۱

B۰۶۶A"۰۶۶"۹F۰۶"۸F۰۶"۷F۰۶"۶F۰۶"۵F۰۶"۴F۰۶"۳F۰۶"۲F۰۶"۱F۰۶"۰F۰۶"

فصل ۱

همسانی بین خم‌های بیضوی

۱.۱ خم بیضوی

فرض کنید k یک میدان بوده و $f(x,y)$ یک چندجمله‌ای با ضرایب روی میدان k باشد، هم‌چنین فرض کنید:

$$C = \{(x,y) \in k^2 \mid f(x,y) = 0\}$$

نقطه‌ی $p = (x,y) \in C$ را نقطه‌ی تکین(منفرد) خم C گوییم هرگاه: $\frac{\partial f}{\partial x}(p) = \frac{\partial f}{\partial y}(p) = 0$ درغیراینصورت p را نامنفرد گوییم.
اگر خم C هیچ نقطه‌ی تکینی نداشته باشد، هموار نامیده می‌شود.

مثال ۱. نقاط تکین خم با معادله‌ی $f(x,y) = y^2 - x^3 + 3x$ را در صورت وجود بیابید.

$$\begin{cases} \frac{\partial f}{\partial x} = -3x + 3 = 0 \longrightarrow x = \pm 1 \\ \frac{\partial f}{\partial y} = 2y = 0 \longrightarrow y = 0 \end{cases}$$

پس $P(1,0)$ و $Q(-1,0)$ کاندیدای نقطه منفرد خم هستند. اما به‌راحتی می‌توان دید که هیچ کدام از نقاط P و Q روی خم قرار ندارند پس این خم نقطه تکین ندارد و لذا هموار است.

تعریف ۱.۱.۱. فرض کنید K یک میدان باشد، معادله‌ی

$$y^2 + a_1xy + a_3 = x^3 + a_2x^2 + a_4x + a_6$$

که درآن $a_1, a_2, a_3, a_4, a_6 \in K$ یک معادله‌ی وایرشتراس نامیده می‌شود.

تعریف ۲.۱.۱. هر خم هموار با معادله‌ی وایرشتراس بالا یک خم بیضوی نامیده می‌شود.

تعریف ۳.۱.۱. معادله‌ی $y^2 = x^3 + Ax + B$ معادله‌ی وایرشتراس کوتاه نامیده می‌شود.
نمودار خم با معادله‌ی وایرشتراس $y^2 = x^3 + 1$ در میدان R به صورت زیر است:

تعریف ۴.۱.۱. مبین یک خم وایرشتراس کوتاه شده به‌صورت زیر می‌باشد:

$$\Delta = -(4A^3 + 27B^2)$$

توجه ۱. خم $y^2 = x^3 + Ax + B$ هموار است اگر و تنها اگر $\Delta \neq 0$. بنابراین یک خم بیضوی را می‌توان خمی با معادله‌ی وایرشتراس بالا و مبین غیرصفر تعریف کرد.

۲.۱ jـ پایای یک خم

در این بخش فرض می‌کنیم k میدانی با مشخصه‌ی مخالف ۲ و ۳ است. همچنین فرض می‌کنیم خم E دارای معادله‌ی وایرشتراس به فرم زیر باشد:

$$E : y^2 = x^3 + Ax + B$$

مبین این خم عبارت است از:

$$\Delta = -(4A^3 + 27B^2)$$

اکنون jـ پایای خم E را به‌صورت زیر تعریف می‌کنیم:

$$j(E) = -1728\frac{4A^3}{\Delta}$$

مثال ۲. فرض کنید

$$E_۱ : y^۲ = x^۳ + x + ۱$$

$$E_۲ : y^۲ = x^۳ + ۴x + ۸$$

در این صورت:

در خم $E_۱$:

$$A = ۱, B = ۱ \Longrightarrow \Delta(E_۱) = -(۴A^۳ + ۲۷B^۲) = -۳۱$$

$$j(E_۱) = -۱۷۲۸\frac{۴A^۳}{\Delta} = -۱۷۲۸\frac{۴}{-۳۱} = \frac{۶۹۱۲}{۳۱}$$

در خم $E_۲$:

$$A = ۴, B = ۸ \Longrightarrow \Delta(E_۲) = -(۴A^۳ + ۲۷B^۲) = -۱۹۸۴$$

$$j(E_۲) = -۱۷۲۸\frac{۴A^۳}{\Delta} = -۱۷۲۸\frac{۲۵۶}{-۱۹۸۴} = \frac{۴۴۲۳۶۸}{۱۹۸۴}$$

همان‌گونه که ملاحظه می‌شود، خم‌های $E_۱$ و $E_۲$ دارای jـ پایای برابرند.

۱.۳ یکریختی خم‌های بیضوی

دو خم $E : y^۲ = x^۳ + Ax + B$ و $E' : y^۲ = x^۳ + A'x + B'$ را یکریخت گوییم هرگاه $\mu \in \bar{K}^*$ موجود باشد که

$$A' = \mu^۴ A\ ,\ B' = \mu^۶ B$$

مثال ۳. فرض کنید E و E' دو خم روی Q با معادلات زیر باشند:

$$E : y^۲ = x^۳ + ۳x + ۵\ ,\ E' : y^۲ = x^۳ + ۱۲x + ۴۰$$

دراینصورت $A = ۳$ ، $B = ۵$ ، $A' = ۱۲$ ، $B' = ۴۰$ ، با قرار دادن $\mu = \sqrt{۲}$ داریم:

$$A' = \mu^۴\ ,\ B' = \mu۶B$$

پس خم‌های E و E' روی Q یکریخت نیستند درحالیکه روی $\bar{Q}$ (یا روی R یا روی $Q\sqrt{۲}$) یکریخت‌اند.

قضیه ۱ . فرض کنید K یک میدان با مشخصه‌ی مخالف ۲ و ۳ باشد و

$$E : y^۲ = x^۳ + Ax + B \ , \ E' : y^۲ = x^۳ + A'x + B'$$

دو خم روی K باشند، در اینصورت:

$$E \text{ و } E' \text{ یکریخت‌اند اگر وتنها اگر } j(E) = J'(E')$$

۴.۱ درون‌ریختی خم‌های بیضوی

اگر E یک خم بیضوی روی میدان K باشد انگاه $\varphi : E(\bar{K}) \to E(\bar{K})$ یک درون‌ریختی است هرگاه:

۱. *φ توسط تابع گویا بیان شده باشد، یعنی: $\varphi(x,y) = (\varphi_۱(x,y), \varphi_۲(x,y))$ که درآن $\varphi_۱$ و $\varphi_۲$ توابع گویا هستند.*

۲. *برای هر P و Q : $\varphi(P+Q) = \varphi(P) + \varphi(Q)$*

۳. $\varphi(\infty) = \infty$

۵.۱ خم‌های بیضوی روی میدان‌های متناهی

فرض کنید E یک خم بیضوی $y^۲ = x^۳ + Ax + B$ روی میدان $\mathbb{F}_q$ باشد. در این‌صورت $\mathbb{F}_q$ـ نقاط روی خم عبارتنداز:

$$E(\mathbb{F}_q) = \{(x,y) \in \mathbb{F}_q \times \mathbb{F}_q | y^۲ = x^۳ + Ax + B\} \cup \{\infty\}$$

واضح است که:

$$E(\mathbb{F}_q) \subseteq (\mathbb{F}_q \times \mathbb{F}_q) \cup \{\infty\}$$

و چون مجموعه‌ی سمت راست متناهی (از مرتبه‌ی $q^۲ + ۱$) است لذا مجموعه‌ی چپ یعنی $E(\mathbb{F}_q)$ نیز متناهی است. پس: $\#E(\mathbb{F}_q) \leq q^۲ + ۱$ بنابراین خم‌های بیضوی روی میدان‌های متناهی، متناهی اند.

قضیه هسه کرانی برای تعداد عناصر $E(\mathbb{F}_q)$ معرفی می‌کند.

قضیه ۲. اگر E یک خم بیضوی روی میدان $\mathbb{F}_q$ باشد آنگاه:

$$|q + ۱ - \#E(\mathbb{F})| \leq ۲\sqrt{q}$$

به‌عبارت دیگر:

$$-۲\sqrt{q} \leq \#E(\mathbb{F}_q) - (q + ۱) \leq +۲\sqrt{q}$$

$$(q + ۱) - ۲\sqrt{q} \leq \#E(\mathbb{F}_q) \leq (q + ۱) + ۲\sqrt{q}$$

$$(\sqrt{q} - ۱)^۲ \leq \#E(\mathbb{F}_q) \leq (\sqrt{q} + ۱)^۲$$

مثال ۴. خم بیضوی با معادله‌ی وایرشتراس $y^۲ = x^۳ + Ax + B$ روی میدان $\mathbb{F}_{۴۹}$ را درنظر بگیرید. در این‌صورت:

$$۴۹ + ۱ - ۲\sqrt{۴۹} \leq \#E(\mathbb{F}_{۴۹}) \leq ۴۹ + ۱ + ۲\sqrt{۴۹}$$

$$۳۶ \leq \#E(\mathbb{F}_{۴۹}) \leq ۶۴$$

تعریف ۱.۵.۱. $a_q = q + ۱ - \#E(\mathbb{F}_q)$ ، اثر خم نامیده می‌شود.

بنا به قضیه هسه، $|a_q| \leq ۲\sqrt{q}$ یعنی $-۲\sqrt{q} \leq a_q \leq ۲\sqrt{q}$ ، بنابراین در مثال قبل: $-۱۴ \leq a_{۴۹} \leq ۴۹$.

۶.۱ نقاط تابی در خم‌های بیضوی

فرض کنید E یک خم بیضوی روی میدان $\mathbb{F}_q$ باشد. برای عدد صحیح n ، درون‌ریختی زیر را در نظر می‌گیریم:

$$[n] : E(\bar{\mathbb{F}}_q) \longrightarrow E(\bar{\mathbb{F}}_q)$$

هسته‌ی این درون‌ریختی عبارت است از: $ker([n]) = \{p \in E(\bar{\mathbb{F}_q}) \mid [n]p = \infty\}$ این مجموعه با $E[n]$ نمایش داده می‌شود، به‌عبارت دیگر:

$$E[n] = \{p \in E(\bar{\mathbb{F}_q}) \mid np = \infty\}$$

به‌راحتی می‌توان دید $E[n]$ زیرگروهی از $E(\bar{\mathbb{F}_q})$ است.

قضیه ۳ . *اگر E یک خم بیضوی رو میدان $\mathbb{F}_q$(q توانی از یک عدد اول) باشد آنگاه برای هر n،*

$$\#E(\mathbb{F}_{q^n}) = q + ۱ - (\alpha^n + \beta^n)$$

که در آن α و β ریشه‌های چندجمله‌ای زیر هستند:

$$x^۲ - a_q x + q = ۰$$

مثال ۵ . فرض کنید خم E روی میدان $\mathbb{F}_۴$ توصیف شده باشد و $\#E(\mathbb{F}_۴) = ۶$. حال می‌خواهیم $\#E(\mathbb{F}_{۱۶})$ را به‌دست آوریم، بنابراین:

$$q = ۴, a_q = q + ۱ - \#E(\mathbb{F}_q) \Longrightarrow a_۴ = ۴ + ۱ - ۶ = -۱$$

در ادامه ریشه‌های چندجمله‌ای $x^۲ - a_q x + q = ۰$ را به‌دست می‌آوریم:

$$x^۲ - (-۱)x + ۴ = ۰ \longrightarrow x^۲ + x + ۴ = ۰, \quad \Delta = b^۲ - ۴aC = ۱ - ۱۶ = -۱۵$$

$$\text{ریشه‌ها} = \frac{-b \pm \sqrt{\Delta}}{۲a} = \frac{-۱ \pm \sqrt{-۱۵}}{۲} \Longrightarrow \begin{cases} \alpha = \frac{-۱-\sqrt{-۱۵}}{۲} \\ \beta = \frac{-۱+\sqrt{-۱۵}}{۲} \end{cases}$$

بنابراین

$$\#E(\mathbb{F}_{۱۶}) = ۱۶ + ۱ - (\alpha^۲ + \beta^۲) = ۱۶ + ۱ - (-۷) = ۲۴$$

قضیه ۴ . *فرض کنید E یک خم بیضوی روی میدان k باشد، دراینصورت:*

۱ . *اگر $char(k) = ۰$ یا $char(k) = p$ که در آن $p \nmid n$، آنگاه:*

$$E[n] \cong \mathbb{Z}_n \times \mathbb{Z}_n$$

۲. اگر $char(k) = p$ و $p \mid n$، آنگاه:

$$E[n] \cong \mathbb{Z}_m \times \mathbb{Z}_m$$

یا

$$E[n] \cong \mathbb{Z}_n \times \mathbb{Z}_m$$

که در آن $n = p^e.m$ و $(p, m) = ۱$

مثال ۶. فرض کنید E یک خم بیضوی روی میدان $\mathbb{F}_۹$ باشد. ساختار $E[۵]$، $E[۶]$ و $E[۶۷۵]$ را به‌صورت زیر به‌دست می‌آوریم:

محاسبه‌ی $E[۵]$:

$char(\mathbb{F}_۹) = ۳$ بنابراین چون $۵ \nmid ۳$ لذا:

$$E[۵] \cong \mathbb{Z}_۵ \times \mathbb{Z}_۵$$

محاسبه‌ی $E[۶]$:

$$p = ۳ \mid ۶ = n \longrightarrow n = ۳ \times ۲ \Rightarrow E[۶] \cong \mathbb{Z}_۲ \times \mathbb{Z}_۲ \text{ یا } \mathbb{Z}_۶ \times \mathbb{Z}_۲$$

محاسبه‌ی $E[۶۷۵]$:

$$p = ۳ \mid ۶۷۵ = n \longrightarrow n = ۳^۳ \times ۲۵ \Rightarrow E[۶۷۵] \cong \mathbb{Z}_{۲۵} \times \mathbb{Z}_{۲۵} \text{ یا } \mathbb{Z}_{۶۷۵} \times \mathbb{Z}_{۲۵}$$

مثال ۷. فرض کنید k میدانی با مشخصه‌ی p باشد. ساختار $E[p]$ به‌صورت زیر خواهد بود:

$$n = p\ ,\ p \mid n \Longrightarrow n = p^۱ \times ۱ \Longrightarrow m = ۱$$

$$\begin{cases} E[p] \cong \mathbb{Z}_۱ \times \mathbb{Z}_۱ = \{(۰, ۰)\} \Longrightarrow E[p] = \{\infty\} \\ E[p] \cong \mathbb{Z}_p \times \mathbb{Z}_p \end{cases}$$

تعریف ۱.۶.۱. اگر E یک خم بیضوی روی میدان k (با مشخصه‌ی p) باشد و $E[p] = \infty$، آنگاه خم را سوپرسینگولار گوییم. در غیراینصورت ($E[p] \cong \mathbb{Z}_p$) خم را معمولی گوییم.

قضیه ۵. فرض کنید E یک خم بیضوی روی میدان $\mathbb{F}_q$ (q توانی از عدد اول p) باشد، دراینصورت:

$$E \text{ سوپرسینگولار است اگر و تنها اگر } a_q \equiv 0 \pmod p$$

توجه ۲.

$$a_q = q + 1 - \#E(\mathbb{F}_q)$$

$$\begin{aligned} a_q \equiv 0 \pmod p &\iff q + 1 - \#E(\mathbb{F}_q) \equiv 0 \pmod p \\ &\iff 1 - \#E(\mathbb{F}_q) \equiv 0 \pmod p \\ &\iff \#E(\mathbb{F}_q) \equiv 1 \pmod p \end{aligned}$$

۷.۱ زوجیت وایل

فرض کنید k یک میدان و n عددی صحیح باشد بطوریکه $char(k) = 0$ یا $char(k) = p \nmid n$ همچنین فرض کنید $E : y^2 = x^3 + Ax + B$ خم بیضوی روی میدان k باشد. در اینصورت نگاشت

$$e_n : E[n] \times E[n] \longrightarrow \mu_n$$

با خواص زیر وجود دارد:

۱. e_n دوخطی است. بدین معنی که:

(آ) $\forall S_1, S_2, T \in E[n] \ : \ e_n(S_1 + S_2, T) = e_n(S_1, T)e_n(S_2, T)$

(ب) $\forall S, T_۱, T_۲ \in E[n] \ : \ e_n(S, T_۱ + T_۲) = e_n(S, T_۱)e_n(S, T_۲)$

لم ۱. اگر $S \in E[n]$ آنگاه $e_n(S, \infty) = ۱$

۲. e_n ناتباهیده است. بدین معنی که :

(آ) اگر $S \in E[n]$ چنان باشد که برای هر $T \in E[n]$، $en(S, T) = ۱$ آنگاه $S = \infty$.

(ب) اگر $T \in E[n]$ چنان باشد که برای هر $S \in E[n]$، $en(S, T) = ۱$ آنگاه $T = \infty$.

۳. برای هر $S \in E[n]$:

$$e_n(S, S) = ۱$$

۴. برای هر $S, T \in E[n]$:

$$e_n(T, S) = e_n(S, T)^{-۱}$$

۵. اگر $\sigma : \bar{k} \to \bar{k}$ یک خودریختی باشد آنگاه برای هر $S, T \in E[n]$:

$$e_n(\sigma(S), \sigma(T)) = \sigma(e_n(S, T))$$

۶. اگر $\varphi : E(\bar{k}) \to E(\bar{k})$ یک درون‌ریختی جداپذیر باشد، آنگاه برای هر $S, T \in E[n]$:

$$e_n(\varphi(S), \varphi(T)) = e_n(S, T)^{deg\varphi}$$

تعریف ۱.۷.۱. نگاشت $e_n \ : \ E[n] \times E[n] \ \longrightarrow \ \mu_n$ تعریف شده در بالا را زوجیت وایل می‌گوییم.

۸.۱ همسانی

فرض کنید $E_۱ : {y_۱}^۲ = {x_۱}^۲ + A_۱x_۱ + B_۱$ و $E_۲ : {y_۲}^۲ = {x_۲}^۲ + A_۲x_۲ + B_۲$ دو خم تعریف شده روی میدان k با مشخصه‌ی مخالف ۲ و ۳ باشند، در این‌صورت یک همسانی از $E_۱$ به $E_۲$ یک همریختی به شکل $\varphi : E_۱(\bar{k}) \longrightarrow E_۲(\bar{k})$ است که توسط توابع گویا تعریف شده باشد. بنابراین توابع گویای $R_۱(x,y)$ و $R_۲(x,y)$ وجود دارند بطوریکه:

$$\varphi : (R_۱(x,y), R_۲(x,y))$$

یک همسانی را می‌توان به‌شکل $\varphi : (r_۱(x), r_۲(x)y)$ که $r_۱(x)$ و $r_۲(x)$ توابع گویا هستند نیز نشان داد.

توجه ۳. φ یک همریختی است یعنی:

$$\forall P, Q \in E_۱(\bar{X}) \ : \ \varphi(P+Q) = \varphi(P) + \varphi(Q)$$

توجه ۴. اگر $E_۱ = E_۲$ آنگاه، همسانی یک درون‌ریختی ناصفر خواهد بود.

تعریف ۱.۸.۱. فرض کنید $\alpha = (r_۱(x), r_۲(x)y)$ و ضرایب $r_۱$ و $r_۲$ در k قرار داشته باشند، گوییم α روی k تعریف شده است:

- اگر $r_۱(x) = \frac{p(x)}{q(x)}$ ، آنگاه: $deg(\alpha) = max\{deg(p), deg(q)\}$
- اگر $r_۱'(x) \neq ۰$ آنگاه گفته می‌شود که α جداپذیر است.

قضیه ۶. فرض کنید $\alpha : E_۱(\bar{k}) \longrightarrow E_۲(\bar{k})$ یک همسانی باشد، در این‌صورت:

۱. *اگر α جداپذیر باشد:*

$$deg(\alpha) = \#ker(\alpha)$$

۲. *اگر α جداناپذیر باشد:*

$$deg(\alpha) \geq \#ker(\alpha)$$

توجه ۵. $ker(\alpha)$ یا هسته همسانی یک زیرگروه متناهی از $E_1(\bar{k})$ است.

۹.۱ دوگان همسانی

قضیه ۷. *فرض کنید $\alpha : E_1 \longrightarrow E_2$ یک همسانی باشد. در این‌صورت همسانی $\hat{\alpha} : E_2 \longrightarrow E_1$ وجود دارد بطوریکه:*

$$\hat{\alpha} \circ \alpha = [deg(\alpha)]$$

به $\hat{\alpha}$ دوگان همسانی α گفته می‌شود. البته قابل ذکر است که $\hat{\alpha}$ یکتاست و

$$deg(\alpha) = deg(\hat{\alpha})$$

و همچنین :

$$\alpha \circ \hat{\alpha} = [deg(\alpha)]$$

۱.۹.۱ خواص دوگان همسانی

اگر $\varphi : E_1 \longrightarrow E_2$ و $\psi : E_2 \longrightarrow E_3$ دو همسانی باشند، آنگاه :

آ. $\widehat{\varphi \circ \psi} = \hat{\varphi} \circ \hat{\psi}$

ب. $\hat{\hat{\varphi}} = \varphi$

ج. $deg(\varphi \circ \hat{\varphi}) = (deg\varphi)^2$

قضیه ۸. فرض کنید E یک خم بیضوی و C زیرگروهی متناهی از E باشد. یک خم بیضوی یکتای E' و یک همسانی جداپذیر $\varphi : E \longrightarrow E' \cong \frac{E(\bar{k})}{ker(\varphi)}$ وجود دارد بطوریکه:

$$ker(\varphi) = C$$

گزاره ۱ . اگر ℓ یک عدد اول متباین با مشخصه‌ی میدان باشد آنگاه $E[\ell]$ دارای $\ell + ۱$ زیرگروه از مرتبه‌ی ℓ خواهد بود. هر یک از این زیرگروه‌ها با توجه به قضیه‌ی قبل می‌تواند هسته‌ی یک همسانی باشند:

$$E[\ell] = \{P \in E(\bar{K}) | \ell P = \infty\}$$

اثبات ۱ . می‌دانیم $E[\ell] \cong \mathbb{Z}_\ell \times \mathbb{Z}_\ell$ ، بنابراین $\ell^۲$ عضو دارد. از آنجایی که $\mathbb{Z}_\ell \times \mathbb{Z}_\ell$ دوری نیست، همه‌ی عناصر غیربدیهی $E[\ell]$ از مرتبه‌ی ℓ هستند(زیرا تنها ℓ ، $\ell^۲$ را می‌شمارد)، بنابراین با حذف عنصر بدیهی $E[\ell]$ ، $\ell^۲ - ۱$ عنصر باقی می‌ماند. هر زیرگروه $E[\ell]$ شامل عنصر بدیهی است، عنصر بدیهی را از هر گروه حذف می‌کنیم. هر زیرگروه $\ell - ۱$ عنصر غیربدیهی خواهد داشت، از آنجایی که $\ell + ۱ = \frac{\ell^۲ - ۱}{\ell - ۱}$ ، پس $E[\ell]$ ، $\ell + ۱$ زیرگروه از مرتبه‌ی ℓ خواهد داشت.

گزاره ۲ . اگر ℓ یک عدد اول متباین با مشخصه‌ی میدان باشد، هر همسانی که هسته‌ی آن زیرگروهی از $E[\ell]$ باشد از درجه‌ی ℓ خواهد بود و به آن یک ℓـ همسانی می‌گوییم.

قضیه ۹. اگر $\phi : E_۱ \longrightarrow E_۲$ یک همسانی باشد آنگاه $E_۱$ سوپرسینگولار است اگر وتنها اگر $E_۲$ سوپرسینگولار باشد (و بالعکس).

قضیه ۱۰ (تیت). دو خم بیضوی $E_۱$ و $E_۲$ روی میدان متناهی $\mathbb{F}_q$ همسان هستند اگر و تنها اگر

$$\#E_۱(\mathbb{F}_q) = \#E_۲(\mathbb{F}_q)$$

به‌عبارت دیگر دو خم بیضوی $E_۱$ و $E_۲$ را همسان گوییم هرگاه یک همسانی از $E_۱$ به $E_۲$ وجود داشته باشد.

نتیجه ۱ . با استفاده از قضیه‌ی تیت می‌توان در زمان چندجمله‌ای مشخص کرد که آیا دو خم روی میدان متناهی $\mathbb{F}_q$ ، همسان هستند یا خیر.

۲.۹.۱ فرمول ولو

قضیه ۱۱ . *فرض کنید E یک خم بیضوی با معادله‌ی وایرشتراس یر روی میدان k باشد:*

$$y^۲ + a_۱xy + a_۳y = x^۳ + a_۲x^۲ + a_۴x + a_۶$$

و C یک زیرگروه متناهی از $E(\bar{k})$ باشد. در اینصورت یک خم بیضوی E' و یک همسانی جداپذیر α از E به E' وجود دارد بطوریکه:

$$C = ker(\alpha)$$

با استفاده از مراحل زیر می‌توان خم E' و همسانی α را به‌دست آورد:

۱. *قرار می‌دهیم:*

$$F(x,y) = x^۳ + a_۲x^۲ + a_۴x + a_۶ - y^۲ + a_۱xy + a_۳y$$

و برای هر $Q = (x_a, y_a) \in c$ که $Q \neq \infty$ تعریف می‌کنیم:

$$g_Q^x = F_x(Q) = ۳x_Q^۲ + ۲a_۲x_Q + a_۴ - a_۱y_Q$$

$$g_Q^x = F_y(Q) = -۲y_Q - a_۱x_Q - a_۳$$

$$V_Q = \begin{cases} g_Q^x & , ۲Q = \infty \text{ اگر} \\ ۲g_Q^x - a_۱g_Q^y & , ۲Q \neq \infty \text{ اگر} \end{cases}$$

$$u_Q = \left(g_Q^y\right)^۲$$

۲. فرض کنید C_2 نقاطی از مرتبه‌ی ۲ در C باشند، $R \subseteq C$ را طوری انتخاب می‌کنیم که یک اجتماع مجزا به شکل زیر داشته باشیم:

$$C = \infty \cup C_2 \cup R \cup (-R)$$

به‌عبارت دیگر هر نقطه‌ی $p, -p \in C$ که تابی از مرتبه‌ی ۲ نباشند را درنظر می‌گیریم، دقیقا یکی از آنها را در R قرار می‌دهیم.
حال فرض کنید $S = R \cup C_2$ قرار می‌دهیم:

$$v = \sum_{Q\in S}(v_Q) \ \ , \ \ w = \sum_{Q\in S}(u_Q + x_Q v_Q)$$

در این‌صورت خم E' دارای معادله‌ی :

$$Y^2 + A_1XY + A_3Y = X^3 + A_2X^2 + A_4X + A_6$$

است که در آن:

$$A_1 = a_1 \ , \ A_2 = a_2 \ , \ A_3 = a_3 \ , \ A_4 = a_4 - 5v \ , A_6 = a_6 - (a_1^2 + 4a_2)v - 7w$$

۳. همسانی $\alpha : E_{(x,y)} \longrightarrow E'_{(X,Y)}$ که در آن:

$$X = x + \sum_{Q\in S}\Big(\frac{v_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2}\Big)$$

$$Y = y - \sum_{Q\in S}\Big(v_Q\cdot\frac{2y + a_1x + a_3}{(x - x_Q)^3} + v_Q\cdot\frac{a_1(x - x_Q) + y - y_Q}{(x - x_Q)^2} + \frac{a_1u_Q - g_Q^x g_Q^y}{(x - x_Q)^2}\Big)$$

را تعریف می‌کنیم. در واقع

$$\alpha : E \longrightarrow E' \cong p \rightsquigarrow \big(X(p), Y(p)\big)$$

که در آن :

$$X(p) = x(p) + \sum_{Q\in C}[\, x(P+Q) - x(Q)\,]$$

$$Y(p) = y(p) + \sum_{Q\in C}[\, y(P+Q) - y(Q)\,]$$

مثال ۸. فرض کنید $E: x^۳+ax^۲+bx$ یک خم بیضوی تعریف شده روی میدان k باشد. نقطه‌ی $(۰,۰)$ یک نقطه‌ای از مرتبه‌ی ۲ روی این خم است. بنابراین $C=\{\infty,(۰,۰)\}$ زیرگروهی از $E(\bar{k})$ است. می‌خواهیم یک همسانی از E به یک خم بیضوی E' تعریف کنیم که هسته‌ی آن C باشد. بنابراین داریم:

$$C_۲=\{(۰,۰)\}\ ,\ R=\phi\ ,\ S=(۰,۰)$$

$$F(x,y)=x^۳+ax^۲+bx-y^۲$$

$$g^x_Q=F_x(Q)=۳x^۲_Q+۲ax_Q+b=b$$

$$g^y_Q=F_y(Q)=-۲y_Q=۰$$

$$v_Q=g^x_Q=b\ \ ,\ \ u_Q=(g^y_Q)^۲=۰$$

$$v=\sum_{Q\in S}v_Q=b\ \ ,\ \ w=\sum_{Q\in S}(u_Q+x_Qv_Q)=۰$$

$$Y^۲+A_۱XY+A_۳Y=X^۳+A_۲X^۲+A_۴X+A_۶$$

$$A_۱=a_۱=۰\ ,\ A_۲=a_۲=a\ ,\ A_۳=a_۳=۰\ ,\ A_۴=a_۴-۵v=b-۵b=-۴b\ ,$$

$$A_۶=a_۶-(a^۲_۱+۴a_۲)v-۷w=-۴b^۲$$

$$\Longrightarrow E':Y^۲=X^۳+a_۲X^۲-۴bY-۴ab$$

$$\alpha:E\longrightarrow E'\cong(x,y)\longrightarrow(X,Y)\Rightarrow\begin{cases}X=x+\frac{b}{x}\\ Y=y-\frac{by}{x^۲}\end{cases}$$

فصل ۲

امضای دیجیتال

[1] فرض کنید شخصی به‌نام پوریا خواهان آن باشد تا سندی(دیجیتالی) را (در محیطی ناامن مانند اینترنت) منتشر کند با این ویژگی که سند گویای آن باشد که از طرف پوریا می‌باشد و درنتیجه موردتایید وی نیز می‌باشد. بدین منظور پوریا باید از پروتکلی به نام امضای دیجیتال که نوعی رمزنگاری نامتقارن می‌باشد، استفاده کند.
در امضای دیجیتال ، شخصی می‌تواند پیام خود را امضا و آن را منتشر کند و هر شخص دیگری با دیدن پیام و امضا پی ببرد که آیا پیام دریافت شده از شخص موردنظر می‌باشد یا خیر. بدین منظور لازم است مکانیزمی وجود داشته باشد تا شخص گیرنده پیام ، صحت هویت فرستنده را تایید کند.
برای اجرای این سیستم پوریا به عنوان امضاکننده می‌تواند از یک کلیدخصوصی برای امضای پیام و از یک کلیدعمومی متناسب با کلید خصوصی (کلید خصوصی و کلیدعمومی با هم وابسته و مربوط هستند و توسط الگوریتمی مشخص تولید می‌شوند) برای تایید امضا استفاده کند.
در ادامه می‌خواهیم نشان دهیم در بطن این پروتکل، دو پروتکل دیگر به نام احرازهویت و اثبات دانش صفر نیز اجرا می‌شوند. به عبارت دیگر با امضای سند، امضاکننده هویت خود را نشان داده است و هر شخصی که سندامضاشده را ببیند متوجه می‌شود که پوریا آن را امضا کرده است و بنابراین پروتکل احرازهویت صورت گرفته است. در طرف دیگر پروتکل اثبات دانش صفر نیز انجام شده است به این صورت که بدون آن‌که دانش امضاکننده یعنی کلیدخصوصی افشا شود، ویکتور(تاییدکننده) توانسته است امضا را با استفاده از کلیدعمومی(تاییدساز) که توسط امضاکننده

[1] Digital Signature

همراه با پیام منتشر شده بود تاییدکند.
در ادامه می‌خواهیم پروتکل امضا را براساس پروتکلی به‌نام اثبات‌دانش‌صفرهویت پیاده‌سازی کنیم.

۱.۲ طرح‌های تعهد

برای نشان‌دادن ایده این پروتکل، این سوال را مطرح می‌کینم که چگونه می‌توان از طریق ایمیل به بازی سنگ، کاغذ، قیچی پرداخت؟
اگر بخواهیم به روش مرسوم بازی کنیم و انتخاب‌های خود را با ایمیل ارسال کنیم، شخص باب می‌تواند پس از دریافت انتخاب آلیس از طریق ایمیل، انتخاب خود را برای به‌دست‌اوردن پیروزی تغییر دهد و انتخابی را از طریق ایمیل ارسال کند که با آگاهی از انتخاب طرف دیگر بازی به‌دست آورده است! و لذا انجام بازی به‌صورت عادلانه میسر نیست.
همچنین لازم به ذکر است که احتمال اینکه هردو انتخاب واقعا در یک زمان به طرف مقابل ارسال شود غیرممکن است. حتی واردکردن فرد سوم به بازی به‌عنوان داور هم نمی‌تواند کمکی به اجرای عادلانه بازی داشته باشد زیرا ممکن است یکی از طرف‌های بازی با داور تبانی کند و از حرکت طرف مقابل آگاه شود.
بنابراین براساس گفته‌های بالا به این نتیجه می‌رسیم که عملا این بازی از طریق ایمیل با روش معمولی امکان‌پذیر نخواهد بود. اما راهی وجود دارد که حتی اگر آلیس شروع‌کننده‌ی بازی باشد و باب پاسخ آن را از طریق ایمیل دریافت کند با این حال باب انتخابی که انجام می‌دهد واقعا تصادفی است و از انتخاب آلیس هیچ‌گونه اطلاعی ندارد. این راه‌حل از طریق طرح تعهد قابل پیاده‌سازی می‌باشد که در ادامه به معرفی آن می‌پردازیم.

۱.۱.۲ معرفی

یک طرح تعهد شامل دو پروتکل به‌نام‌های تعهد و آشکارسازی می‌باشد که معمولا بین دو بخش که به فرستنده و دریافت‌کننده شناخته می‌شوند شکل می‌گیرد. در بیشتر حالت‌ها پروتکل‌های تعهد و آشکارسازی در یک الگوریتم تجمیع می‌شوند. همچنین لازم به ذکر است که برای ارتباط بین فرستنده و دریافت‌کننده نیاز به هیچ تعامل دو سویه نمی‌باشد و درنتیجه این طرح اساسا یک طرح غیرتعاملی می‌باشد.

تعریف ۱.۱.۲ . اگر تعهد یک الگوریتم با زمان چندجمله‌ای و معین به صورت زیر باشد:

$$commit : \{۰, ۱\}^k \times \{۰, ۱\}^* \to \{۰, ۱\}^*$$

که k پارامتر امنیتی می‌باشد، آنگاه یک طرح تعهد(غیرتعاملی) شامل دو پروتکل بین فرستنده و دریافت‌کننده می‌باشد که به صورت زیر بیان می‌شوند:

مرحله‌ی تعهد . در این پروتکل، فرستنده مقدار انتخابی خود یعنی $x \in \{۰, ۱\}^*$ را معین و تابع $C = commit(u, x)$ را محاسبه می‌کند تا مقدار انتخابی خود یعنی x با مقدار تصادفی اما معین $u \in_R \{۰, ۱\}^k$ تلفیق شود و درنتیجه انتخابش، مقداری تصادفی به خود بگیرد. در این مرحله، فرستنده به‌جای مقدار x ، مقدار C را به‌عنوان تعهدی بر انتخاب اصلی خود ارسال می‌کند. و در طرف دیگر دریافت‌کننده مقدار C را برای ادامه طرح، ذخیره می‌کند.
درمثال بالا می‌توان گفت، x بیانگر حرکت و انتخاب هر بازیکن می‌باشد و u یک مقدار تصادفی است که هیچ ارتباطی با انتخاب x آن ندارد. C تعهد و مقداری است که انتخاب x در آن مخفی شده است و از طریق ایمیل برای طرف مقابل بازی ارسال می‌شود.

مرحله‌ی آشکارسازی . پروتکلی است که در طی آن فرستنده با ارسال u و x برای دریافت‌کننده، با کمک تابع $C = commit\ (u, x)$ توانایی آشکارسازی مقدار x از تعهد C را به دریافت‌کننده می‌سپارد. به‌عبارت‌دیگر در این مرحله دریافت‌کننده به محاسبه‌ی تابع $commit(u, x)$ می‌پردازد و بررسی می‌کند که آیا خروجی این تابع با مقدار C که قبلا دریافت کرده است برابر هستند یا خیر.
برای شبیه‌سازی این مرحله در مثال بالا می‌توان گفت که پس از ارسال تعهدها از طرف آلیس و باب برای یکدیگر در مرحله‌ی قبل، در این مرحله بازی هریک از بازیکن‌ها به بررسی تعهد طرف مقابل پرداخته و درانتها با نمایش انتخاب‌ها پیروز بازی مشخص می‌شود.

لازم به ذکر است که یک حالت خاص از طرح بالا زمانی است که مقدار تعهد شده تک بیتی یعنی $x \in \{۰, ۱\}$ باشد که به **طرح تعهد بیتی** معروف است. به‌عبارت دیگر برای انجام این پروتکل فرستنده تنها دو انتخاب دارد که ارزش آن یا صفر یا یک می‌باشد.

برای امنیت کامل این طرح لازم است تا ویژگی‌های زیر در هر طرح تعهدی وجود داشته باشد:

انقیاد .[2] فرستنده نباید بعد از ارسال تعهد C قادر به تغییر مقدار x شود. در مثال بالا می‌توان گفت که آلیس بعد از ارسال حرکت خود از طریق ایمیل، نمی‌تواند بعد از انتخاب حرکت باب، انتخاب خود را تغییر دهد.

مخفی‌سازی .[3] بیانگر آن است که هنگام دریافت تعهد توسط دریافت‌کننده، مقدار موردنظر x از طریق تعهد C قابلیت کشف نداشته باشد. در مثال بالا می‌توان گفت باب با دریافت تعهد آلیس از طریق ایمیل قابلیت کشف مقدار انتخاب شده‌ی آلیس را ندارد و انتخاب خود را کاملا تصادفی انتخاب می‌کند.

اگر بخواهیم ویژگی‌های بالا را به‌صورت یک تابع ریاضی پیاده‌سازی کنیم، این دو ویژگی به صورت زیر خواهند بود:

مقاومت ثانویه . برای هر ورودی داده شده، ورودی دیگری که خروجی مشابهی را موجب شود، دشوار باشد. به عبارت دیگر برای هر متخاصم ε احتمال تولید $u, u' \in \{۰, ۱\}^k$ که رابطه‌ی $commit(u, ۰) = commit(u', ۱)$ را موجب شود، ناچیز باشد.

معکوس‌سازی سخت (یکطرفه) با داشتن ورودی محاسبه خروجی آسان است اما از طریق خروجی محاسبه ورودی آن مشکل است. به‌عبارت دیگر می‌توان گفت توزیع‌های ناشی از $commit(u, ۰)$ و $commit(u, ۱)$ زمانیکه $u \in_R \{۰, ۱\}^k$ ،غیرقابل‌تشخیص(یکنواخت) باشند.

مثال ۹. اگر یک تابع هش رمزنگاری H را دراختیار داشته باشیم آنگاه طرح تعهد بیتی خود را می‌توانیم به صورت زیر به‌دست بیاوریم:

$$commit_{.}(u, x) = H(u, x)$$

که درآن مقدار تعهد $x \in \{۰, ۱\}$ و $u \in \{۰, ۱\}^k$ می‌باشند.

[2] Binding
[3] Hiding

- ویژگی مقاوم‌ـ تصادم تابع هش H ضمانت می کند که شخص متعهدشده نمی‌تواند u ، x و u' ، $x - ۱$ را به‌دست آورد بطوریکه

$$H(u, x) = H(u', ۱ - x)$$

بنابراین طرح ما خاصیت انقیاد را دارا می‌باشد.

- ویژگی مقاوم در برابر تضاهر برای خاصیت مخفی‌سازی لازم می‌باشد ولی با این ویژگی هیچ ضمانتی نیست که مقدار x مخفی(به اندازه مقدار u) بماند.بدین منظور لازم است که از مدل اوراکل تصادفی استفاده کنیم.

۲.۱.۲ طرح تعهد در اثبات دانش صفر

طرح تعهد در بسیاری از پروتکل‌های رمزنگاری مورداستفاده قرار می‌گیرند. از جمله‌ی این پروتکل‌ها می‌توان به اثبات دانش صفر اشاره کرد. در ادامه به دلایل استفاده اثبات دانش صفر از طرح تعهد می‌پردازیم.

۲.۲ اثبات دانش صفر هویت

به‌طور رسمی، یک سیستم اثبات دانش صفر یک رویه است که طی آن پگی(به‌عنوان شخص اثبات‌کننده)، ویکتور(به‌عنوان شخص تاییدکننده) را متقاعد می‌کند که به یک حقیقت معین [۴] اشراف دارد بطوریکه هیچ اطلاعات اضافی نسبت به دانش خود در اختیار ویکتور قرار نمی‌دهد تا بدین‌منظور خود ویکتور نتواند به عنوان یک مدعی، دیگران را متقاعد کند که به حقیقت مورد بحث اشراف دارد. برای توضیح بیشتر این پروتکل مثالی ارائه می‌کنیم.

مثال ۱۰ . **اثبات دانش صفر**

فرض کنید دو لیوان شفاف در اختیار داریم که یکی حاوی آبِ خالص و دیگری حاوی آب با مخلوطی شفاف می‌باشد که فقط پگی فرق این دو لیوان را می‌داند. حال برای آن‌که پگی به ویکتور ثابت کند که دانش لازم را برای تشخیص لیوان حاوی آب خالص و لیوان آب ناخالص

[۴] حقیقت می‌تواند هویت اثبات‌کننده (پگی) باشد.

را در اختیار دارد می‌بایست به چالش‌هایی که از طرف ویکتور مورد سوال قرار می‌گیرد به درستی جواب بدهد. ویکتور برای اطمینان از اینکه پگی واقعا دانش لازم این اثبات را می‌داند می‌تواند چالش‌های خود را چندین بار تکرار کند و اگر پگی در تمامی چالش‌ها به درستی جواب بدهد آنگاه مطمئن می‌شود که پگی دانش لازم را دراختیار دارد. پگی برای آن‌که مستقیما دانش خود را افشا نکند لیوان حاوی آب خالص را به ویکتور نشان نمی‌دهد و در عوض مراحل زیر به تعداد مشخصی تکرار می‌شود

۱. ابتدا پگی به عنوان اثبات‌کننده ادعا می‌کند که مکان لیوان حاوی آب ناخالص را می‌داند.

۲. پگی چشمان خود را با چشم‌بند می‌بندد و سپس ویکتور به‌عنوان یک چالش ، یا جای دو ظرف آب را باهم جابه‌جا می‌کند یا بدون تغییر آن‌ها آماده پاسخ چالش خود می‌شود

۳. پگی چشم‌بند را از چشمان خود برمی‌دارد و با اتکا به دانشی که در اختیار دارد مشخص می‌کند که آیا جای این دو لیوان عوض شده است یا خیر

۴. اگر پگی به درستی تشخیص دهد که جای لیوان‌ها عوض شده است یا خیر آنگاه ویکتور برای اطمینان از شانسی نبودن جواب پگی می‌تواند بار دیگر مراحل را با همکاری پگی تکرارکند اما اگر حتی یک بار پگی به اشتباه جواب چالش پگی را بدهد آنگاه ویکتور با اطمینان ادعای پگی را نمی‌پذیرد.

ذکر این نکته لازم است که اگر پگی به صورت شانسی به چالش جواب بدهد به احتمال ۱/۲ به طور صحیح جواب داده است ، حال اگر که رویه اثبات به تعداد n بار تکرار شود آنگاه به احتمال $۱/۲^n$ پگی به صورت شانسی جواب چالش‌ها را به درستی داده است ، همان‌گونه که مشخص است تقریبا محال است که همچین اتفاقی رخ دهد و پگی قادر باشد که تمام جواب‌ها را به صورت شانسی جواب داده باشد. بنابریان ویکتور با تکرار رویه اثبات و جواب صحیح پگی در هر مرحله، کاملا قانع می‌شود که پگی به دانش ادعا شده اشراف دارد.

برای پیاده‌سازی پروتکل‌اثبات‌دانش‌صفر به صورت ریاضی، از همسانی بین خم‌های سوپرسینگولار استفاده می‌کنیم.

در طرحی که خواهان ارائه آن هستیم به خم‌های سوپرسینگولار با درجه‌ای هموار [۵] نیاز می‌باشد بنابراین ابتدا عدد اولی به فرم ۱ $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm$ را انتخاب می‌کنیم که ℓ_A و ℓ_B اعداد اول کوچک (معمولا ۲ و ۳) می‌باشند با این خاصیت که طول ارقام $\ell_A^{e_A}$ و $\ell_B^{e_B}$ برابر باشد (در بخش ۶ به طور مفصل آن را بررسی خواهیم کرد) و همچنین f یک عامل کوچک است که باعث می‌شود p یک عدد اول شود(از آنجا که $\ell_A^{e_A}$ و $\ell_B^{e_B}$ دیگر اول نیستند). در ادامه با روش بروکر [۳] ، یک خم سوپرسینگولار E را روی میدان F_{p^2} با مرتبه‌ی $(\ell_A^{e_A}\ell_B^{e_B})^2$ به دست می‌آوریم. سپس دو زیرگروه $E[\ell_A^{e_A}]$ و $E[\ell_B^{e_B}]$ که مولدهای آن به ترتیب به صورت زوج نقاط $\langle P_A, Q_A\rangle$ و $\langle P_B, Q_B\rangle$ می‌باشند را روی خم E محاسبه می‌کنیم.
طرح اثبات دانش صفر براساس همسانی‌ها مطابق شکل **؟؟** صورت می‌گیرد. پگی به عنوان اثبات کننده، نقطه S که تولیدکننده هسته همسانی $\phi : E \longrightarrow E/\langle S\rangle$ می‌باشد را به عنوان دانش خود، به صورت مخفی نزد خود نگه می‌دارد. و بر این اساس، کلید خصوصی و کلیدعمومی پگی به‌صورت زیر معرفی می‌شود :

- **کلید خصوصی :** هسته همسانی ϕ یعنی S
- **کلید عمومی :** خم بیضوی $E/\langle S\rangle$ و تصویر نقاط P_B و Q_B یعنی $\phi(P_B)$ و $\phi(Q_B)$

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

شکل ۱.۲: هر فلش با همسانی و و هسته‌اش نشانه گذاری شده است

حال پگی برای آن که به ویکتور (تاییدکننده) ثابت کند که دانش $\langle S\rangle$ را می‌داند ، مراحل زیر به ترتیب انجام می‌شود:

---

[۵] در مبحث خم‌های سوپرسینگولار، ساخت خم‌های با درجه هموار آسان می‌باشد و با استفاده از این خم‌ها می‌توان تعداد زیادی همسانی بین آنها ساخت که خیلی سریع قابل محاسبه هستند.

۱. • نقطه تصادفی R را از مرتبه‌ی $\ell_B^{e_B}$ انتخاب می‌کند.

• همسانی $\psi : E \rightarrow E/\langle R \rangle$ را محاسبه می‌کند.

• در ادامه همسانی‌های $\phi' : E/\langle R \rangle \rightarrow E/\langle R, S \rangle$ را باهسته $\langle \psi(S) \rangle$ و همچنین همسانی $\psi' : E/\langle S \rangle \rightarrow E/\langle R, S \rangle$ را با هسته‌ی $\langle \phi(R) \rangle$ از طریق فرمول ولو محاسبه می‌کند.

• پس از محاسبات بالا ، پگی تعهد [6] $com = (E_۱, E_۲)$ را برای ویکتور ارسال می‌کند که $E_۱ = E/\langle R \rangle$ و $E_۲ = E/\langle R, S \rangle$ می‌باشند.

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

۳. پگی پاسخ $resp$ را برای ویکتور ارسال می‌کند :

• اگر $ch = ۰$ آنگاه $resp = (R, \phi(R))$

• اگر $ch = ۱$ آنگاه $resp = \psi(S)$

۴. • اگر $ch = ۰$ ، ویکتور ابتدا بررسی می‌کند که آیا R و $\phi(R)$ هردو از مرتبه‌ی $\ell_B^{e_B}$ هستند یا خیر . در ادامه بررسی می‌کند که آیا این دو، هسته‌ی همسانی‌های $E \rightarrow E_۱$ و $E/\langle S \rangle \rightarrow E_۲$ را تولید می‌کنند یا خیر.

• اگر $ch = ۱$ ، ویکتور ابتدا بررسی می‌کند که آیا $\psi(S)$ از مرتبه‌ی $\ell_A^{e_A}$ می‌باشد یا خیر و همچنین درادامه بررسی می‌کند که آیا هسته‌ی همسانی $E_۱ \rightarrow E_۲$ را تولید می‌کند یا خیر.

توجه ۶. قابل ذکر است که در مرحله‌ی ۴، اگر نتیجه‌ی بررسی تاییدکننده مثبت باشد، ادعای اثبات‌کننده تایید و در غیر اینصورت ادعای وی رد می‌شود.

توجه ۷. همچنین قابل ذکر است که رابطه‌ی زیر همواره برقرار است:

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

[6] 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 ۲.۲: همسانی‌های مخفی با خط‌های مقطع نمایش 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.

برای دست‌یابی به λ بیت امنیت، لازم است که عدد اول p دقیقا 6λ بیت باشد(دلیل این امر در بخش ۶ ذکر شده است) و پروتکل λ بار تکرار شود. اگر ویکتور(تاییدکننده) تمام λ بار تکرار پروتکل را با موفقیت تایید کند آنگاه پگی(اثبات‌کننده) توانسته است ادعای خود مبنی بر دانش کلیدخصوصی S را برای ویکتور اثبات کند، در غیراینصورت ادعا توسط ویکتور موردقبول قرار نمی‌گیرد.

لازم به ذکر است که در هربار اجرای این طرح، نقطه R کاملا به صورت تصادفی انتخاب می‌شود و براساس این نقطه، شکل **؟؟** حاصل می‌شود و بنابراین همسانی‌ها و خم‌های هر بار اجرای این پروتکل با دفعه قبل کاملا فرق دارد و فقط خم $E/\langle S\rangle$ ثابت باقی می‌ماند.
همچنین چنان‌که در بخش ۶ خواهیم دید اگر همسانی‌های ψ و ψ' و ϕ' همزمان معلوم شوند، دانش S قابل کشف است بنابراین برای جلوگیری از افشای S ، در هر بار تکرار رویه اثبات، درمقابل چالش $ch = 0$ همسانی‌های ψ و ϕ' عمومی می‌شوند و درمقابل چالش $ch = 1$ تنها همسانی ϕ' عمومی می‌شود تا طرح ما ایمن باقی بماند.

۳.۲ پروتکل‌های شناسایی

پروتکل‌های شناسایی [۷] بین دو بخش به‌نام‌های تاییدکنننده و اثبات‌کننده رخ می‌دهد که در طی آن اثبات‌کننده خواهان متقاعدکردن تاییدکننده برای ادعای خود می‌باشد. یک مثال معمولی زمانی است که کاربری خواهان دسترسی به حساب کامپیوتری خود می‌باشد(ورود امن).
عموما پروتکل‌های شناسایی ممکن است براساس یک یا بیشتر فاکتورهای زیر بناشده باشد:

- **آنچه شما هستید.** مانند اثرانگشت، اسکن چشم و ..
- **آنچه شما در اختیار دارید.** مانند کارت‌هوشمند، سیم‌کارت یا هر توکن‌های سخت‌افزاری
- **آنچه شما می‌دانید.** مانند گذرواژه، کلیدهای مخفی و ..

درادامه خواهان آن هستیم تا پروتکل‌های شناسایی رمزنگاری را که درآن یک اثبات‌کننده تنها نیاز به داشتن(دانستن) یک کلیدمخفی است را بیان کنیم. ساختارهای رمزنگاری بسیار زیادی برای پروتکل‌های شناسایی موجود می‌باشد و هدف عمومی این است که مقدار محاسبات برای بخش‌های تاییدکننده و همینطور اثبات‌کننده کاهش یابد.

۱.۳.۲ معرفی

یک پروتکل شناسایی دراصل قسمتی از یک طرح شناسایی می‌باشد. یک طرح کامل شناسایی شامل دو پروتکل به‌نام‌های ثبت‌نام و شناسایی است که میان دو بخش به‌نام‌های اثبات‌کننده و تاییدکننده صورت می‌پذیرد. طرح‌های شناسایی به دو صورت متقارن و نامتقارن پیاده‌سازی می‌شوند. دریک طرح شناسایی متقارن ، با اشتراک یک کلیدخصوصی بین هردو بخش، پروتکل ثبت‌نام پایان می‌یابد. در یک طرح شناسایی نامتقارن ، پروتکل ثبت‌نام پس از اشتراک یک کلیدعمومی بین تمام بخش‌ها پایان می‌یابد و تنها اثبات‌کننده از کلیدخصوصی مطلع است.(البته در طرح‌های پیشرفته، ممکن است تاییدکننده نیز دارای کلیدخصوصی باشد)
یکی از مزیت‌های بزرگ طرح نامتقارن این است که اثبات‌کننده ممکن است چندین مرتبه از کلیدعمومی‌اش در ارتباط با تاییدکننده استفاده کند.

[7] Identification Protocols

از انواع پروتکل‌های ارائه‌شده برای طرح شناسایی، می‌توان به **طرح گذرواژه مبنا** ، **زنجیره‌ی هش یکطرفه** و **پروتکل‌های چالش-پاسخ** اشاره کرد. از آنجا که در طرح‌امضای ارائه شده در این پایان‌نامه از پروتکل‌های چالش-پاسخ استفاده شده است لذا در ادامه به معرفی این پروتکل خواهیم پرداخت.

۲.۳.۲ امنیت طرح شناسایی

در این قسمت به حمله‌هایی که ممکن است به طرح شناسایی صورت پذیرد را بررسی می‌کنیم. با درنظرگرفتن اینکه پروتکل ثبت‌نام در یک محیط امن انجام می‌شود درنتیجه تنها به بررسی حملات رمزنگاری به پروتکل شناسایی می‌پردازیم. اساسی‌ترین پیش‌نیاز امنیت برای پروتکل شناسایی، متوقف کردن حملات جعل‌هویت [8] می‌باشد، که در این صورت برای یک مهاجم غیرممکن است تا هویت خودش را (در برابر با تاییدکننده یا اثبات‌کننده)با موفقیت جعل کند. در ادامه به معرفی چندین حمله‌ی جعل‌هویت از نوع فعالانه و غیرفعالانه می‌پردازیم.

از اصلی‌ترین حملات جعل‌هویت غیرفعال می‌توان به استراق‌سمع روی یک ارتباط بین اثبات‌کننده و تاییدکننده در یک اجرای قانونی پروتکل شناسایی اشاره کرد. نوع دیگر حملات غیرفعال، حمله‌ی کلید در طرح غیرمتقارن می‌باشد، که درآن حمله‌کننده تلاش دارد تا از طریق کلیدعمومی به کلیدخصوصی دسترسی یابد.

یک فرم ساده از حمله‌ی جعل‌هویت فعال، حمله‌ی حدس [9] می‌باشد که درآن حمله‌کننده در نقش اثبات‌کننده درآمده و امیدوار است تا بدون دانستن کلیدخصوصی یا کلیدمخفی اثبات‌کننده، حدس درستی درنظر گرفته باشد.میزان موفقیت حمله‌ی حدس می‌تواند با ترکیب با حمله‌ی تاییدکننده‌ی متقلب [10] افزایش یابد. در حمله‌ی تاییدکننده‌ی متقلب، متخاصم خود را درنقش تاییدکننده جا می‌زند و امیدوار است تا اطلاعات مفیدی را از طریق اثبات‌کننده با انحراف پروتکل استخراج کند. در نهایت متخاصم می‌تواند حمله‌ی مردمیانی [11] را اجرا کند. در این حمله، اثبات‌کننده صادق $\mathcal{P}$ فکر می‌کند که پروتکل شناسایی را با تاییدکننده $\mathcal{V}^*$ اجرا می‌کند، اما درواقع امر، $\mathcal{V}^*$ تمام پیام‌های ردوبدل شده بین خود و تاییدکننده صادق $\mathcal{P}$ را به یک تاییدکننده $\mathcal{V}$ می‌فرستد که خود تاییدکننده

[8]impersonation attacks
[9]guessing attack
[10]cheating verifier
[11]man-in-the-middle

$\mathcal{V}$ نیز فکر می‌کند که در اصل پروتکل را با یک تاییدکننده $\mathcal{P}$ اجرا می‌کند. به بیان ساده‌تر متخاصم بین تاییدکننده و اثبات‌کننده قرار می‌گیرد و تمام تعاملات از طریق کانال متخاصم انجام شده و هردو بخش تاییدکننده و اثبات‌کننده به این موضوع پی نمیبرند.
حمله مردمیانی یادآور حمله‌ای در شطرنج به‌نام استادبزرگ می‌باشد. در این حمله، یک بازیکن آماتور شطرنج تلاش دارد با بازی کردن همزمان با دو استادشطرنج در یک زمان، مهارت و امتیاز خود را افزایش دهد. در این حمله، یازیکن آماتور دریک بازی مهره‌سفید و در بازی دیگر مهره‌سیاه می‌باشد و با شروع بازی با استادبزرگ اول با مهره مشکی و کپی کردن حرکات شطرنج بین این دواستادبزرگ درانتظار پایان یازی می‌ماند. در پایان بازی‌ها یا بازیکن آماتور در یک بازی پیروزشده و متعاقبا در بازی دیگر شکست خورده و یا هردو بازی به نتیجه مساوی خاتمه یافته است.در پایان با هر کدام از حالت‌های فوق امتیاز بازیکن آماتور به‌طورقابل‌توجهی افزایش یافته است.

۳.۳.۲ پروتکل‌های پرسش-پاسخ اصلی

چهارنوع پروتکل پرسش-پاسخ اصلی وجود دارد. درهریک از این پروتکل‌های شناسایی، تاییدکننده با ارسال یک چالش تصادفی برای تاییدکننده، پروتکل را آغاز می‌کند. درادامه اثبات‌کننده با ارسال یک پاسخ برای تاییدکننده به‌منظور بررسی آن واکنش نشان می‌دهد. طرح کلی این پروتکل در شکل ۱۱۱ به‌صورت خلاصه آمده است. درادامه برای هر طرح، حملات استراق‌سمع و تاییدکننده‌متقلب را بررسی خواهیم کرد.

۱.۳.۳.۲ رمزنگاری متقارن

فرض کنید اثبات‌کننده و تاییدکننده یک کلیدمتقارن $K \in_R \{۰,۱\}^k$ را بین خود به اشتراک گذاشته‌اند. اگر E_K نشان‌دهنده‌ی الگوریتم رمزگذاری با استفاده از کلید K و D_K نشان‌دهنده‌ی الگوریتم رمزگشایی مربوطه باشد آنگاه به‌سادگی روشن است که:

$$E_K, D_K : \{۰,۱\}^k \to \{۰,۱\}^k$$

اگر به شکل ۱/۱/آ توجه شود پیداست که پروتکل شناسایی با ارسال چالش $c \in_R \{۰,۱\}^k$ توسط تاییدکننده آغاز می‌شود و در ادامه اثبات‌کننده پاسخ $r = E_K(c)$ را برای تاییدکننده ارسال می‌کند. و درنهایت تاییدکننده تساوی $D_K(r) = c$ را بررسی می‌کند. اگر تساوی برقرار باشد، پاسخ

اثبات‌کننده تایید و درنتیجه شناسایی با موفقیت انجام می‌پذیرد و درغیراینصورت پاسخ اثبات‌کننده رد و پروتکل با موفقیت اتمام نمی‌یابد.

برای ایستادگی در برابر حمله‌ی استراق‌سمع در یک طرح رمزگذاری باید از حمله‌ی متن ساده‌ی شناخته شده [12] جلوگیری کرد. و برای ایستادگی در برابر حمله‌ی تاییدکننده‌متقلب در این طرح لازم است از حمله‌ی متن‌ساده‌ی انتخابی [13] تطبیقی جلوگیری کرد.
لازم به ذکر است که حمله‌ی متن ساده‌ی شناخته شده یک نوع مدل حمله است که درآن حمله‌کننده به زوج متن ساده و ورژن رمزگذاری‌شده‌ی آن دسترسی دارد و درصدد است با استفاده از این زوج‌ها به کشف اطلاعات مخفی همچون کلیدهای خصوصی بپردازد.و حمله‌ی پیام ساده‌ی انتخابی نیز اشاره به مدلی از حمله دارد که در آن حمله‌کننده خواهان آن است تا یک متن ساده‌ی تصادفی را انتخاب و آن را رمزگذاری کند و درنتیجه نسخه‌ی رمزگذاری شده‌ی آن پیام را به‌دست آورد.

۲.۳.۳.۲ احرازهویت متقارن

فرض کنید اثبات‌کننده و تاییدکننده یک کلیدمتقارن $K \in_R \{۰,۱\}^k$ را بین خود به اشتراک گذاشته‌اند. همچنین $H : \{۰,۱\}^* \to \{۰,۱\}^k$ معرف یک تابع هش رمزنگاری می‌باشد.
با توجه به شکل ۱.۱.ب پروتکل شناسایی با ارسال چالش $c \in_R \{۰,۱\}^k$ توسط تاییدکننده برای اثبات‌کننده آغاز می‌شود.و متناسب با آن اثبات‌کننده نیز پاسخ $r = H(K, c)$ را برای تاییدکننده ارسال می‌کند. درنهایت تاییدکننده رابطه‌ی $r = H(K, c)$ را بررسی می‌کند.

در مدل‌های اوراکل تصادفی، این طرح در برابر حملات استراق‌سمع و تاییدکننده‌متقلب مقاوم می‌باشد.

۳.۳.۳.۲ رمزگذاری نامتقارن

فرض کنید اثبات‌کننده و تاییدکننده یک کلیدعمومی pk را بین خود به‌اشتراک گذاشته‌اند که تنها اثبات‌کننده از کلیدخصوصی sk متناظر با کلیدعمومی مطلع است.همچنین E_{pk} معرف یک

[12] known-plaintext attack (KPA)
[13] chosen-plaintext attack(CPA)

الگوریتم رمزگذاری بااستفاده از کلیدعمومی pk می‌باشد و D_{sk} نشان‌دهنده‌ی الگوریتم رمزگشای مرتبط با الگوریتم رمزگذاری است که ازکلیدخصوصی sk استفاده می‌کند.
چنانکه در شکل ج مشاهده می‌شود ابتدا تاییدکننده پیام $M \in_R \{۰, ۱\}^k$ را انتخاب کرده و سپس با استفاده از الگوریتم رمزگذار، چالش $c = E_{pk}(M)$ را برای اثبات‌کننده ارسال می‌کند و در ادامه اثبات‌کننده پاسخ موردانتظار $r = D_{sk}(c)$ را تولید می‌کند. در انتها تاییدکننده تساوی $r = M$ را بررسی می‌کند.

برای مقاومت در برابر حمله‌ی استراق‌سمع لازم است طرح رمزگذار ایمن معنایی باشد. و همچنین برای مقاومت دربرابر حمله‌ی تاییدکننده‌متقلب لازم است که این طرح در برابر حمله‌ی متن‌ساده‌انتخابی انطباقی ایمن باشد.

۴.۳.۳.۲ احرازهویت نامتقارن

فرض کنید اثبات‌کننده و تاییدکننده یک کلیدعمومی pk را به‌اشتراک گذاشته و تنها اثبات‌کننده از کلیدخصوصی sk وابسته به کلیدعمومی آگاهی دارد. همچنین فرض کنید S_{sk} بیانگر الگوریتم امضا با کلیدخصوصی sk و V_{pk} الگوریتم تاییدساز وابسته به الگوریتم امضا می‌باشد که از کلیدعمومی pk استفاده می‌کند. مطابق با شکل د، در آغاز پروتکل شناسایی ، تاییدکننده چالش $c \in_R \{۰, ۱\}^k$ را برای اثبات‌کننده ارسال می‌کند و درادامه اثبات‌کننده در جواب چالش دریافت‌شده، مقدار $r = S_{sk}(c)$ را برمی‌گرداند. درادامه تاییدکننده بررسی می‌کند که آیا خروجی $V_{pk}(c, r)$ معتبر می‌باشد یا خیر. اگر خروجی معتبر باشد بیانگر آن است که مقدار r درواقع امضای پیام c تحت کلبدعمومی pk می‌باشد.

برای مقاومت دربرابر حمله‌ی استراق‌سمع لازم است که طرح امضای دیجیتال در برابر حمله‌ی پیام شناخته شده ایمن باشد و برای جلوگیری از حمله‌ی تاییدکننده‌متقلب لازم است تا این طرح در برابر حمله‌های پیام‌انتخابی انطباقی ایمن باشد.

۴.۳.۲ پروتکل شناسایی دانش صفر

۱۴

طرح بخش قبل زمانی ایمن است که از طرح‌های احرازهویت یا رمزنگاری بسیارقوی(مثل طول کلید بزرگ و ..) استفاده شود.همچنین هزینه‌ی مقاومت در برابر حمله‌ی پیام انتخابی تطبیقی برای یک طرح امضای دیجیتال نیز بسیار بالا می‌باشد.علاوه‌براین محاسبه‌ی پاسخ چالش نیز برای اثبات‌کننده ممکن است هزینه‌بر باشد. بنابراین برای رفع مشکلات طرح‌های قبلی به معرفی یک پروتکل جدید می‌پردازیم.

در این بخش به معرفی پروتکل شناسایی دانش‌صفر می‌پردازیم که یکی از ویژگی‌های باارزش آن این است که با هر تلاش یک تاییدکننده‌متقلب، هیچ اطلاعات مفیدی از اثبات‌کننده(ی‌معتبر) استخراج نمی‌شود. به‌بیان‌دیگر واژه‌ی دانش‌صفر بیانگر این حقیقت است که تاییدکننده‌متقلب با کسب هر اطلاعاتی که از طریق تعاملاتی که با اثبات‌کننده دارد، با اطلاعاتی که تاییدکننده‌متقلب از طریق خودش بدون تعامل با اثبات‌کننده تولیدشده است تفاوتی ندارد. به‌عبارت‌دیگر ممکن است پیامی که که‌بوسیله‌ی اثبات‌کننده ارسال شده است قابل شبیه‌سازی [15] باشد که درعمل اثبات‌کننده در آن نقشی نداشته باشد. یک تاییدکننده‌صدق به‌هرحال متقاعد می‌شود که اثبات‌کننده کلیدخصوصی را می‌داند.؟؟؟

۱.۴.۳.۲ پروتکل دانش‌صفر اشنور

۱۶

از نمونه‌های موجود پروتکل دانش‌صفر می‌توان به پروتکل اشنور اشاره کرد که براساس لگاریتم گسسته طراحی شده است.

فرض کنیم $\langle g \rangle$ یک گروه از درجه‌ی n (که یک عدد اول بسیار بزرگ است) می‌باشد و همچنین $\boldsymbol{x} \in_R \mathbb{Z}_n$ کلیدخصوصی و $\boldsymbol{h} = g^x$ نیز کلیدعمومی اثبات‌کننده می‌باشد.تاییدکننده درحین پروتکل ثبت‌نام، کلیدعمومی h را دریافت می‌کند.یک دور از پروتکل اشنور در شکل ۲۰ نشان داده شده است. درمجموع این پروتکل، به تعداد k بار به‌صورت پی‌درپی بین اثبات‌کننده و تاییدکننده تکرار

[14] ZERO-KNOWLEDGE IDENTIFICATION PROTOCOLS

[15] simulated

[16] SCHNORE ZERO-KNOWLEDGE PROTOCOL

می‌شود. k یک پارامتر امنیتی می‌باشد.یک ساختار سه قسمتی به‌مانند اکثر پروتکل‌های دانش‌صفر در هربار تکرار پروتکل اشنور به‌صورت انتقال پیام انجام می‌شود؛

- پیام اول، a می‌باشد که دراصل تعهدی برای u می‌باشد
- پیام دوم، c است که چالش نامیده می‌شود
- و پیام سوم، r می‌باشد که پاسخ نامیده می‌شود

۱.۱.۴.۳.۲ ویژگی صداقت

درابتدا می‌خواهیم این بحث را شروع کنیم که چگونه پروتکل اشنور، تاییدکننده را متقاعد می‌سازد که اثبات‌کننده واقعا کلیدخصوصی یعنی $x = log_g^h$ را می‌داند. این خاصیت، ویژگی **صداقت** نامیده می‌شود. و اگر تاییدکننده مجاب شود که اثبات‌کننده راست می‌گوید بنابراین ویژگی صداقت این پروتکل برآورده می‌شود.

اگر اثبات‌کننده مقدار x را نداند، بهترین کاری که می‌تواند انجام دهد این است که اعلان a را چنان تولید کند که پاسخ r برای حالت‌های $c = ۰$ و $c = ۱$ درست عمل کند. بدین منظور یک اثبات‌کننده متقلب، برای مهیا کردن پاسخ چالش $c = ۰$ ، مقدار اعلان را به‌صورت $a = g^u$ درنظر می‌گیرد و $r = u$ را ارسال می‌کند. همچنین برای پاسخ به چالش $c = ۱$ می‌تواند اعلان $a = g^u/h$ را تنظیم و $r = u$ را ارسال کند؛ درادامه هنگام تاییدسازی رابطه‌ی $g^r = ah$ رخ خواهد داد.

نکته مهمی که باید در اینجا ذکر شود آن است که اثبات‌کننده بدون دانستن کلیدمخفی x هیچ‌گاه نمی‌تواند پاسخی هم برای $c = ۰$ و هم برای $c = ۱$ مهیا کند. به‌عبارت‌دیگر فرض کنید بعد از ارسال اعلان a ، اثبات‌کننده‌ای قادر باشد به هر دو چالش $c = ۰$ و $c = ۱$ به‌درستی پاسخ دهد. این مطلب به این معنی است که اثبات‌کننده قادر است تا دو پاسخ $r_۰$ و $r_۱$ را تولید کند که به‌ترتیب متناسب با چالش‌های $c = ۰$ و $c = ۱$ می‌باشد. با این حساب، مقادیر a و $r_۰$ و $r_۱$ رابطه‌ی

$$g^{r_۰} = a, \quad g^{r_۱} = ah$$

را موجب می‌شوند که اشاره به رابطه‌ی زیر دارد:

$$h = g^{r_1 - r_0}$$

و رابطه‌ی بالا دراصل نشان‌دهنده‌ی آن است که اثبات‌کننده عملا مقدار x را می‌داند. دلیل این امر هم آن است که رابطه‌ی زیر برقرار است:

$$x \equiv r_1 - r_0 \pmod{n}$$

درنتیجه با هربار تکرار پروتکل اشنور، اثبات‌کننده‌متقلب به‌احتمال ۵۰٪ موفق می‌شود تا به یکی از چالش‌ها به درستی جواب دهد ولی با تکرار k بار، احتمال موفقیت اثبات‌کننده‌متقلب حداکثر به‌احتمال 2^{-k} خواهد بود که با این احتمال بسیار کم عملا مشخص می‌شود که اثبات‌کننده‌ی متقلب هیچ شانسی برای پیروز شدن در این پروتکل را ندارد و درنتیجه ویژگی صداقت ضمانت حقیقی بودن اثبات‌کننده و اشراف به دانش موردنظر رادر این پروتکل تضمین می‌کند.

۲.۱.۴.۳.۲ ویژگی دانش‌صفر

دراین قسمت می‌خواهیم نشان دهیم که پروتکل اشنور ویژگی دانش‌صفر را نیز برآورده می‌کند. یک تاییدکننده‌ی متقلب می‌تواند چندین‌بار برای به‌دست‌آوردن گفتگوهای $(a; c; r)$ در هریک از مراحل پروتکل شناسایی مشغول شود. عبارت چندین‌بار به‌معنی حداکثر $\mathcal{O}(k^{\lambda})$ برای ثابت $\lambda \in \mathbb{N}$ (محدود چندجمله‌ای در پارامترامنیتی k) می‌باشد. علاوه‌براین تاییدکننده‌ی متقلب می‌تواند تعاملات $(a; c; r)$ زیادی را به‌دست آورد.بااین حال تاییدکننده‌ی متقلب می‌تواند تعاملات بالا را بدون تعامل با اثبات‌کننده به‌دست آورد. دراین حالت تعاملات به‌دست‌آمده‌ی $(a; c; r)$ را نسخه‌ی شبیه‌سازی شده می‌نامیم که تاییدکننده‌متقلب،نقش تاییدکننده‌ی واقعی و اثبات‌کننده‌ی واقعی را خود تنها بازی می‌کند.

در قدم اول ویژگی دانش‌صفر را برای یک تاییدکننده‌ی صادق $\mathcal{V}$ نشان می‌دهیم که در این حالت تاییدکننده یک چالش c را به‌صورت اتفاقی از $\{0, 1\}$ انتخاب می‌کند.درادامه دو الگوریتم چندجمله‌ای احتمالاتی یکی برای تعاملات واقعی و دیگری برای تعاملات شبیه‌سازی شده ارائه می‌کنیم:

sdsdsd

هردو الگوریتم، تعاملات $(a; c; r)$ را به‌صورت تصادفی و موردپذیرش تولید می‌کنند با این حال احتمال آن‌که تعامل شبیه‌سازی شده چندتایی $(A; C; R) \in \langle g \rangle \times \{۰, ۱\} \times \mathbb{Z}_n$ که در عبارت $g^R = Ah^C$ صدق کند به‌صورت زیر می‌باشد:؟؟

$$Pr[(a; c; r) = (A; C; R)] = \frac{۱}{۲n}$$

با این تفاوت که که تعاملات واقعی دسترسی به کلیدخصوصی x را منجر می‌شود در صورتیکه تعاملات شبیه‌سازی شده منجر به تولید کلیدعمومی h می‌شود.

توجه ۸. حقیقت این است که یک پروتکل شناسایی در برابر یک تاییدکننده صادق(حقیقی)، ویژگی دانش‌صفر را در خود دارد که درنتیجه‌ی آن هر حمله به کلید کاهش می‌یابد؟؟؟. به‌خصوص آن‌که استراق‌سمع تعاملات میان اثبات‌کننده صادق با تاییدکننده هیچ اطلاعاتی در مورد کلیدخصوصی اثبات‌کننده افشا نمی‌کندُ حتی اگر کلیدعمومی را نیز دراختیار داشته باشیم؟؟؟.

در قدم بعدی ویژگی دانش‌صفر را برای هر حالت عمومی برای هر تاییدکننده‌ی متقلب $\mathcal{V}^*$ با زمان چندجمله‌ای احتمالاتی درنظر می‌گیریم. در اینجا از تاییدکننده‌ی متقلب $\mathcal{V}^*$ که یک نوع ماشین تورینگ احتمالاتی می‌باشد به‌عنوان یک جعبه‌سیاه [۱۷] قابل چاپ مجدد [۱۸] استفاده خواهیم کرد، به‌این معنی که :

- دسترسی $\mathcal{V}^*$ فقط در درون جعبه سیاه می‌باشد، به این معنی که تبادل پیام‌ها با $\mathcal{V}^*$ از طریق ورودی خودش و خروجی نوار می‌باشد.
- حالت $\mathcal{V}^*$ را می‌توانیم به هر حالت قبلی بازگردانی کنیم.

از قسمت ۱٫۲٫۱ یادآوری می‌کنیم که تنظیمات یک ماشین تورینگ احتمالاتی از طریق حالت قسمت کنترل متناهی معین می‌شود. محتویات نوار به موقعیت هدر نوار بستگی دارد.از طریق بازگردانی $\mathcal{V}^*$ می‌توانیم برای چندین ورودی تست کنیم تا خروجی دلخواه ما به‌دست آید.
fsdfsdfsdf

[17] black-box
[18] rewindable

در مرحله‌ی شش‌ام شبیه‌سازی، از آنجا که $c \in_R \{۰,۱\}$ ، احتمال آن‌که $c = c'$ دقیقا برابر با ۱/۲ می‌باشد.بنابراین به‌طور متوسط بعداز دو مرحله تکرار می‌توان تعاملات $(a; c; r)$ را شبیه‌سازی کرد. نتیجه آن‌که، مهم نیست تاییدکننده‌متخاصم $\mathcal{V}^*$ با چه الگوریتمی سعی در استخراج اطلاعات مهم از اثبات‌کننده می‌کند، از همین الگوریتم می‌توان به تولید تعامل توزیع‌شده منحصربه‌فرد بدون نیاز داشتن به همکاری با تاییدکننده نیز پرداخت. البته قابل ذکر است این وقایع درحالی رخ می‌دهد که تعاملات واقعی از طریق کلیدخصوصی x به‌عنوان ورودی تولید شده‌اند و تعاملات شبیه‌سازی شده از طریق کلیدعمومی h به‌عنوان ورودی تولید شده‌اند.

۲.۴.۳.۲ پروتکل اشنور

پروتکل شکل ۴٫۲ یک پروتکل دانش‌صفر می‌باشد.بنابرآنچه قبلا بحث کردیم، احتمال موفقیت اثبات‌کننده‌ی متقلب ٪۵۰ می‌باشد.با استفاده از k تکرار پی‌درپی این پروتکل، ویژگی دانش‌صفر قاعدتا حفظ خواهد شد اما احتمال موفقیت اثبات‌کننده‌ی متقلب به ۲^{-k} کاهش پیدا خواهد کردکه این عدد مقدار بسیار ناچیز به عنوان یک تابع از مقدار امنیتی k خواهد بود.

به‌دلیل آنکه هم اثبات‌کننده و هم تاییدکننده نیاز به محاسبه‌ی $\mathcal{O}(k)$ عمل توان در گروه $\langle g \rangle$ خواهند داشت بنابریان پیچیدگی محاسباتی پروتکل نتیجه نسبتا بالا می‌باشد. بنابریان اشنور همچنین استفاده ؟؟ را در شکل ۴٫۳ ارائه کرده است(که به عنوان پروتکل اشنور سناخته می‌شود). در این پروتکل، تاییدکننده چالش خود را از یک میدان بسیار بزرگ به عنوان مثال $c \in \mathbb{Z}_n$ انتخاب می‌کند.

ویژگی صداقت پروتکل اشنور می‌تواند همچون بالا آنالیز شود.فرض کنید یک اثبات‌کننده پس از ارسال a به‌عنوان اعلان، قادر باشد تا به حداقل دو چالش c و c' با این فرض که $c \neq c'$ ،به‌درستی جواب دهد. درنتیجه اثبات‌کننده قادر خواهد بود تا دو مبادله‌ی $(a; c; r)$ و $(a; c'; r')$ را تولید کند. البته ذکر این نکته لازم است که این اوامر در صورتی رخ می‌دهد که اثبات‌کننده واقعا لگاریتم گسسته زیر را بداند:

$$x = \log_g h$$

همچنین

$$g^r = ah^c, \quad g^{r'} = ah^{c'}$$

نشان‌دهنده‌ی آن است که:

$$h = g^{(r-r')/(c-c')}$$

بنابراین بعد از ارسال اعلان a اگر اثبات‌کننده کلیدخصوصی x را نداند آنگاه حداکثر می‌تواند به یک چالش به‌درستی پاسخ درست دهد.از آنجا که n حالت ممکن برای چالش وجود دارد(چالش‌ها از یک میدان بسیار بزرگ انتخاب می‌شوند)، احتمال موفقیت $۱/n$ می‌باشد که عدد بسیار کوچکی می‌باشد.

۳.۴ figure

ویژگی دانش‌صفر نیز همچنین مي تواند در پروتکل اشنور همچون بالا برای یک تاییدکننده صادق $\mathcal{V}$ اثبات شود. مراودات دو گفگتگوی واقعی(تولیدشده توسط کلیدخصوصی $x \in \mathbb{Z}_n$) و گفتگوی شبیه‌سازی شده(تولیدشده توسط کلیدعمومی $h \in \langle g \rangle$) به‌صورت زیر می‌باشند:

$$\{(a;c;r) : u, c \in_R \mathbb{Z}_n;\ a \leftarrow g^u;\ r \leftarrow_n u + cx\}$$

$$\{(a;c;r) : c, r \in_R \mathbb{Z}_n;\ a \leftarrow g^r h^{-c}\}$$

این مراودات یکتا هستند،

۵.۳.۲ اثبات دانش‌صفر

اثبات دانش‌صفر یک کلاس عمومی از پروتکل‌هایی است که بین دو بخش به‌نام های اثبات‌کننده و تاییدکننده صورت می‌پذیرد.از طریق اثبات دانش‌صفر، اثبات‌کننده تاییدکننده را متقاعد می‌کند که یک عبارت داده شده معتبر است بدون آن‌که هیچ اطلاعات ارزشمندی از حقیقت عبارت را افشا کند.

در پروتکل‌های شناسایی دانش‌صفر، عبارت اثبات‌شده شبیه چیزی مثل "من کلیدخصوصی این کلیدعمومی را می‌دانم" می‌باشد. البته عبارت می‌تواند بزرگتر و بیشتر هم باشد،به‌عنوان مثال "من کلیدخصوصی را برای این کلید عمومی یا آن کلیدعمومی می‌دانم،که در هر حالت کلیدهای خصوصی می‌توانند متفاوت باشند."

درحقیقت، نظریه‌ی اثبات دانش‌صفر گویای این است که هر جمله‌ی چندجمله‌ای غیرمعین [۱۹] می‌تواند به طور موثر در دانش‌صفر اثبات شود.

۱.۵.۳.۲ پروتکل زیگما

مفهوم یک پروتکل زیگما، پروتکل شناسایی (ارائه شده توسط اشنور) را تعمیم می‌دهد. یک پروتکل زیگما فقط باید دربرابر تاییدکننده‌ی صادق، ویژگی دانش‌صفر را دارا باشد.؟؟
به‌دلایل فنی، یک پروتکل زیگما در واقع نیاز دارد تا یک دانش‌صفرِ تاییدکننده‌ی صادقِ ویژه باشد، به‌این‌معنی که شبیه‌ساز بتواند چالش c را به‌عنوان یک ورودی اضافی انتخاب و یک گفگتگو براساس چالش مشخص c تولید کند.تمام پروتکل‌های ذکرشده در بالا همگی دانش‌صفر تاییدکننده‌صادق ویژه می‌باشند(بررسی آن آسان است.)

$R = \{(v; w) \subseteq V \times W\}$ را یک رابطه‌ی باینری فرض کنیم که $v \in V$ معرف ورودی عمومی اثبات‌کننده و تاییدکننده می‌باشد و $w \in W$ معرف **شاهد** است که ورودی خصوصی اثبات‌کننده می‌باشد.

---
[19] NP-statment

تعریف ۱.۳.۲ . یک پروتکل زیگما برای رابطه‌ی R ، پروتکلی طبق شکل ۵/۱ است بین یک اثبات‌کننده $\mathcal{P}$ و یک تاییدکننده $\mathcal{V}$ که دارای ویژگی‌های زیر می‌باشد:

- **کامل بودن.** اگر اثبات‌کننده $\mathcal{P}$ و تاییدکننده $\mathcal{V}$ از پروتکل پیروی کنند آنگاه تاییدکننده همواره خروجی **موافقت** را تولید می‌کند.
- **صداقت ویژه.** یک الگوریتم چندجمله‌ای احتمالاتی E (استخراج‌کننده) وجود دارد که با دریافت هر $v \in V$ و هر جفت گفتگوهای موردپذیرش $(a; c; r)$ و $(a'; c'; r')$ که $c \neq c'$ همواره شاهد w را محاسبه می‌کند که رابطه‌ی $(v; w) \in R$ برقرار می‌باشد.
- **دانش‌صفرِ تاییدکننده‌صادقِ ویژه.** یک الگوریتم چندجمله‌ای احتمالاتی S (شبیه‌ساز) وجود دارد که با دریافت هر $v \in L_R$ و چالش $c \in C$ ، یک گفتگوی $(a; c; r)$ را با همان توزیع احتمال تولید گفتگو بین اثبات‌کننده‌ی صادق $\mathcal{P}$ و تاییدکننده‌ی صادق $\mathcal{V}$ با ورودوی عمومی v و چالش c تولید می کند. البته قابل ذکر است که اثبات‌کننده‌ی $\mathcal{P}$ از شاهد w برای برقراری رابطه‌ی $(v; w) \in R$ استفاده می‌کند.

همچنین ذکر این نکته نیز لازم است که اگر C شامل یک عضو باشد آنگاه پروتکل زیگما را بدیهی می‌نامیم.

ویژگی صداقت ویژه اشاره یه این نکته دارد که احتمال موفقیت اثبات‌کننده‌ی متقلب حداکثر $۱/n$ است که در اینجا n نشان‌دهنده‌ی اندازه‌ی فضای چالش‌ها یا C می‌باشد.ازاین‌رو بااین فرض که n نسبتا بزرگ می‌باشد آنگاه یک پروتکل زیگما دانش یک شاهد w برای یک ورودی عمومی v را اثبات می‌کند.

گزاره ۳. پروتکل شکل ۴/۳ یک پروتکل زیگما برای رابطه‌ی $\{(h; x) : h = g^x\}$ می‌باشد.

اثبات ۲ .

- *کامل بودن. این ویژگی به‌راحتی قابل بیان است:*

$$g^r = g^{u+cx} = g^u(g^x)^c = ah^c$$

قابل ذکر است که اثبات‌کننده $r = u + cx$ را محاسبه می‌کند.

- صداقت ویژه. با دریافت دو گفتگوی موردپذیرش $(a; c; r)$ و $(a'; c'; r')$ درحالیکه $c \neq c'$ ، آنگاه داریم:

$$g^r = ah^c, g^{r'} = ah^{c'} \quad (۱.۲)$$

$$\Rightarrow g^{r-r'} = h^{c-c'} \quad (۲.۲)$$

$$\iff h = \frac{r-r'}{c-c'} \quad (۳.۲)$$

ازاین‌روز شاهد به صورت $x = (r-r')/(c-c')$ به‌دست می‌آید.

- دانش‌صفرِ تاییدکننده‌ی صادقِ ویژه. برای نشان دادن این ویژگی نیز، با دریافت دو توزیع از گفتگوها، یکی با تاییدکننده‌ی صادق(که با اجرای پروتکل رخ می‌دهد و شاهد x به‌عنوان ورودی استفاده می‌شود) و دیگری گفتگوی شبیه‌سازی شده(که با انحراف پروتکل، از کلیدعمومی h به‌عنوان ورودی استفاده می‌شود) داریم:

$$\{(a; c; r) : u \in_R \mathbb{Z}_n; a \leftarrow g^u; r \leftarrow_n u + cx\}, \quad (۴.۲)$$

$$\{(a; c; r) : r \in_R \mathbb{Z}_n; a \leftarrow g^r h^{-c}\} \quad (۵.۲)$$

که با دریافت چالش c دلخواه، این توزیع‌ها یکتا هستند(هر توزیع دقیقا با احتمال $۱/n$ رخ می‌دهند).

۲.۵.۳.۲ پروتکل زیگمای غیرتعاملی

یادآوری می‌کنیم که اساسا دو نمونه از طرح‌های احرازهویت وجود دارند:طرح‌های احرازهویت تعاملی(مانند طرح‌های شناسایی) و طرح‌های احرازهویت غیرتعاملی(مانند امضای دیجیتال). به‌طور مشابه در اثبات دانش‌صفر نیز دو فرم وجود دارد:طرح‌های اثبات دانش‌صفر تعاملی و طرح‌های اثبات دانش‌صفر غیرتعاملی. یک طرح اثبات غیرتعاملی شامل یک پروتکل است که بوسیله‌ی آن یک

اثبات‌کننده، یک تاییدکننده را متقاعد می‌سازد که یک بیانه خاص دارد. در مقابل یک طرح اثبات غیرتعاملی شامل دو الگوریتم است یکی برای آن‌که اثبات‌کننده یک اثبات را برای یک بیانه خاص تولید کند و الگوریتم دیگر برای آن است که تاییدکننده صحت اثبات داده شده را بررسی کند.

یک روش ساده و موثر وجود دارد که هر پروتکل زیگما را غیرتعاملی می‌سازد، این روش توسط فیات و شمیر ارائه شده‌اند و به طرح فیات‌ـ شمیر شناخته می‌شود.
؟؟؟؟
یک ویژگی متمایز اثبات زیگمای غیرتعاملی این است که هر موجودیتی می‌تواند نقش اثبات‌کننده را ایفا کند. به‌عنوان نتیجه می‌توان گفت که اثبات زیگمای غیرتعاملی می‌تواند توسط هر موجودیتی مستقلا تایید شود(مانند امضای دیجیتال که توسط هرشخصی که علاقه به تاییدسازی آن دارد انجام می‌شود) .

۳.۵.۳.۲ امضای دیجیتال ازطریق پروتکل زیگما

یک طرح امضای دیجیتال شامل سه الگوریتم می‌باشد:

- الگوریتم تولید کلید
- الگوریتم تولید امضا
- الگوریتم تاییدسازی امضا

با یک مثال می‌خواهیم نشان دهیم تا چگونه از هر پروتکل زیگما (که برای شناسایی استفاده می‌شود) یک طرح امضای دیجیتال معادل بسازیم. برای این منظور لازم است تا از تابع هش H استفاده کنیم.

در پروتکل اشنور(شکل ۴/۳)، برای اثبات دانش $\boldsymbol{x = \log_g h}$ از کلیدخصوصی x و کلیدعمومی h استفاده می‌کنیم. طرح امضای دیجیتال اشنور از طریق اعمال پروتکل فیات‌ـ شمیر به پروتکل اشنور به‌دست می‌آید. این به‌معنی ان است که چالش c ازطریق مقدار هش به‌دست آمده

از طریق تابع هش با دو ورودی پیام M و اعلان a به‌دست می‌آید، $c \leftarrow H(a, M)$. البته قابل ذکر است که چالش c به‌صورت کاملا تصادفی و مستقل از اعلان a به‌دست می‌آید. در این روش هیچ تعاملی بین تاییدکننده و تاییدکننده نیاز نیست و تاییدکننده خود به تولید چالش c می‌پردازد. امنیت این طرح امضا به تابع هش H به‌عنوان یک اوراکل تصادفی می‌باشد.از آنجا که H یک تابع تصادفی می‌باشد، بنابریان به‌نظر می‌رسد مقدار چالش $c = H(a, M)$ با هر پیام M متفاوتی برای امضا، به همان میزان تصادفی باشد که اثبات‌کننده چالش را از تاییدکننده‌ی صادق با تعامل دریافت می‌کند. علاوه بر این لازم است تا اثبات‌کننده قبل از محاسبه‌ی چالش c ، اعلان a را انتخاب کند. همچنین فرض می‌شود که $n \leq ۲^k$. درنتیجه رشته بیتی در $\{۰, ۱\}^k$ می‌تواند با مقدارهای عددی در $\{۰, ۱, \ldots, ۲^k - ۱\} \subseteq \mathbb{Z}_n$ یکتا باشد؟؟. طرح امضای اشنور از سه الگوریتم زیر تشکیل شده است، که تمام کاربران گروه $\langle g \rangle$ با مرتبه‌ی عدد اول n و تابع هش $H : \{۰, ۱\}^* \rightarrow \{۰, ۱\}^k$ به اشتراک می‌گذارند.

- ***الگوریتم تولید کلید.*** یک جفت کلید $(h; x)$ با انتخاب کلیدخصوصی $x \in_R \mathbb{Z}_n$ و مقدار کلیدعمومی h که از طریق $h \leftarrow g^x$ به‌دست می‌آید، ساخته می‌شوند.

- ***الگوریتم تولید امضا.*** با ورودی پیام M و کلیدخصوصی x و انتخاب $u \in_R \mathbb{Z}_n$ و محاسبه‌ی $a \leftarrow g^u$ ، $c \leftarrow H(a, M)$ ، $r \leftarrow_n u + cx$ ، امضای پیام M ، زوج (c, r) می‌باشد.

- ***الگوریتم تاییدسازامضا.*** با دریافت پیام M و زوج (c, r) به‌عنوان امضا و یک کلیدعمومی h ،امضای (c, r) بر پیام M مورد پذیرش است اگر تساوی زیر برقرار باشد:

$$c = H(g^r h^{-c}, M)$$

کاملا واضح است که طرح امضای اشنور بسیار کارآمد است.هزینه‌ی محاسباتی تولید امضا وابسته به زمان انجام یک عمل توان برای تشکیل g^u می‌باشد ؟؟ . هزینه محاسباتی تاییدسازی امضا وابسته به زمان انجام دوعمل توان رسانی $g^r h^{-c}$ می‌باشد

فصل ۳

امضای دیجیتال بر اساس همسانی‌ها

۱.۳ پروتکل‌های اثبات

۱.۱.۳ پروتکل زیگما

[1]

در یک سیستم اثبات ، اثبات کننده $\mathcal{P}$ خواهان آن است تا اظهار x را برای تاییدکننده $\mathcal{V}$ اثبات کند با این ویژگی که برای متقاعد کردن تاییدکننده برای اظهار x ، شاهد w را برای ادعای خود در اختیار دارد. در هر پروتکل اثبات یک رابطه باینری به نام R وجود دارد، به این معنی که اگر اظهار [2] x ادعا شود آنگاه باید شاهدی [3] به نام w برای آن موجود باشد که در این صورت آن را به صورت $(x,w) \in R$ نمایش خواهیم داد. به طور مثال در امضای دیجیتال اگر کلید عمومی v برای امضای S ادعا شود آنگاه کلیدخصوصی s به عنوان شاهدی برای کلیدعمومی می‌باشد که رابطه‌ی این دو به صورت زیر تعریف می‌شود:

$$(v,s) \in R \,; \quad R : \{\, sv \equiv 1 \mod ((p-1)(q-1))\}$$

به عبارت دیگر اگر کلید عمومی (ادعا) منتشر شده با کلیدخصوصی(شاهد) امضاکننده واقعا

[1] sigma protocol
[2] Statement
[3] Witness

رابطه‌ای داشته باشند آنگاه برای هر چالشی که توسط تاییدکننده (در اینجا چالش ، پیام m می‌باشد) برای امضاکننده (اثبات‌کننده)ارسال می‌شود، باید بعداز دریافت امضای پیام بتوانیم با کلیدعمومی به پیام ارسال شده(m) برسیم. بنابراین زمانی ادعای امضاکننده مبنی بر داشتن شاهد یا کلیدخصوصی محرز می‌شود که بتوانیم با کلیدعمومی منتشر شده توسط وی به پیام برسیم ،در غیراینصورت ادعا موردقبول واقع نمی‌شود و درنتیجه آشکار می‌شود که رابطه‌ای بین کلیدها برقرار نبوده و مهمتر آنکه امضا متعلق به امضاکننده نیست.

۱.۱.۱.۳ تعریف

پروتکل زیگما $\Sigma = ((P^{1}, P^{2}), V)$ ، یک سیستم اثبات تعاملی است که شامل چهار قسمت به ترتیب زیر می‌باشد:

- تعهد [4] $com = P^{1}(x, w)$ توسط اثبات کننده ارائه می‌شود و به معنی آن است که اثبات‌کننده برای ادعای خود یعنی x ، شاهد w را دراختیار دارد.

- چالش ch که به‌صورت تصادفی و یکنواخت از یک دامنه‌ی مجاز N_{ch} ، توسط تاییدکننده برای به چالش کشیدن ادعای مطرح‌شده توسط اثبات‌کننده انتخاب می‌شود.

- پاسخ $resp = P^{2}(x, w, com, ch)$ ، براساس چالش دریافتی ch از طرف تاییدکننده ، توسط اثبات‌کننده محاسبه می‌شود.

- خروجی $V(x, com, ch, resp)$ توسط تاییدکننده محاسبه می‌شود و مقدارآن صفر یا یک می‌باشد و معین آن است که اثبات مورد پذیرش واقع شده است یا خیر، بنابراین اگر خروجی صفر باشد به معنی رد و نپذیرفتن اثبات و خروجی یک به معنی تایید اثبات می‌باشد.

یک پروتکل زیگما علاوه بر قسمت‌های بالا که در هر سیستم اثبات دانش‌صفرتعاملی وجود دارد، باید ویژگی‌های زیر را نیز دارا باشد :

[4] commitment

تمامیت [۵]:

اگر اثبات کننده $\mathcal{P}$ واقعا شاهد w را برای اظهار x بداند آنگاه طبق این پروتکل ، تاییدکننده $\mathcal{V}$ ادعای اثبات کننده را می‌پذیرد. به‌عبارت دیگر احتمال آنکه اثبات‌کننده، شاهد w را بداند ولی تاییدکننده، متقاعد نشود برابر با صفردرصد می‌باشد.که به‌صورت زیر نمایش می‌دهیم:

$$Pr(P(x,y) \leftrightarrow V(x) \rightarrow ۱) = ۱$$

صداقت ویژه [۶]:

الگوریتم چندجمله‌ای استخراج [۷] E_Σ وجود دارد که با دریافت هر جفتی از تعاملات معتبر $(com, ch, resp)$ و $(com, ch', resp')$ با شرط آنکه $ch \neq ch'$ و هر دو تعامل مورد پذیرش تاییدکننده باشد، می‌تواند یک شاهد w که $(x,w) \in R$ را محاسبه کند.
این ویژگی تضمین می‌کند که تاییدکننده حتما با یک اثبات‌کننده صادق روبرو است که دانش موردنظر و شاهد را می‌داند زیرا درغیراینصورت الگوریتم هیچ شاهدی را استخراج نمی‌کند که در این صورت تاییدکننده پی می‌برد که تاییدکننده، متقلب است و به دانش ادعایی دسترسی ندارد.

توجه ۹. توجه به این نکته لازم است که در اینجا برای یک اظهار com دو چالش ch و ch' همزمان برای آن ارسال شده و جواب‌های $resp$ و $resp\ '$ دریافت می‌شود که در هر پروتکل اثبات همچون زیگما هیچ‌گاه این اتفاق مجاز به انجام نیست و برای هر تعهد com باید فقط یک چالش مورد سوال قرار گیرد. اما در اینجا فرض شده است اگر برای هر تعهد بتوانیم دو چالش متفاوت ارسال و دو پاسخ دریافت کنیم آنگاه الگوریتمی موجود هست که باعث افشای شاهد می‌شود.
همچنین قابل ذکر است که دامنه‌ی چالش‌ها N_{ch} ، حداقل دو یا بیشتر فرض شده است.

دانش صفر تاییدکننده صادق [۸] :

---

[5] Completeness

[6] Special soundness

[7] polynomial time extractor

[8] Honest-verifier zero-knowledge (HVZK)

الگوریتم چندجمله‌ای شبیه ساز S_Σ وجود دارد که یک خروجی شبیه‌سازی شده‌ای به فرم $(com, ch, resp)$ تولید می‌کند که نسبت به خروجی معتبر دریافت شده توسط تعاملات حقیقی بین اثبات‌کننده و تاییدکننده هیچ نوع تمایز و فرقی ندارد. این ویژگی نیز بیانگر آن است که تاییدکننده هیچ اطلاعاتی از دانش موردنظر اثبات‌کننده دریافت نمی‌کند و تنها متقاعد می‌شود که تاییدکننده به‌راستی دانش را می‌داند. البته در اینجا فرض شده است که تاییدکننده، صادق باشد.

توجه ۱۰. اگر پروتکل زیگما را به صورت خلاصه‌شده‌ی $\Sigma = (P, V)$ در نظر بگیریم آنگاه $P = (P^{۱}, P^{۲})$ خواهد بود.

توجه ۱۱. اثبات‌دانش‌صفرهویت همسانی مبنای گفته شده در مثال بالا **??** در اصل یک پروتکل زیگما می‌باشد. در بخش ۵ نشان خواهیم داد که تمام ویژگی‌های یک پروتکل زیگما را برآورده می‌کند.

۲.۱.۳ سیستم اثبات غیرتعاملی

[9]

یک سیستم اثبات غیرتعاملی شامل دو الگوریتم

- اثبات‌کننده $P(x, w)$ که یک اثبات π را برای اظهار x (که دارای شاهد w می‌باشد) تولید می‌کند.
- تاییدکننده $V(x, \pi)$ که با دریافت اثبات π ، خروجی **تایید** یا **انکار** زا تولید می‌کند.

و همچنین سه ویژگی زیر می‌باشد:

تمامیت :

[9] Non-interactive Proof System

اگر شاهد w برای اظهار x واقعا وجود داشته باشد آنگاه تاییدکننده V ، اثبات $\pi = P(x, w)$ را می‌پذیرد.

دانش صفر[۱۰] :

الگوریتم چندجمله‌ای شبیه ساز S که به یک اوراکل تصادفی دسترسی دارد، موجود است که می‌تواند اثبات‌هایی مشابه و غیرقابل تمایز با اثبات‌های تولید شده توسط اثبات کننده $\mathcal{P}$ را تولید(یا شبیه‌سازی) کند.

شبیه‌ساز صداقت با ویژگی استخراج آنلاین[۱۱]

الگوریتم چندجمله‌ای استخراج E وجود دارد که توانایی تولید یک شاهد w برای ادعای x مطرح شده توسط اثبات‌کننده را دارا می‌باشد.
لازم به ذکر است که اثبات π متناظر با ادعای x و شاهد w با الگوریتم شبیه‌ساز S به‌دست می‌آید.

۳.۱.۳ ساخت آنره

ساخت آنره ، پروتکل زیگما (Σ) را به یک سیستم اثبات غیرتعاملی (P_{OE}, V_{OE}) تغییرشکل می‌دهد بطوریکه اگر پروتکل زیگما (Σ) شامل ویژگی‌های تمامیت ، صداقت ویژه و دانش صفر باشد آنگاه نتیجه یک سیستم اثبات دانش‌صفر غیرتعاملی با ویژگی شبیه ساز صداقت با استخراج آنلاین خواهد بود.
فرض کنید یک پروتکل زیگما به صورت $\Sigma = (P_\Sigma, V_\Sigma)$ که $P_\Sigma = (P_\Sigma^1, P_\Sigma^2)$ داشته باشیم که c چالش ممکن متمایز در دامنه‌ی چالش ها (N_{ch}) داشته باشد و قرار است پروتکل به تعداد t بار(که به پارامتر امنیتی λ بستگی دارد) اجرا شود. و همچنین فرض کنید اوراکل‌های تصادفی

[10] Zero-knowledge (NIZK)
[11] Simulation-sound online-extractability

کوانتومی G و H را نیز در اختیار داریم.
(P_{OE}, V_{OE}) یک سیستم اثبات غیرتعاملی براساس پروتکل زیگما می‌باشد که P_{OE} به عنوان اثبات‌کننده و V_{OE} به عنوان تاییدکننده از طریق الگوریتم های ۱.۳.۲.۳ و ۲.۳.۲.۳ به دست می‌آیند.

لازم به ذکر است در طرح امضای دیجیتال معرفی شده در این پایان نامه، مقادیر فرض شده در پروتکل زیگما به صورت $N_{ch} = \{۰, ۱\}$ ، $c = ۲$ و $t = ۲\lambda$ می‌باشند.

۱.۳.۱.۳ الگوریتم اثبات‌کننده

۱. همان‌طور که در خط اول الگوریتم اثبات‌کننده ذکر شده است خواهان آن هستیم که تعداد $t \cdot c$ اثبات (از طریق پروتکل زیگما) تولید کنیم. دلیل استفاده از این مقدار به این خاطر است که طبق آنچه قبلا در پروتکل زیگما اشاره کردیم برای امنیت کامل لازم است تا پروتکل زیگما به تعداد t بار تکرار شود. اما مقدار c ، به این دلیل است که در پروتکل زیگما، تاییدکننده به انتخاب خود یک چالش را از میان چالش‌های مجاز(دامنه‌ی چالش‌ها) انتخاب می‌کند ولی به دلیل آنکه در ساخت آنره هدف آن است که ارتباط با تاییدکننده حذف شود بنابراین در این سیستم تمام چالش‌ها شبیه‌سازی می‌شود تا هیچ دانش خاصی در به‌کارگرفتن چالش در هر مرحله صورت نگیرید و اثبات کاملا تصادفی و یکنواخت باشد.

۲. بنابراین برای ایجاد امنیت کامل لازم است تا به تعداد t مرتبه رویه اثبات شبیه‌سازی شود که بدین منظور از یک حلقه‌ی تکرار استفاده می‌کنیم.

۳. در هر بار تکرار حلقه، ادعای x با الگوریتم $P_{\Sigma}^{۱}$ مطرح می‌شود و به عنوان یک تعهد خروجی الگوریتم در متغیر com_i ذخیره می‌گردد

۴. به منظور شبیه‌سازی تمام حالت‌های ممکن برای ارسال درخواست یک چالش توسط تاییدکننده ، تمام مقادیر دامنه‌ی چالش‌ها(که c حالت ممکن برای آن وجود دارد) توسط حلقه‌ی تکرار تولید می‌شود

۵. در این مرحله یک چالش به صورت کاملا تصادفی و یکنواخت از دامنه‌ی چالش‌های مجاز انتخاب می‌شود و مقدار انتخاب شده از دامنه‌ی چالش‌ها حذف می‌گردد

۶. براساس چالش دریافتی می‌بایست یک پاسخ از طرف تاییدکننده ارسال شود بنابراین الگوریتم $P_{\Sigma}^{۲}$ اجرا می‌شود و خروجی در متغیر $resp_{i,j}$ ذخیره می‌شود

۷. در این مرحله، از پاسخ هر چالش با الگوریتم G هش گرفته می‌شود و در متغیر $h_{i,j}$ ذخیره می‌شود. دلیل این امر آن است که مطمئن باشیم که پاسخ هر چالش بدون تغییر ذخیره شده است.

تا پایان این مرحله به تعداد $t \cdot c$ پاسخ داریم که البته هش شده است و غیرقابل‌تغییر می‌باشند.

۸. مهمترین قسمت پروتکل آنره در این قسمت از الگوریتم انجام می‌پذیرد. از آنجا که با پروتکل آنره خواهان آن هستیم که از یک اثبات دانش‌صفر تعاملی به یک اثبات‌دانش‌صفر غیرتعاملی برسیم بنابراین با استفاده از مجموع تعهدها،چالش‌ها و پاسخ‌های به هر چالش که توسط خود اثبات‌کننده در مراحل قبلی الگوریتم شبیه‌سازی شد به تولید چالش‌هایی می‌پردازیم که شبیه به چالش‌هایی باشد که از طرف تاییدکننده دریافت می‌شود با این ویژگی که کاملا تصادفی تولید شوند. در نتیجه برای تولید چالش‌ها از متغیرهای به‌دست آمده از مراحل قبلی الگوریتم با تابع H هش می‌گیریم و از آنجا که خروجی هش کاملا تصادفی می‌باشد بنابراین در هریک از J_i ها مقدار یک یا صفر خواهیم داشت با این ویژگی که تابع هش H و ورودی‌های آن عمومی می‌باشند به این منظور که رشته $(J_۱ \parallel \cdots \parallel J_t)$ توسط تاییدکننده قابل بررسی باشد.

۹. بعد از اتمام تمام مراحل بالا، اثبات π به عنوان خروجی الگوریتم اثبات‌کننده تولید می‌شود. اثبات موردنظر شامل یک چندتایی شامل تمام تعهدها(com_i) ،چالش‌ها($ch_{i,j}$) و همچنین هش‌شده‌ی پاسخ‌های اثبات‌کننده($h_{i,j}$) می‌باشدو آخرین قطعه‌ی این اثبات شامل پاسخ‌های شفاف ارائه‌شده توسط اثبات‌کننده می‌باشد با این تفاوت که چینش پاسخ‌ها براساس چینش تولید شده نمی‌باشد به عبارت دیگر پاسخ‌های $resp_{i,j}$ متناسب با چالش‌های ساخته شده توسط J_i ها به دست می‌آید یعنی به جای ارسال $resp_{i,j}$ ، الگوریتم $resp_{i,J_i}$ را به دست آورده و آن‌ها را در اثبات π قرار می‌دهد.

الگوریتم ۱ Prover : input on P_{OE} (x, w)

// Create t.c proofs and hash each response $i = 1$ **to** t $com_i \leftarrow P^1_\Sigma(x, w)$ $j = 1$ **to** c $ch_{i,j} \leftarrow_R N_{ch} \setminus \{ch_{i,1}, \cdots, ch_{i,j-1}\}$ $resp_{i,j} \leftarrow P^2_\Sigma(x, w, com, ch_{i,j})$ $h_{i,j} \leftarrow G(resp_{i,j})$ // Get challenge by hashing $J_1 \parallel \cdots \parallel j_t \leftarrow H(x(com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$ Get challenge by hashing // return proof **return** $\pi \leftarrow ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$ return proof

۲.۳.۱.۳ الگوریتم تاییدکننده

۱. در آغاز این الگوریتم، تاییدکننده خود جدای از آن‌که چالش‌ها را دریافت کند به تولید چالش‌ها می‌پردازد به عبارت دیگر چالش موجود در پروتکل آنره به صورت تصادفی ولی با یک روش مشخص به دست می‌آید چنان‌که هم اثبات‌کننده و هم تاییدکننده می‌توانند با یک تابع مشخص(H) به آن برسند. لازم به ذکر است که تمام ورودی‌های این تابع بوسیله اثبات‌کننده به‌عنوان اثبات (π) برای تاییدکننده فرستاده شده است.

۲. برای آن‌که اثبات π توسط تاییدکننده تایید شود لازم است که بررسی‌های زیر به‌تعداد ادعاهای مطرح‌شده توسط اثبات‌کننده صورت بپذیرد بنابراین از یک حلقه تکرار استفاده می‌کنیم

۳. تمام چاش‌های تولید شده در الگوریتم قبلی باید نسبت به هم متفاوت باشند بنابراین بررسی می‌شود که آیا هر دو زوج متفاوت از چالش‌ها باهم متفاوت هست یا خیر

۴. در این خط از الگوریتم بررسی می‌شود که آیا مقدار هش پاسخ‌ها با هش دریافتی در اثبات باهم برابر هستند یا خیر. دلیل این امر آن است که تابع G عمومی است و تاییدکننده باید از هش دریافتی مطمئن شود

۵. تاییدکننده بررسی می‌کند که آیا خروجی الگوریتم V_Σ براساس ورودی‌های متناظر با آن یک می‌شود یا خیر. ذکر این نکته لازم است به دلیل آن‌که هنگام دریافت اثبات π از طرف اثبات‌کننده، پاسخ‌ها با چینش متناظر با چالش‌های به دست آمده از طریق J_i ها بود یعنی $resp_{i,J_i}$ ها را دریافت کردیم بنابراین برای تایید پاسخ متناظر با چالش‌ها لازم است که به‌جای $ch_{i,j}$ از مقدار ch_{i,J_i} که متناظر با $resp_{i,J_i}$ می‌باشد استفاده کنیم

۶. اگر تمام بررسی‌های بالا صحیح باشد آنگاه خروجی الگوریتم یک می‌باشد به این معنی که اثبات توسط تاییدکننده پذیرفته شده است

الگوریتم ۲ Verifier : V_{OE} on input (x, π) where

$\pi = ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$

// Compute the challenge hash $i = 1$ **to** t **check** $ch_{i,1}, \cdots ch_{i,m} pairwisedistinct$ **check** $h_{i,J_i} = G(resp_i)$ **check** $V_\Sigma(x, com_i, ch_{i,J_i}, resp_i) = 1$ all checks succeed **return** 1

۲.۳ ساخت آنره

[۱۲]

برای ساخت یک سیستم امضای دیجیتال برپایه‌ی پروتکل اثبات‌دانش‌صفر معرفی شده در بخش قبل لازم است در پروتکل تغییراتی صورت گیرد. به عبارت دیگر چنان‌که در توضیحات بخش قبلی ملاحظه شد در پروتکل اثبات‌دانش‌صفر نیاز است که بین اثبات‌کننده و تاییدکننده یک تعامل دوسویه برقرار باشد یعنی هنگامی که پگی قصد دارد تا دانشی را به ویکتور اثبات کند لازم است که به چالش‌هایی که از طرف ویکتور می‌آید پاسخی مناسب داده شود اما همان‌طور

[12] Unruh's Construction

که در امضای دیجیتال مشاهده شد برای اثبات یک دانش (هویت امضاکننده) لزومی به ارتباط دوسویه بین اثبات‌کننده و تاییدکننده نمی‌باشد به عبارت دیگر برای آنکه تاییدکننده قانع شود که امضا(اثبات) از طرف امضاکننده است لزومی ندارد تا با امضاکننده در ارتباط باشد و تنها کافی است با کلیدعمومی منتشر شده توسط امضاکننده به بررسی و تایید امضا بپردازد.
برای آن که بتوانیم از طریق پروتکل اثبات دانش صفر هویت به امضای دیجیتال برسیم لازم است طرح اولیه خود که به سیستم اثبات تعاملی [13] معروف است را به یک سیستم اثبات غیرتعاملی [14] تبدیل کنیم. اولین بار فیات و شمیر طرحی با نام خود ارائه کردند [۹] که یک سیستم اثبات دانش صفر تعاملی را به یک سیستم اثبات دانش صفر غیرتعاملی تبدیل می‌کرد ولی به دلیل آن که این طرح در برابر حملات کوانتومی ایمن نبود، اخیرا آنره در [۱۹] سیستمی جدید را معرفی کرده است که برخلاف طرح فیات‌ـشمیر در برابر حملات کوانتومی نیز ایمن می باشد. بنابراین در ادامه به معرفی طرح ارائه شده توسط آنره می‌پردازیم که یک سیستم اثبات تعاملی با ویژگی‌های خاص به نام پروتکل زیگما را به یک سیستم اثبات دانش صفر غیرتعاملی تبدیل می‌کند که برای ساخت طرح امضای دیجیتالی مقاوم کوانتومی خود از این طرح سود می‌بریم.

در یک سیستم اثبات ، اثبات کننده $\mathcal{P}$ خواهان آن است تا اظهار x را برای تاییدکننده $\mathcal{V}$ اثبات کند با این ویژگی که برای متقاعد کردن تاییدکننده برای اظهار x ، شاهد w را برای ادعای خود در اختیار دارد. در هر پروتکل اثبات یک رابطه باینری به نام R وجود دارد، به این معنی که اگر اظهار [15] x ادعا شود آنگاه باید شاهدی [16] به نام w برای آن موجود باشد که در این صورت آن را به صورت $(x, w) \in R$ نمایش خواهیم داد. به طور مثال در امضای دیجیتال اگر کلید عمومی v برای امضای S ادعا شود آنگاه کلیدخصوصی s به عنوان شاهدی برای کلیدعمومی می‌باشد که رابطه‌ی این دو به صورت زیر تعریف می‌شود:

$$(v, s) \in R\,; \quad R : \{\, sv \equiv ۱ \mod ((p-۱)(q-۱))\}$$

به عبارت دیگر اگر کلید عمومی (ادعا) منتشر شده با کلیدخصوصی(شاهد) امضاکننده واقعا رابطه‌ای داشته باشند آنگاه برای هر چالشی که توسط تاییدکننده (در اینجا چالش ، پیام m می‌باشد)

[13] Interactive Proof System
[14] Non-interactive Proof System
[15] Statement
[16] Witness

برای امضاکننده (اثبات‌کننده)ارسال می‌شود، باید بعداز دریافت امضای پیام بتوانیم با کلیدعمومی به پیام ارسال شده برسیم . بنابراین زمانی ادعای امضاکننده مبنی بر داشتن شاهد یا کلیدخصوصی محرز می‌شود که بتوانیم با کلیدعمومی منتشر شده توسط وی به پیام برسیم ،در غیراینصورت ادعا موردقبول واقع نمی‌شود و درنتیجه آشکار می‌شود که رابطه‌ای بین کلیدها برقرار نبوده و مهمتر آنکه امضا متعلق به امضاکننده نیست.

۳.۲.۱ پروتکل زیگما

[۱۷]

پروتکل زیگما $\Sigma = ((P^1, P^2), V)$ ، یک سیستم اثبات تعاملی است که شامل چهار قسمت به ترتیب زیر می‌باشد:

- تعهد [۱۸] $com = P^1(x, w)$ توسط اثبات کننده ارائه می‌شود و به معنی آن است که اثبات‌کننده برای ادعای خود یعنی x ، شاهد w را دراختیار دارد.
- چالش ch که به‌صورت تصادفی و یکنواخت از یک دامنه‌ی مجاز N_{ch} ، توسط تاییدکننده برای به چالش کشیدن ادعای مطرح‌شده توسط اثبات‌کننده انتخاب می‌شود.
- پاسخ $resp = P^2(x, w, com, ch)$ ، براساس چالش دریافتی ch از طرف تاییدکننده ، توسط اثبات‌کننده محاسبه می‌شود.
- خروجی $V(x, com, ch, resp)$ توسط تاییدکننده محاسبه می‌شود و مقدار آن صفر یا یک می‌باشد و معین آن است که اثبات مورد پذیرش واقع شده است یا خیر، بنابراین اگر خروجی صفر باشد به معنی رد و نپذیرفتن اثبات و خروجی یک به معنی تایید اثبات می‌باشد.

توجه. اگر پروتکل زیگما را به صورت خلاصه‌شده‌ی $\Sigma = (P, V)$ در نظر بگیریم آنگاه

[17] Sigma Protocols
[18] commitment

$P = (P^{۱}, P^{۲})$ خواهد بود.

پروتکل زیگما علاوه بر قسمت‌های بالا که در هر سیستم اثبات دانش‌صفرتعاملی وجود دارد باید ویژگی‌های زیر را نیز دارا باشد :

تمامیت [19]:

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

صداقت ویژه [20]:

الگوریتم چندجمله‌ای استخراج [21] E_Σ وجود دارد که با دریافت هر جفتی از تعاملات معتبر $(com, ch, resp)$ و $(com, ch', resp')$ با شرط آنکه $ch \neq ch'$ و هر دو تعامل مورد پذیرش تاییدکننده باشد، می‌تواند یک شاهد w که $(x, w) \in R$ را محاسبه کند.

توجه به این نکته لازم است که در اینجا برای یک اظهار com دو چالش ch و ch' همزمان برای آن ارسال شده و جواب‌های $resp$ و $resp'$ دریافت می‌شود که در پروتکل زیگما هیچ‌گاه این اتفاق مجاز به انجام نیست و برای هر تعهد com باید فقط یک چالش انجام بپذیرد اما در اینجا فرض شده است اگر فرض کنیم برای هر تعهد بتوانیم دو چالش متفاوت ارسال و دو پاسخ دریافت کنیم آنگاه الگوریتمی موجود هست که باعث افشای شاهد می‌شود.

دانش صفر تاییدکننده صادق [22] :

الگوریتم چندجمله‌ای شبیه ساز S_Σ وجود دارد که خروجی آن $(com, ch, resp)$ می‌باشد که نسبت به خروجی دریافت شده توسط تعاملات حقیقی بین اثبات‌کننده و تاییدکننده هیچ نوع تمایزی وجود ندارد.

[19] Completeness

[20] Special soundness

[21] polynomial time extractor

[22] Honest-verifier zero-knowledge (HVZK)

توجه. اثبات‌دانش‌صفرهویت همسانی مبنای گفته شده در مثال بالا **؟؟** در اصل یک پروتکل زیگما می‌باشد.

۲.۲.۳ سیستم اثبات غیرتعاملی

[۲۳] یک سیستم اثبات غیرتعاملی شامل دو قسمت می‌باشد :

- اثبات کننده $P(x, w)$ ، یک اثبات π برای اظهار x (که دارای شاهد w می‌باشد) ادعا می‌کند.
- تاییدکننده $V(x, \pi)$ اگر با اثبات π متقاعد به اظهار x شود، اثبات را **تایید** می‌کند و درغیراینصورت آن را **انکار** یا رد می‌کند.

یک سیستم اثبات غیرتعاملی (P, V) شامل سه ویژگی زیر می‌باشد:

تمامیت :

اگر شاهد w برای اظهار x واقعا وجود داشته باشد آنگاه تاییدکننده V ، اثبات

$\pi = P(x, w)$ را می‌پذیرد.

دانش صفر[۲۴] :

الگوریتم چندجمله‌ای شبیه ساز S که به یک اوراکل تصادفی دسترسی دارد، موجود است که می‌تواند اثبات‌هایی مشابه و غیرقابل تمایز با اثبات‌های تولید شده توسط اثبات کننده $\mathcal{P}$ را تولید(یا شبیه‌سازی) کند.

[23] Non-interactive Proof System
[24] Zero-knowledge (NIZK)

شبیه‌ساز صداقت با ویژگی استخراج آنلاین [25]

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

۳.۲.۳ ساخت آنره

ساخت آنره ، پروتکل زیگما (Σ) را به یک سیستم اثبات غیرتعاملی (P_{OE}, V_{OE}) تغییرشکل می‌دهد بطوریکه اگر پروتکل زیگما (Σ) شامل ویژگی‌های تمامیت ، صداقت ویژه و دانش صفر باشد آنگاه نتیجه یک سیستم اثبات دانش‌صفر غیرتعاملی با ویژگی شبیه ساز صداقت با استخراج آنلاین خواهد بود.
فرض کنید یک پروتکل زیگما به صورت $\Sigma = (P_\Sigma, V_\Sigma)$ که $P_\Sigma = (P_\Sigma^1, P_\Sigma^2)$ داشته باشیم که c چالش ممکن متمایز در دامنه‌ی چالش ها (N_{ch}) داشته باشد و قرار است پروتکل به تعداد t بار(که به پارامتر امنیتی λ بستگی دارد) اجرا شود. و همچنین فرض کنید اوراکل‌های تصادفی کوانتومی G و H را نیز در اختیار داریم.
(P_{OE}, V_{OE}) یک سیستم اثبات غیرتعاملی براساس پروتکل زیگما می‌باشد که P_{OE} به عنوان اثبات‌کننده و V_{OE} به عنوان تاییدکننده از طریق الگوریتم های ۱.۳.۲.۳ و ۲.۳.۲.۳ به دست می‌آیند.

لازم به ذکر است در طرح امضای دیجیتال معرفی شده در این پایان نامه، مقادیر فرض شده در پروتکل زیگما به صورت $N_{ch} = \{0, 1\}$ ، $c = 2$ و $t = 2\lambda$ می‌باشند.

[25] Simulation-sound online-extractability

۱.۳.۲.۳ الگوریتم اثبات‌کننده

۱. همان‌طور که در خط اول الگوریتم اثبات‌کننده ذکر شده است خواهان آن هستیم که تعداد $t \cdot c$ اثبات (از طریق پروتکل زیگما) تولید کنیم. دلیل استفاده از این مقدار به این خاطر است که طبق آنچه قبلا در پروتکل زیگما اشاره کردیم برای امنیت کامل لازم است تا پروتکل زیگما به تعداد t بار تکرار شود. اما مقدار c ، به این دلیل است که در پروتکل زیگما، تاییدکننده به انتخاب خود یک چالش را از میان چالش‌های مجاز(دامنه‌ی چالش‌ها) انتخاب می‌کند ولی به دلیل آنکه در ساخت آنره هدف آن است که ارتباط با تاییدکننده حذف شود بنابراین در این سیستم تمام چالش‌ها شبیه‌سازی می‌شود تا هیچ دانش خاصی در به‌کارگرفتن چالش در هر مرحله صورت نگیرید و اثبات کاملا تصادفی و یکنواخت باشد.

۲. بنابراین برای ایجاد امنیت کامل لازم است تا به تعداد t مرتبه رویه اثبات شبیه‌سازی شود که بدین منظور از یک حلقه‌ی تکرار استفاده می‌کنیم.

۳. در هر بار تکرار حلقه، ادعای x با الگوریتم P_{Σ}^{1} مطرح می‌شود و به عنوان یک تعهد خروجی الگوریتم در متغیر com_i ذخیره می‌گردد

۴. به منظور شبیه‌سازی تمام حالت‌های ممکن برای ارسال درخواست یک چالش توسط تاییدکننده ، تمام مقادیر دامنه‌ی چالش‌ها(که c حالت ممکن برای آن وجود دارد) توسط حلقه‌ی تکرار تولید می‌شود

۵. در این مرحله یک چالش به صورت کاملا تصادفی و یکنواخت از دامنه‌ی چالش‌های مجاز انتخاب می‌شود و مقدار انتخاب شده از دامنه‌ی چالش‌ها حذف می‌گردد

۶. براساس چالش دریافتی می‌بایست یک پاسخ از طرف تاییدکننده ارسال شود بنابراین الگوریتم P_{Σ}^{2} اجرا می‌شود و خروجی در متغیر $resp_{i,j}$ ذخیره می‌شود

۷. در این مرحله، از پاسخ هر چالش با الگوریتم G هش گرفته می‌شود و در متغیر $h_{i,j}$ ذخیره می‌شود. دلیل این امر آن است که مطمئن باشیم که پاسخ هر چالش بدون تغییر ذخیره شده است.

تا پایان این مرحله به تعداد $t \cdot c$ پاسخ داریم که البته هش شده است و غیرقابل‌تغییر می‌باشند.

۸. مهمترین قسمت پروتکل آنره در این قسمت از الگوریتم انجام می‌پذیرد. از آنجا که با پروتکل آنره خواهان آن هستیم که از یک اثبات دانش‌صفر تعاملی به یک اثبات‌دانش‌صفر غیرتعاملی برسیم بنابراین با استفاده از مجموع تعهدها،چالش‌ها و پاسخ‌های به هر چالش که توسط خود اثبات‌کننده در مراحل قبلی الگوریتم شبیه‌سازی شد به تولید چالش‌هایی می‌پردازیم که شبیه به چالش‌هایی باشد که از طرف تاییدکننده دریافت می‌شود با این ویژگی که کاملا تصادفی تولید شوند. در نتیجه برای تولید چالش‌ها از متغیرهای به‌دست آمده از مراحل قبلی الگوریتم با تابع H هش می‌گیریم و از آنجا که خروجی هش کاملا تصادفی می‌باشد بنابراین در هریک از J_i ها مقدار یک یا صفر خواهیم داشت با این ویژگی که تابع هش H و ورودی‌های آن عمومی می‌باشند به این منظور که رشته $(J_1 \parallel \cdots \parallel J_t)$ توسط تاییدکننده قابل بررسی باشد.

۹. بعد از اتمام تمام مراحل بالا، اثبات π به عنوان خروجی الگوریتم اثبات‌کننده تولید می‌شود. اثبات موردنظر شامل یک چندتایی شامل تمام تعهدها(com_i) ،چالش‌ها($ch_{i,j}$) و همچنین هش‌شده‌ی پاسخ‌های اثبات‌کننده($h_{i,j}$) می‌باشدو آخرین قطعه‌ی این اثبات شامل پاسخ‌های شفاف ارائه‌شده توسط اثبات‌کننده می‌باشد با این تفاوت که چینش پاسخ‌ها براساس چینش تولید شده نمی‌باشد به عبارت دیگر پاسخ‌های $resp_{i,j}$ متناسب با چالش‌های ساخته شده توسط J_i ها به دست می‌آید یعنی به جای ارسال $resp_{i,j}$ ، الگوریتم $resp_{i,J_i}$ را به دست آورده و آن‌ها را در اثبات π قرار می‌دهد.

الگوریتم ۳ Prover : P_{OE} input on (x, w)

// Create t.c proofs and hash each response $i = 1$ **to** t $com_i \leftarrow P_\Sigma^1(x, w)$ $j = 1$ **to** c $ch_{i,j} \leftarrow_R N_{ch} \setminus \{ch_{i,1}, \cdots, ch_{i,j-1}\}$ $resp_{i,j} \leftarrow P_\Sigma^2(x, w, com, ch_{i,j})$ $h_{i,j} \leftarrow G(resp_{i,j})$ // Get challenge by hashing $J_1 \parallel \cdots \parallel j_t \leftarrow H(x(com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$ Get challenge by hashing // return proof **return** $\pi \leftarrow ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$ return proof

---

۲.۳.۲.۳ الگوریتم تاییدکننده

۱. در آغاز این الگوریتم، تاییدکننده خود جدای از آن‌که چالش‌ها را دریافت کند به تولید چالش‌ها می‌پردازد به عبارت دیگر چالش موجود در پروتکل آنره به صورت تصادفی ولی با یک روش مشخص به دست می‌آید چنان‌که هم اثبات‌کننده و هم تاییدکننده می‌توانند با یک تابع مشخص(H) به آن برسند. لازم به ذکر است که تمام ورودی‌های این تابع بوسیله اثبات‌کننده به‌عنوان اثبات (π) برای تاییدکننده فرستاده شده است.

۲. برای آن‌که اثبات π توسط تاییدکننده تایید شود لازم است که بررسی‌های زیر به‌تعداد ادعاهای مطرح‌شده توسط اثبات‌کننده صورت بپذیرد بنابراین از یک حلقه تکرار استفاده می‌کنیم

۳. تمام چاش‌های تولید شده در الگوریتم قبلی باید نسبت به هم متفاوت باشند بنابراین بررسی می‌شود که آیا هر دو زوج متفاوت از چالش‌ها باهم متفاوت هست یا خیر

۴. در این خط از الگوریتم بررسی می‌شود که آیا مقدار هش پاسخ‌ها با هش دریافتی در اثبات باهم برابر هستند یا خیر. دلیل این امر آن است که تابع G عمومی است و تاییدکننده باید از هش دریافتی مطمئن شود

۵. تاییدکننده بررسی می‌کند که آیا خروجی الگوریتم V_Σ براساس ورودی‌های متناظر با آن یک می‌شود یا خیر. ذکر این نکته لازم است به دلیل آن‌که هنگام دریافت اثبات π از طرف اثبات‌کننده، پاسخ‌ها با چینش متناظر با چالش‌های به دست آمده از طریق J_i ها بود یعنی $resp_{i,J_i}$ ها را دریافت کردیم بنابراین برای تایید پاسخ متناظر با چالش‌ها لازم است که به‌جای $ch_{i,j}$ از مقدار ch_{i,J_i} که متناظر با $resp_{i,J_i}$ می‌باشد استفاده کنیم

۶. اگر تمام بررسی‌های بالا صحیح باشد آنگاه خروجی الگوریتم یک می‌باشد به این معنی که اثبات توسط تاییدکننده پذیرفته شده است

الگوریتم ۴ Verifier : V_{OE} input on (x, π) where

$\pi = ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$

// Compute the challenge hash $i = 1$ **to** t **check** $ch_{i,1}, \cdots ch_{i,m} pairwise distinct$ **check** $h_{i,J_i} = G(resp_i)$ **check** $V_\Sigma(x, com_i, ch_{i,J_i}, resp_i) = 1$ all checks succeed **return** 1

۴.۲.۳ امضا براساس اثبات دانش صفر غیرتعاملی

[26]

هر طرح امضای دیجیتالی شامل سه الگوریتم تولیدکلید [27] برای ساخت کلیدخصوصی و کلیدعمومی امضاکننده، الگوریتم امضا [28] برای امضای پیام‌های دریافتی و همچنین الگوریتم تاییدامضا [29] برای بررسی اعتبار امضا، می‌باشد که این الگوریتم‌ها در طرح ما به صورت زیر تعریف می‌شوند :

- **KeyGen**(λ)

این الگوریتم یک پارامتر امنیتی λ (تا امنیت کامل طرح برآورده شود) به عنوان ورودی گرفته و یک زوج کلید (pk, sk) را به عنوان زوج کلیدعمومی وکلیدخصوصی تولید می‌کند وبرای تولید امضا و تایید آن مورداستفاده قرار می‌گیرد. این الگوریتم توسط امضاکننده اجرا می‌شود و عمومی نیست.

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

[26] Signature from Non-interactive Zero-Knowledge Proofs
[27] Key Generation Algorithm
[28] Signature Algorithm
[29] Verification Signature Algorithm

این الگوریتم، پیام m وکلید خصوصی sk را به عنوان ورودی گرفته و خروجی آن امضای σ می‌باشد که توسط امضاکننده منتشر می‌شود. این الگوریتم نیز توسط امضاکننده اجرا می‌شود و خصوصی است.

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

این الگوریتم با داشتن کلید عمومی امضاکننده $\mathbf{pk}$ تایید می کند که آیا امضای دریافتی σ متعلق به پیام $\mathbf{m}$ می‌باشد یا حیر. این الگوریتم توسط هرشخصی که خواهان بررسی امضا می‌باشد قابل اجرا خواهد بود و بنابراین یک الگوریتم عمومی است.

یک طرح امضای دیجیتال، قویا تحت حمله متن انتخاب شده غیرقابل جعل [۳۰] است اگر برای هر متخاصم $\mathcal{A}$ [۳۱] 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 پیام‌ـامضای جدید تولید کند. در طرح ارائه شده در این پایان‌نامه قصد داریم طرح خودرا با این ویژگی تشریح کنیم لذا در ادامه مرحله‌به‌مرحله به تشریح این طرح می‌پردازیم.
فرض کنیم یک تابع تولید کلید $\mathbf{KeyGen}$ ، در اختیار داریم که یک جفت کلید عمومی‌ـخصوصی $(\mathbf{sk}, \mathbf{pk})$ 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 نطر گرفته شود که می‌تواند توسط تابع $\mathbf{KeyGen}$ تولید شده باشد.
در این صورت ، امضای دیجیتال اساسا یک اثبات دانش صفر غیرتعاملی هویت خواهد بود. البته در این حالت لازم است پیام موردنظر را 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 عبارت دیگر به طور خلاصه ،

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

[30] SUF-CMA
[31] Adversary

بنابراین از طریق یک اثبات دانش صفر غیرتعاملی هویت [۳۲] (P, V) ، یک طرح امضای دیجیتال $\mathcal{DS}$ به دست می‌آید که شامل تعاریف زیر می‌باشد:

$$\mathcal{DS} = (KeyGen, Sign, Verify)$$

که دراین حالت الگوریتمِ امضا با الگوریتم اثبات‌کننده پروتکل زیگما شبیه‌سازی می‌شود:

$$Sign(sk, m) = P((pk, m), sk)$$

و الگوریتم تاییدِامضا با الگوریتم تاییدکننده پروتکل زیگما شبیه‌سازی می‌شود:

$$Verify(pk, m, \sigma = V((pk, m, \sigma)))$$

قضیه ۱۲. اگر (P, V) یک اثبات هویت $NIZK$ [۳۳] با ویژگی‌های شبیه سازی- صداقت و استخراج-آنلاین باشد آنگاه طرح امضای $\mathcal{DS}$ ذکرشده در بالا یک امضای دیجیتال SUF-CMA در مدل اوراکل تصادفی کوانتومی خواهد بود.

اثبات. content... □

[32] NIZK proof of identity

[33] Non-Interactive Zero-Knowledge

۳.۳ امضای دیجیتال همسانی مبنا

حال که در بخش قبلی توانستیم از یک طرح اثبات‌دانش‌صفرهویت‌غیرقابل‌انکار یک طرح امضای دیجیتال غیرقابل‌جعل معرفی کنیم، در این بخش خواهان پیاده‌سازی این امضا بر اساس مسئله‌ی همسانی‌ها در خم‌های بیضوی می‌باشیم.
برای معرفی طرح‌امضای‌دیجیتال برا اساس همسانی‌ها لازم است الگوریتم‌هایی که در هر طرح امضایی وجود دارد را به زبان همسانی‌ها بازتعریف کنیم. بنابریان در ادامه به معرفی الگوریتم‌های تولیدکلید،امضا و تاییدسازی می‌پردازیم. اما از آنجا که طرح ما در یک فضای‌خاصی تعریف می‌شود لازم است که ابتدا به تعریف محیطی که امضا درآن تعریف می‌شود بپردازیم پس ابتدا به عنوان تعریف محیط طرح خود، پارامترهای عمومی را تعریف می‌کنیم:

پارامترهای عمومی [۳۴]

پارامترهای عمومی ما همان پارامترهای عمومی معرفی شده در پروتکل زیگما می‌باشد:

- عدد اول به فرم $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$
- خم بیضوی سوپرسینگولار E از مرتبه‌ی $(\ell_A^{e_A}\ell_B^{e_B})^2$ در میدان $\mathbb{F}_{p^2}$
- زوج نقاط (P_B, Q_B) مولد زیرگروه تابی $E[\ell_B^{e_B}]$

تولیدکلید[۳۵]

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.

[34] Public Parameters
[35] Key Generation

امضا[36]

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

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

تاییدسازی[37]

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 تولیدکلید ، امضا و تاییدسازی را بر اساس مسئله همسانی بیان می‌کنند.

۱.۳.۳ الگوریتم تولیدکلید

همان‌طور که در الگوریتم ۳ مشاهده می‌شود

۱. ابتدا یک نقطه‌تصادفی S ازمرتبه‌ی $\ell_A^{e_A}$ انتخاب می‌کنیم.

۲. در قدم بعدی همسانی $\phi : E \to E/\langle S \rangle$ را از طریق فرمول ولو و با هسته $\langle S \rangle$ به دست می‌آوریم

۳. حال خم $E/\langle S \rangle$ و تصویر نقاط ϕP_B و ϕQ_B را به عنوان کلیدعمومی pk در نظر می‌گیریم

۴. و همچنین کلیدخصوصی را نقطه تصادفی S درنظر می‌گیریم

۵. در پایان خروجی این الگوریتم زوج کلیدعمومی و خصوصی (pk, sk) می‌باشد

[36] Signing
[37] Verification

۲.۳.۳ الگوریتم امضا

الگوریتم امضا با ورودی پیام m و کلیدخصوصی sk در طرح ما در یک مرحله انجام نمی‌شود و لازم است که برای تولید خروجی الگوریتم عملیاتی بارها انجام شود بنابراین در این الگوریتم روال زیر طی خواهد شد :

۱. همان‌طور که قبلا توضیح دادیم، الگوریتم امضای ما براساس الگوریتم اثبات‌کننده در ساخت آنره به دست می‌آید بنابراین برای امنیت کامل لازم است که الگوریتم به تعداد مشخصی تکرار شود و طبق آنچه قبلا ذکر شد لازم است که ایت تکرار به اندازه ۲λ باشد، بنابراین از یک حلقه تکرار استفاده می‌کنیم

۲. عدد R به صورت کاملا تصادفی از مرتبه‌ی $\ell_B^{e_B}$ انتخاب می‌شود

۳. درادامه همسانی ψ از طریق فرمول ولو و با زیرگروه $\langle R\rangle$ به صورت $\psi : E \to E/\langle R\rangle$ محاسبه می‌شود

۴. سپس دو همسانی ϕ' و ψ' مستقلا توسط زیرگروه‌های مشخص زیر توسط فرمول ولو محاسبه می‌شود :

$$\phi' : E/\langle R\rangle \to E/\langle R, S\rangle$$

$$\psi' : E/\langle S\rangle \to E/\langle R, S\rangle$$

۵. خم‌های به دست آمده از دوهمسانی ϕ' و ψ' را به صورت زیر نامگذاری می‌کنیم :

$$(E_۱, E_۲) \leftarrow (E/\langle R\rangle, E/\langle R, S\rangle)$$

۶. همان‌طور که در ساخت آنره(برگرفته از پروتکل زیگما) مشاهده کردیم لازم است اثبات‌کننده (در اینجا امضاکننده) تعهدی را برای اثبات ارائه کند، بنابراین تعهد مطرح شده در الگوریتم امضا به صورت زیر خواهد بود :

$$com_i \leftarrow (E_۱, E_۲)$$

۷. برای به چالش کشیدن امضاکننده براساس تعهدی که ارائه کرده است لازم است چالشی توسط ساخت آنره تولید شود و چون دامنه‌ی چالش‌ها محدود به دو انتخاب است بنابراین چالش ما به صورت تصادفی از بین دو مقدار صفرو یک انتخاب می‌شود :

$$ch_{i,\cdot} \leftarrow_R \{۰, ۱\}$$

۸. در این مرحله بدون درنظر گرفتن چالش انتخابی مرحله قبل، پاسخ‌های را براساس زیر تولید می‌کنیم. دلیل این امر این است که پاسخ‌های امضاکننده در هرمرحله ثابت ولی نسبت به چالش ارائه شده جایگشتی در ارائه آنها در مرحله‌ی بعدی صورت می‌گیرد. با این اوصاف پاسخ‌های الگوریتم به صورت زیر تعیین می‌شوند:

$$(resp_{i,\cdot}, resp_{i,۱}) \leftarrow ((R, \phi(R)), \psi(S))$$

۹. در این مرحله مشخص می‌شود براساس چالش دریافتی پاسخ چه باشد. اگر بیت چالش صفر باشد پاسخ همان ترتیب بالا خواهد بود وگرنه جای پاسخ‌ها با یکدیگر عوض شده و سپس ارسال می‌شود

۱۰. در این مرحله به منظور اطمینان از ثبت درست مراحل بالا طبق الگوریتم امضا لازم است که از پاسخ‌ها با کمک تابع G هش گرفته شود بنابراین :

$$h_{i,j} \leftarrow G(resp_{i,j})$$

۱۱. همان‌طور که در الگوریتم اثبات‌کننده مطرح شد لازم است که چالش توسط خود الگوریتم به صورت تصادفی و بدون هرگونه دخالت دانش قبلی تولید شود بنابراین ۲λ چالش ممکن توسط کد زیر تولید می‌شود:

$$J_۱ \parallel \cdots \parallel J_{۲\lambda} \leftarrow H(pk, m, (com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$$

لازم به یادآوری است که تابع H تابع هش می‌باشد

۱۲. خروجی الگوریتم، امضای امضاکننده (اثبات اثبات‌کننده) می‌باشد که به صورت زیر تولید می‌شود:

$$\sigma \leftarrow ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$$

و برای تاکید بیشتر لازم به یادآوری است که پاسخ‌های $resp$ تولید شده در مراحل قبلی کاملا به صورت تصادفی و بدون هیچ ترتیب خاصی و وابسته به چالش‌های J_i در اثبات جاسازی می‌شوند

۳.۳.۳ الگوریتم تایید امضا

با داشتن کلیدعمومی pk ،پیام m و امضای دریافتی σ از طرف امضاکننده، هر شخصی می‌تواند بدون تعامل با امضاکننده به تایید امضا بپردازد. تاییدکننده باید طبق الگوریتم تایید مراحل زیر را انجام دهد :

۱. از آنجا که طرح امضای ما براساس ساخت آنره می‌باشد ودر این پروتکل تعاملی بین اثبات‌کننده(امضاکننده) و تاییدکننده برقرار نیست باید راهی وجود داشته باشد تا تاییدکننده نیز به چالش‌هایی که اثبات‌کننده نسبت به آن‌ها پاسخ داده است اشراف داشته باشد بنابراین همان‌طور که در الگوریتم امضا مشاهده شد، چالش‌ها از قسمت‌هایی به‌دست می‌آید که آن قسمت‌ها توسط امضاکننده منتشر و عمومی می‌شود بنابراین تاییدکننده نیز می‌تواند خود مستقلا به تولید چالش‌ها بپردازد. این عمل نشان می‌دهد که چالش‌ها واقعا به صورت تصادفی خلق شده‌اند. بنابراین در اولین قدم اجرای الگوریتم، چالش‌ها مستقلا بازتولید می‌شوند تا در قدم‌های بعدی الگوریتم مورد استفاده قرار گیرد:

$$J_۱ \parallel \cdots \parallel J_{۲\lambda} \leftarrow H(pk, m, (com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$$

۲. از آنجا که برای تولید امضا π ، به ۲λ راند نیاز بود بنابراین برای تایید آن نیز به ۲λ مرحله نیاز است که برای این امر از حلقه تکرار استفاده می‌کنیم

۳. چون تابع هش G عمومی است پس می‌توان بررسی کرد که آیا هش پاسخ‌ها $h_{i,j}$ به عنوان بخشی از اثبات π از پاسخ‌های $resp_{i,j}$ الگوریتم امضا به‌دست آمده است یا خیر. درنتیجه این بررسی در الگوریتم تایید مستقلا انجام می‌شود:

$$\textbf{check } h_{i,J_i} = G(resp_{i,J_i})$$

۴. از آنجا که برای تولیدامضا ۲λ مرحله نیاز بود برای تاییدامضا نیز باید این تعداد مرحله انجام شود. همچنین می‌دانیم در طرح خود چالش یا صفر است یا یک، درنتیجه اگر چالش صفر باشد عملیات خاصی انجام می‌شود و اگر چالش یک باشد عملیاتی متناظر با آن انجام خواهد شد.

۵. اگر چالش صفر انتخاب شده باشد آنگاه عملیات زیر به ترتیب انجام می‌شود:

- پاسخ $resp_{i,J_i}$ به عنوان مقدار $(R, \phi(R))$ انتخاب می‌شود ؟؟؟؟؟؟؟
- در قدم بعدی بررسی می‌شود که آیا $(R, \phi(R))$ از مرتبه‌ی $\ell_B^{e_B}$ می‌باشد یا خیر
- با داشتن دو خم E و $E_۱$ و همچنین فرمول ولو بررسی می‌شود که آیا R تولیدکننده‌ی هسته‌ی همسانی $\psi : E \to E_۱$ می‌باشد یا خیر
- درآخرین قسمت این مرحله بررسی می‌شود که آیا $\phi(R)$ تولیدکننده هسته‌ی همسانی $\psi' : E/\langle S\rangle \to E_۲$ می‌باشد یا خیر. این بررسی نیز با داشتن دو خم $E/\langle S\rangle$ و $E_۲$ و همچنین فرمول ولو و الگوریتم ؟؟؟ قابل انجام می‌باشد

۶. اگر چالش حاوی مقدار یک باشد آنگاه مراحل زیر انجام می‌شود:

- پاسخ $resp_{i,J_i}$ به عنوان $\psi(S)$ درنظرگرفته می‌شود؟؟؟
- بررسی می‌شود که آیا $\psi(S)$ از مرتبه‌ی $\ell_A^{e_A}$ می‌باشد یاخیر
- همچنینی بررسی می‌شود که آیا $\psi(S)$
  تولیدکننده‌ی هسته‌ی همسانی $\phi' : E_۱ \to E_۲$ می‌باشد یا خیر

۷. اگر در تمام بررسی‌های بالا پاسخ بلی باشد آنگاه الگوریتم یک رو به عنوان خروجی تولید می‌کند به این معنی که امضا پذیرفته شده است.

۴.۳ جنبه‌های الگوریتمی

۱.۴.۳ تولید پارامترها

اعداد اولی که در طرح خود استفاده می‌کنیم به فرم $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm ۱$ می باشند. دلیل این امر آن است که برای تامین امنیت طرح خود لازم است تا ابتدا اعداد اول ثابت ℓ_A و ℓ_B را به‌صورت مجزا از عدد اول p ، با ویژگی $\ell_A^{e_A} \approx \ell_B^{e_B}$ (به این معنی که از نظر بیتی هم اندازه هستند) انتخاب کنیم. مطمئنا با ضرب مقادیر $\ell_A^{e_A}$ و $\ell_B^{e_B}$ عدد اولی خاصل نخواهد شد، در بهترین حالت با اضافه یا کم کردن مقدار یک به این حاصلضرب می‌توان به یک عدد اول رسید و اگر نتیجه حاصل نشد می‌توانیم مقدار یک را با مضربی از $\ell_A^{e_A}\ell_B^{e_B}$ جمع یا تفریق کنیم. این مضرب در فرم بالا همان مقدار متغیر f می‌باشد. بنابراین عدد اول استفاده شده در طرح ما به‌یکی از فرم‌های زیر خواهد بود:

$$p = \ell_A^{e_A}\ell_B^{e_B} \cdot f + ۱ \quad \text{یا} \quad p = \ell_A^{e_A}\ell_B^{e_B} \cdot f - ۱$$

بروکر در [۳] نشان داده است برای هر عدد اول $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm ۱$ ، می‌توان به راحتی یک خم بیضوی سوپرسینگولار E روی میدان $\mathbb{F}_{p^۲}$ با مرتبه $(\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲ = (p \mp ۱)^۲$ به دست آورد. دلیل این امر هم آن است که اگر E یک خم بیضوی روی میدان $\mathbb{F}_p$ باشد، آنگاه $\mathbb{F}_p$ ـ نقاط روی خم به شکل زیر می‌باشد:

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^۲ = x^۳ + Ax + B\} \cup \{\infty\}$$

که نتیجه می‌شود که :

$$E(\mathbb{F}_p) \subseteq (\mathbb{F}_p \times \mathbb{F}_p) \cup \{\infty\}$$

و چون مجموعه‌ی سمت راست متناهی(از مرتبه‌ی $p^۲ + ۱$) است لذا مجموعه‌ی سمت چپ یعنی $E(\mathbb{F}_p)$ نیز متناهی است. بنابراین :

$$\#E(\mathbb{F}_p) \leq p^۲ + ۱$$

حال اگر یک تغییر کوچک به معادله‌ی بالا اعمال کنیم، خواهیم داشت:

$$\#E(\mathbb{F}_p) - ۱ \leq p^۲$$

حال، اگر

$$p = \ell_A^{e_A}\ell_B^{e_B}\cdot f + ۱ \quad \text{یا} \quad p = \ell_A^{e_A}\ell_B^{e_B}\cdot f - ۱$$

آنگاه

$$p^۲ = (\ell_A^{e_A}\ell_B^{e_B}\cdot f)^۲ + ۲(\ell_A^{e_A}\ell_B^{e_B}\cdot f) + ۱ \quad \text{یا} \quad p^۲ = (\ell_A^{e_A}\ell_B^{e_B}\cdot f)^۲ - ۲(\ell_A^{e_A}\ell_B^{e_B}\cdot f) - ۱$$

که به‌طور خلاصه خواهیم داشت:

$$p^۲ \mp ۱ = (\ell_A^{e_A}\ell_B^{e_B}\cdot f)^۲ \pm ۲\ell_A^{e_A}\ell_B^{e_B}\cdot f$$

و از آنجا که $(\ell_A^{e_A}\ell_B^{e_B}\cdot f) = p$ ، بنابراین فرم بالا به‌صورت زیر خلاصه می‌شود:

$$p^۲ \mp ۲p \mp ۱ = (\ell_A^{e_A}\ell_B^{e_B}\cdot f)^۲$$

که اگر دوباره آن را خلاصه کنیم، خواهیم داشت:

$$(p \mp ۱)^۲ = (\ell_A^{e_A}\ell_B^{e_B}\cdot f)^۲$$

که همان مرتبه‌ی گفته شده با روش بروکر می‌باشد.

۲.۴.۳ پیدا کردن مولدهای زیرگروه تابی

برای انتخاب نقاط مولد زیرگروه $E.[\ell_A^{e_A}]$ ، می‌توان یک نقطه تصادفی $P \in_R E.(\mathbb{F}_{p^۲})$ انتخاب و آن را در $(\ell_B^{e_B}\cdot f)^۲$ ضرب کرد تا نقطه P' با مرتبه توانی از ℓ_A حاصل شود. از آنجا که عامل‌های عدد اول p، $\ell_A^{e_A}$ و $\ell_B^{e_B}$ می‌باشند، احتمالا P' از مرتبه $\ell_A^{e_A}$ خواهد بود؛ 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 P می‌شویم. برای به دست آوردن نقطه دوم مولد یعنی 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 و محاسبه $e(P_A, Q_A)$ در میدان $E[\ell_A]$ ، بررسی کرد که آیا نتیجه از مرتبه ℓ_A می‌باشد یا خیر ؛ برای اطمینان از اینکه نقطه‌ي Q_A متفاوت از نقطه‌ی P_A می‌باشد می‌توانیم از گزاره زیر استفاده کنیم:

گزاره ۴. اگر $P_A, Q_A \in E[\ell_A]$ و ℓ_A عددی اول باشد آنگاه

$$e_n(P_A, Q_A) = ۱ \text{ اگر و تنها اگر } Q_A = kP_A$$

اثبات ۳.

۱. اگر فرض شود به ازای یک k ، $Q_A = kP_A$ در این‌صورت:

$$e_n(P_A, Q_A) = e_n(P_A, kP_A) = e_n(P_A, P_A)^k = ۱^k = ۱$$

۲. همچنین اگر $e_n(P_A, Q_A) = ۱$ ، در اینصورت $R \in E[\ell_A]$ را چنان اختیار می‌کنیم که $E[\ell_A] = \langle P, R\rangle$ ، بنابراین $\partial = e_n(P_A, R)$ یک ریشه n ـ‌ام اولیه واحد است. پس:

$$Q_A \in E[n] = \langle P_A, R\rangle \longrightarrow \exists ۰ \leqslant k, l \leqslant \ell_A - ۱ \ , \quad Q_A = kP_A + \ell R$$

اکنون :

$$۱ = e_n(P_A, Q_A) = e_n(P, kP + \ell R) = e_n(P, P)^k r_n(P, R)^\ell = \partial^\ell$$

بنابراین $\partial^\ell = ۱$ و درنتیجه $\ell = ۰$ ، پس $Q_A = kP_A$.

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، قابل تبدیل به یکدیگر می‌باشند. چنانچه در [۱۷] اشاره شده است این محاسبه به راحتی در زیرگروه $E[\ell_A]$ قابل انجام می‌باشد.

۳.۴.۳ تبادل کلید

یک از مهمترین ارکان هر سیستم رمزنگاری مربوط به تبادل کلید می‌باشد. به‌عبارت دیگر زمانی‌که یک سیستم رمزنگاری معرفی می‌شود در ابتدا بررسی می‌شود که آیا می‌توان پروتکل تبادل کلید را

پیاده‌سازی کرد یا خیر. بعد از معرفی سیستم رمزنگاری همسانی‌ـ مبنا، آقای جائو در [] به معرفی تبادل کلید در همسانی‌ها پرداخت. از آنجا که این موضوع در ادامه بحث مورد استفاده قرار می‌گیرد، این پروتکل رو تشریح می‌کنیم.

پروتکل تبادل کلید نوعی از پروتکل دیفای‌ـ هلمن است که طبق شکل ۱ صورت می‌پذیرد. ایده‌ی کلی این پروتکل آن است که شخصی مانند آرش، همسانی ϕ و شخص دیگری همچون بابک همسانی ψ را به عنوان کلیدخصوصی انتخاب می‌کنند و با یک خم سوپرسینگولار عمومی همچون $E.$ بااستفاده از پروتکل قدم‌زدن روی گراف به تولید یک خم خصوصی همچون E_A و E_B می‌پردازند که بیانگر کلیدخصوصی آنها همچون دیگر پروتکل‌های رمزنگاری می‌باشد. در ادامه با انجام محاسباتی که در ادامه تشریح خواهد شد به‌طور مجزا به تولید یک خم سوپرسینگولار همچون E_{AB} خواهند پرداخت که به عنوان کلیدعمومی‌شان درنظر گرفته می‌شود.

۱. در ابتدا، یک خم سوپرسینگولار دلخواه در میدان $\mathbb{F}_{p^2}$ همچون $E.$ و دو جفت نقاط $\{P_A, Q_A\}$ و $\{P_B, P_B\}$ که مولد زیرگروه‌های تابی $E_[\ell_A^{e_A}]$ و $E_[\ell_B^{e_B}]$ می‌باشند را به‌عنوان پارامترهای عمومی پروتکل درنظر می‌گیریم.

۲. (آ) درادامه آرش دو عنصر تصادفی همچون $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (که هردو همزمان به ℓ_A بخش‌پذیر نیستند) را انتخاب می‌کند و همسانی $\phi_A : E. \to E_A$ را که هسته‌ی آن $K_A := \langle [m_A]P_A + [n_A]Q_A \rangle$ می‌باشد را محاسبه می‌کند.درادامه آلیس $\{\phi_A(P_B), \phi_A(Q_B)\} \subset E_A$ را نیز محاسبه می‌کند.

(ب) به‌طورمشابه، بابک نیز دو عنصر تصادفی همچون $m_B, n_B \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ را انتخاب و همسانی $\phi_B : E. \to E_B$ با هسته‌ی $K_B := \langle [m_B]P_B + [n_B]Q_B$ را به‌همراه نقاط $\{\phi_B(P_A), \phi_B(Q_A)\}$ محاسبه می‌کند.

۳. (آ) در ادامه آرش پس از دریافت E_B و $\phi_B(P_A), \phi_B(Q_A) \in E_B$ از جانب بابک، به محاسبه‌ی همسانی $\phi'_A : E_B \to E_{AB}$ با هسته‌ی $\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$ می‌پردازد.

(ب) بابک نیز با دریافت E_A به محاسبه‌ی همسانی $\phi'_B : E_A \to E_{AB}$ با هسته‌ی $\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$ می‌پردازد.

۴. آرش و بابک برای دسترسی به یک کلیدخصوصی مشترک می‌توانند j ـ پایای خم زیر را محاسبه کنند:

$$E_{AB} = \phi'_B(\phi_A(E_\cdot)) = \phi'_A(\phi_B(E_\cdot)) = E_\cdot/\langle [m_A]P_A{+}[n_A]Q_A, [m_B]P_B{+}[n_B]Q_B \rangle$$

۴.۴.۳ ساده‌سازی نقاط تاب‌دار

الگوریتم ۵ نردبان سه‌ـ نقطه‌ای برای محاسبه‌ی $P + [t]Q$

Input : t,P,Q

Set $A = 0, B = Q, C = P$ Compute $Q - P$; $i = |t|$ **to** 1 Let t_i be the i-th bit of t; $t_i = 0$ $A = 2A, B = dadd(A, B, Q), C = dadd(A, C, P)$; $A = dadd(A, B, Q), B = 2B, C = dadd(B, C, Q - P)$;

Output : $C = P + [t]Q$

در طرح خود از نقاط تصادفی متنوعی با مرتبه‌ی خاص همچون $\ell_A^{e_A}$ و $\ell_B^{e_B}$ به‌مراتب برای ساخت یک زیرگروه استفاده می‌کنیم. از آنجا که نقاط موردنظر ما از زیرگروه‌های تابی $E[\ell_A^{e_A}]$ و $E[\ell_B^{e_B}]$ با دو مولد P و Q ساخته می‌شوند لذا این نقاط به فرم $\langle [m]P + [n]Q \rangle$ خواهند بود. البته m و n همزمان نباید با ℓ^e موردنظر بخش‌پذیر باشند. بنابراین نتیجه می‌گیریم که که اگر روشی ارائه دهیم تا ساخت این زیرگروه‌ها با محاسبات کمتری انجام پذیرند، به‌طور کلی طرح ما بهینه‌تر خواهد شد.

می‌توان فرض کرد که هر m ، دارای عنصر وارون در پیمانه‌ی مرتبه‌ی گروه می‌باشد(این فرض هیچ خدشه‌ای به زیرگروه وارد نمی‌کند). در این حالت $R' = P + [m^{-1}n]Q$ زیرگروهی همانند دیگر مولدها خواهد بود. محاسبه R' با روش استاندارد **دوبرابرـ وـ جمع** [۳۸] نیاز به نصف عملیات محاسبات $[m]P + [n]Q$ معمولی را دارا می‌باشد(برای روش های بهتر محاسبه عملیات معمولی به مراجعه [۱۶،۷،۱] شود). با این حال، محاسبه $P + [m^{-1}n]Q$ با روش دوبرابرـ وـ جمع، یک

[38] double-and-add

حفره امنیتی (اشکال بزرگ) را داراست : در برابر حملات ***آنالیز قدرت ساده*** *یا* ***SPA*** *[۱۳] آسیب پذیر می‌باشد. برای جلوگیری از این حمله می‌توان از روش* ***نردبان مونت‌گومری*** *[39] [۱۴] برای محاسبه $[m^{-1}n]Q$ استفاده کرد و سپس P را به آن اضافه کرد، اما این روش به طور قابل توجهی کند می‌باشد. به‌منظور رفع دو مشکل کندی و حمله‌ی SPA ، الگوریتم* **??** *را ارائه می‌دهیم که بیانگر یک روش بسیار موثرتری می‌باشد و مستقیما $P + [m^{-1}n]Q$ را محاسبه می‌کند. 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 $dadd(A,B,C)$ مورد استفاده شده در الگوریتم نیز معرف جمع تفاضلی [40] [۱۴] می‌باشد.*
پیاده سازی جمع تفاضلی در خم‌های مونت‌گومری به کارآمدی روش دوبرابر- و- جمع ساده روی خم های دوقولوی ادوارد [41] ۴.۴.۶ می‌باشد، بنابراین درادامه به معرفی خم‌های مونت‌گومری خواهیم پرداخت.

۵.۴.۳ محاسبه همسانی‌های با درجه هموار

[42]

محاسبه‌ی همسانی یکی از پرهزینه‌ترین محاسبات در سیستم‌های همسانی- مبنا می‌باشند. از آنجا که در طرح خود نیز به‌مراتب به محاسبه‌ی همسانی‌ها با درجه‌ی معینی می‌پردازیم بنابراین لازم است تا روشی سریع برای این امر معرفی کنیم. البته این روش می‌تواند در تمامی طرح‌ها مورداستفاده قرار گیرد، به‌عنوان مثال می‌توان در محاسبات همسانی در پروتکل تبادل کلیدی که در بخش قبل بین آرش و بابک انجام می‌پذیرد اشاره کرد.
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 ۱.۳: ساختمان محاسبات ساخت $\phi = \phi_۵ \circ \cdots \circ \phi$.

[39]Montgomery ladder
[40]differential addition
[41]twisted Edwards curves
[42]Computing smooth degree isogenies

اگر درجه‌ی نگاشت ϕ هموار باشد، می‌توان آن را به زنجیره‌ای از ℓ ـ همسانی ها تجزیه کرد. اگر $E_\circ = E$ و $R_\circ = R$ در نظر بگیریم ، آنگاه برای هر $\circ \leq i < e$ 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 $\phi = \phi_{e-1} \circ \cdots \circ \phi_\circ$ می‌باشد.

توجه به این نکته لازم است که از آنجا که زیرگروه ℓ ـ تابی $\langle R_i \rangle$ خم E_i مشخص می‌باشند ، خم بیضوی E_{i+1} و همسانی ϕ_i می‌توانند توسط فرمول ولو [43] [۲۰] به راحتی محاسبه شوند. در [۱۲] ، دو پیشنهاد برای داشتنی پیچیدگی درجه دو برای e بیان شده است ؟؟؟؟؟.
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 $R_\circ$ را در اختیار داریم. به بیان دیگر هدف ما در این الگوریتم محاسبه تمام نقاط روی خط پایانی توسط نقطه آغازین $R_\circ$ می‌باشد (ورودی این الگوریتم نقطه $R_\circ$ و خروجی این الگوریتم نقاط $[\ell^5]R_\circ$ ، $[\ell^4]R_1$ ، $[\ell^3]R_2$ ، $[\ell^2]R_3$ ، $[\ell^1]R_4$ و R_5 می‌باشد). در واقع با دانستن نقطه $[\ell^{e-i-1}]R_i$ ، 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+1} را محاسبه کنیم.
برای فهم بیشتر این الگوریتم مراحل ذکر شده در مثال $e = ۶$ را مرحله به مرحله نمایش می‌دهیم :

$$i = \circ \Rightarrow \quad E_1 = E_\circ/\langle \ell^4 R_\circ \rangle, \quad \phi_\circ : E_\circ \to E_1, \quad R_1 = \phi_\circ(R_\circ)$$

$$i = 1 \Rightarrow \quad E_2 = E_1/\langle \ell^3 R_1 \rangle, \quad \phi_1 : E_1 \to E_2, \quad R_2 = \phi_1(R_1) = \phi_1(\phi_\circ(R_\circ))$$

[43] Velu's formulas

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

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

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

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

۶.۴.۳ انتخاب مدل

بعد از ارائه‌ی یک طرح لازم است تا آن طرح بهینه‌سازی شود. به‌عبارت‌دیگر زمانی که یک طرح ارائه و تایید شد برای پیاده‌سازی آن باید سرعت اجرای طرح را تا جای ممکن افزایش داد. به‌طورمثال می‌توانیم تعداد عملیاتی که در یک الگوریتم انجام می‌شود را به حداقل رساند یا عملیات‌های سنگین تر را به عملیات های با بار پیچیدگی کمتر از لحاظ زمان اجرا جایگزین کرد. معمولا در زمان محاسبه پیچیدگی یک الگوریتم از عملیات تفریق، جمع و مقایسه صرف‌نظر می‌کنند. دلیل این امر آن است که این عملبات ، پیچیدگی محاسبات بالا یی ندارند و به‌عنوان عمل‌های پایه در هر الگوریتم فرض می شوند. از جمله اعمالی که در سرعت اجرای یک الگوریتم می‌تواند تاثیرگذار باشد می‌توان به عملیات وارون، ضرب و توان اشاره کرد. برای بهینه‌سازی یک الگوریتم تلاش می‌شود که این عملیات ها به حداقل برسند. در ادامه از علائم I ، M و S به‌منظور عملیات وارون، ضرب و توان استفاده می‌کنیم. از جمله عملیات سنگین و زمان‌بری که در طرح امضای خود می‌توانیم نام ببریم عملیات ***دوبرابرکردن*** *،* ***جمع*** *،* ***محاسبه*** *و* ***ارزیابی*** *همسانی‌ها خواهد بود. چنانچه ذکر شد پس از ارائه‌ی طرح خود مصمم هستیم تا سرعت اجرای الگوریتم ها در طرح خود را افزایش دهیم. یکی از راه حل‌های ممکن این است که به‌جای استفاده از یک خم بیضوی معمولی با معادله‌ی وایرشتراس، از خم بیضوی مونت‌گومری استفاده کنیم. مزیت استفاده از خم مونت‌گومری را پس از تعریف آن ذکر خواهیم کرد.*

تعریف ۳.۴.۱ . یک خم مونت‌گومری در میدان $\mathbb{F}_q$ یک خم بیضوی به فرم زیر می‌باشد:

$$M_{B,A} : By^۲ = x^۳ + Ax^۲ + x$$

در این خم برای نقاط $P = (x_P, y_P)$ و $Q = (x_Q, y_Q)$ ، نقطه‌ی $R = P + Q = (x_R, y_R)$ به‌صورت زیر محاسبه می‌شود:

$$x_R = B\lambda^۲ - (x_P + x_Q) - A$$
$$y_R = \lambda(x_P - x_Q) - y_P$$

که در آن

$$\lambda = \begin{cases} \frac{y_Q - y)P}{x_Q - x_P} & P \neq Q, -Q \text{ اگر } \\ \frac{۳x_P{}^۲ + ۲Ax_P + ۱}{۲By_P} & P = Q \text{ اگر}. \end{cases}$$

j- پایای این فرم از خم بیضوی برابر با مقدار

$$j(M_{B,A}) = \frac{۲۵۶(A^۲ - ۳)^۳}{A^۲ - ۴}$$

است که تنها به پارامتر A بستگی دارد.
همچنین خم تصویری مونت‌گومری در میدان $\mathbb{F}_q$ یک خم بیضوی به فرم

$$M_{B,A} : BY^۲Z = X^۳ + AX^۲Z + XZ^۲$$

است که در آن $A, B \in \mathbb{F}_q$ ، $B \neq ۰$ و $A^۲ \neq ۴$. مجموعه نقاطی که روی این خم هستند همراه با نقطه‌ی همانی $(۰ : ۱ : ۰) = \infty$ گروه نقاط $M_{B,A}$ را تشکیل می‌دهند.

برای آن‌که بتوانیم خم مونت‌گومری را جایگزین خم وایرشتراس کنیم لازم است تا یک یکیریختی بین آنها پیدا کنیم. چناچه در [۵] ذکر شده است اگر در میدان $\mathbb{F}_q$ ، q توانی از ۳ نباشد، خم مونت‌گومری $M_{B,A}$ با نگاشت گویای

$$\phi : M_{B,A} \longrightarrow E$$
$$(x, y) \mapsto (X, Y) = (B(x + A/۳), B^۲y)$$

با خم وایرشتراس کوتاه

$$E : Y^۲ = X^۳ + (B^۲ \frac{۱ - A^۲}{۳})X + \frac{B^۳ A}{۳(۲A^۲/۹ - ۱)}$$

یکریخت است.

همچنین وارون نگاشت ϕ برابر است با :

$$\phi^{-۱} : E \longrightarrow M_{B,A}$$
$$(X, Y) \mapsto (x, y) = (X/B - A/۳, Y/B^۲)$$

اگر فرض کنیم $E : Y^۲ = X^۳ + aX + b$ یک خم بیضوی باشد در این‌صورت E با یک خم مونت‌گومری یکیریخت است اگر و تنها اگر $\alpha \in \mathbb{F}_q$ وجود داشته باشد که $\alpha^۳ + a\alpha + b = ۰$ و $\sqrt{۳\alpha^۲ + a} \in \mathbb{F}_q$. حال اگر $\beta = \sqrt{۳\alpha^۲ + a}$ آنگاه نگاشت گویای زیر یک یکریختی بین این دو خم خواهد بود:

$$\phi : E \longrightarrow M_{۳\alpha/\beta, ۱/\beta}$$
$$(X, Y) \mapsto (x, y) = ((X - \alpha)/\beta, Y/\beta)$$

اعمال جمع و ضرب اسکالر روی نقاط خم بیضوی به فرم مونت‌گومری بااستفاده از یک نگاشت x انجام می‌شود. این نگاشت روی نقطه‌ی $P = (x : y : z) \in M_{B,A}$ به‌صورت زیر تعریف می‌شود:

$$x : M_{B,A} \longrightarrow \mathbb{P}^۱$$
$$P \mapsto \begin{cases} (x : z) & P \neq \infty \text{ اگر } \\ (۱ : ۰) & P = \infty \text{ اگر } \end{cases}$$

در [۱۵] نشان داده شده است که رابطه‌های

$$x_{P+Q}(x_P - x_Q)^۲ x_P x_Q = B(x_P y_Q - x_Q y_Q)^۲$$

$$۴x_{۲P} x_P(x_P^۲ + Ax_P + ۱) = (x_P^۲ - ۱)^۲$$

و

$$xP-Q(x_P-x_Q)^۲x_Px_Q=B(x_Py_Q+x_Qy_Q)^۲$$

روی نقاط $P,Q\in M_{B,A}$ برقرار است. از این معادلات می‌توان نتیجه گرفت

$$x_{P+Q}x_{P-Q}=\frac{(x_Px_Q-۱)^۲}{(x_P-x_Q)^۲}$$

$$x_{۲P}=\frac{(x_P^۲-۱)^۲}{۴x_P(x_P^۲+Ax_P+۱)}$$

و لذا

$$x_{P+Q}=\frac{(x_Px_Q-۱)^۲}{(x_P-x_Q)^۲xP-Q}$$

$$x_{۲P}=\frac{(x_P^۲-۱)^۲}{۴x_P(x_P^۲+Ax_P+۱)}$$

ار این معادلات می‌توان مولفه‌ی x نقاط $P+Q$ و $۲P$ را با مولفه‌های x نقاط $P,Q,P-Q\in M_{B,A}$ محاسبه کرد.

۷.۴.۳ اندازه پارامتر

[۴۴] همان‌طور که قبلا بررسی شد، اعداد اولی که برای ساخت همسانی‌ها از آن استفاده می‌کنیم به فرم $p=\ell_A^{e_A}\ell_B^{e_B}\cdot f\pm ۱$ می‌باشد با این ویژگی که $\ell_A^{e_A}\approx\ell_B^{e_B}$. همچنین یادآوری می‌کنیم که برای داشتن λ بیت امنیت پساکوانتومی لازم است تا اعداد اول مورداستفاده در طرح امضا به طول ۶λ بیت باشند(**??**)، درنتیجه اندازه عامل‌های اعداد اول به‌صورت $\ell_A^{e_A}\approx\ell_B^{e_B}\approx ۲^{۳\lambda}$ خواهد بود. ازآنجا که در طرح امضای خود، خم‌های سوپرسینگولار را در میدان $\mathbb{F}_{p^۲}$ تعریف می‌کنیم درنتیجه اندازه عناصر میدان، ۱۲λ بیت طول خواهند داشت.

خم‌هایی که در طرح امضای خود استفاده می‌کنیم به فرم خم‌های مونت‌گومری $By^۲=x^۳+Ax^۲+x$ می‌باشد. از مزیت‌های خم مونت‌گومری می‌توان به محاسبات همسانی‌ها اشاره کرد که فقط به ضریب A نیاز می‌باشد. از طرف دیگر، یک نقطه روی خط کامر [۴۵] نیز می‌تواند بوسیله‌ی ضریب X اش نشان داده شود. با این اوصاف،برای نمایش هر عنصر میدان در

[44] Parameter Sizes

[45] Kummer line

خم به فرم مونت‌گومری و خط کامر، نیاز به ۱۲λ بیت می‌باشد.

فشرده‌سازی. آذردرخش و همکارانش در [۲] نشان داده‌اند که نقاط تابی(که مولد زیرگروه‌های تابی می‌باشند) می‌توانند بوسیله‌ی ضریب‌هایشان فشرده شوند. از آن‌جا که پیاده سازی این روش زیادی کند می‌باشد اخیرا کاستللو و همکارانش در [۴] یک الگوریتم جدیدتری ارائه داده‌اند که نسبت به روش آذردرخش هم سریع‌تر است و هم اندازه کلیدعمومی آن به نسبت طرح قبلی کوچکتر می‌باشد. در مورد اجرای این الگوریتم می‌توان گفت تقریبا برابر با اجرای یک مرحله از پروتکل اثبات‌دانش‌صفر می‌باشد.
همچنین در بخش قبلی اشاره شد که می‌توانیم زیرگروه تولید شده توسط یک نقطه تابی را تنها با یک مولد و ضریبش نشان دهیم. از آنجا که نقاط مولد زیرگروه ها در طرح ما عمومی می‌باشند درنتیجه می‌توانیم برای نمایش یک زیرگروه تابی تنها از یک ضریب برای اختصار استفاده کنیم، به عبارت دیگر:

$$R = mP_A+nQ_A \xrightarrow{m^{-1}} m^{-1}R = m^{-1}mP_A+m^{-1}nQ_A = P_A+m^{-1}nQ_A = P_A+kQ_A$$

با توجه به مطالب بالا برای نمایش R فقط لازم است که k را در اختیار داشته باشیم چون P_A و Q_A عمومی هستند.
در محاسبه ترکیبات خطی، برای فشرده‌سازی دو مولد یک گروه تابی، نیاز به سه ضریب می‌باشد که برای هر ضریب تقریبا ۳λ بیت نیاز می‌باشد.

۱.۷.۴.۳ فشرده‌سازی امضا

به دو روش می‌توانیم، طرح امضای خود را فشرده کنیم:

- فشرده‌سازی کلیدعمومی
- فشرده سازی پاسخ $\psi(S)$ زمانیکه در مرحله‌ای از الگوریتم امضا، $ch = ۱$ انتخاب شده باشد

لازم به ذکر است، کلیدخصوصی S و پاسخ $ch = ۰$ یعنی $(R, \phi(R))$ به دلیل آنکه با ضریب ۳λ

بیتی قابل نمایش‌اند، لذا نیازی به فشرده‌سازی ندارند.

- **کلیدعمومی.** از آنجا که از خم مونتگومری استفاده می‌کنیم بنابراین کلیدعمومی ما به فرم $pk = (a, x(P_B), x(Q_B), x(P_B - Q_B))$ می‌باشد که a بیانگر ضریب A در خم عمومی $E/\langle S\rangle$ می‌باشد. این چهار عنصر میدان به $(۱۲\lambda \times ۴ =)۴۸\lambda$ بیت برای نمایش نیاز دارند.
کلیدعمومی را می‌توانیم با فشرده‌سازی نقاط تابی $(\phi(P_B), \phi(Q_B))$ ، که نیاز به سه ضریب ۳λ بیتی دارند را فشرده کنیم. به‌دلیل آنکه مختصات نقاط P_B و Q_B از طریق ضرایب فشرده‌شان قابل تولید می‌باشد بنابراین نیازی به ضریب X نقطه‌ی $\phi(P_B - Q_B)$ نمی‌باشد. بنابراین به‌طورکلی در کلیدعمومی برای نمایش خم، ۱۲λ بیت و برای مولدها نیز ۹λ بیت نیاز داریم که جمعا ۲۱λ بیت می‌شود.

- **کلیدخصوصی.** کلیدخصوصی S می‌تواند تنها با یک ضریب n که نیاز به ۳λ بیت می‌باشد ذخیره شود. دلیل این امر هم این است که کلیدخصوصی S از مرتبه‌ی $\ell_A^{e_A}$ می‌باشد و $S = P_A + [n]Q_A$.

- **امضا.** برای هر مرحله‌ی i ام از پروتکل اثبات‌دانش‌صفر، امضا شامل چندتایی $(com_i, ch_{i,j}, h_{i,j}, resp_{i,J_i})$ می‌باشد. بنابراین:

  - هر تعهد شامل دو خم $(E_۱, E_۲)$ می‌باشد که هر کدام از این خم‌ها به یک عنصر میدان که همان ضریب A می‌باشد، نیاز دارند.
  - یک بیت برای نمایش بیت چالشی $ch_{i,\cdot}$ نیازاست. البته قابل ذکر است که اگر مقدار $ch_{i,\cdot}$ را داشته باشیم نیازی به ارسال $ch_{i,۱}$ نمی‌باشد، دلیل این امر هم تساوی $ch_{i,۱} = ۱ - ch_{i,\cdot}$ می‌باشد.
  - چنانچه در **؟؟** توضیح داده شده است، برای هش $h_{i,j} = G(resp_{i,J_I})$ نیز به ۳λ بیت فضا نیاز می‌باشد.
  ذکر این نکته نیز لازم است که با $resp_{i,J_i}$ ، می‌توان $h_{i,j}$ را محاسبه کرد و بنابراین نیازی به ارسال این هش وجود ندارد.

- براساس بیت چالشی J_i جواب‌های متفاوتی خواهیم داشت و ازاین‌رو طول بیت متفاوتی نیز برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۰$ آنگاه پاسخ موردنظر $(R, \phi(R))$ خواهد بود که دراین صورت با توجه به وجود مولدهای عمومی، بدون هیچ هزینه محاسباتی نیاز به ۳λ بیت برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۱$ آنگاه پاسخ، $\psi(S)$ است که به ۱۲λ بیت به‌عنوان یک عنصر میدان لازم خواهد بود که با فشرده سازی به ۳λ بیت تقلیل می‌یابد.

در مجموع، برای هر مرحله از اثبات دانش صفر تقریبا به طورمتوسط به

$$۲۴\lambda + ۱ + ۳\lambda + \frac{۳\lambda + ۱۲\lambda}{۲} \approx ۳۴/۵\lambda$$

بیت فضا بدون فشرده سازی نیازاست که با فشرده‌سازی تقریبا به‌طور متوسط به

$$۲۴\lambda + ۱ + ۳\lambda + ۳\lambda \approx ۳۰\lambda$$

بیت نیاز خواهد بود.

اگرچه برای تامین λ بیت امنیت پساکوانتومی کفایت می‌کند تا پروتکل اثبات دانش صفر، λ بار تکرار شود اما به‌دلیل آنکه هش چالش‌ها دربرابر الگوریتم گراور [۱۱] آسیب‌پذیر نباشد(بخش ۵/۳)، لازم است که پروتکل امضا، ۲λ بار پروتکل اثبات دانش صفر را تکرار کند. با این اوصاف در کل، امضا تقریبا به‌طورمتوسط $(۳۴/۵\lambda \times ۲\lambda =)۶۹\lambda^۲$ بیت در حالت عادی و $۶۰\lambda^۲$ بیت درحالت فشرده‌سازی لازم دارد.
به‌عنوان مثال برای دست‌یابی به ۱۲۸ بیت امنیت پساکوانتومی(تعداد بیتی که در حالت پساکوانتومی ایمن باشد) برای طرح امضای ارائه شده، به‌طور متوسط به $۴۸\lambda = ۶۱۴۴$ (۲۶۸۸ در حالت فشرده) بیت برای کلیدعمومی، $۳\lambda = ۳۸۴$ بیت برای کلیدخصوصی و $۶۹\lambda^۲ = ۱,۱۳۰,۴۹۶$ (۱۲۲,۸۸۰ برای حالت فشرده) بیت برای امضا لازم است.

۲.۷.۴.۳ سنجش

دراین قسمت می‌خواهیم سایز پارامترهای لازم در طرح خود را با سایر طرح‌های امضای پساکوانتومی مقایسه می‌کنیم.

همان‌طور که از جدول زیر قابل مشاهده است، طرح امضای معرفی شده در این پایان‌نامه دربرابر سایر طرح‌های امضای پساکوانتومی موجود دارای کلید با طول سایز کوچکتر می‌باشد. البته قابل ذکر است که گونه‌هایی از طرح امضای مرکل وجود دارد که دارای طول کلید کوچکتری(۳۲ بیت) با همان درجه امنیت می‌باشد اما؟؟.

جدول ۱.۳: سنجش سایز پارامترها(به بایت) در طرح‌های امضاهای پساکوانتومی متفاوت در سطح امنیتی ۱۲۸ بیت کوانتومی

طرح امضا	سایز کلیدعمومی	سایز کلیدخصوصی	سایز امضا
هش مبنا	۱،۰۵۶	۱،۰۸۸	۴۱،۰۰۰
کد مبنا	۱۹۲،۱۹۲	۱،۴۰۰،۲۸۸	۳۷۰
مشبکه مبنا	۷،۱۶۸	۲،۰۴۸	۵،۱۲۰
حلقه مبنا	۷،۱۶۸	۴،۶۰۸	۳،۴۸۸
چندمتغیره مبنا	۹۹،۱۰۰	۷۴،۰۰۰	۴۲۴
همسانی مبنا	۷۶۸	۴۸	۱۴۱،۳۱۲
همسانی مبنای فشرده	۳۳۶	۴۸	۱۲۲،۸۸۰

فصل ۴

امنیت طرح امضای دیجیتال همسانی مبنا

امنیت سیستم های رمزنگاری براساس مسائل سخت ریاضی بنا شده است. منظور از مسائل سخت می‌توان به مسائلی اشاره کرد که پیچیدگی محاسباتی آن‌ها برای حل مسئله فرض شده زمان‌بر بوده و بنابراین حل آن مسئله با فرض داشتن جواب مقرون‌به‌صرفه نمی‌باشد. برپایه‌ی این نوع مسائل سیستم‌های رمزنگاری متفاوتی طراحی شده‌اند. به‌عنوان مثال تجزیه‌اعداد یک مسئله سخت ریاضی محسوب می‌شود که برای حل آن حتی با بهترین الگوریتم‌های ارائه شده زمانی نمایی لازم است به این معنی که با افزایش طول ارقام مسئله زمان اجرای برنامه برای حل مسئله به صورت نمایی افزایش می‌یابد که برای اعداد بزرگ حتی به چندین سال زمان برای حل آن مورد نیاز می‌باشد. در نتیجه برپایه‌ی این مسئله‌ی سخت سیستم رمزنگاری RSA طراحی شد که هنوز در بسیاری از پروتکل‌های رمزنگاری مورداستفاده قرار می‌گیرد.

اما با ورود کامپیوترهای کوانتومی زمان مورد نیاز برای حل چنین مسائلی به شدت کاهش یافت به طوری‌که سیستم‌های رمزنگاری ارائه شده برمبنای چنین مسائلی همچون سیستم رمزنگاری RSA کارایی خود را در برابر کامپیوترهای کوانتومی از دست داد. بنابراین رمزنگاران برای آینده رمزنگاری یعنی زمانی که به جای کامپیوترهای کلاسیک از کامپیوترهای کوانتومی که به طور قابل توجهی دارای کارایی بهتر و پردازش سریع‌تری هستند استفاده شود به دنبال مسائلی برآمدند که حتی با وجود کامپیوترهای کوانتومی نیز مسائلی سخت طلقی شوند. از جمله معتبرترین مسائلی که با وجود کامپیوترها و الگوریتم‌های کوانتومی به عنوان مسائل سخت درنظر گرفته می‌شوند مسائلی است که در حوزه‌ی همسانی بین خم‌های بیضوی سوپرسینگولار ارائه شده اند. این مسائل به طور

کامل در [۶] ارائه شده‌اند. از آنجا که در طرح خود از دو تا از این مسائل استفاده می‌کنیم بنابراین در ادامه به تشریح این دو مسئله می‌پردازیم.

برای بیان این دو مسئله لازم است یک عدد اول p به فرم $\ell_A^{e_A}\ell_B^{e_B}.f \pm 1$ انتخاب کنیم سپس یک خم بیضوی سوپرسینگولار $E.$ روی میدان $\mathbb{F}_{p^2}$ با روش [۳] به‌دست آوریم که زوج نقاط $\{P_A, Q_A\}$ و $\{P_B, Q_B\}$ مولدهای زیرگروه های $E.[\ell_A^{e_A}]$ و $E.[\ell_B^{e_B}]$ باشند.

با فرض عدد اول p ، خم سوپرسینگولار $E.$ و زیرگروه‌های $E.[\ell_A^{e_A}]$ و $E.[\ell_B^{e_B}]$ و همچنین مولدهای آن به تعریف این دو مسئله می‌پردازیم:

مساله همسانی سوپرسینگولار محاسباتی : [1]

فرض کنیم $\phi_A : E. \to E_A$ 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 $(\mathbb{Z}/\ell_A^{e_A}\mathbb{Z})$ هستند با این ویژگی که هردو همزمان عاملی از ℓ_A نمی‌باشند.
با داشتن خم سوپرسینگولار E_A و همچنین نقاط $\phi_A(P_B)$ و $\phi_A(Q_B)$ یافتن مولد هسته‌ی 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 $E.$ و E_A و همسانی بین آنها یعنی ϕ_A و همچنین نقاط کمکی $\phi_A(P_B)$ و $\phi_A(Q_B)$ پیداکردن هسته همسانی مشکل است.
این مسئله به عنوان مسئله‌ی **CSSI** شناخته می‌شود.

توجه. ذکر این نکته لازم است که با داشتن مولد $R_A(= [m_A]P_A + [n_A]Q_A)$ و زوج نقاط P_A و Q_A ، یافتن نقاط m_A و n_A به سادگی توسط لگاریتم گسسته توسیع‌یافته [2] با این فرض که خم $E.$ هموار باشد، امکان‌پذیر است [۱۸] ، در نتیجه در فرض بالا نگاشت نقاط یعنی $\phi_A(P_B)$ و $\phi_A(Q_B)$ را به عنوان نقاط کمکی درنظر گرفتیم.

[1] Computational Supersingular Isogeny (CSSI) problem
[2] extended discrete logarithms

مسئله ساخت خم سوپرسینگولار تصمیم‌پذیر : [3]

برای فهم بیشتر این مسئله لازم است ابتدا به شکل زیر دقت شود :

$$\begin{tikzcd} E_0 \arrow[r, "\phi"] \arrow[d] & E_3 \arrow[d] \\ E_1 \arrow[r, "\phi'"] & E_2 \end{tikzcd}$$

شکل ۱.۴

اگر $\phi : E_0 \to E_3$ یک همسانی با مرتبه $\ell_A^{e_A}$ باشد. با دریافت چندتایی (E_1, E_2, ϕ') ، مشخص کردن اینکه کدامیک از دو توزیع زیر (که به احتمال ۱/۲ رخ می‌دهند) موجب تشکیل آنها شده است به عنوان یک مسئله سخت در همسانی‌ها شناخته می‌شود:

- انتخاب یک نقطه تصادفی R از مرتبه‌ی $\ell_B^{e_B}$ و ساخت خم‌های $E_1 = E_0/\langle R\rangle$ و $E_2 = E_3/\langle \phi_R\rangle$ و تولید همسانی $\phi' : E_1 \to E_2$ با درجه‌ی $\ell_A^{e_A}$

- انتخاب تصادفی خم E_1 در میان خم‌های هم‌مرتبه با E_0 و همچنین انتخاب تصادفی همسانی $\phi' : E_1 \to E_2$ با درجه‌ی $\ell_A^{e_A}$

این مسئله به عنوان مسئله **DSSP** شناخته می‌شود.

در ادامه برای آن‌که امنیت طرح امضای دیجیتال خود را براساس مسائل سختی که معرفی شد تشریح کنیم لازم است ابتدا به امنیت اثبات دانش صفر بپردازیم. علت این امر آن است که همان‌طور که در بخش‌های قبلی بیان شد، طرح امضای ما براساس یک نوع سیستم اثبات دانش صفر غیرتعاملی بناشده است که خود از طریق پروتکل زیگما و ساخت آنره تشکیل شده است. بنابراین در قدم اول به امنیت اثبات دانش صفر و در قدم بعدی به امنیت امضای دیجیتال می‌پردازیم.
امنیت ساخت آنره به طور کامل در [۱۹] بیان شده است.

[3] Decisional Supersingular Product (DSSP problem)

۱.۴ امنیت اثبات دانش صفر

۴

برای تشریح امنیت پروتکل زیگما کافی است قضیه زیر را اثبات کنیم:

قضیه ۱۳ . اثبات دانش صفر هویت همسانی مبنا ، ویژگی‌های تمامیت ، صداقت ویژه و دانش صفر تاییدکننده صادق را برآورده می‌کند.

اثبات. با بهره‌گیری از تکنیک‌های کلاسیک ارائه شده در [۸، ۱۰] ، در سه مرحله مستقلا به اثبات ویژگی‌های پروتکل زیگما می‌پردازیم :

تمامیت . اثبات‌کننده با استفاده از الگوریتم ارائه شده در بخش [۴] می‌تواند به راحتی و در زمان چندجمله‌ای دیاگرام **??** را به راحتی تشکیل داده و تاییدکننده نیز در زمان چندجمله‌ای می‌تواند ادعای اثبات‌کننده را تایید کند.

صداقت . برای اثبات این قسمت اجازه می‌دهیم شخصی به‌نام چارلز به عنوان یک متخاصم چندجمله‌ای با احتمال نه چندان کمی توانایی متقاعد کردن ویکتور به عنوان تاییدکننده را داشته باشد

صداقت ویژه . فرض کنیم دو رونوشت از تعاملات بین تاییدکننده و اثبات‌کننده به شکل $(com, ۰, resp_۰)$ و $(com, ۱, resp_۱)$ که $com = (E_۱, E_۲)$ را در اختیار داریم. بنابراین با استفاده از $resp_۰ = (R, \phi(R))$ ، می‌توانیم همسانی $\psi : E \to E/\langle R\rangle$ را محاسبه کنیم. از آنجا که $resp_۱ = \psi(S)$ مولد هسته‌ی ϕ' می‌باشد، درنتیجه می‌توانیم دوگان همسانی ψ (که یکتا نیز می‌باشد) یعنی $\psi' : E/\langle R\rangle \to E$ را محاسیه کنیم. و در آخر با محاسبه‌ی $\psi'(resp_۱)$ می‌توانیم یک مولد برای $\langle S\rangle$ تولید کنیم. برای فهم بیشتر مطالب بالا لازم است به شکل زیر دقت شود:

---
[4] Security of the Zero-Knowledge Proof

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

شکل ۲.۴

نتیجه ۲ . با داشتن همزمان هر دو همسانی ϕ' و ψ ، می‌توانیم زیرگروه مخفی $\langle S\rangle$ را به دست آوریم.

دانش صفر . برای اثبات این ویژگی، یک تاییدکننده متقلب [5] که به‌عنوان یک جعبه‌سیاه طلقی می‌شود، یک شبیه‌ساز (S) را می‌سازد. درهربار تکرار(پرسش و پاسخ بین تاییدکننده و اثبات‌کننده)، شبیه‌ساز S به‌صورت کاملا تصادفی و یکنواخت به تولید یک حدسی که انتظار دارد در مرحله‌ی بعد، تاییدکننده آن را به عنوان چالش موردسوال قرار دهد، می‌پردازد.

اگر $b = ۰$ ، حدس شبیه‌ساز S باشد آنگاه یک نقطه‌ی $R \in E$ از زیرگروه $E[\ell_B^{e_B}]$ لنتخاب کرده و نگاشت $\phi(R)$ را محاسبه می‌کند(لازم به یادآوری است که نگاشت ϕ روی زیرگروه $E[\ell_B^{e_B}]$ قسمتی از داده‌ی عمومی می‌باشد). درادامه شبیه‌ساز همسانی‌های $\psi : E \to E/\langle R\rangle$ $\psi' : E/\langle S\rangle \to E/\langle S,R\rangle$ را محاسبه کرده:

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

شکل ۳.۴

و در انتها $E_۱ = E/\langle R\rangle$ و $E_۲ = E/\langle S,R\rangle$ را برای تاییدکننده‌ی متقلب ارسال می‌کند.

[5] cheating verifier

اگر $b = ۱$ حدس شبیه‌ساز باشد آنگاه یک خم بیضپی سوپرسینگولار تصادفی E' هم‌مرتبه با خم E و همچنین یک نقطه تصادفی $R \in E'$ از زیرگروه $E[\ell_A^{e_A}]$ انتخاب کرده و سپس همسانی $\phi' : E' \to E'/\langle R \rangle$ را تشکیل داده:

$$E \overset{\phi}{\dashrightarrow} E/\langle S \rangle$$

$$E/\langle R \rangle \xrightarrow{\phi'} E/\langle R, S \rangle$$

شکل ۴.۴

و در پایان $E_۱ = E'$ و $E_۲ = E'/\langle R \rangle$ را برای تاییدکننده‌ی متقلب ارسال می‌کند.

اگر تاییدکننده‌ی متقلب هیچ سوال موردانتظاری (چالش)را مورد پرسش قرار ندهد آنگاه شبیه‌ساز به‌سادگی تلاشش را متوقف می‌کند. اما اگر تاییدکننده‌ی متقلب سوال موردانتظار را بپرسد، شبیه‌ساز در جواب چندتایی $(E_۱, E_۲, b, R)$ را به‌عنوان خروجی‌اش نشان می‌دهد. شبیه‌ساز بعداز m بار تعامل موفق یا تاییدکننده‌ی متقلب و یا درصورت امتناع تاییدکننده، عملیات را متوقف می‌کند.

برای اثبات ویژگی دانش‌صفر لازم است نشان دهیم که شبیه‌ساز S در زمان چندجمله‌ای اجرا شده است و خروجی‌اش نیز به‌صورت چندجمله‌ای غیرقابل تشخیص نسبت به تعاملات واقعی بین تاییدکننده‌ی متقلب و اثبات‌کننده‌ی واقعی می‌باشد:

- برای نشان دادن آن‌که شبیه‌ساز S در زمان چندجمله‌ای اجرا می‌شود کافی است تا نشان دهیم در هر تکرار برای هر حدس بیت b توسط شبیه‌ساز S احتمال آن‌که تاییدکننده‌ی متقلب سوالی بپرسد مقدار $b - ۱$ خواهد بود که به‌طور معکوس نزدیک به ۱/۲ خواهد بود. اگر فرض کنیم چنینی حالتی رخ نمی‌دهد پس اثبات‌کننده‌ی متقلب می‌تواند از یک اوراکل برای مسئله‌ی $DSSP$ استفاده کند.
- برای اثبات غیرقابل تمایز بودن لازم است از تکنیک هیبریدی گفته شده در [] استفاده

کنیم. بنابراین کافی است تا اثبات کنیم که هیچ متمایزکننده‌ی چندجمله‌ای دیگری برای یک مرحله از روند طرح تاییدهویت وجود ندارد. بنابراین به روشنی معلوم است که هیچ چنین متمایزکننده‌ای برای پرسش حالت $b = ۰$ وجود ندارد. دلیلی این امر هم این است که خروجی شبیه‌ساز S و اثبات‌کننده‌ی واقعی در این حالت کاملا یکتاست.

حال فرض کنید که یک متمایزکننده‌ی D موجود است که روی ورودی $\phi' : E_۱ \rightarrow E_۲$ به احتمال خیلی زیاد می‌تواند مشخص کند که آیا تعاملات از طرف شبیه‌ساز S رخ داده یا تعامل بین تاییدکننده‌ی متقلب و اثبات‌کننده‌ی واقعی انجام شده است یا خیر. بنابراین متمایزکننده‌ی D می‌تواند به عنوان یک اوراکل از مس‌سله‌ی $DSSP$ استفاده کند.

۲.۴ امنیت امضا

همان‌گونه که در قضیه ۲ بیان شد، طرح امضای دیجیتال به دست‌آمده از بخش ۴ ، یک امضای دیجیتال مقاوم دربرابر حمله‌ی جعل متن انتخابی یا به اختصار SUF-CMA می‌باشد. یک بخش مهم طرح امضای ما مربوط به اثبات آنره می‌باشد که اساس این اثبات برپایه‌ی اوراکل تصادفی کوانتومی G پایه‌گذاری شده است. یک ویژگی الزامی و پایه‌ای این اوراکل این است که دامنه و برد یکسانی برای هر دو نوع پاسخ داشته باشد.
همچنین در بخش ۲.۴ تکنیکی معرفی کردیم که باعث فشرده‌سازی امضا می‌شد(کلیدعمومی و پاسخ‌ها فشرده می‌شوند) که درنتیجه‌ی آن امضای ما حجم کمتری در مقایسه با حالت اولیه به‌دست می‌آورد. همچنین تابع G را اوراکل تصادفی کوانتومی‌امی درنظر گرفتیم که هش‌هایی به طول $k \approx ۳\lambda$ تولید می کند.
با این اوصاف به دلیل آنکه اثبات آنره تابع G را اوراکلی درنظر می‌گیرد که دامنه و بردی از یک نوع و اندازه دارد بنابراین برخلاف حالت فشرده، در امضای حالت نافشرده به‌دلیل آنکه پاسخ‌های ما طول‌های متفاوت k و $۴k$ دارند و دامنه‌ي تابع G در بعضی حالت‌ها متفاوت با برد آن می‌شود بنابراین اثبات آنره غیرمعتبر و درنتیجه امضای ما غیرمعتبر می گردد. تنها راه‌حلی که برای این موضوع می‌توان به کار برد این است که پاسخ‌های کوتاه‌تر را از k بیت به $۴k$ بیت افزایش دهیم و از طرف دیگر لازم است تا تابع G ، هش‌هایی به‌اندازه $۴k$ بیت تولید کند تا تابع G دامنه و برد یکسانی به‌صورت $\{۰,۱\}^{۴k}$ داشته باشد که درنتیجه‌ی آن اثبات آنره و بالطبع آن امضای ما معتبر گردد. البته با این روش سایز امضای ما دقیقا ۱۸λ بیت افزایش خواهد یافت.
درادامه با یک استدلال موقت نشان خواهیم داد که در طرح امضای خود نیازی به فشرده‌سازی نداریم و اگر تابع G هش‌هایی به‌اندازه $k \approx ۳\lambda$ تولید کند آنگاه امضای نافشرده‌ی ما ایمن باقی می‌ماند. در ادامه بحث $\mathcal{DS}_u$ معرف امضای نافشرده و $\mathcal{DS}_c$ بیان‌کننده‌ی امضای فشرده(پاسخ $\psi(S)$ فشرده می‌شود) خواهد بود.

قضیه ۱۴ . $\mathcal{DS}_c$ یک SUF-CMA در مدل اوراکل تصادفی می‌باشد.

اثبات ۴ . ازآنجا که تمام پاسخ‌ها به‌اندازه k بیت ارائه می‌شوند درنتیجه ورودی و خروجی الگوریتم امضا یکسان و k می‌باشد، بنابراین امنیت $\mathcal{DS}_c$ وابسته به امنیت قضیه ۲ می‌باشد.

قضیه ۱۵. $\mathcal{DS}_u$ یک SUF-CMA در مدل اوراکل تصادفی می‌باشد.

اثبات ۵. برای اثبات این قضیه از روش حل مسائل کاهشی استفاده می‌کنیم. به‌عبارت دیگر اگر $\mathcal{DS}_u$ و $\mathcal{DS}_c$ را دومسئله درنظربگیریم و $\mathcal{DS}_u$ قابل کاهش به $\mathcal{DS}_c$ باشد، آنگاه اگر امنیت مسئله‌ی اول نقض شود مطمئنا امنیت مسئله‌ی دوم نیز به‌خطر می‌افتد و درطرف دیگر، اگر مسئله‌ی دوم در برابر هر متخاصمی ایمن باشد اطمینان می‌یابیم که مسئله‌ی اول نیز کاملا ایمن می‌باشد. در ادامه می‌خواهیم این امر را اثبات کنیم که $\mathcal{DS}_u$ قابل کاهش به $\mathcal{DS}_c$ است و از آنجا که در قضیه **؟؟** اثبات شد که $\mathcal{DS}_c$ ایمن است بنابراین به این نتیجه برسیم که $\mathcal{DS}_u$ نیز ایمن می‌باشد.

فرض کنید متخاصم چندجمله‌ای کوانتومی A موجود می‌باشد که قادر به شکستن امنیت $\mathcal{DS}_u$ می‌باشد. می‌خواهیم نشان دهیم اگر این متخاصم به یک اوراکل امضای کلاسیک برای $\mathcal{DS}_c$ و یک اوراکل تصادفی کوانتومی $G_c : \{۰,۱\}^k \to \{۰,۱\}^k$ دسترسی داشته باشد، می‌تواند یک زوج پیام-امضای معتبر برای $\mathcal{DS}_c$ جعل کند.
بدین منظور فرض کنید کلیدعمومی pk و یک اوراکل امضا برای نمونه‌ی $\mathcal{DS}_c$ را به‌همراه اوراکل‌های تصادفی کوانتومی G_c و H دراختیار داریم. همچنین فرض کنید $C_۰$ و $C_۱$ نیز به‌ترتیب معرف مجموعه جواب‌های ممکن چالش‌های $ch = ۰,۱$ در $\mathcal{DS}_c$ باشند(البته قابل ذکر است که تعداد اعضای هر دو مجموعه دقیقا ۲^k و هر یک از اعضای مجموعه، رشته‌بیت‌هایی به طول k می‌باشند). حال می‌خواهیم از طریق الگوریتم‌های بیان شده در بخش‌های قبلی و همچنین موارد فرض شده در بالا یک نمونه امضای $\mathcal{DS}_u$ را تولید کنیم با این تفاوت که تابع تصادفی کوانتومی G_u به‌صورت زیر تعریف می‌شود.

قبل از معرفی تابع G_u ، ابتدا به معرفی چندمتغیر می‌پردازیم.

- مجموعه پاسخ‌های ممکن چالش‌های $ch = ۰,۱$ در امضای $\mathcal{DS}_u$ را به ترتیب $U_۰$ و $U_۱$ می‌نامیم. همان‌طور که قابل بررسی است روابط $C_۰ = U_۱$ و $|C_۱| = |U_۱|$ بین این متغیرها برقرار می‌باشد با این نکته که عناصر $U_۱$ ، $۴k$ بیتی می‌باشند.

- تابع $\mathcal{C} : U_1 \to C_1$ نمایش یک نگاشت فشرده‌سازی است که نقاط $\psi(s)$ در U_1 را به ضریب فشرده‌سازی معادلش در C_1 نگاشت می‌کند.
  تابع $\mathcal{C} : U_1 \to C_1$ یک تابع دوطرفه می‌باشد که .. ؟؟؟
- تابع $G'_u : \{0,1\}^{4k} \to \{0,1\}^k$ یک اوراکل تصادفی کوانتومی می‌باشد که

$$\forall x \in \{0,1\}^{4k} \ : \quad G'_u(z \parallel x) = G_c(x)$$

و z نشان‌دهنده‌ی رشته‌هایی تماما صفر به طول $3k$ می‌باشد.

با توجه به تعاریف بالا، تابع $G_u : \{0,1\}^{4k} \to \{0,1\}^k$ را به‌صورت زیر تعریف می‌کنیم:

$$G_u(x) = \begin{cases} G'_u(z \parallel \mathcal{C}(x)) & \text{اگر } x \in U_1 \\ G'_u(\mathcal{C}^{-1}(y)) & \text{اگر } x = z \parallel y \text{ زمانیکه } y \in C_1 \\ G'_u(x) & \text{درغیراینصورت} \end{cases}$$

از آنجا که تابع G_u تنها به تغییر ورودی‌ها طبق تابع دوطرفه‌ی $\mathcal{C}$ قبل از اعمال اوراکل تصادفی کوانتومی G'_u می‌پردازد بنابراین تابع G_u همانند تابع G'_u می‌باشد و قابل تفکیک نیستند. درنتیجه متخاصم $\mathcal{A}$ می‌تواند زمانیکه G_u تشکیل می‌شود، امضای $\mathcal{DS}_u$ را بشکند.

اگر متخاصم $\mathcal{A}$ به کلیدعمومی pk و اوراکل‌های تصادفی کوانتومی G_u و H دسترسی داشته باشد و تقاضای امضا برای پیام m را داشته باشد، برای آن‌که بتوانیم امضای $\mathcal{DS}_c$ را جعل کنیم(می‌دانیم که طبق آنچه درابتدا فرض کردیم امضای $\mathcal{DS}_u$ توسط متخاصم قابل جعل می‌باشد) کافیست درخواستش را به اوراکل امضای $\mathcal{DS}_c$ ارسال کنیم و امضای

$$\sigma \leftarrow ((com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$$

را دریافت کنیم. دراین حالت رابطه‌های زیر را خواهیم داشت:

$$J_1 \parallel \cdots \parallel J_{2\lambda} \leftarrow H(pk, m, (com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$$

$$h_{i,J_i} = G(resp_{i,J_i})$$

از آنجا که امضای به‌دست آمده امضای فشرده می‌باشد و متخاصم $\mathcal{A}$ ، قادر به جعل امضای نافشرده می‌باشد بنابراین لازم است تا تمام $resp_{i,J_i}$ در σ را زمانیکه $ch_{i,J_i} = ۱$ می‌باشند را از حالت فشرده خارج کرده و امضای σ اصلاح‌شده که متناسب با طرح امضای $\mathcal{DS}_u$ می‌باشد را برای $\mathcal{A}$ ارسال کنیم. البته باتوجه به روابط G_u پایین، $h_{i,j}$ به‌دست آمده متناظر با یک هش واقعی در امضای $\mathcal{DS}_u$ خواهند بود و درنتیجه σ اصلاح‌شده، یک امضای معتبر برای پیام m در امضای $\mathcal{DS}_u$ خواهد بود:

$$\begin{cases} G_u(x) = G'_u(z \parallel x) = G_c(x) & \text{اگر } x \in C_۰, U_۰ \\ G_u(\mathcal{C}^{-۱}(x)) = G'_u(z \parallel x) = G_c(x) & \text{اگر } x \in C_۱ \\ G_u(x) = G'_u(z \parallel \mathcal{C}(x)) = G_c(\mathcal{C}(x)) & \text{اگر } x \in U_۱ \end{cases}$$

بنابراین با توجه به مطالب بالا، به‌راحتی می‌توانیم درخواست‌های متخاصم $\mathcal{A}$ را از طریق اوراکل‌های $\mathcal{DS}_c$ جواب دهیم(ونه با $\mathcal{DS}_u$) و می‌دانیم که متخاصم $\mathcal{A}$ می‌تواند یک جفت پیام_امضای معتبر (m, σ) از نوع $\mathcal{DS}_u$ جعل کند.حال اگر این امضای جعلی را بدون محاسبه‌ی هش‌ها، فشرده کنیم آنگاه یک زوج پیام_امضای معتبر برای $\mathcal{DS}_c$ به‌دست آوریم که این امر مخالف با قضیه‌ی **؟؟** می‌باشد و نتیجه می‌گیریم به‌دلیل آنکه امضای $\mathcal{DS}_c$ امن است پس امضای $\mathcal{DS}_u$ نیز ایمن است و دربرابر هر حمله‌ی کوانتومی مقاوم خواهد بود.

۴.۳ تعداد مراحل

همان‌طور که قبلا بیان شد برای دست‌یابی به λ بیت امنیت، لازم است تا پروتکل حداقل $t = ۲\lambda$ بار تکرار شود. در ادامه این بخش، به اثبات این ادعا می‌پردازیم.

با توجه به تابع H که ورودی و خروجی‌های آن به شکل زیر است :

$$J_۱ \parallel \cdots \parallel J_t \longleftarrow H\big(\, pk, m, (com_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j} \,\big)$$

فرض کنید یک متخاصم کوانتومی می‌تواند رشته‌ی دلخواه $(J_1 \parallel \cdots \parallel J_t)$ را به‌عنوان چالش انتخاب و با استفاده از الگوریتم گراور [۱۱] ، به‌جستجوی ؟؟ روی تابع H بپردازد تا بتواند یک پیام m متناسب با هش به‌دست آمده تولید کند. و از آنجا که بقیه‌ی پارامترها عمومی هستند به تولید اثبات شبیه‌سازی شده‌ی π اقدام کند. یک حمله‌ی ؟؟؟

درنتیجه برای داشتن λ بیت امنیت در برابر حمله‌ی متخاصم کوانتومی لازم است تا طرح امضای ما، اثبات‌دانش‌صفر را $t = ۲\lambda$ بار تکرار کند.

همان‌طور که قبلا بیان شد در اثبات دانش صفر، اگر پاسخ هر دو چالش $b = ۰, ۱$ همزمان داده شود آنگاه هرکسی توانایی محاسبه کلیدخصوصی را خواهد داشت. بنابراین یک امرمسلم در طرح امضای دیجیتال ما این است تا یک تعهد، دوبار استفاده نشود. درادامه می‌خواهیم نشان دهیم رخ دادن این اتفاق در طرح امضای ما، احتمال بسیار کوچکی دارد که قابل چشم‌پوشی می‌باشد و امنیت امضا برقرار می‌باشد.

چنانچه می‌دانیم، عدداول انتخابی ما به فرم $p = \ell_A^{e_A}\ell_B^{e_B}.f \pm ۱ \approx ۲^{۶\lambda}$ که $\ell_A^{e_A} \approx \ell_B^{e_B} \approx ۲^{۳\lambda}$ می‌باشد. با این حال، دقیقا $\ell_B^{e_B-۱} - ۱ \approx ۲^{۳\lambda}$ زیرگروه دوری متفاوت برای $E[\ell_B^{e_B}]$ وجود خواهد داشت که هر یک از ادعاها از این زیرگروه‌ها به صورت تصادفی انتخاب می‌شوند. برای هر امضا، پروتکل دانش‌صفر به تعداد ۲λ بار تکرار می‌شود، حال اگر فرض کنیم که ۲^s پیام را می‌خواهیم امضا کنیم آنگاه $۲^{s+۱}\lambda$ زیرگروه دوری به‌صورت تصادفی از $E[\ell_B^{e_B}]$ انتخاب خواهیم کرد. یک احتمال بالا از اینکه یک زیرگروه را حداقل دو بار انتخاب می‌کنیم با ؟؟:

$$\frac{۲^{s+۱}\lambda(۲^{s+۱}\lambda - ۱)}{۲.۲^{۳\lambda}} \frac{۲^{۲s+۲}\lambda^۲}{۲^{۳\lambda+۱}} \frac{\lambda^۲}{۲^{\lambda-۱}}$$

که برای $s\lambda$

۴.۴ جنبه‌های الگوریتمی

۱.۴.۴ تولید پارامترها

اعداد اولی که در طرح خود استفاده می‌کنیم به فرم $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm ۱$ می باشند. دلیل این امر آن است که برای تامین امنیت طرح خود لازم است تا ابتدا اعداد اول ثابت ℓ_A و ℓ_B را به‌صورت مجزا از عدد اول p ، با ویژگی $\ell_A^{e_A} \approx \ell_B^{e_B}$ (به این معنی که از نظر بیتی هم اندازه هستند) انتخاب کنیم. مطمئنا با ضرب مقادیر $\ell_A^{e_A}$ و $\ell_B^{e_B}$ عدد اولی خاصل نخواهد شد، در بهترین حالت با اضافه یا کم کردن مقدار یک به این حاصلضرب می‌توان به یک عدد اول رسید و اگر نتیجه حاصل نشد می‌توانیم مقدار یک را با مضربی از $\ell_A^{e_A}\ell_B^{e_B}$ جمع یا تفریق کنیم. این مضرب در فرم بالا همان مقدار متغیر f می‌باشد. بنابراین عدد اول استفاده شده در طرح ما به‌یکی از فرم‌های زیر خواهد بود:

$$p = \ell_A^{e_A}\ell_B^{e_B} \cdot f + ۱ \quad \text{یا} \quad p = \ell_A^{e_A}\ell_B^{e_B} \cdot f - ۱$$

بروکر در [۳] نشان داده است برای هر عدد اول $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm ۱$ ، می‌توان به راحتی یک خم بیضوی سوپرسینگولار E روی میدان $\mathbb{F}_{p^۲}$ با مرتبه $(p \mp ۱)^۲ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲$ به دست آورد. دلیل این امر هم آن است که اگر E یک خم بیضوی روی میدان $\mathbb{F}_p$ باشد، آنگاه $\mathbb{F}_p$ ـ نقاط روی خم به شکل زیر می‌باشد:

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^۲ = x^۳ + Ax + B\} \cup \{\infty\}$$

که نتیجه می‌شود که :

$$E(\mathbb{F}_p) \subseteq (\mathbb{F}_p \times \mathbb{F}_p) \cup \{\infty\}$$

و چون مجموعه‌ی سمت راست متناهی(از مرتبه‌ی $p^۲ + ۱$) است لذا مجموعه‌ی سمت چپ یعنی $E(\mathbb{F}_p)$ نیز متناهی است. بنابراین :

$$\#E(\mathbb{F}_p) \leq p^۲ + ۱$$

حال اگر یک تغییر کوچک به معادله‌ی بالا اعمال کنیم، خواهیم داشت:

$$\#E(\mathbb{F}_p) - ۱ \leq p^۲$$

حال، اگر

$$p = \ell_A^{e_A}\ell_B^{e_B} \cdot f + ۱ \quad \text{یا} \quad p = \ell_A^{e_A}\ell_B^{e_B} \cdot f - ۱$$

آنگاه

$$p^۲ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲ + ۲(\ell_A^{e_A}\ell_B^{e_B} \cdot f) + ۱ \quad \text{یا} \quad p^۲ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲ - ۲(\ell_A^{e_A}\ell_B^{e_B} \cdot f) - ۱$$

که به‌طور خلاصه خواهیم داشت:

$$p^۲ \mp ۱ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲ \pm ۲\ell_A^{e_A}\ell_B^{e_B} \cdot f$$

و از آنجا که $p = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)$ ، بنابراین فرم بالا به‌صورت زیر خلاصه می‌شود:

$$p^۲ \mp ۲p \mp ۱ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲$$

که اگر دوباره آن را خلاصه کنیم، خواهیم داشت:

$$(p \mp ۱)^۲ = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^۲$$

که همان مرتبه‌ی گفته شده با روش بروکر می‌باشد.

۲.۴.۴ پیدا کردن مولدهای زیرگروه تابی

برای انتخاب نقاط مولد زیرگروه $E.[\ell_A^{e_A}]$ ، می‌توان یک نقطه تصادفی $P \in_R E.(\mathbb{F}_{p^۲})$ انتخاب و آن را در $(\ell_B^{e_B} \cdot f)^۲$ ضرب کرد تا نقطه P' با مرتبه توانی از ℓ_A حاصل شود. از آنجا که عامل‌های عدد اول p، $\ell_A^{e_A}$ و $\ell_B^{e_B}$ می‌باشند، احتمالا P' از مرتبه $\ell_A^{e_A}$ خواهد بود؛ 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 P می‌شویم. برای به دست آوردن نقطه دوم مولد یعنی 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 و محاسبه $e(P_A, Q_A)$ در میدان $E[\ell_A]$ ، بررسی کرد که آیا نتیجه از مرتبه ℓ_A می‌باشد یا خیر؛ برای اطمینان از اینکه نقطه‌ی Q_A متفاوت از نقطه‌ی P_A می‌باشد می‌توانیم از گزاره زیر استفاده کنیم:

گزاره ۵. اگر $P_A, Q_A \in E[\ell_A]$ و ℓ_A عددی اول باشد آنگاه

$$e_n(P_A, Q_A) = ۱ \text{ اگر و تنها اگر } Q_A = kP_A$$

اثبات ۶.

۱. اگر فرض شود به ازای یک k ، $Q_A = kP_A$ در این‌صورت:

$$e_n(P_A, Q_A) = e_n(P_A, kP_A) = e_n(P_A, P_A)^k = ۱^k = ۱$$

۲. همچنین اگر $e_n(P_A, Q_A) = ۱$ ، در اینصورت $R \in E[\ell_A]$ را چنان اختیار می‌کنیم که $E[\ell_A] = \langle P, R \rangle$ ، بنابراین $\partial = e_n(P_A, R)$ یک ریشه n ـ ام اولیه واحد است. پس:

$$Q_A \in E[n] = \langle P_A, R \rangle \longrightarrow \exists ۰ \leqslant k, l \leqslant \ell_A - ۱ \ , \quad Q_A = kP_A + \ell R$$

اکنون :

$$۱ = e_n(P_A, Q_A) = e_n(P, kP + \ell R) = e_n(P, P)^k r_n(P, R)^\ell = \partial^\ell$$

بنابراین $\partial^\ell = ۱$ و درنتیجه $\ell = ۰$ ، پس $Q_A = kP_A$.

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، قابل تبدیل به یکدیگر می‌باشند. چنانچه در [۱۷] اشاره شده است این محاسبه به راحتی در زیرگروه $E[\ell_A]$ قابل انجام می‌باشد.

۳.۴.۴ تبادل کلید

یک از مهمترین ارکان هر سیستم رمزنگاری مربوط به تبادل کلید می‌باشد. به‌عبارت دیگر زمانی‌که یک سیستم رمزنگاری معرفی می‌شود در ابتدا بررسی می‌شود که آیا می‌توان پروتکل تبادل کلید را

پیاده‌سازی کرد یا خیر. بعد از معرفی سیستم رمزنگاری همسانی- مبنا، آقای جائو در [] به معرفی تبادل کلید در همسانی‌ها پرداخت. از آنجا که این موضوع در ادامه بحث مورد استفاده قرار می‌گیرد، این پروتکل رو تشریح می‌کنیم.

پروتکل تبادل کلید نوعی از پروتکل دیفای- هلمن است که طبق شکل ۱ صورت می‌پذیرد. ایده‌ی کلی این پروتکل آن است که شخصی مانند آرش، همسانی ϕ و شخص دیگری همچون بابک همسانی ψ را به عنوان کلیدخصوصی انتخاب می‌کنند و با یک خم سوپرسینگولار عمومی همچون $E.$ بااستفاده از پروتکل قدم‌زدن روی گراف به تولید یک خم خصوصی همچون E_A و E_B می‌پردازند که بیانگر کلیدخصوصی آنها همچون دیگر پروتکل‌های رمزنگاری می‌باشد. در ادامه با انجام محاسباتی که در ادامه تشریح خواهد شد به‌طور مجزا به تولید یک خم سوپرسینگولار همچون E_{AB} خواهند پرداخت که به عنوان کلیدعمومی‌شان درنظر گرفته می‌شود.

۱. در ابتدا، یک خم سوپرسینگولار دلخواه در میدان $\mathbb{F}_{p^2}$ همچون $E.$ و دو جفت نقاط $\{P_A, Q_A\}$ و $\{P_B, P_B\}$ که مولد زیرگروه‌های تابی $E_[\ell_A^{e_A}]$ و $E_[\ell_B^{e_B}]$ می‌باشند را به‌عنوان پارامترهای عمومی پروتکل درنظر می‌گیریم.

۲. (آ) درادامه آرش دو عنصر تصادفی همچون $m_A, n_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (که هردو همزمان به ℓ_A بخش‌پذیر نیستند) را انتخاب می‌کند و همسانی $\phi_A : E. \to E_A$ را که هسته‌ی آن $K_A := \langle [m_A]P_A + [n_A]Q_A \rangle$ می‌باشد را محاسبه می‌کند.درادامه آلیس $\{\phi_A(P_B), \phi_A(Q_B)\} \subset E_A$ را نیز محاسبه می‌کند.

(ب) به‌طورمشابه، بابک نیز دو عنصر تصادفی همچون $m_B, n_B \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ را انتخاب و همسانی $\phi_B : E. \to E_B$ با هسته‌ی $K_B := \langle [m_B]P_B + [n_B]Q_B \rangle$ را به‌همراه نقاط $\{\phi_B(P_A), \phi_B(Q_A)\}$ محاسبه می‌کند.

۳. (آ) در ادامه آرش پس از دریافت E_B و $\phi_B(P_A), \phi_B(Q_A) \in E_B$ از جانب بابک، به محاسبه‌ی همسانی $\phi'_A : E_B \to E_{AB}$ با هسته‌ی $\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$ می‌پردازد.

(ب) بابک نیز با دریافت E_A به محاسبه‌ی همسانی $\phi'_B : E_A \to E_{AB}$ با هسته‌ی $\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B)\rangle$ می‌پردازد.

۴. آرش و بابک برای دسترسی به یک کلیدخصوصی مشترک می‌توانند j ـ پایای خم زیر را محاسبه کنند:

$$E_{AB} = \phi'_B(\phi_A(E.)) = \phi'_A(\phi_B(E.)) = E./\langle [m_A]P_A+[n_A]Q_A, [m_B]P_B+[n_B]Q_B\rangle$$

۴.۴.۴ ساده‌سازی نقاط تاب‌دار

الگوریتم ۶ نردبان سه‌ـ نقطه‌ای برای محاسبه‌ی $P + [t]Q$

Input : t,P,Q

Set $A = 0, B = Q, C = P$ Compute $Q - P$; $i = |t|$ **to** 1 Let t_i be the i-th bit of t; $t_i = 0$ $A = 2A, B = dadd(A, B, Q), C = dadd(A, C, P)$; $A = dadd(A, B, Q), B = 2B, C = dadd(B, C, Q - P)$;

Output : $C = P + [t]Q$

در طرح خود از نقاط تصادفی متنوعی با مرتبه‌ی خاص همچون $\ell_A^{e_A}$ و $\ell_B^{e_B}$ به‌مراتب برای ساخت یک زیرگروه استفاده می‌کنیم. از آنجا که نقاط موردنظر ما از زیرگروه‌های تابی $E[\ell_A^{e_A}]$ و $E[\ell_B^{e_B}]$ با دو مولد P و Q ساخته می‌شوند لذا این نقاط به فرم $\langle [m]P + [n]Q\rangle$ خواهند بود. البته m و n همزمان نباید با ℓ^e موردنظر بخش‌پذیر باشند. بنابراین نتیجه می‌گیریم که که اگر روشی ارائه دهیم تا ساخت این زیرگروه‌ها با محاسبات کمتری انجام پذیرند، به‌طور کلی طرح ما بهینه‌تر خواهد شد.

می‌توان فرض کرد که هر m ، دارای عنصر وارون در پیمانه‌ی مرتبه‌ی گروه می‌باشد(این فرض هیچ خدشه‌ای به زیرگروه وارد نمی‌کند). در این حالت $R' = P + [m^{-۱}n]Q$ زیرگروهی همانند دیگر مولدها خواهد بود. محاسبه R' با روش استاندارد **دوبرابرـ وـ جمع** [۶] نیاز به نصف عملیات محاسبات $[m]P + [n]Q$ معمولی را دارا می‌باشد(برای روش های بهتر محاسبه عملیات معمولی به مراجعه [۱، ۷، ۱۶] شود). با این حال، محاسبه $P + [m^{-۱}n]Q$ با روش دوبرابرـ وـ جمع، یک

[6] double-and-add

*حفره امنیتی (اشکال بزرگ) را داراست : در برابر حملات **آنالیز قدرت ساده** یا **SPA** [۱۳] آسیب پذیر می‌باشد. برای جلوگیری از این حمله می‌توان از روش **نردبان مونت‌گومری** [7] [۱۴] برای محاسبه $[m^{-1}n]Q$ استفاده کرد و سپس P را به آن اضافه کرد، اما این روش به طور قابل توجهی کند می‌باشد. به‌منظور رفع دو مشکل کندی و حمله‌ی SPA ، الگوریتم **؟؟** را ارائه می‌دهیم که بیانگر یک روش بسیار موثرتری می‌باشد و مستقیما $P + [m^{-1}n]Q$ را محاسبه می‌کند. 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 $dadd(A, B, C)$ مورد استفاده شده در الگوریتم نیز معرف جمع تفاضلی [8] [۱۴] می‌باشد.*
پیاده سازی جمع تفاضلی در خم‌های مونت‌گومری به کارآمدی روش دوبرابر- و- جمع ساده روی خم های دوقلوی ادوارد [9] ۶.۴.۴ می‌باشد، بنابراین درادامه به معرفی خم‌های مونت‌گومری خواهیم پرداخت.

۵.۴.۴ محاسبه همسانی‌های با درجه هموار

[10]

محاسبه‌ی همسانی یکی از پرهزینه‌ترین محاسبات در سیستم‌های همسانی- مبنا می‌باشند. از آنجا که در طرح خود نیز به‌مراتب به محاسبه‌ی همسانی‌ها با درجه‌ی معینی می‌پردازیم بنابراین لازم است تا روشی سریع برای این امر معرفی کنیم. البته این روش می‌تواند در تمامی طرح‌ها مورداستفاده قرار گیرد، به‌عنوان مثال می‌توان در محاسبات همسانی در پروتکل تبادل کلیدی که در بخش قبل بین آرش و بابک انجام می‌پذیرد اشاره کرد.
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 ۵.۴: ساختمان محاسبات ساخت $\phi = \phi_5 \circ \cdots \circ \phi$.

[7] Montgomery ladder
[8] differential addition
[9] twisted Edwards curves
[10] Computing smooth degree isogenies

اگر درجه‌ی نگاشت ϕ هموار باشد، می‌توان آن را به زنجیره‌ای از ℓ ـ همسانی ها تجزیه کرد. اگر $E_{۰} = E$ و $R_{۰} = R$ در نظر بگیریم، آنگاه برای هر $۰ \leq i < e$ می‌توان مقادیر زیر را در نظر گرفت :

$$E_{i+۱} = E_i/\langle \ell^{e-i-۱} R_i\rangle, \quad \phi_i : E_i \to E_{i+۱}, \quad R_{i+۱} = \phi_i(R_i).$$

چنانکه $E/\langle R\rangle = E_e$ و $\phi = \phi_{e-۱} \circ \cdots \circ \phi_{۰}$ می‌باشد.

توجه به این نکته لازم است که از آنجا که زیرگروه ℓ ـ تابی $\langle R_i\rangle$ خم E_i مشخص می‌باشند ،

خم بیضوی $E_{i+۱}$ و همسانی ϕ_i می‌توانند توسط فرمول ولو [11] [۲۰] به راحتی محاسبه شوند.

در [۱۲] ، دو پیشنهاد برای داشتنی پیچیدگی درجه دو برای e بیان شده است ؟؟؟؟؟.

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 :

$$i = ۰ \Rightarrow \quad E_۱ = E_{۰}/\langle \ell^۴ R_{۰}\rangle, \quad \phi_{۰} : E_{۰} \to E_۱, \quad R_۱ = \phi_{۰}(R_{۰})$$

$$i = ۱ \Rightarrow \quad E_۲ = E_۱/\langle \ell^۳ R_۱\rangle, \quad \phi_۱ : E_۱ \to E_۲, \quad R_۲ = \phi_۱(R_۱) = \phi_۱(\phi_{۰}(R_{۰}))$$

[11] Velu's formulas

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

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

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

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

۶.۴.۴ انتخاب مدل

*بعد از ارائه‌ی یک طرح لازم است تا آن طرح بهینه‌سازی شود. به‌عبارت‌دیگر زمانی که یک طرح ارائه و تایید شد برای پیاده‌سازی آن باید سرعت اجرای طرح را تا جای ممکن افزایش داد. به‌طورمثال می‌توانیم تعداد عملیاتی که در یک الگوریتم انجام می‌شود را به حداقل رساند یا عملیات‌های سنگین تر را به عملیات های با بار پیچیدگی کمتر از لحاظ زمان اجرا جایگزین کرد. معمولا در زمان محاسبه پیچیدگی یک الگوریتم از عملیات تفریق، جمع و مقایسه صرف‌نظر می‌کنند. دلیل این امر آن است که این عملبات ، پیچیدگی محاسبات بالایی ندارند و به‌عنوان عمل‌های پایه در هر الگوریتم فرض می شوند. از جمله اعمالی که در سرعت اجرای یک الگوریتم می‌تواند تاثیرگذار باشد می‌توان به عملیات وارون، ضرب و توان اشاره کرد. برای بهینه‌سازی یک الگوریتم تلاش می‌شود که این عملیات ها به حداقل برسند. در ادامه از علائم I ، M و S به‌منظور عملیات وارون، ضرب و توان استفاده می‌کنیم. از جمله عملیات سنگین و زمان‌بری که در طرح امضای خود می‌توانیم نام ببریم عملیات **دوبرابرکردن** ، **جمع** ، **محاسبه** و **ارزیابی** همسانی‌ها خواهد بود. چنانچه ذکر شد پس از ارائه‌ی طرح خود مصمم هستیم تا سرعت اجرای الگوریتم ها در طرح خود را افزایش دهیم. یکی از راه حل‌های ممکن این است که به‌جای استفاده از یک خم بیضوی معمولی با معادله‌ی وایرشتراس، از خم بیضوی مونت‌گومری استفاده کنیم. مزیت استفاده از خم مونت‌گومری را پس از تعریف آن ذکر خواهیم کرد.*

تعریف ۱.۴.۴ . یک خم مونت‌گومری در میدان $\mathbb{F}_q$ یک خم بیضوی به فرم زیر می‌باشد:

$$M_{B,A} : By^۲ = x^۳ + Ax^۲ + x$$

در این خم برای نقاط $P = (x_P, y_P)$ و $Q = (x_Q, y_Q)$ ، نقطه‌ی $R = P + Q = (x_R, y_R)$ به‌صورت زیر محاسبه می‌شود:

$$x_R = B\lambda^۲ - (x_P + x_Q) - A$$
$$y_R = \lambda(x_P - x_Q) - y_P$$

که در آن

$$\lambda = \begin{cases} \frac{y_Q - y)P}{x_Q - x_P} & \text{اگر } P \neq Q, -Q \\ \frac{۳x_P{}^۲ + ۲Ax_P + ۱}{۲By_P} & \text{اگر } P = Q. \end{cases}$$

j- پایای این فرم از خم بیضوی برابر با مقدار

$$j(M_{B,A}) = \frac{۲۵۶(A^۲ - ۳)^۳}{A^۲ - ۴}$$

است که تنها به پارامتر A بستگی دارد.
همچنین خم تصویری مونت‌گومری در میدان $\mathbb{F}_q$ یک خم بیضوی به فرم

$$M_{B,A} : BY^۲Z = X^۳ + AX^۲Z + XZ^۲$$

است که در آن $A, B \in \mathbb{F}_q$ ، $B \neq ۰$ و $A^۲ \neq ۴$. مجموعه نقاطی که روی این خم هستند همراه با نقطه‌ی همانی $(۰ : ۱ : ۰) = \infty$ گروه نقاط $M_{B,A}$ را تشکیل می‌دهند.

برای آن‌که بتوانیم خم مونت‌گومری را جایگزین خم وایرشتراس کنیم لازم است تا یک یکیریختی بین آنها پیدا کنیم. چناچه در [۵] ذکر شده است اگر در میدان $\mathbb{F}_q$ ، q توانی از ۳ نباشد، خم مونت‌گومری $M_{B,A}$ با نگاشت گویای

$$\phi : M_{B,A} \longrightarrow E$$
$$(x, y) \mapsto (X, Y) = (B(x + A/۳), B^۲y)$$

با خم وایرشتراس کوتاه

$$E : Y^{۲} = X^{۳} + (B^{۲}\frac{۱ - A^{۲}}{۳})X + \frac{B^{۳}A}{۳(۲A^{۲}/۹ - ۱)}$$

یکریخت است.
همچنین وارون نگاشت ϕ برابر است با :

$$\phi^{-۱} : E \longrightarrow M_{B,A}$$
$$(X,Y) \mapsto (x,y) = (X/B - A/۳, Y/B^{۲})$$

اگر فرض کنیم $E : Y^{۲} = X^{۳} + aX + b$ یک خم بیضوی باشد در این‌صورت E با یک خم مونت‌گومری یکیریخت است اگر و تنها اگر $\alpha \in \mathbb{F}_q$ وجود داشته باشد که $\alpha^{۳} + a\alpha + b = ۰$ و $\sqrt{۳\alpha^{۲} + a} \in \mathbb{F}_q$. حال اگر $\beta = \sqrt{۳\alpha^{۲} + a}$ آنگاه نگاشت گویای زیر یک یکریختی بین این دو خم خواهد بود:

$$\phi : E \longrightarrow M_{۳\alpha/\beta, ۱/\beta}$$
$$(X,Y) \mapsto (x,y) = ((X - \alpha)/\beta, Y/\beta)$$

اعمال جمع و ضرب اسکالر روی نقاط خم بیضوی به فرم مونت‌گومری بااستفاده از یک نگاشت x انجام می‌شود. این نگاشت روی نقطه‌ی $P = (x : y : z) \in M_{B,A}$ به‌صورت زیر تعریف می‌شود:

$$x : M_{B,A} \longrightarrow \mathbb{P}^{۱}$$
$$P \mapsto \begin{cases} (x : z) & P \neq \infty \text{ اگر } \\ (۱ : ۰) & P = \infty \text{ اگر } \end{cases}$$

در [۱۵] نشان داده شده است که رابطه‌های

$$x_{P+Q}(x_P - x_Q)^{۲}x_P x_Q = B(x_P y_Q - x_Q y_Q)^{۲}$$

$$۴x_{۲P}x_P(x_P^{۲} + Ax_P + ۱) = (x_P^{۲} - ۱)^{۲}$$

و

$$xP - Q(x_P - x_Q)^{۲} x_P x_Q = B(x_P y_Q + x_Q y_Q)^{۲}$$

روی نقاط $P, Q \in M_{B,A}$ برقرار است. از این معادلات می‌توان نتیجه گرفت

$$x_{P+Q} x_{P-Q} = \frac{(x_P x_Q - ۱)^{۲}}{(x_P - x_Q)^{۲}}$$

$$x_{۲P} = \frac{(x_P^{۲} - ۱)^{۲}}{۴x_P(x_P^{۲} + Ax_P + ۱)}$$

و لذا

$$x_{P+Q} = \frac{(x_P x_Q - ۱)^{۲}}{(x_P - x_Q)^{۲} xP - Q}$$

$$x_{۲P} = \frac{(x_P^{۲} - ۱)^{۲}}{۴x_P(x_P^{۲} + Ax_P + ۱)}$$

ار این معادلات می‌توان مولفه‌ی x نقاط $P + Q$ و $۲P$ را با مولفه‌های x نقاط $P, Q, P - Q \in M_{B,A}$ محاسبه کرد.

۷.۴.۴ اندازه پارامتر

[۱۲] همان‌طور که قبلا بررسی شد، اعداد اولی که برای ساخت همسانی‌ها از آن استفاده می‌کنیم به فرم $p = \ell_A^{e_A} \ell_B^{e_B} \cdot f \pm ۱$ می‌باشد با این ویژگی که $\ell_A^{e_A} \approx \ell_B^{e_B}$. همچنین یادآوری می‌کنیم که برای داشتن λ بیت امنیت پساکوانتومی لازم است تا اعداد اول مورداستفاده در طرح امضا به طول ۶λ بیت باشند(**؟؟**)، درنتیجه اندازه عامل‌های اعداد اول به‌صورت $\ell_A^{e_A} \approx \ell_B^{e_B} \approx ۲^{۳\lambda}$ خواهد بود. ازآنجا که در طرح امضای خود، خم‌های سوپرسینگولار را در میدان $\mathbb{F}_{p^{۲}}$ تعریف می‌کنیم درنتیجه اندازه عناصر میدان، ۱۲λ بیت طول خواهند داشت.

خم‌هایی که در طرح امضای خود استفاده می‌کنیم به فرم خم‌های مونت‌گومری $By^{۲} = x^{۳} + Ax^{۲} + x$ می‌باشد. از مزیت‌های خم مونت‌گومری می‌توان به محاسبات همسانی‌ها اشاره کرد که فقط به ضریب A نیاز می‌باشد. از طرف دیگر، یک نقطه روی خط کامر [۱۳] نیز می‌تواند بوسیله‌ی ضریب X اش نشان داده شود. با این اوصاف،برای نمایش هر عنصر میدان در

[12] Parameter Sizes

[13] Kummer line

خم به فرم مونت‌گومری و خط کامر، نیاز به ۱۲λ بیت می‌باشد.

فشرده‌سازی. آذردرخش و همکارانش در [۲] نشان داده‌اند که نقاط تابی(که مولد زیرگروه‌های تابی می‌باشند) می‌توانند بوسیله‌ی ضریب‌هایشان فشرده شوند. از آن‌جا که پیاده سازی این روش زیادی کند می‌باشد اخیرا کاستللو و همکارانش در [۴] یک الگوریتم جدیدتری ارائه داده‌اند که نسبت به روش آذردرخش هم سریع‌تر است و هم اندازه کلیدعمومی آن به نسبت طرح قبلی کوچکتر می‌باشد. در مورد اجرای این الگوریتم می‌توان گفت تقریبا برابر با اجرای یک مرحله از پروتکل اثبات‌دانش‌صفر می‌باشد.
همچنین در بخش قبلی اشاره شد که می‌توانیم زیرگروه تولید شده توسط یک نقطه تابی را تنها با یک مولد و ضریبش نشان دهیم. از آنجا که نقاط مولد زیرگروه ها در طرح ما عمومی می‌باشند درنتیجه می‌توانیم برای نمایش یک زیرگروه تابی تنها از یک ضریب برای اختصار استفاده کنیم، به عبارت دیگر:

$$R = mP_A + nQ_A \xrightarrow{m^{-1}} m^{-1}R = m^{-1}mP_A + m^{-1}nQ_A = P_A + m^{-1}nQ_A = P_A + kQ_A$$

با توجه به مطالب بالا برای نمایش R فقط لازم است که k را در اختیار داشته باشیم چون P_A و Q_A عمومی هستند.
در محاسبه ترکیبات خطی، برای فشرده‌سازی دو مولد یک گروه تابی، نیاز به سه ضریب می‌باشد که برای هر ضریب تقریبا ۳λ بیت نیاز می‌باشد.

۴.۴.۷.۱ فشرده‌سازی امضا

به دو روش می‌توانیم، طرح امضای خود را فشرده کنیم:

- فشرده‌سازی کلیدعمومی
- فشرده سازی پاسخ $\psi(S)$ زمانیکه در مرحله‌ای از الگوریتم امضا، $ch = ۱$ انتخاب شده باشد

لازم به ذکر است، کلیدخصوصی S و پاسخ $ch = ۰$ یعنی $(R, \phi(R))$ به دلیل آنکه با ضریب ۳λ

بیتی قابل نمایش‌اند، لذا نیازی به فشرده‌سازی ندارند.

- **کلیدعمومی.** از آنجا که از خم مونتگومری استفاده می‌کنیم بنابراین کلیدعمومی ما به فرم $pk = (a, x(P_B), x(Q_B), x(P_B - Q_B))$ می‌باشد که a بیانگر ضریب A در خم عمومی $E/\langle S\rangle$ می‌باشد. این چهار عنصر میدان به $(۱۲\lambda \times ۴ =)۴۸\lambda$ بیت برای نمایش نیاز دارند.

  کلیدعمومی را می‌توانیم با فشرده‌سازی نقاط تابی $(\phi(P_B), \phi(Q_B))$ ، که نیاز به سه ضریب ۳λ بیتی دارند را فشرده کنیم. به‌دلیل آنکه مختصات نقاط P_B و Q_B از طریق ضرایب فشرده‌شان قابل تولید می‌باشد بنابراین نیازی به ضریب X نقطه‌ی $\phi(P_B - Q_B)$ نمی‌باشد. بنابراین به‌طورکلی در کلیدعمومی برای نمایش خم، ۱۲λ بیت و برای مولدها نیز ۹λ بیت نیاز داریم که جمعا ۲۱λ بیت می‌شود.

- **کلیدخصوصی.** کلیدخصوصی S می‌تواند تنها با یک ضریب n که نیاز به ۳λ بیت می‌باشد ذخیره شود. دلیل این امر هم این است که کلیدخصوصی S از مرتبه‌ی $\ell_A^{e_A}$ می‌باشد و $S = P_A + [n]Q_A$.

- **امضا.** برای هر مرحله‌ی i ام از پروتکل اثبات‌دانش‌صفر، امضا شامل چندتایی $(com_i, ch_{i,j}, h_{i,j}, resp_{i,J_i})$ می‌باشد. بنابراین:

  - هر تعهد شامل دو خم $(E_۱, E_۲)$ می‌باشد که هر کدام از این خم‌ها به یک عنصر میدان که همان ضریب A می‌باشد، نیاز دارند.
  - یک بیت برای نمایش بیت چالشی $ch_{i,۰}$ نیازاست. البته قابل ذکر است که اگر مقدار $ch_{i,۰}$ را داشته باشیم نیازی به ارسال $ch_{i,۱}$ نمی‌باشد، دلیل این امر هم تساوی $ch_{i,۱} = ۱ - ch_{i,۰}$ می‌باشد.
  - چنانچه در **؟؟** توضیح داده شده است، برای هش $h_{i,j} = G(resp_{i,J_I})$ نیز به ۳λ بیت فضا نیاز می‌باشد.

    ذکر این نکته نیز لازم است که با $resp_{i,J_i}$ ، می‌توان $h_{i,j}$ را محاسبه کرد و بنابراین نیازی به ارسال این هش وجود ندارد.

- براساس بیت چالشی J_i جواب‌های متفاوتی خواهیم داشت و ازاین‌رو طول بیت متفاوتی نیز برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۰$ آنگاه پاسخ موردنظر $(R, \phi(R))$ خواهد بود که دراین صورت با توجه به وجود مولدهای عمومی، بدون هیچ هزینه محاسباتی نیاز به ۳λ بیت برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۱$ آنگاه پاسخ، $\psi(S)$ است که به ۱۲λ بیت به‌عنوان یک عنصر میدان لازم خواهد بود که با فشرده سازی به ۳λ بیت تقلیل می‌یابد.

در مجموع، برای هر مرحله از اثبات دانش صفر تقریبا به طورمتوسط به

$$۲۴\lambda + ۱ + ۳\lambda + \frac{۳\lambda + ۱۲\lambda}{۲} \approx ۳۴/۵\lambda$$

بیت فضا بدون فشرده سازی نیازاست که با فشرده‌سازی تقریبا به‌طور متوسط به

$$۲۴\lambda + ۱ + ۳\lambda + ۳\lambda \approx ۳۰\lambda$$

بیت نیاز خواهد بود.

اگرچه برای تامین λ بیت امنیت پساکوانتومی کفایت می‌کند تا پروتکل اثبات دانش صفر، λ بار تکرار شود اما به‌دلیل آنکه هش چالش‌ها دربرابر الگوریتم گراور [۱۱] آسیب‌پذیر نباشد(بخش ۵/۳)، لازم است که پروتکل امضا، ۲λ بار پروتکل اثبات دانش صفر را تکرار کند. با این اوصاف در کل، امضا تقریبا به‌طورمتوسط $(۳۴/۵\lambda \times ۲\lambda =)۶۹\lambda^۲$ بیت در حالت عادی و $۶۰\lambda^۲$ بیت درحالت فشرده‌سازی لازم دارد.
به‌عنوان مثال برای دست‌یابی به ۱۲۸ بیت امنیت پساکوانتومی(تعداد بیتی که در حالت پساکوانتومی ایمن باشد) برای طرح امضای ارائه شده، به‌طور متوسط به $۶۱۴۴ = ۴۸\lambda$ (۲۶۸۸ در حالت فشرده) بیت برای کلیدعمومی، $۳۸۴ = ۳\lambda$ بیت برای کلیدخصوصی و $۶۹\lambda^۲ = ۱,۱۳۰,۴۹۶$ (۱۲۲,۸۸۰ برای حالت فشرده) بیت برای امضا لازم است.

۲.۷.۴.۴ سنجش

دراین قسمت می‌خواهیم سایز پارامترهای لازم در طرح خود را با سایر طرح‌های امضای پساکوانتومی مقایسه می‌کنیم.

همان‌طور که از جدول زیر قابل مشاهده است، طرح امضای معرفی شده در این پایان‌نامه دربرابر سایر طرح‌های امضای پساکوانتومی موجود دارای کلید با طول سایز کوچکتر می‌باشد. البته قابل ذکر است که گونه‌هایی از طرح امضای مرکل وجود دارد که دارای طول کلید کوچکتری(۳۲ بیت) با همان درجه امنیت می‌باشد اما؟؟.

جدول ۱.۴: سنجش سایز پارامترها(به بایت) در طرح‌های امضاهای پساکوانتومی متفاوت در سطح امنیتی ۱۲۸ بیت کوانتومی

طرح امضا	سایز کلیدعمومی	سایز کلیدخصوصی	سایز امضا
هش مبنا	۱،۰۵۶	۱،۰۸۸	۴۱،۰۰۰
کد مبنا	۱۹۲،۱۹۲	۱،۴۰۰،۲۸۸	۳۷۰
مشبکه مبنا	۷،۱۶۸	۲،۰۴۸	۵،۱۲۰
حلقه مبنا	۷،۱۶۸	۴،۶۰۸	۳،۴۸۸
چندمتغیره مبنا	۹۹،۱۰۰	۷۴،۰۰۰	۴۲۴
همسانی مبنا	۷۶۸	۴۸	۱۴۱،۳۱۲
همسانی مبنای فشرده	۳۳۶	۴۸	۱۲۲،۸۸۰

۸.۴.۴ اندازه پارامتر

[14] همان‌طور که قبلا بررسی شد، اعداد اولی که برای ساخت همسانی‌ها از آن استفاده می‌کنیم به فرم $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm ۱$ می‌باشد با این ویژگی که $\ell_A^{e_A} \approx \ell_B^{e_B}$. همچنین یادآوری می‌کنیم که برای داشتن λ بیت امنیت پساکوانتومی لازم است تا اعداد اول مورداستفاده در طرح امضا به طول ۶λ بیت باشند(??)، درنتیجه اندازه عامل‌های اعداد اول به‌صورت $\ell_A^{e_A} \approx \ell_B^{e_B} \approx ۲^{۳\lambda}$ خواهد بود. ازآنجا که در طرح امضای خود، خم‌های سوپرسینگولار را در میدان $\mathbb{F}_{p^۲}$ تعریف می‌کنیم درنتیجه اندازه عناصر میدان، ۱۲λ بیت طول خواهند داشت.
خم‌هایی که در طرح امضای خود استفاده می‌کنیم به فرم خم‌های مونت‌گومری
$By^۲ = x^۳ + Ax^۲ + x$ می‌باشد. از مزیت‌های خم مونت‌گومری می‌توان به محاسبات همسانی‌ها اشاره کرد که فقط به ضریب A نیاز می‌باشد. از طرف دیگر، یک نقطه روی خط کامر [15] نیز می‌تواند بوسیله‌ی ضریب X اش نشان داده شود. با این اوصاف،برای نمایش هر عنصر میدان در خم به فرم مونت‌گومری و خط کامر، نیاز به ۱۲λ بیت می‌باشد.

فشرده‌سازی. آذردرخش و همکارانش در [۲] نشان داده‌اند که نقاط تابی(که مولد زیرگروه‌های تابی می‌باشند) می‌توانند بوسیله‌ی ضریب‌هایشان فشرده شوند. از آن‌جا که پیاده سازی این روش زیادی کند می‌باشد اخیرا کاستللو و همکارانش در [۴] یک الگوریتم جدیدتری ارائه داده‌اند که نسبت به روش آذردرخش هم سریع‌تر است و هم اندازه کلیدعمومی آن به نسبت طرح قبلی کوچکتر می‌باشد. در مورد اجرای این الگوریتم می‌توان گفت تقریبا برابر با اجرای یک مرحله از پروتکل اثبات‌دانش‌صفر می‌باشد.
همچنین در بخش قبلی اشاره شد که می‌توانیم زیرگروه تولید شده توسط یک نقطه تابی را تنها با یک مولد و ضریبش نشان دهیم. از آنجا که نقاط مولد زیرگروه ها در طرح ما عمومی می‌باشند درنتیجه می‌توانیم برای نمایش یک زیرگروه تابی تنها از یک ضریب برای اختصار استفاده کنیم، به عبارت دیگر:

$$R = mP_A+nQ_A \xrightarrow{m^{-۱}} m^{-۱}R = m^{-۱}mP_A+m^{-۱}nQ_A = P_A+m^{-۱}nQ_A = P_A+kQ_A$$

[14] Parameter Sizes

[15] Kummer line

با توجه به مطالب بالا برای نمایش R فقط لازم است که k را در اختیار داشته باشیم چون P_A و Q_A عمومی هستند.
در محاسبه ترکیبات خطی، برای فشرده‌سازی دو مولد یک گروه تابی، نیاز به سه ضریب می‌باشد که برای هر ضریب تقریبا ۳λ بیت نیاز می‌باشد.

۱.۸.۴.۴ فشرده‌سازی امضا

به دو روش می‌توانیم، طرح امضای خود را فشرده کنیم:

- فشرده‌سازی کلیدعمومی
- فشرده سازی پاسخ $\psi(S)$ زمانیکه در مرحله‌ای از الگوریتم امضا، $ch = ۱$ انتخاب شده باشد

لازم به ذکر است، کلیدخصوصی S و پاسخ $ch = ۰$ یعنی $(R, \phi(R))$ به دلیل آنکه با ضریب ۳λ بیتی قابل نمایش‌اند، لذا نیازی به فشرده‌سازی ندارند.

- **کلیدعمومی.** از آنجا که از خم مونتگومری استفاده می‌کنیم بنابراین کلیدعمومی ما به فرم $pk = (a, x(P_B), x(Q_B), x(P_B - Q_B))$ می‌باشد که a بیانگر ضریب A در خم عمومی $E/\langle S \rangle$ می‌باشد. این چهار عنصر میدان به $۴۸\lambda (= ۴ \times ۱۲\lambda)$ بیت برای نمایش نیاز دارند.
  کلیدعمومی را می‌توانیم با فشرده‌سازی نقاط تابی $(\phi(P_B), \phi(Q_B))$ ، که نیاز به سه ضریب ۳λ بیتی دارند را فشرده کنیم. به‌دلیل آنکه مختصات نقاط P_B و Q_B از طریق ضرایب فشرده‌شان قابل تولید می‌باشد بنابراین نیازی به ضریب X نقطه‌ی $\phi(P_B - Q_B)$ نمی‌باشد. بنابراین به‌طورکلی در کلیدعمومی برای نمایش خم، ۱۲λ بیت و برای مولدها نیز ۹λ بیت نیاز داریم که جمعا ۲۱λ بیت می‌شود.
- **کلیدخصوصی.** کلیدخصوصی S می‌تواند تنها با یک ضریب n که نیاز به ۳λ بیت می‌باشد ذخیره شود. دلیل این امر هم این است که کلیدخصوصی S از مرتبه‌ی $\ell_A^{e_A}$ می‌باشد و $S = P_A + [n]Q_A$.

- **امضا.** برای هر مرحله‌ی i ام از پروتکل اثبات‌دانش‌صفر، امضا شامل چندتایی $(com_i, ch_{i,j}, h_{i,j}, resp_{i,J_i})$ می‌باشد. بنابراین:

– هر تعهد شامل دو خم $(E_۱, E_۲)$ می‌باشد که هر کدام از این خم‌ها به یک عنصر میدان که همان ضریب A می‌باشد، نیاز دارند.

– یک بیت برای نمایش بیت چالشی $ch_{i,۰}$ نیازاست. البته قابل ذکر است که اگر مقدار $ch_{i,۰}$ را داشته باشیم نیازی به ارسال $ch_{i,۱}$ نمی‌باشد، دلیل این امر هم تساوی $ch_{i,۱} = ۱ - ch_{i,۰}$ می‌باشد.

– چنانچه در **??** توضیح داده شده است، برای هش $h_{i,j} = G(resp_{i,J_I})$ نیز به ۳λ بیت فضا نیاز می‌باشد.
ذکر این نکته نیز لازم است که با $resp_{i,J_i}$ ، می‌توان $h_{i,j}$ را محاسبه کرد و بنابراین نیازی به ارسال این هش وجود ندارد.

– براساس بیت چالشی J_i جواب‌های متفاوتی خواهیم داشت و ازاین‌رو طول بیت متفاوتی نیز برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۰$ آنگاه پاسخ موردنظر $(R, \phi(R))$ خواهد بود که دراین صورت با توجه به وجود مولدهای عمومی، بدون هیچ هزینه محاسباتی نیاز به ۳λ بیت برای ذخیره‌سازی لازم خواهد بود. اگر $J_i = ۱$ آنگاه پاسخ، $\psi(S)$ است که به ۱۲λ بیت به‌عنوان یک عنصر میدان لازم خواهد بود که با فشرده سازی به ۳λ بیت تقلیل می‌یابد.

در مجموع، برای هر مرحله از اثبات دانش صفر تقریبا به طورمتوسط به

$$۲۴\lambda + ۱ + ۳\lambda + \frac{۳\lambda + ۱۲\lambda}{۲} \approx ۳۴/۵\lambda$$

بیت فضا بدون فشرده سازی نیازاست که با فشرده‌سازی تقریبا به‌طور متوسط به

$$۲۴\lambda + ۱ + ۳\lambda + ۳\lambda \approx ۳۰\lambda$$

بیت نیاز خواهد بود.

اگرچه برای تامین λ بیت امنیت پساکوانتومی کفایت می‌کند تا پروتکل اثبات دانش صفر، λ بار تکرار شود اما به‌دلیل آنکه هش چالش‌ها دربرابر الگوریتم گراور [۱۱] آسیب‌پذیر نباشد(بخش ۵/۳)، لازم است که پروتکل امضا، ۲λ بار پروتکل اثبات دانش صفر را تکرار کند. با این اوصاف در کل، امضا تقریبا به‌طورمتوسط $(۳۴/۵\lambda \times ۲\lambda =)۶۹\lambda^۲$ بیت در حالت عادی و $۶۰\lambda^۲$ بیت درحالت فشرده‌سازی لازم دارد.
به‌عنوان مثال برای دست‌یابی به ۱۲۸ بیت امنیت پساکوانتومی(تعداد بیتی که در حالت پساکوانتومی ایمن باشد) برای طرح امضای ارائه شده، به‌طور متوسط به $۶۱۴۴ = ۴۸\lambda$ (۲۶۸۸ در حالت فشرده) بیت برای کلیدعمومی، $۳۸۴ = ۳\lambda$ بیت برای کلیدخصوصی و
$۱,۱۳۰,۴۹۶ = ۶۹\lambda^۲$ (۱۲۲,۸۸۰ برای حالت فشرده) بیت برای امضا لازم است.

۲.۸.۴.۴ سنجش

دراین قسمت می‌خواهیم سایز پارامترهای لازم در طرح خود را با سایر طرح‌های امضای پساکوانتومی مقایسه می‌کنیم.
همان‌طور که از جدول زیر قابل مشاهده است، طرح امضای معرفی شده در این پایان‌نامه دربرابر سایر طرح‌های امضای پساکوانتومی موجود دارای کلید با طول سایز کوچکتر می‌باشد. البته قابل ذکر است که گونه‌هایی از طرح امضای مرکل وجود دارد که دارای طول کلید کوچکتری(۳۲ بیت) با همان درجه امنیت می‌باشد اما؟؟.

جدول ۲.۴: سنجش سایز پارامترها(به بایت) در طرح‌های امضاهای پساکوانتومی متفاوت در سطح امنیتی ۱۲۸ بیت کوانتومی

طرح امضا	سایز کلیدعمومی	سایز کلیدخصوصی	سایز امضا
هش مبنا	۱،۰۵۶	۱،۰۸۸	۴۱،۰۰۰
کد مبنا	۱۹۲،۱۹۲	۱،۴۰۰،۲۸۸	۳۷۰
مشبکه مبنا	۷،۱۶۸	۲،۰۴۸	۵،۱۲۰
حلقه مبنا	۷،۱۶۸	۴،۶۰۸	۳،۴۸۸
چندمتغیره مبنا	۹۹،۱۰۰	۷۴،۰۰۰	۴۲۴
همسانی مبنا	۷۶۸	۴۸	۱۴۱،۳۱۲
همسانی مبنای فشرده	۳۳۶	۴۸	۱۲۲،۸۸۰

کتاب‌نامه

[1] 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.

[2] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel, and Christopher Leonardi. Key compression for isogeny-based cryptosystems. In *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography*, pages 1–10. ACM, 2016.

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

[4] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. Efficient compression of sidh public keys. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 679–706. Springer, 2017.

[5] Craig Costello and Benjamin Smith. Montgomery curves and their arithmetic. *Journal of Cryptographic Engineering*, 8(3):227–240, 2018.

[6] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

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

[8] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[9] 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.

[10] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728, 1991.

[11] Lov K Grover. A fast quantum mechanical algorithm for database search. *arXiv preprint quant-ph/9605043*, 1996.

[12] 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.

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

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

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

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

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

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

[19] 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.

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