‌های ظاهرشده هیچ اطلاعاتی درباره همسانی مخفی ϕ افشا نمی‌کند.

۴.
- اگر $ch = 0$ ، ویکتور تایید می‌کند که R و $\phi(R)$ هردو از مرتبه‌ی $\ell_B^{e_B}$ هستند و هسته‌های همسانی‌های $E \to E_1$ و $E/\langle S\rangle \to E_2$ را تولید می‌کنند.
- اگر $ch = 1$ ، ویکتور تایید می‌کند که $\psi(S)$ از مرتبه‌ی $\ell_A^{e_A}$ است و هسته‌ی همسانی $E_1 \to E_2$ را تولید می‌کند.

برای دست‌یابی به λ بیت امنیت ، لازم است که عدد اول p انتخابی ، حتما 6λ بیت باشد و پروتکل بالا حتما λ بار تکرار شود. اگر ویکتور تمام λ مرحله از پروتکل را تایید کند ، آنگاه اثبات هویت پگی مورد قبول قرار می‌گیرد(ادعای او مبنی بر دانش کلید خصوصی S اثبات می‌شود) و در غیر اینصورت ویکتور متقاعد نمی‌شود و آن را رد می‌کند.

۲ ساخت آنره

۳

ساخت آنره [۲۹] یک سیستم اثبات دانش صفر تعاملی را به سیستم اثبات دانش صفر غیرتعاملی متناظر با آن انتقال می‌دهد. این ساخت ، ویژگی استخراج آنلاین ۴ را که اجازه می‌دهد شاهد(کلید خصوصی) را از یک متخاصم موفق بدون چرخش ۵ استخراج کنیم ، را دارا می‌باشد.

در این پروتکل ما یک رابطه باینری به نام R استفاده خواهیم کرد. یک اظهار x رخ می‌دهد اگر شاهدی به نام w برای آن موجود باشد که در این صورت آن را به صورت $(x, w) \in R$ نمایش خواهیم داد.در یک سیستم اثبات ، یک اثبات کننده $\mathcal{R}$ خواهان آن است تا اظهار x را برای تاییدکننده $\mathcal{V}$ اثبات کند (به عبارت دیگر اثبات کننده درصدد متقاعد کردن تاییدکننده است که شاهد w را برای x در اختیار دارد).در ابتدای امر این فرص را داریم که تمام بخش های این پروتکل به یک اوراکل تصادفی کوانتومی ۶ H دسترسی دارند.

۱.۲ پروتکل زیگما

۷

یک پروتکل زیگما $\Sigma = ((P^{۱}, P^{۲}), V)$ ، یک سیستم اثبات تعاملی است که شامل سه قسمت به ترتیب زیر می‌باشد:

- یک تعهد ۸ $com = P^{۱}(x, w)$ ارائه شده توسط اثبات کننده
- یک چالش ch یکنواخت و به طور تصادفی انتخاب شده توسط تاییدکننده
- یک پاسخ $resp = P^{۲}(x, w, com, ch)$ محاسبه شده توسط اثبات کننده بر اساس چالش دریافتی ch

[3] Unruh's Construction
[4] online extractability
[5] rewinding
[6] quantum random oracle
[7] Sigma Protocols
[8] commitment

- خروجی $V(x, com, ch, resp)$ توسط تاییدکننده که بر اساس آن ، اثبات یا پذیرفته می‌شود یا مورد قبول واقع نمی‌شود.

اگر پروتکل زیگما را به صورت $\Sigma = (P, V)$ در نظر بگیریم آنگاه $P = (P^۱, P^۲)$ خواهد بود.

پروتکل زیگما شامل ویژگی‌هایی می‌باشد که آنها را به صورت زیر بیان می‌کنیم :

تمامیت [۹]:

اگر اثبات کننده $\mathcal{R}$ واقعا شاهد w را برای اظهار x بداند آنگاه طبق این پروتکل ، تاییدکننده $\mathcal{V}$ ادعای اثبات کننده را می‌پذیرد.

صداقت ویژه [۱۰]:

یک الگوریتم چندجمله‌ای استخراج [۱۱] E_Σ وجود دارد که با دریافت هر جفتی از تعاملات معتبر $(com, ch, resp)$ و $(com, ch', resp')$ که هم $ch \neq ch'$ و هم مورد پذیرش تاییدکننده می‌باشد، می‌تواند یک شاهد w محاسبه کند که $(x, w) \in R$

دانش صفر تاییدکننده صادق [۱۲] :

یک الگوریتم چندجمله‌ای شبیه ساز S_Σ وجود دارد که خروجی $(com, ch, resp)$ را تولید می‌کند به طوری که نسبت به خروجی تعامل انجام شده میان اثبات کننده و تاییدکننده صادق، توسط هیچ الگوریتم چندجمله‌ای کوانتومی قابل تشخیص نمی‌باشد.

ذکر این نکته لازم است که اثبات دانش صفر هویت همسانی مبنای گفته شده در بخش قبلی در اصل یک پروتکل زیگما می‌باشد.

[9] Completeness
[10] Special soundness
[11] polynomial time extractor
[12] Honest-verifier zero-knowledge (HVZK)

۲.۲ سیستم اثبات غیرتعاملی

[13]

یک سیستم اثبات غیرتعاملی شامل دو الگوریتم می‌باشد :

- یک اثبات کننده $P(x, w)$ ، یک اثبات π برای اظهار x (که دارای شاهد w می‌باشد) تولید می‌کند.
- تاییدکننده $V(x, \pi)$ ، یا خروجی **تایید** برای پذیرش ادعا یا خروجی **انکار** را برای اثبات π مطرح شده توسط اثبات کننده تولید می‌کند.

یک سیستم اثبات غیرتعاملی (P, V) شامل سه ویژگی لازم است که در زیر آنها را تشریح می‌کنیم :

تمامیت :

اگر $R \ni (x, w)$ آنگاه تاییدکننده V ، اثبات $\pi = P(x, w)$ را می‌پذیرد.

دانش صفر[14] :

یک الگوریتم چندجمله‌ای شبیه ساز S که به یک اوراکل تصادفی دسترسی دارد (می‌تواند یک اوراکل را اجرا کند) ، وجود دارد که می‌تواند اثبات‌هایی متفاوت از اثبات‌های تولید شده توسط اثبات کننده P را تولید کند. الگوریتم شبیه‌ساز به وسیله دو الگوریتم $S = (S_{init}, S_P)$ بیان می‌شود.

شبیه‌ساز صداقت با ویژگی استخراج آنلاین [15]

یک الگوریتم چندجمله‌ای استخراج E وجود دارد که توانایی تولید یک شاهد w برای ادعای x مطرح شده توسط اثبات‌کننده ، را دارا می‌باشد.

[13] Non-interactive Proof System
[14] Zero-knowledge (NIZK)
[15] Simulation-sound online-extractability

۳.۲ ساخت آنره

ساخت آنره ، یک پروتکل زیگما (Σ) را به یک سیستم اثبات غیرتعاملی (P_{OE}, V_{OE}) منتقل می‌کند چنانکه اگر پروتکل (Σ) شامل ویژگی‌های تمامیت ، صداقت خاص و دانش صفر باشد آنگاه نتیجه یک سیستم اثبات با ویژگی تمامیت ، دانش صفر به همراه ویژگی شبیه ساز صداقت با استخراج آنلاین خواهد بود.

اگر فرض کنیم یک پروتکل زیگما به صورت $\Sigma = (P_\Sigma, V_\Sigma)$ که $P_\Sigma = (P_\Sigma^۱, P_\Sigma^۲)$ داشته باشیم که c چالش ممکن در دامنه‌ی چالش ها N_{ch} داشته باشیم و بخش‌ها خواهان اجرای پروتکل به تعداد t بار باشد (که t به پارامتر امنیتی λ بستگی دارد. در طرح امضای دیجیتال معرفی شده در این پایان نامه : $t = ۲\lambda$,$c = ۲$,$N_{ch} = \{۰, ۱\}$) در این صورت اگر G و H را اوراکل‌های تصادفی کوانتومی در نظر بگیریم که G در همان دامنه باشد آنگاه سیستم اثبات غیرتعاملی (P_{OE}, V_{OE}) را تعریف می‌کنیم که P_{OE} و V_{OE} را به صورت الگوریتم های ۱ و ۲ به دست می‌آیند.

الگوریتم ۱ اثبات‌کننده : P_{OE} بر اساس ورودی (x, w)

1: **for** $i = 1tot$ **do**

ایده آن است که تعامل Σ را بوسیله چالش $J = J_۱||\cdots||J_t$ به عنوان خروجی تابع تصادفی H شبیه سازی کرد.

۴.۲ امضا براساس اثبات دانش صفر غیرتعاملی

[16]

یک طرح امضای دیجیتال شامل سه الگوریتم زیر می‌باشد :

- $\mathbf{KeyGen}(\lambda)$

این الگوریتم یک پارامتر امنیتی λ به عنوان ورودی گرفته و یک زوج کلید (pk, sk) تولید می‌کند.

- $\mathbf{Sign(sk, m)}$

[16] Signature from Non-interactive Zero-Knowledge Proofs

این الگوریتم پیام m وکلید خصوصی sk را به عنوان ورودی گرفته و خروجی آن امضای σ می‌باشد.

- $\mathbf{Verify}(\mathbf{pk}, \mathbf{m}, \sigma)$

این الگوریتم با داشتن کلید عمومی امضاکننده $\mathbf{sk}$ تایید می کند که آیا امضای دریافتی σ متعلق به پیام $\mathbf{m}$ می‌باشد یا نه

یک طرح امضای دیجیتال قویا تحت حمله متن انتخاب شده [۱۷] ، غیرقابل جعل [۱۸] است اگر برای هر متخاصم $\mathcal{A}$ [۱۹] با داشتن الگوریتم زمان چندجمله‌ای کوانتومی و دسترسی کلاسیک به اوراکل امضای $\mathbf{sig} : \mathbf{m} \mapsto \mathbf{Sign}(\mathbf{sk}, \mathbf{m})$ ، حتی با احتمال خیلی کم هم نتواند یک زوج پیام-امضای جدید تولید کند.

فرض کنیم یک تابع تولید کلید $\mathbf{KeyGen}$ ، در اختیار داریم که یک جفت کلید عمومی-خصوصی $(\mathbf{sk}, \mathbf{pk})$ را تولید می‌کند و هیچ الگوریتم چندجمله‌ای کوانتومی حتی با احتمال خیلی کوچک هم نتواند از طریق کلید عمومی $\mathbf{pk}$ ، یک کلید خصوصی $\mathbf{sk}$ معتبر (متناظر با کلید عمومی) بازیابی کند. در این صورت یک اثبات هویت می‌تواند به صورت اثبات اظهار $\mathbf{x} = \mathbf{pk}$ با شاهد $\mathbf{w} = \mathbf{sk}$ در نظر گرفته شود که $(\mathbf{x}, \mathbf{w}) \in \mathbf{R}$ اگر و تنها اگر $(\mathbf{x}, \mathbf{w})$ یک زوج کلید معتبر در نظر گرفته شود که می‌تواند توسط تابع $\mathbf{KeyGen}$ تولید شده باشد.

در این صورت ، یک امضای دیجیتال اساسا یک اثبات دانش صفر غیرتعاملی هویت می‌باشد به جز آنکه لازم است یک پیام مشخص داخل **اثبات(امضا)** وارد کنیم ، این عمل را به این صورت انجام می‌دهیم که متن موردنظر را به عنوان بخشی از اظهار $\mathbf{x}$ درنظر می‌گیریم به عبارت دیگر اظهار جدید ما به صورت $\mathbf{x} = (\mathbf{pk}, \mathbf{m})$ درنظر گرفته می‌شود که در این صورت رابطه $\mathbf{R}$ پیام را در نظر نمی‌گیرد؛ به طور خلاصه ،

$$((\mathbf{pk}, \mathbf{m}), \mathbf{w}) \in \mathbf{R} \quad \text{اگر و تنها اگر} \quad (\mathbf{pk}, \mathbf{m}) \text{ یک زوج کلید معتبر باشند}$$

بنابراین از طریق یک اثبات هویت $(\mathbf{P}, \mathbf{V})$ با ویژگی $\mathbf{NIZK}$ ، یک طرح امضای دیجیتال $\mathcal{DS} = (\mathbf{KeyGen}, \mathbf{Sign}, \mathbf{Verify})$ که $\mathbf{Sign}(\mathbf{sk}, \mathbf{m}) = \mathbf{P}((\mathbf{pk}, \mathbf{m}), \mathbf{sk})$ و $\mathbf{Verify}(\mathbf{pk}, \mathbf{m}, \sigma = \mathbf{V}((\mathbf{pk}, \mathbf{m}, \sigma)))$ به دست می‌آید.

[17] chosen message attack
[18] SUF-CMA
[19] Adversary

قضیه ۲ . اگر (P, V) یک اثبات هویت $NIZK$ [20] با ویژگی‌های شبیه سازی_صداقت و استخراج_آنلاین باشد آنگاه طرح امضای $\mathcal{DS}$ ذکرشده در بالا یک امضای دیجیتال SUF-CMA در مدل ارواکل تصادفی کوانتومی خواهد بود.

۳ امضای دیجیتال همسانی مبنا

در این بخش قصد داریم طرح امضای دیجیتال همسانی مبنای خود را بر اساس نتیجه‌ی خود از بخش قبلی بیان کنیم. اگر Σ را به عنوان اثبات دانش صفر هویت همسانی مبنای توصیف شده در بخش [۱] نظر بگیریم آنگاه با اعمال ساخت آنره روی این پروتکل(زیگما) ، یک اثبات هویت غیرتعاملی (P_{OE}, V_{OE}) به دست می‌آید که از این طریق یک طرح امضای دیجیتال معرفی می‌کنیم:

پارامترهای عمومی [21]

پارامترهای عمومی ما همان پارامترهای عمومی معرفی شده در پروتکل زیگما می‌باشد: یک عدد اول به فرم $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ ، یک خم بیضوی سوپرسینگولار E از مرتبه‌ی $(\ell_A^{e_A}\ell_B^{e_B})^2$ در میدان $\mathbb{F}^2$ و نقاط (P_B, Q_B) مولد زیرگروه تابی $E[\ell_B^{e_B}]$.

تولیدکلید [22]

برای تولید کلید ، یک نقطه تصادفی S از مرتبه‌ی $\ell_A^{e_A}$ انتخاب و همسانی $\phi : E \to E/\langle S \rangle$ را محاسبه می‌کنیم و زوج کلید (pk, sk) که

$$pk = \Big(E/\langle S\rangle, \phi(P_B), \phi(Q_B)\Big)$$

و

$$sk = S$$

را به عنوان خروجی نمایش می‌دهیم.

[20] Non-Interactive Zero-Knowledge
[21] Public Parameters
[22] Key Generation

امضا[23]

برای امضای پیام m ، الگوریتم امضا را به صورت زیر انجام می‌دهیم:

$$Sign(sk, m) = P_{OE}((pk, m), sk)$$

تاییدسازی[24]

برای تایید امضای σ برای پیام مشخص m ، الگوریتم تایید را به صورت زیر انجام می‌دهیم:

$$Verify(pk, m, \sigma) = V_{OE}((pk, m), \sigma)$$

الگوریتم های ۳و۴و۵ به طور صریح الگوریتم‌های تولبدکلید ، امضا و تاییدسازی را بیان می‌کنند.

۴ جنبه های الگوریتمیک

۲۵

۱.۴ تولید پارامترها

۲۶

برای انتخاب‌های مشخص $\ell_A^{e_A}$ و $\ell_B^{e_B}$ ، می‌توان هر مقدار تصادفی برای f (با هر اندازه رمزنگاری دلخواه) آزمایش کرد تا مقداری به دست آید که $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f - 1$ یا $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f + 1$ یک عدد اول شود. قضیه عدد اول در پیشرفت حساب ؟؟؟؟؟؟؟(به طور مشخص نسخه اثرگذار لاگارایز و اودلیزکو [۱۷]) یک کران پایین کافی چنین اعداد اولی مهیا می‌کنند؟؟؟؟؟.

[23] Signing
[24] Verification
[25] Algorithmic Aspects
[26] Parameter generation

بروکر در [۴] نشان داده است برای هر عدد اول $p = \ell_A^{e_A}\ell_B^{e_B}\cdot f \pm 1$ مشخص ، می‌توان به راحتی یک خم بیضوی سوپرسینگولار E روی میدان $\mathbb{F}_{p^2}$ با مرتبه $(\ell_A^{e_A}\ell_B^{e_B}\cdot f^2) = (p \mp 1^2)$ به دست آورد.

با شروع از خم E می‌توان یک خم سوپرسینگولار E_0 روی میدان $\mathbb{F}_{p^2}$ با استفاده از گام تصادفی روی گراف همسانی انتخاب کرد. به طور معادل می‌توان به سادگی $E = E_0$ در نظر گرفت. در هر دو مورد ، E_0 ساختار گروهی $(\mathbb{Z}/(p\mp 1)\mathbb{Z})^2$ را دارا می‌باشد.
برای انتخاب نقاط مولد زیرگروه $E_0[\ell_A^{e_A}]$ ، می‌توان یک نقطه تصادفی $P \in_R E_0(\mathbb{F}_{p^2})$ انتخاب و آن را در $(\ell_B^{e_B}\cdot f)^2$ ضرب کرد تا نقطه P' با مرتبه توانی از ℓ_A حاصل شود.با احتمال بسیار بالایی P' دقیقا از مرتبه $\ell_A^{e_A}$ می‌باشد؛ برای اثبات این ادعا می‌توان با ضرب P' در توان‌هایی از ℓ_A آن را بررسی کرد. اگر بررسی موفقیت آمیز بود آنگاه $P_A = P'$ در نظر می‌گیریم در غیر اینصورت به دنبال یافتن نقطه‌ای دیگر برای یافتن P می‌شویم. برای به دست آوردن نقطه دوم ، Q_A از مرتبه‌ی ℓ_A می‌توان از همین روش استفاده کرد.برای بررسی این که آیا نقطه Q_A از نقطه P_A متفاوت است ، می‌توان به راحتی با استفاده از زوجیت وایل [27] و محاسبه $e(P_A, Q_A)$ در میدان $E[\ell_A]$ بررسی کرد که آیا نتیجه از مرتبه ℓ_A می‌باشد یا خیر ؛ مثل قبل با احتمال بسیار زیادی ممکن است این دو نقطه متفاوت از هم باشند ولی در صورتیکه این گونه نباشد می‌توان از نقطه Q_A دیگری استفاده کرد.

توجه . انتخاب نقاط مولد ، هیچ گونه تاثیری روی امنیت این طرح ندارد ؛ از آنجا که هر کدام از نقاط مولد با استفاده از لگاریتم گسسته توسیع یافته [28] ، قابل تبدیل به یکدیگر می‌باشند . چنانچه در [۲۷] اشاره شده است این محاسبه به راحتی در زیرگروه $E[\ell_A]$ قابل انجام می‌باشد.

[27] Weil Pairing

[28] extended discrete logarithms

۲.۴ تبادل کلید و دیگر پروتکل‌ها

[۲۹]

تبادل کلید در دو مرحله انجام می‌پذیرد. در هر مرحله آرش و بابک عملیات زیر را در هر طرف انجام می‌دهند:

۱. محاسبه زیرگروه (هسته همسانی) $\langle R\rangle = \langle [m]P + [n]Q\rangle$ برای نقاط مشخص P و Q.

۲. محاسبه همسانی $\phi : E \to E/\langle R\rangle$ برای خم بیضوی E.

۳. (فقط)در مرحله اول ، محاسبه $\phi(R)$ و $\phi(S)$ برای بعضی نقاط R و S؛

چنان که خم E و نقاط P ، Q ، R ، S وابسته به هر مرحله و هر بازیکنی که در یک طرف پروتکل می‌باشند. عملیات مشابه نیاز دیگر پروتکل های قسمت **۳** دارند. در ادامه پیاده سازی موثر هر مرحله را نشان خواهیم داد.

۳.۴ محاسبه $\langle [m]P + [n]Q\rangle$

؟؟؟ بدون کوچکترین خدشه‌ای به این زیرگروه ، می‌توان فرض کنیم که m دارای عنصر وارون در پیمانه‌ی مرتبه‌ی گروه می‌باشد ، در این حالت $R' = P + [m^{-1}n]Q$ زیرگروهی همانند دیگر مولدها می‌باشد. محاسبه R' با روش استاندارد رویکرد دوبرابرکردن-و-جمع [۳۰] نیاز یه نصف عملیات محاسبات $[m]P + [n]Q$ معمولی را دارا می‌باشد(برای روش های بهتر محاسبه عملیات معمولی به مراجعه [۲، ۱۰، ۲۵] شود).

با این حال ، محاسبه $P + [m^{-1}n]Q$ با روش دوبرابرکردن-و-جمع یک حفره امنیتی (اشکال بزرگ) را داراست : در برابر حملات آنالیز قدرت ساده [۳۱] [۱۶] آسیب پذیر می‌باشد. برای جلوگیری از این حمله می‌توان از نردبان مونتگومری [۳۲] [۲۱] برای محاسبه $[m^{-1}n]Q$ استفاده کرد و سپس P را به آن اضافه کرد ، اما این روش به طور قابل ملاحظه‌ای کند می‌باشد.

[29] Key Exchange and other protocols
[30] double-and-add
[31] simple power analysis (SPA)
[32] Montgomery ladder

در عوض در الگوریتم ۱ ، یک نردبان بسیار موثرتری ارائه می‌دهیم و مستقیما $P+[m^{-1}n]Q$ را محاسبه می‌کنیم. ایده اصلی این طرح ساده است : در هر تکرار ، ثبات های A و B و C محتوی مقدارهای به ترتیب $[x]Q$ و $[x+1]Q$ و $P+[x]Q$ می باشند ، که x حاوی ارزش چپ ترین بیت $m^{-1}n$ می‌باشد.تابع $dadd(A,B,C)$ معرف جمع تفاضلی [۳۳] [۲۱] می‌باشد.تاثیر پیاده سازی نردبان معرفی شده در این قسمت به کارآمدی روش دوبرابرکردن_و_جمع ساده روی خم های دوقولوی ادوارد [۳۴] ۵.۴ می‌باشد.

۴.۴ محاسبه همسانی‌های با درجه هموار

[۳۵]

در این قسمت به تشریح چگونگی محاسبه و ارزیابی همسانی‌ها توسط آرش و بابک می‌پردازیم. فرض کنیم E یک خم بیضوی و R یک نقطه از مرتبه ℓ^e باشد. هدف ما محاسبه تصویر خم $E/\langle R\rangle$ و ارزیابی همسانی $\phi : E \to E/\langle R\rangle$ در بعضی نقاط روی خم E می‌باشد.

شکل ۳: ساختمان محاسبات ساخت. $\phi = \phi_5 \circ \cdots \circ \phi.$

زمانیکه درجه ϕ هموار باشد بهتر است آن را به زنجیره‌ای از ℓ _همسانی ها تجزیه کرد. اگر $E. = E$ و $R. = R$ در نظر بگیریم ، آنگاه برای هر $0 \leq i < e$ می‌توان مقادیر زیر را در نظر گرفت :

$$E_{i+1} = E_i/\langle \ell^{e-i-1} R_i\rangle, \quad \phi_i : E_i \to E_{i+1}, \quad R_{i+1} = \phi_i(R_i).$$

چنانکه $E/\langle R\rangle = E_e$ و $\phi = \phi_{e-1} \circ \cdots \circ \phi.$ می‌باشد.

توجه به این نکته لازم است که از آنجا که زیرگروه ℓ _تابی $\langle R_i\rangle$ خم E_i مشخص می‌باشند ، خم بیضوی E_{i+1} و همسانی ϕ_i می‌توانند توسط فرمول ولو [۳۶] [۳۰] به راحتی محاسبه شوند. در [۱۴] ، دو پیشنهاد برای داشتنی پیچیدگی درجه دو برای e بیان شده است ؟؟؟؟؟.

[33] differential addition
[34] twisted Edwards curves
[35] Computing smooth degree isogenies
[36] Velu's formulas

شکل بالا خلاصه‌ای از ساختار محاسباتی مسئله برای $e = ۶$ می‌باشد. نقطه‌های توپر این گراف نشان دهنده نقاط می‌باشد. نقطه‌های موجود در یک سطح افقی نشان دهنده آن است که این نقاط از یک مرتبه می‌باشند و همچنین نقطه‌های روی خط مورب چپین نشان دهنده آن است که این نقطه‌ها همگی متعلق به یک خم می‌باشند. یال‌های نقطه‌چین همگی جهت‌دار و به سمت پایین می‌باشند ؛ یال‌های چپین معرف آن هستند که نقطه‌ها ℓ برابر شده‌اند و یال‌های راست‌چین هم یک ℓ ـ همسانی را نشان می‌دهند. در ابتدای اجرای الگوریتم ، تنها نقطه $R_۰$ را در اختیار داریم. به بیان دیگر هدف ما در این الگوریتم محاسبه تمام نقاط روی خط پایانی توسط نقطه آغازین $R_۰$ می‌باشد (ورودی این الگوریتم نقطه $R_۰$ و خروجی این الگوریتم نقاط $[\ell^۵]R_۰$ ، $[\ell^۴]R_۱$ ، $[\ell^۳]R_۲$ ، $[\ell^۲]R_۳$ ، $[\ell^۱]R_۴$ و $R_۵$ می‌باشد). در واقع با دانستن نقطه $[\ell^{e-i-۱}]R_i$ ، می‌توانیم هسته همسانی ϕ_i را به تعداد $\mathcal{O}(\ell)$ جمع نقاط ، محاسبه کنیم ؛ که در این صورت پیچیدگی محاسبات به طور قابل توجهی کم می‌شود. در ادامه می‌توانیم از طریق فرمول ولو ، همسانی ϕ_i و خم $E_{i+۱}$ را محاسبه کنیم.

برای فهم بیشتر این الگوریتم مراحل ذکر شده در مثال $e = ۶$ را مرحله به مرحله نمایش می‌دهیم :

$$i = ۰ \Rightarrow \quad E_۱ = E_۰/\langle \ell^۴ R_۰\rangle, \quad \phi_۰ : E_۰ \to E_۱, \quad R_۱ = \phi_۰(R_۰)$$

$$i = ۱ \Rightarrow \quad E_۲ = E_۱/\langle \ell^۳ R_۱\rangle, \quad \phi_۱ : E_۱ \to E_۲, \quad R_۲ = \phi_۱(R_۱) = \phi_۱(\phi_۰(R_۰))$$

$$i = ۲ \Rightarrow \quad E_۳ = E_۲/\langle \ell^۲ R_۲\rangle, \quad \phi_۲ : E_۲ \to E_۳, \quad R_۳ = \phi_۲(R_۲) = \phi_۲(\phi_۱(\phi_۰(R_۰)))$$

$$i = ۳ \Rightarrow \quad E_۴ = E_۳/\langle \ell^۱ R_۳\rangle, \quad \phi_۳ : E_۳ \to E_۴, \quad R_۴ = \phi_۳(R_۳) = \phi_۳(\phi_۲(\phi_۱(\phi_۰(R_۰))))$$

$$i = ۴ \Rightarrow \quad E_۵ = E_۴/\langle \ell R_۴\rangle, \quad \phi_۴ : E_۴ \to E_۵, \quad R_۵ = \phi_۴(R_۴) = \phi_۴(\phi_۳(\phi_۲(\phi_۱(\phi_۰(R_۰)))))$$

$$i = ۵ \Rightarrow \quad E_۶ = E_۵/\langle R_۵\rangle, \quad \phi_۵ : E_۵ \to E_۶, \quad R_۶ = \phi_۵(R_۵) = \phi_۵(\phi_۴(\phi_۳(\phi_۲(\phi_۱(\phi_۰(R_۰))))))$$

۵.۴ انتخاب مدل

[37]

۶.۴ ساده سازی نقاط تاب دار

[38]

۷.۴ محاسبه‌ی همسانی‌ها

[39]

۸.۴ سایز پارامترها

[40]

۵ امنیت

[41]

امنیت سیستم های رمزنگاری همسانی مبنا بر اساس سختی مسائلی همچون دو مساله بیان شده در زیر بنا شده است که حتی در برابر کامپیوترهای کوانتومی نیز کاملا ایمن می‌باشند. در طرح امضای دیجیتال ارائه شده در این پایان نامه ، امنیت بر اساس این دو مساله بنا شده است.

اگر عدد اول p را به فرم $\ell_A^{e_A}\ell_B^{e_B}.f \pm ۱$ در نظر بگیریم ، آنگاه یک خم بیضوی سوپرسینگولار $E.$ روی میدان $\mathbb{F}_{p^۲}$ وجود دارد که زوج نقاط $\{P_A, Q_A\}$ و $\{P_B, Q_B\}$ مولدهای زیرگروه های $E.[\ell_A^{e_A}]$ و $E.[\ell_B^{e_B}]$ می‌باشند.

[37] choice of the model
[38] Sampling Torsion Points
[39] Computing Isogenies
[40] Parameter Sizes
[41] Security

مساله همسانی سوپرسینگولار محاسباتی : [۴۲]

فرض کنیم $\phi_A : E_۰ \to E_A$ یک همسانی با هسته $\langle [m_A]P_A + [n_A]Q_A \rangle$ می‌باشد که m_A و n_A نقاط تصادفی از میدان $(\mathbb{Z}/\ell_A^{e_A}\mathbb{Z})$ است که هر دو همزمان عاملی از ℓ_A نمی‌باشند. با داشتن E_A ، $\phi_A(P_B)$ و $\phi_A(Q_B)$ یافتن مولد همسانی، یعنی $\langle R_A \rangle = \langle [m_A]P_A + [n_A]Q_A \rangle$ یک مسئله سخت محاسباتی در همسانی‌ها می‌باشد. به عبارت دیگر با داشتن دو خم $E_۰$ و E_A و همسانی بین آنها یعنی ϕ_A و همچنین نقاط کمکی گفته شده در بالا نمی‌توان زیرگروهی که از طریق فرمول ولو امکان‌پذیر است را به دست آورد ؟؟؟.

توجه. ذکر این نکته لازم است که با داشتن مولد $R_A = [m_A]P_A + [n_A]Q_A$ ، یافتن نقاط m_A و n_A به سادگی توسط لگاریتم گسسته توسیع‌یافته [۴۳] با این فرض که خم $E_۰$ هموار باشد، امکان‌پذیر است. [۲۸]

مسئله ساخت خم سوپرسینگولار تصمیم‌پذیر : [۴۴]

فرض کنیم $\phi : E_۰ \to E_۳$ یک همسانی با مرتبه $\ell_A^{e_A}$ باشد. با داشتن $(E_۱, E_۲, \phi')$ ساده سازی شده با احتمال ۱/۲ از طریق دو توزیع زیر ، فهمیدن اینکه کدام رخ می‌دهد یک مسپله سخت می‌باشد :

-
-

۱.۵ امنیت اثبات دانش صفر

[۴۵]

در [۱۲, S۶/۲] اثبات شده است که طرح اثبات دانش صفر هویت همسانی مبنای معرفی شده در قسمت ۱ دارای ویژگی‌های تمامیت ، صداقت ویژه و دانش صفر تاییدکننده صادق می‌باشد اگر

[42] Computational Supersingular Isogeny (CSSI) problem
[43] extended discrete logarithms
[44] Decisional Supersingular Product (DSSP problem)
[45] Security of the Zero-Knowledge Proof

فرض کنیم که مسائل $CSSI$ و $DSSP$ مسائلی سخت می‌باشند.با این وجود برای امنیت کامل ، ساخت آنره باید دارای ویژگی صداقت ویژه باشد.

قضیه ۳ .

اثبات دانش صفر هویت همسانی مبنا ، ویژگی‌های تمامیت ، صداقت ویژه و دانش صفر تاییدکننده صادق را دارا می‌باشد.

اثبات. در این قسمت تنها به اثبات ویژگی صداقت خاص می‌پردازیم.فرض کنید دو رونوشت معتبر $(com, 0, resp_0)$ و $(com, 1, resp_1)$ که $(com = (E_1, E_2))$ را دریافت کرده ایم. پس می‌توانیم از $resp_0 = (R, \phi(R))$ استفاده کنیم تا همسانی $\psi : E \to E/\langle R\rangle$ را محاسبه کنیم. از آنجا که $resp_1 = \psi(S)$ یک مولد هسته ψ' می‌باشد ، بنابراین می‌توانیم دوگان همسانی یعنی $\psi' : E/\langle R\rangle \to E$ را به دست آوریم و $\psi'(resp_1)$ را به عنوان مولد $\langle S\rangle$ به دست آوریم.

$$\begin{tikzcd} E \arrow[r, dashed, "\phi"] \arrow[d, "\psi"] & E/\langle S\rangle \arrow[d, "\psi'"] \\ E/\langle R\rangle \arrow[r, "\phi'"] & E/\langle R, S\rangle \end{tikzcd}$$

شکل ۴: اگر ψ و ϕ' هر دو همزمان معلوم باشند آنگاه می‌توان زیرگروه مخفی $\langle S\rangle$ را به دست آوریم.

۲.۵ امنیت امضا

۶ امضای غیرقابل انکار

[۱۵] [46] مفهوم طرح امضای غیرقابل انکار اولین بار توسط چام و آنترپن [۷] معرفی شده است. در یک طرح امضای غیرقابل انکار، امضاکننده یک امضای غیرقابل انکار σ را تولید می‌کند که توسط هرکسی(به صورت عمومی) قابل تایید نمی‌باشد. بنابراین تاییدکننده برای تایید امضا نیاز به تعاملاتی با امضاکننده دارد که برای تایید یا انکار امضای σ ، امضاکننده یک اثبات دانش صفر را بوسیله اجرای پروتکل تایید یا پروتکل انکار انجام می‌دهد.
طرح امضای غیرقابل انکار موجب پیدایش برنامه‌های کاربردی فراوانی در رمزنگاری شده است. از جمله‌ی این کاربردها می‌توان به نرم‌افزار صدور مجوز [47] ،پول الکترونیکی [48] ، رای‌گیری الکترونیکی [49] و حراج الکترونیکی [50] اشاره کرد.

۱.۶ تعریف

مطابق با تعریف رسمی ارائه شده در [۹] ، یک طرح امضای غیرقابل انکار بوسیله چندتایی زیر مشخص شده است:

$$\Sigma = \big(G_{sign}, Sign, Check, Sim, \pi_{con}, \pi_{dis}\big).$$

الگوریتم G_{sign} تولیدکننده کلید، الگوریتم $Sign$ یک الگوریتم امضا، الگوریتم $Check$ یک الگوریتم بررسی اعتبار، الگوریتم Sim یک شبیه‌ساز امضا ، پروتکل π_{con} یک پروتکل تایید و پروتکل π_{dis} یک پروتکل انکار می‌باشد.
الگوریتم تولیدکننده کلید G_{sign} ، یک الگوریتم چندجمله‌ای احتمالاتی [51] می‌باشد که خروجی آن زوج کلید (vk, sk) می‌باشد که vk یک کلید تاییدساز و sk یک کلید امضا [52] می‌باشد.
فضای پیام $\mathcal{M}$ توسط vk مشخص شده است.

---

[46] Undeniable Signature
[47] licensing software
[48] electronic cash
[49] voting electronic
[50] electronic auction
[51] PPT(probabilistic polynomial-time)
[52] فرض می‌کنیم که sk به طور منحصر به فرد توسط vk تعیین شده است.

الگوریتم امضای $Sign$ ، یک الگوریتم چندجمله‌ای احتمالاتی می‌باشد که امضای σ را از طریق پیام $m \in \mathcal{M}$ و کلید امضای sk به عنوان ورودی‌هایش تولید می‌کند.
اگر σ ، خروجی الگوریتم $Sign(sk, m)$ با رشته تصادفی r باشد، آنگاه زوج (m, σ) را معتبر می‌گوییم، در غیر این صورت آن را نامعتیر می‌گوییم.
الگوریتم بررسی اعتبار $Check$ ، یک الگوریتم چندجمله‌ای قطعی می‌باشد که:

$$Check\big((vk, m), \sigma\big) = \begin{cases} ۱ & \text{اگر خروجی زوج } (m,\sigma) \text{ معتبر باشد.} \\ ۰ & \text{اگر خروجی زوج } (m,\sigma) \text{ نامعتبر باشد.} \end{cases}$$

الگوریتم شبیه‌ساز Sim یک الگوریتم چندجمله‌ای احتمالاتی است که یک امضای شبیه‌سازی‌شده‌ی $\sigma' = Sim(vk, m)$ را تولید می‌کند.

یک طرح امضای غیرقابل انکار باید ویژگی‌های غیرقابل جعلی [۵۳] و غیرقابل دسترس‌پذیری (نامرئی بودن) [۵۴] را داشته باشد. غیرقابل دسترس‌پذیری به معنای آن است که برای یک پیام m ، دریافت‌کننده نمی‌تواند متوجه شود که σ ، یک امضای معتبر است یا یک امضای شبیه‌سازی شده. این بدین معنی است که دریافت کننده نمی‌تواند اعتبار زوج (m, σ) را به تنهایی تایید کند.درعوض با همکاری امضاکننده می‌توان اعتبار و عدم اعتبار زوج (m, σ) را با اجرای پروتکل تاییدساز π_{con} و پروتکل انکار π_{dis} و خروجی متناظر با آن پروتکل به دست آورد. پروتکل π_{con} ، یک سیستم اثبات دانش صفر تعاملی [۵۵] روی یک زبان $L_۰ = \{(vk, m, \sigma) |$ زوج (m, σ) معتبر هستند$\}$ و پروتکل π_{dis} ، یک سیستم اثبات دانش صفر تعاملی روی یک زبان $L_۱ = \{(vk, m, \sigma) |$ زوج (m, σ) معتبر نیستند$\}$ می‌باشد. هر سیستم اثبات دانش صفر تعاملی باید ویژگی‌های تمامیت ، صداقت و اثبات صفر را داشته باشند.

[53] unforgeability
[54] invisibility
[55] zero-knowledge interactive proof system (ZKIP)

۲.۶ امنیت امضای غیرقابل انکار

۵۶

غیرقابل جعل بودن . مفهوم غیرقابل جعل بودن را توسط بازی زیر بین یک چالشگر $\mathcal{CH}$[۵۷] و یک متخاصم $\mathcal{A}$[۵۸] تشریح می‌کنیم.

۱. چالشگر یک زوج کلید (vk, sk) را به صورت تصادفی تولید و کلید تاییدساز vk را به متخاصم می‌دهد.

۲. برای $i = ۱, ۲, \cdots, q_s$ وبرای بعضی q_s ،متخاصم برای امضای پیام m_i درخواستی به اوراکل امضا می‌فرستد و متعاقبا یک امضای σ_i دریافت می‌کند.

۳. در پایان، متخاصم زوج جعلی (m^*, σ^*) را به عنوان خروجی نمایش می‌دهد.

متخاصم این اجازه را دارد تا درخواست (m_j, σ_j) را در مرحله دوم برای اوراکل تایید/انکار ارسال کند و پاسخ اوراکل تایید/انکار به صورت زیر می‌باشد:

- اگر (m_j, σ_j) یک زوج معتبر باشد آنگاه اوراکل بیت $\mu = ۱$ را به عنوان خروجی برمی‌گرداند و اجرای پروتکل تایید π_{con} را با متخاصم در جریان می‌گذارد.

- در غیراینصورت، اوراکل بیت $\mu = ۰$ را برمی‌گرداند و بر این اساس پروتکل انکار π_{dis} را با متخاصم در جریان می‌گذارد.

گوییم متخاصم در جعل(قوی) موفق شده است اگر زوج (m^*, σ^*) معتبر باشد و این زوج در میان زوج‌های (m_i, σ_i) تولید شده در میان درخواست‌های امضای اوراکل نباشد. [۵۹]

تعریف ۱ . گوییم Σ قویا غیرقابل‌جعل است اگر احتمال آنکه متخاصم در جعل(قوی) موفق شود (برای هر متخاصم چندجمله‌ای احتمالاتی در بازی بالا)، ناچیز باشد.

---

[56] Security of Undeniable Signature

[57] challenger

[58] adversary

[۵۹] گوییم متخاصم در جعل(ضعیف) موفق شده است اگر (m^*, σ^*) معتبر باشد و m^* هرگز برای امضا از اوراکل درخواست نشده باشد.غیرقابل‌جعلی(ضعیف) و غیرقابل‌جعلی(قوی) یکی هستند اگرالگوریتم امضا قطعی باشد و درنتیجه برای هر پیام یک امضای منحصر به فرد وجود دارد که به درستی تایید می‌شود.

غیرقابل دسترس‌پذیری. دامگارد و پدرسون بوسیله بازی زیر بین چالشگر و متخاصم در [۹] به معرفی مفهوم غیرقابل‌دسترس‌پذیری پرداخته‌اند.

۱. چالشگر یک زوج کلید (vk, sk) را به صورت تصادفی تولید و کلید تاییدساز vk را به متخاصم می‌دهد.

۲. متخاصم مجاز است یک سری درخواست برای امضای پیام m_i به اوراکل امضا ارسال کند و امضای σ_i را دریافت کند.

۳. در برخی موارد، متخاصم یک پیام m^* را انتخاب و برای چالشگر ارسال می‌کند.

۴. چالشگر یک بیت تصادفی b را انتخاب می‌کند.

۵. اگر $b = 1$ آنگاه چالشگر امضای واقی $\sigma^* = Sign(sk, m^*)$ را محاسبه می‌کند. در غیر اینصورت امضای ساختگی(جعلی) $\sigma^* = Sim(sk, m^*)$ را محاسبه می‌کند. و در ادامه امضای σ^* را برای متخاصم برمی‌گرداند.

۶. متخاصم دوباره چند درخواست امضا را انجام می‌دهد.

۷. در انتهای بازی، متخاصم یک بیت حدسی b' را برمی‌گرداند.
متخاصم مجاز است در مراحل ۲ و ۵ ، درخواست (m_j, σ_j) را برای اوراکل تایید/انکار ارسال کند.

با این حال متخاصم اجازه ندارد تا چالش (m^*, σ^*) را در مرحله‌ی ۵ از اوراکل تایید/انکار درخواست کند. همچنین متخاصم مجاز نیست تا درخواست m^* را برای اوراکل امضا ارسال کند.
تعریف ۲. گوییم Σ غیرقابل‌دسترس است اگر برای هر متخاصم با زمان چندجمله‌ای احتمالاتی در بازی بالا، احتمال آن‌که $b = b'$ خیلی ناچیز باشد.

۳.۶ پروتکل

[60]

[60] Protocol

برای پیاده‌سازی این طرح امضا به روی خم‌های سوپرسینگولار لازم است تا عدد اول p به فرم $\ell_A^{e_A}\ell_M^{e_M}\ell_C^{e_C}\cdot f \pm ۱$ داشته باشیم و سپس یک خم بیضوی سوپرسینگولار E روی میدان $\mathbb{F}_{p^۲}$ معرفی کنیم چنانکه مرتبه‌ی خم ($\#E(\mathbb{F}_{p^۲})$) ، مقدار $(\ell_A^{e_A}\ell_M^{e_M}\ell_C^{e_C})^۲$ را عاد کند.همچنین لازم است تا مولدهای زیرگروه‌های $E[\ell_A^{e_A}]$ ، $E[\ell_M^{e_M}]$ و $E[\ell_C^{e_C}]$ را که به ترتیب شامل $\{P_A, Q_A\}$ ، $\{P_M, Q_M\}$ و $\{P_C, Q_C\}$ می‌باشد را نیز به دست آوریم.در طراحی این پروتکل معمولا نقاط $\{P_A, Q_A\}$ ساخت کلید و نقاط $\{P_M, Q_M\}$ برای داده‌ی پیام و نقاط $\{P_C, Q_C\}$ برای داده‌های تعهد مورد استفاده قرار می‌گیرند.

امضاکننده به صورت تصادفی دو عدد صحیح m_A و n_A را از میدان $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ انتخاب می‌کند($m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$) . و سپس زیرگروه $K_A = [m_A]P_A + [n_A]Q_A$ را به دست آورده و خم بیضوی $E_A = E/\langle K_A \rangle$ را محاسبه می‌کند. در انتها همسانی ϕ_A که از E به E_A می‌باشد ($\phi_A : E \to E_A$) را محاسبه می‌کند.

پارامترهای عمومی: p ، E ، $\{P_A, Q_A\}$ ، $\{P_M, Q_M\}$ ، $\{P_C, Q_C\}$ و تابع هش $H : \{۰, ۱\}^* \to \mathbb{Z}$.

کلید عمومی : E_A ، $\phi_A(P_C)$ و $\phi_A(Q_C)$.

کلید خصوصی : m_A و n_A .

برای امضای پیام M لازم است تا با استفاده از تابع هش به مقدار $h = H(M)$ دست بیابیم. هسته همسانی به شکل $K_M = P_M + [h]Q_M$ خواهد بود. در ادامه امضاکننده همسانی‌های زیر

- $\phi_M : E \to E_M = E/\langle K_M \rangle$
- $\phi_{M,AM} : E_M \to E_{AM} = E_M/\langle \phi_M(K_A) \rangle$
- $\phi_{A,AM} : E_A \to E_{AM} = E_A/\langle \phi_A(K_M) \rangle$

همراه با نقاط کمکی $\phi_{M,AM}(\phi_M(P_C))$ و $\phi_{MaAM}(\phi_M(Q_C))$ محاسبه می‌کند.امضاکننده سپس این دو نقطه کمکی را به همراه خم بیضوی E_{AM} به عنوان امضا منتشر می‌کند.(شکل ۵).

پروتکل تایید به شکل زیر انجام می‌شود.در ابتدا خم E_{AM} را بدون افشای همسانی‌های که آن را ساخته‌اند تایید می‌کنیم، برای این منظور خم E_{AM} را بوسیله همسانی ϕ_C کور می‌کنیم و سپس همسانی‌های کورشده را نمایش می‌دهیم.(شکل ۶).

۱. امضاکننده به صورت مخفی اعداد تصادفی m_C و n_C را از میدان $\mathbb{Z}/\ell_C^{ec}\mathbb{Z}$ انتخاب می‌کند($m_C, n_C \in \mathbb{Z}/\ell_C^{ec}\mathbb{Z}$) ، و نقطه $K_C = [m_C]P_C + [n_C]Q_C$ را به همراه خم‌ها و همسانی‌های شکل ۳ محاسبه می‌کند.چنانچه در شکل گویاست داریم:

- $E_C = E/\langle K_C \rangle$
- $E_{MC} = E_M/\langle \phi_M(K_C) \rangle = E_C/\langle \phi_C(K_M) \rangle$
- $E_{AC} = E_A/\langle \phi_A(K_C) \rangle = E_C/\langle \phi_C(K_A) \rangle$
- $E_{AMC} = E_{MC}/\langle \phi_{C,MC}(K_A) \rangle$

۲. امضاکننده خم‌های E_C ، E_{AC} ، E_{MC} ، E_{AMC} و همچنین $ker(\phi_{C,MC})$ را به عنوان تعهد منتشر می‌کند.

۳. تاییدکننده به طور تصادفی بیت $b \in \{۰, ۱\}$ را انتخاب می‌کند.

۴. اگر $b = ۰$ آنگاه امضاکننده $ker(\phi_C)$ را منتشر می‌کند. تاییدکننده به همراه کلیدعمومی امضاکننده $ker(\phi_{A,AC})$ را محاسبه می‌کند.با دانستن $ker(\phi_M)$ ،تاییدکننده می‌تواند $\phi_{M,MC}$ را محاسبه کند. همچنین تاییدکننده با کمک نقاط کمکی داده شده در امضا ، می‌تواند $\phi_{AM,AMC}$ را محاسبه کند. تاییدکننده همچنین هر نگاشت همسانی بین دو خم اشاره شده در تعهد را بررسی می‌کند. با اطلاع از $ker(\phi_C)$ ، همچنین به طور مستقل می‌تواند $\phi_{C,MC}$ را دوباره محاسبه و بررسی کند که آیا با تعهد ارائه شده همخوانی دارد یا نه.

۵. اگر $b = ۱$ آنگاه امضاکننده $ker(\phi_{C,AC})$ را نمایش می‌دهد. در ادامه تاییدکننده همسانی‌های $\phi_{MC,AMC}$ و $\phi_{AC,AMC}$ را محاسبه می‌کند و نگاشت‌های $\phi_{C,AC}$ ، $\phi_{MC,AMC}$ و $\phi_{AC,AMC}$ را بین دو خم معرفی شده متناظر در تعهد را بررسی می‌کند.

حال به تشریح پروتکل انکار می‌پردازیم. فرض کنید امضاکننده یک امضای جعلی (E_F, F_P, F_Q) برای پیام M ارائه کند، که E_F خم جعلی E_{AM} ، $\{F_P, F_Q\}$ نقاط کمکی جعلی به‌جای نقاط

معادل کمکی صحیح $\phi_{M,AM}(\phi_M(P_C))$ و $\phi_{M,AM}(\phi_M(Q_C))$ باشند.پس طبق طرح ارائه شده ما موظفیم تا خم E_F را بدون افشای خم E_{AM} ، انکار کنیم.بدین منظور قبل از به دست آوردن خم E_{AMC} ، خم E_{AM} را کور می‌کنیم. و اطلاعاتی به اندازه کافی در اختیار تاییدکننده می‌گذاریم تا بتواند خم E_{FC} را محاسبه و رابطه $E_{FC} \neq E_{AMC}$ را بررسی کند.

۱. امضاکننده به صورت مخفی اعداد تصادفی m_C و n_C را از میدان $\mathbb{Z}/\ell_C^{ec}\mathbb{Z}$ انتخاب می‌کند، و $K_C = [m_C]P_C + [n_C]Q_C$ را به همراه تمام خم‌ها و همسانی‌های نشان داده شده در شکل ۷ محاسبه می‌کند.

۲. امضاکننده خم‌های E_C ، E_{AC} ، E_{MC} و E_{AMC} را به همراه $ker(\phi_C)$ به عنوان تعهد منتشر می‌کند.

۳. تاییدکننده یک بیت تصادفی $b \in \{۰, ۱\}$ انتخاب می‌کند.

۴. اگر $b = ۰$ آنگاه امضاکننده $ker(\phi_C)$ را منتشر می‌کند. در ادامه تاییدکننده همسانی‌های ϕ_C ، $\phi_{M,MC}$ ، $\phi_{A,AC}$ را به‌همراه همسانی $\phi_F : E_F \to E_{FC} = E_F/\langle[m_C]F_P + [n_C]F_Q\rangle$ محاسبه کرده و هر نگاشت همسانی بین دوخ مشخص شده در در تعهد را بررسی می‌کند. تاییدکننده به طور مستقل همسانی $\phi_{C,MC}$ را محاسبه و بررسی می‌کند که آیا خروجی، همان همسانی ذکرشده در تعهد می‌باشد یا خیر

۵. اگر $b = ۱$ آنگاه امضاکننده $ker(\phi_{C,AC})$ را منتشر می‌کند و در ادامه تاییدکننده همسانی‌های $\phi_{AC,AMC}$ و $\phi_{MC,AMC}$ را محاسبه و بررسی میکند که آیا این همسانی‌ها نگاشتی به خم E_{AMC} دارند یا خیر.

شکل ۵: تولید امضا

$$
\begin{tikzcd}
 & E_A \arrow[rr, dashed, "\phi_{A,AC}"] \arrow[dd, dashed] & & E_{AC} \arrow[dd, dashed, "\phi_{AC,AMC}"] \\
E \arrow[ur, dashed, "\phi_A"] \arrow[rr, dashed, "\phi_C"] \arrow[dd, "\phi_M"] & & E_C \arrow[ur, dashed, "\phi_{C,AC}"] \arrow[dd, "\phi_{C,MC}"] & \\
 & E_{AM} \arrow[rr, dashed, "\phi_{AM,AMC}"] & & E_{AMC} \\
E_M \arrow[ur, dashed, "\phi_{M,AM}"] \arrow[rr, dashed, "\phi_{M,MC}"] & & E_{MC} \arrow[ur, dashed, "\phi_{MC,AMC}"] &
\end{tikzcd}
$$

شکل ۶: پروتکل تایید

$$
\begin{tikzcd}
 & E_A \arrow[rr, dashed, "\phi_{A,AC}"] \arrow[dd, dashed] & & E_{AC} \arrow[dd, dashed, "\phi_{AC,AMC}"] \\
E \arrow[ur, dashed, "\phi_A"] \arrow[rr, dashed, "\phi_C"] \arrow[dd, "\phi_M"] & & E_C \arrow[ur, dashed, "\phi_{C,AC}"] \arrow[dd, "\phi_{C,MC}"] & \\
 & E_{AM} \arrow[rr, dashed, "\phi_{AM,AMC}"] & & E_{AMC} \\
E_M \arrow[ur, dashed, "\phi_{M,AM}"] \arrow[rr, dashed, "\phi_{M,MC}"] & & E_{MC} \arrow[ur, dashed, "\phi_{MC,AMC}"] &
\end{tikzcd}
$$

شکل ۷: پروتکل انکار

۴.۶ اثبات‌های امنیت

۶۱

۱.۴.۶ پروتکل تایید

۶۲

[61] Security Proofs
[62] Confirmation Protocol

۲.۴.۶ پروتکل انکار

[63]

[63] Disavowal Protocol

۷ امضای کور غیرقابل انکار

۶۴

طرح امضای کور پروتکلی است که طی آن درخواست‌کننده بدون افشای محتوای سند از امضاکننده در خواست می‌کند تا سند را امضا کند.در سال ۱۹۸۲ اولین بار چام طرح امضای کور رامعرفی کرد. [۶] این طرح براساس مسئله RSA بنا شده است. [۲۲] از آنجا که اکثر طرح‌های امضای کور و تغییرات آن براساس سختی مسائل متفاوتی از جمله مسئله لگاریتم گسسته [۶۵] ، مسائل زوجیت‌مبنا [۶۶] و مسائل مشکبه‌مبنا [۶۷] ارائه شده است [۵، ۲۳، ۳۱] ، ولی تمام این طرح‌ها یک مشکل اساسی دارند و مشکل این است که در برابر متخاصم کوانتومی ایمن نمی‌باشند. امضاهای کور معرفی شده توسط چام [۶] ، کامنیش [۵] و ژانگ‌وکیم [۳۱] به دلیل الگوریتم شور [۶۸] در برابر حملات کوانتومی ایمن نیستند. چنان‌که در [۸] نشان داده شده است ، امضای کور مشبکه‌مبنای معرفی شده توسط روکرت [۲۳] که از مدل فیات‌شمیر [۱۱] استفاده می‌کند در برابر مدل اوراکل تصادفی کوانتومی ایمن [۶۹] نمی‌باشد.

امضای کور هر دو ویژگی ناشناس‌بودن [۷۰] و احرازهویت [۷۱] را در خود دارد. [۱۳، ۱۸] درنتیجه این طرح در بسیاری از پروتکل‌های حفظ حریم‌خصوصی [۷۲] ازجمله پول‌الکترونکی [۷۳] و رای‌گیری‌الکترونیکی [۷۴] استفاده می‌شود. [۱۹، ۲۰] چنان‌چه در ابتدا گفته‌شد امضاکننده هیچ کنترلی بر محتوای سندی که قرار است امضا شود را ندارد ، علاوه‌براین امضاکننده هیچ کنترلی در نحوه استفاده از امضا را هم ندارد. با این اوصاف احساس می‌شود اعطای درجه‌ای از کنترل به امضاکننده نیاز است. یک از راه‌های ممکن آن است که امضاکننده و درخواست کننده(امضا) روی بخشی از محتوای سند توافق کنند. این راه توسط تکنیکی که آبه و فوجیساکی در [۱] ارائه

[64] Undeniable Blind Signature

[65] Discrete Logarithm Problem (DLP)

[66] pairing-based problems

[67] lattice-based problems

[۶۸] در زمان چندجمله‌ای مسائل لگاریتم گسسته و تجزیه‌اعداد را در کامپیوترهای کوانتومی حل می‌کند

[69] quantum random oracle model

[70] anonymity

[71] authentication

[72] privacy-preserving

[73] e-cash

[74] e-voting

کرده‌اند قابل دستیابی می‌باشد.

راه دیگر آن است که این اختیار به امضاکننده داده شود تا تصمیم بگیرد چه‌کسی مجاز به تایید امضا می‌باشد.این روش ؟؟؟؟. طرح امضای غیرقابل‌انکار معرفی‌شده توسط چام و ون‌آنترپن [۷] دقیقا مطالب بالا [۷۵] را دربرمی‌گیرد.

بنابراین مطلوب است طرحی داشته باشیم که ناشناس‌بودن و تاییدسازی‌کنترل‌شده را درخود داشته باشد که ویژگی‌های هر دو طرح امضای کور و امضای غیرقابل‌انکار را برآورده کند. در سال ۱۹۹۶ ، ساکوری و یامانه [۲۴] یک طرح امضای کور غیرقابل‌انکار را براساس مساله لگاریتم گسسته ارائه دادند. چنان‌که در [۷] گفته‌شده‌است با این تکنیک می‌توان یک طرح امضای کور غیرقابل‌انکار بر اساس مسئله آر‌اس‌آ [۷۶] طراحی کرد. ذکر این نکته لازم است که تمام این طرح‌ها در برابر حملات کوانتموی ایمن نیستند.

در این پایان‌نامه در نظر داریم یک طرح امضای کور غیرقابل‌انکار مقاوم کوانتومی بر اساس سختی مسائل همسانی روی خم‌های بیضوی سوپرسینگولار ارائه کنیم.

سوخارو و همکارانش در [۲۶] پیشنهادی درباره‌ی ساخت یک طرح امضا با تاییدکننده معین‌شده براساس سختی مسائل همسانی که مقاوم کوانتومی نیز می‌باشد ارائه کرده است.آنها همچنین یک ساخت عمومی از طرح رمزگذاری تاییداعتبار کلید نامتقارن ؟؟؟ را نشان داده‌اند.جائو و سوخارو در [۱۵] یک طرح امضای غیرقابل‌انکار همسانی‌مبنا ارائه کرده‌اند.در این پایان‌نامه قصد داریم طرح جائو و سوخارو را به یک طرح امضای کور غیرقابل‌انکار توسعه دهیم.

۱.۷ تعریف استاندارد

[۷۷]

انتظار می‌رود طرح امضای کور غیرقابل‌انکار $(UBSS)$ [۷۸] ، ویژگی‌های طرح امضای غیرقابل‌انکار و طرح امضای کور را همزمان داشته باشد.در نتیجه این طرح باید ویژگی‌های ناخوانابودن محتوای پیام‌اولیه(قبل از امضا) [۷۹] و تاییدسازی کنترل شده [۸۰] را دارا باشد.

[۷۵] در یک طرح امضای غیرقابل‌انکار، امضاکننده تصمیم می‌گیرد تا چه کسی امضا را تایید کند

[76] RSA

[77] Formal Definition

[78] Undeniable Blind Signature Scheme

[79] anonoymity of the message origination

[80] controlled verification

تعریف۱. طرح امضای کور غیرقابل‌انکار ، یک طرح امضای تعاملاتی است که‌بوسیله چندتایی زیر معرفی می‌شود:

$$\mathcal{UBSS} = \big(KeyGen, Blind, Sign, Unblind, Check, \mathcal{CON}, \mathcal{DIS}\big)$$

۱. الگوریتم تولید کلید تصادفی $KeyGen$ ، پارامتر امنیتی ۱^λ را به عنوان ورودی گرفته و زوج کلیدهای (vk, sk) را که به عنوان کلیدتاییدساز و کلیدمخفی نامیده می‌شوند، به عنوان خروجی تولید می‌کند. شکل شماتیک این الگوریتم به‌صورت زیر می‌باشد:

$$(vk, sk) \longleftarrow KeyGen(۱^\lambda)$$

۲. الگوریتم کورسازی تصادفی $Blind$ ، پیام m را به عنوان ورودی گرفته و خروجی آن کورشده‌ی پیام، یعنی m' می‌باشد. شکل شماتیک این الگوریتم به شکل زیر می‌باشد که r کاملا به صورت تصادفی توسط الگوریتم ساخته می‌شود:

$$m' \longleftarrow {}_r Blind(m)$$

۳. الگوریتم امضای قطعی یا تصادفی $Sign$ ، کلید مخفی sk و پیام m را به عنوان ورودی گرفته و امضای σ را به عنوان خروجی تولید می‌کند. این الگوریتم را می‌توان به صورت زیر نشان داد:

$$\sigma \longleftarrow Sign_{sk}(m)$$

۴. الگوریتم شفاف‌ساز قطعی $Unblind$ ،امضای کور σ' و عددتصادفی r (انتخاب شده توسط الگوریتم کورسازی) را به عنوان ورودی گرفته و امضای شفاف σ را به عنوان خروجی تولید می‌کند. این الگوریتم را می‌توان به شکل زیر نمایش داد:

$$\sigma := Unblind_r(\sigma')$$

۵. الگوریتم قطعی بررسی $Check$ ، پیام m ، امضای شفاف σ و زوج کلیدهای (vk, sk) را به عنوان ورودی گرفته و بیت b را به عنوان خروجی تولید می‌کند. $b = ۱$ به معنای آن است

که امضا متعلق به پیام می‌باشد و $b = ۰$ نیز به این معناست که امضا غیرمعتبر می‌باشد. این الگوریتم به‌صورت زیر قابل نمایش است:

$$b := Check_{(vk,sk)}(m, \sigma)$$

۶. پروتکل تایید π_{con} توسط امضاکننده اجرا می‌شود تا تاییدکننده اطمینان یابد که امضا معتبر است.

۷. پروتکل انکار π_{dis} نیز توسط امضاکننده اجرا می‌شود و تاییدکننده متقاعد می‌شود که امضا نامعتبر است.

برای هر زوج کلید (vk, sk) که توسط الگوریتم $KeyGen(۱^\lambda)$ تولید می‌شود و همچنین هر m از میان فضای پیام و هر عددتصادفی r که توسط الگوریتم $Blind$ تولید شده است، باید تساوی زیر برقرار باشد:

$$Check_{(vk,sk)}(m, Unblind_r(Sign_{sk}({}_rBlind(m)))) = ۱$$

علاوه‌براین، اگر الگوریتم امضا قطعی باشد آنگاه می‌توان فرض کرد اثر مراحل الگوریتم‌های کورسازی_امضا_شفافیت روی پیام دقیقا مشابه اجرای مستقیم الگوریتم امضا روی پیام می‌باشد. برای درک این مطلب آن را به صورت زیر نمایش می‌دهیم:

$$Unblind_r(Sign_{sk}({}_rBlind(m))) = Sign_{sk}(m)$$

۲.۷ کارکرد UBSS

[۸۱]

برای درک بهتر نقش الگوریتم‌های گفته شده در بخش قبلی، پروتکل را به صورت کامل اجرا می‌کنیم.

در ابتدا امضاکننده یک پارامتر امنیتی λ را انتخاب و الگوریتم $KeyGen(1^{\lambda})$ را برای به دست آوردن زوج کلید (vk, sk) اجرا می‌کند.کلید امضای sk به صورت مخفی پیش امضاکننده حفظ می‌شود و کلید تاییدساز vk توسط امضاکننده منتشر می‌شود. m پیامی است که درخواست‌کننده خواهان امضای آن به صورت ناخوانا است؟؟. به این منظور ، درخواست‌کننده ابتدا m[۸۲] را با اجرای الگوریتم $Blind(m)$ به m'[۸۳] تبدیل می‌کند. [۸۴] درادامه درخواست‌کننده m' را به همراه شناسه هویتی خود Id_R ، ارسال می‌کند. امضاکننده ابتدا شناسه درخواست‌کننده را تایید (۲.۷) و سپس الگوریتم $Sign_{sk}$ را روی m' اجرا می‌کند تا امضای کور σ' به دست آید. دریافت‌کننده پس از دریافت امضای کور از امضاکننده ، توسط الگوریتم $Unblind$ و مقدارتصادفی r انتخاب شده در مرحله کورسازی، امضا را از حالت کور خارج کرده وسپس زوج پیام اصلی و امضای شفاف (m, σ) پیام رابرای بخش تایید؟؟ ارسال می‌کند.

هربخشی که خواهان تایید امضا باشد، شناسه خود Id_V را به همراه زوج پیام و امضا (m, σ) برای امضاکننده ارسال می‌کند. امضاکننده در ابتدا شناسه تاییدکننده را بررسی می‌کند(۲.۷) آنگاه اگر Id_V یک شناسه معتبر در میان تاییدکنندگان احراز شده(مجاز) نباشد، امضاکننده از ادامه ارتباط خودداری می‌کند. در غیراینصورت الگوریتم بررسی $Check$ را اجرا می‌کند. اگر خروجی این الگوریتم معتبر باشد آنگاه پروتکل تایید $\mathcal{CON}$ توسط امضاکننده آغاز می‌شود؛ درغیراینصورت پروتکل انکار $\mathcal{DIS}$ اجرا می‌شود(شکل ۸ تمام مفاهیم طرح $UBSS$ را نشان می‌دهد).

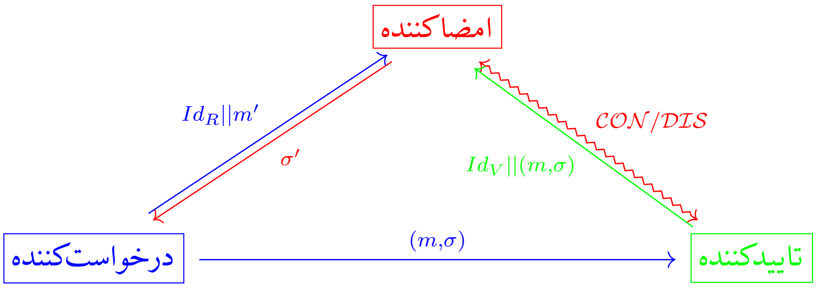
---

[81] Workinf of UBSS

[۸۲] پیام خوانا

[۸۳] پیام ناخوانا

[۸۴] در زمان اجرای الگوریتم ، یک انتخاب تصادفی r توسط خود الگوریتم تولید می‌شود.



شکل ۸: اطلاعات کامل طرح امضای کور غیرقابل‌انکار

توجه۱. در این پایان‌نامه عمدا چگونگی احراز هویت بین درخواست‌کننده و تاییدکننده با امضاکننده را مشخص نمی‌کنیم. این امر مستلزم آشنایی با احراهویت متقابل می‌باشد. این طرح در [۳، ۱۲] به‌صورت کامل آورده شده است که در مقابل حملات کوانتومی نیز ایمن می‌باشند.

۳.۷ ویژگی‌ها [۸۵]

[85] Properties

مراجع

[1] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 244–251. Springer, 1996.

[2] Adrian Antipa, Daniel Brown, Robert Gallant, Rob Lambert, René Struik, and Scott Vanstone. Accelerated verification of ecdsa signatures. In *International Workshop on Selected Areas in Cryptography*, pages 307–318. Springer, 2005.

[3] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Annual Cryptology Conference*, pages 361–379. Springer, 2013.

[4] Reinier Bröker. Constructing supersingular elliptic curves. *J. Comb. Number Theory*, 1(3):269–273, 2009.

[5] Jan L Camenisch, Jean-Marc Piveteau, and Markus A Stadler. Blind signatures based on the discrete logarithm problem. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 428–432. Springer, 1994.

[6] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[7] David Chaum and Hans Van Antwerpen. Undeniable signatures. In *Conference on the Theory and Application of Cryptology*, pages 212–216. Springer, 1989.

[8] Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The fiat–shamir transformation in a quantum world. In *International Confer-*

ence on the Theory and Application of Cryptology and Information Security, pages 62–81. Springer, 2013.

[9] Ivan Damgård and Torben Pedersen. New convertible undeniable signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 372–386. Springer, 1996.

[10] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[11] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 186–194. Springer, 1986.

[12] Sebastianus A Goorden, Marcel Horstmann, Allard P Mosk, Boris Škorić, and Pepijn WH Pinkse. Quantum-secure authentication of a physical unclonable key. *Optica*, 1(6):421–424, 2014.

[13] Min-Shiang Hwang, Cheng-Chi Lee, and Yan-Chi Lai. An untraceable blind signature scheme. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 86(7):1902–1906, 2003.

[14] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *International Workshop on Post-Quantum Cryptography*, pages 19–34. Springer, 2011.

[15] David Jao and Vladimir Soukharev. Isogeny-based quantum-resistant undeniable signatures. In *International Workshop on Post-Quantum Cryptography*, pages 160–179. Springer, 2014.

[16] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.

[17] Jeffrey C Lagarias. Effective versions of the chebotarev density theorem. In *Algebraic number fields: L-functions and Galois properties (Proc. Sympos., Univ. Durham, Durham, 1975)*. Academic Press, 1977.

[18] Cheng-Chi Lee, Min-Shiang Hwang, and Wei-Pang Yang. A new blind signature based on the discrete logarithm problem for untraceability. *Applied Mathematics and Computation*, 164(3):837–841, 2005.

[19] Chun-Ta Li, Min-Shiang Hwang, and Chi-Yu Liu. An electronic voting protocol with deniable authentication for mobile ad hoc networks. *Computer Communications*, 31(10):2534–2540, 2008.

[20] Iuon-Chang Lin, Min-Shiang Hwang, and Chin-Chen Chang. Security enhancement for anonymous secure e-voting over a network. *Computer Standards & Interfaces*, 25(2):131–139, 2003.

[21] Peter L Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987.

[22] RL Rivest and B Kaliski. Rsa problem, encyclopedia of cryptography and security, 2005.

[23] Markus Rückert. Lattice-based blind signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 413–430. Springer, 2010.

[24] Kouichi Sakurai and Yoshinori Yamane. Blind decoding, blind undeniable signatures, and their applications to privacy protection. In *Inter-*

national Workshop on Information Hiding, pages 257–264. Springer, 1996.

[25] Jerome A Solinas. Low-weight binary representations for pairs of integers. 2001.

[26] Vladimir Soukharev, David Jao, and Srinath Seshadri. Post-quantum security models for authenticated encryption. In *Post-Quantum Cryptography*, pages 64–78. Springer, 2016.

[27] Edlyn Teske. The pohlig–hellman method generalized for group structure computation. *Journal of Symbolic Computation*, 27(6):521–534, 1999.

[28] Edlyn Teske. The pohlig–hellman method generalized for group structure computation. *Journal of Symbolic Computation*, 27(6):521–534, 1999.

[29] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 755–784. Springer, 2015.

[30] Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris, Séries A*, 273:305–347, 1971.

[31] Fangguo Zhang and Kwangjo Kim. Id-based blind signature and ring signature from pairings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 533–547. Springer, 2002.