۱ امضای غیرقابل انکار

[۳] [۱] مفهوم طرح امضای غیرقابل انکار اولین بار توسط چام و آنترپن [۱] 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 [۲] ،پول الکترونیکی [۳] ، رای‌گیری الکترونیکی [۴] و حراج الکترونیکی [۵] اشاره کرد.

۱.۱ تعریف

مطابق با تعریف رسمی ارائه شده در [۲] ، 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 [۶] می‌باشد که خروجی آن زوج کلید (vk, sk) می‌باشد که vk یک کلید تاییدساز و sk یک کلید امضا [۷] می‌باشد.
فضای پیام $\mathcal{M}$ توسط vk مشخص شده است.

[1] Undeniable Signature
[2] licensing software
[3] electronic cash
[4] voting electronic
[5] electronic auction
[6] PPT(probabilistic polynomial-time)
[۷] 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 نامعتبر 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 [۸] و غیرقابل دسترس‌پذیری (نامرئی بودن) [۹] 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 [۱۰] 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.

[8] unforgeability
[9] invisibility
[10] zero-knowledge interactive proof system (ZKIP)

۲.۱ امنیت امضای غیرقابل انکار

[11]

غیرقابل جعل بودن . مفهوم غیرقابل جعل بودن را توسط بازی زیر بین یک چالشگر $\mathcal{CH}$[12] و یک متخاصم $\mathcal{A}$[13] 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. [14]

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.

---

[11] Security of Undeniable Signature

[12] challenger

[13] adversary

[14] 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 [۲] 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 $b = ۱$ 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.

۳.۱ پروتکل

۱۵

[15] Protocol

برای پیاده‌سازی این طرح امضا به روی خم‌های سوپرسینگولار لازم است تا عدد اول p به فرم $p = \ell_A^{e_A}\ell_M^{e_M}\ell_C^{e_C} \cdot f \pm 1$ داشته باشیم و سپس یک خم بیضوی سوپرسینگولار E روی میدان $\mathbb{F}_{p^2}$ معرفی کنیم چنانکه مرتبه‌ی خم ($\#E(\mathbb{F}_{p^2})$) ، مقدار $(\ell_A^{e_A}\ell_M^{e_M}\ell_C^{e_C})^2$ 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 $H : \{0,1\}^* \to \mathbb{Z}$.

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 ۱).

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 ۲).

۱. امضاکننده به صورت مخفی اعداد تصادفی m_C و n_C را از میدان $\mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$ انتخاب می‌کند($m_C, n_C \in \mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$) ، 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 $\mathbb{Z}/\ell_C^{e_C}\mathbb{Z}$ 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 ۳ محاسبه می‌کند.

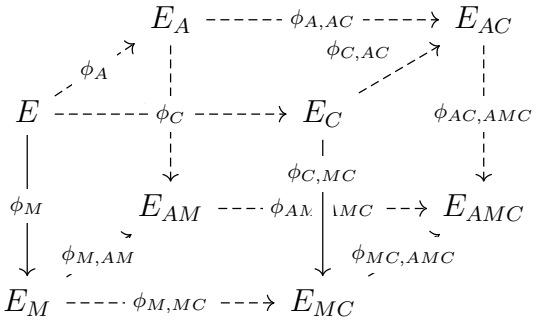
۲. امضاکننده خم‌های E_C ، E_{AC} ، E_{MC} و E_{AMC} را به همراه $ker(\phi_C)$ به عنوان تعهد منتشر می‌کند.

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

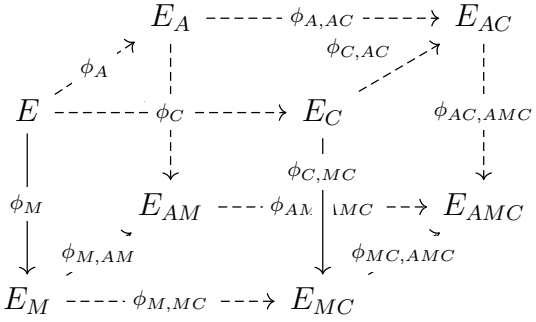
۴. اگر $b = ۰$ آنگاه امضاکننده $ker(\phi_C)$ را منتشر می‌کند. در ادامه تاییدکننده همسانی‌های ϕ_C ، $\phi_{M,MC}$ ، $\phi_{A,AC}$ را به‌همراه همسانی $\phi_F : E_F \to E_{FC} = E_F/\langle [m_C]F_P + [n_C]F_Q \rangle$ محاسبه کرده و هر نگاشت همسانی بین دوخ مشخص شده در در تعهد را بررسی می‌کند. تاییدکننده 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 ۱: تولید امضا



شکل ۲: پروتکل تایید



شکل ۳: پروتکل انکار

۴.۱ اثبات‌های امنیت

۱۶

۱.۴.۱ پروتکل تایید

۱۷

[16]Security Proofs

[17]Confirmation Protocol

۲.۴.۱ پروتکل انکار

۱۸

[18] Disavowal Protocol

مراجع

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

[2] 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.

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