



## فاز اول (بیت کوین چگونه کار می کند؟)

بیت کوین<sup>۱</sup> یک واحد پول دیجیتالی است که در سال 2009 توسط ساتوشی ناکاموتو معرفی شد. بیت کوین نوعی پول الکترونیکی است که به کاربران اجازه می دهد تا دارایی خود را از حسابی به حساب دیگر منتقل کنند؛ بدون اینکه نیاز به مداخله موسسات مالی و اعتباری مانند بانکها وجود داشته باشد.

در سیستم های سنتی، موسسات مالی به عنوان محلی امن برای ذخیره و نگهداری دارایی افراد محسوب می شوند. در این سیستم ها، مدیریت انتقال دارایی و پول بین مشتری ها و سرویس گیرنده ها برعهده بانک است. سیستم سنتی مدیریت دارایی معایبی دارد از جمله اینکه انتقال الکترونیکی موجودی ها و تراکنش ها بین شعب مختلف بانکها هزینه<sup>۲</sup> زیادی را تحمیل می کند. این تراکنش علاوه بر هزینه فرآیندی زمان بر است. از همه مهم تر انتقال تراکنش ها نیاز به بستری امن دارد تا اطلاعات توسط افراد ناشناس سرقت نشود.

سایر سیستم های مدیریت دارایی مانند Visa، MasterCard و PayPal نیز هزینه زیادی را برای انتقال پول تحمیل می کنند. در مقابل بیت کوین به عنوان یک سیستم انتقال پول و ارز، واسطه ها را حذف کرده است و در عوض تراکنش ها را در یک شبکه نظیر به نظیر<sup>۳</sup> انجام می دهد. در شبکه بیت کوین به جای استفاده از ابزارهای کمکی<sup>۴</sup> برای امن کردن مسیر انتقال، از رمزنگاری استفاده می شود. برای ساخت و تصدیق امضای دیجیتال که کاربران برای انجام تراکنش های خود از آن استفاده می کنند، از رمزنگاری مبتنی بر کلید عمومی و خصوصی استفاده می شود. این رمزنگاری اساس کار شبکه بیت کوین را تشکیل می دهد.

انتقال اطلاعات بین گیرنده و فرستنده نهایی بیت کوین در شبکه، زنجیره ای از بلاک ها را تشکیل می دهد که هر بلاک شامل رکورد انتقال بیت کوین مابین دو آدرس مشخص در شبکه است. هر بلاک قبل پیوستن به زنجیره نیاز به تصدیق دارد. در این زنجیره اگر یکی از بلاک ها دستکاری یا دچار تغییر شود، تمامی بلاک های بعد نیاز به تصدیق مجدد دارند. نکته دیگر اینکه زمانی که بلاک تراکنش بین دو کاربر به زنجیره بلاک متصل می شود، گیرنده مطمئن است که این تراکنش توسط تمامی کامپیوترهای شبکه رکورد شده است. این باعث می شود تا فرستنده نتواند بیت کوین مشابه را مجدداً<sup>۵</sup> برای کاربر دیگری ارسال کند.

در شبکه بیت کوین (مجموعه ای از سیستم ها (Bitcoin Clients) که نرم افزار بیت کوین را اجرا می کنند و به اینترنت متصل هستند) هر بلاک قبل از پیوستن به زنجیره، نیاز به تأیید و تصدیق دارد تا از اعتبار بلاک

<sup>1</sup> Bitcoin

<sup>2</sup> Transaction Fee

<sup>3</sup> Peer-to-Peer

<sup>4</sup> Third Parity

<sup>5</sup> Double spending



اطمینان حاصل شود که به این فرآیند Proof of Work می‌گویند. این فرآیند نیاز به زمان محاسبات زیادی دارد. شبکه بیت‌کوین برای فرآیند Proof of Work از تابع SUDO-SHA-256 استفاده می‌کند. ضمناً لازم به ذکر است که هر کلاینت در شبکه بیت‌کوین علاوه بر اینکه می‌تواند بلاک جدیدی در شبکه تولید کند، می‌تواند عملیات Proof of Work را بر روی بلاک‌های موجود در شبکه بیت‌کوین انجام دهد. نکته دیگر آنکه هر کلاینت در شبکه تمامی بلاک‌های تأیید شده را نگهداری می‌کند (دانلود و اجرای نرم افزار بیت‌کوین به معنی پیوستن شما به شبکه بیت‌کوین بوده و وجود این بلاک‌های تأیید شده بر روی کلاینت را نشان می‌دهد).

حاصل فرآیند Proof of Work استخراج<sup>۶</sup> نامیده می‌شود که در نتیجه آن تراکنش جدیدی به زنجیره بلاک افزوده می‌شود. همه کلاینت‌ها در شبکه، اطلاعات مربوط به تراکنش‌های جدید را به اشتراک می‌گذارند، یعنی اگر تراکنش جدید رخ دهد هر کلاینت در شبکه این تراکنش را بر روی سیستم خود دریافت و نگهداری می‌کند. در این هنگام هر کلاینت این اختیار را دارد تا به استخراج بیت‌کوین بپردازد و تراکنش‌های جدید را به زنجیره بلاک اضافه کند. همانطور که قبلاً گفته شد، اضافه کردن تراکنش، به لیست تراکنش‌های تأیید شده (زنجیره بلاک) نیاز به قدرت پردازشی زیادی دارد که در ادامه دلیل این نیاز ذکر می‌شود. اما بعد از اضافه کردن بلاک به زنجیره، به عنوان پاداش 25 بیت‌کوین به شخص برنده اهدا می‌شود، که موجب افزایش دارایی شخص می‌شود و نام استخراج (مشابه استخراج طلا از معدن) به همین دلیل اقتباس شده است.

تابع رمزنگاری sudo-SHA-256 (در این تابع با داشتن ورودی می‌توان به خروجی دست یافت ولی با داشتن خروجی به دست آوردن ورودی تقریباً غیرممکن است) یک پیام با طول متغیر (به عنوان ورودی) را به یک رشته با طول ثابت (به عنوان خروجی) نگاشت می‌کند (پیام با طول متغیر نشان دهنده تراکنشی است که در نظر داریم تا با فرآیند Hash آن را به زنجیره اضافه کنیم و پیام با طول ثابت تراکنشی است که باید به زنجیره اضافه شود). خروجی الگوریتم sudo-SHA-256 یک پیام 256 بیتی است.

این الگوریتم شامل سه مرحله است:

### مرحله اول: چسباندن و تجزیه<sup>۷</sup>

به پیام خود یک عدد '1' و به تعداد کافی '0' اضافه می‌کنیم به طوری که طول کل پیام به پیمانه 512 برابر 448 گردد ( $L + 1 + k = 448 \bmod 512$ ). که  $L$  طول پیام اصلی و  $k$  تعداد صفرهای اضافه شده است. فرض کنید که می‌خواهیم پیام "abc" را رمزنگاری کنیم.

<sup>۶</sup> Mining

<sup>۷</sup> Padding and Parsing



تحویل مطابق جدول زمانبندی

طول پیام برحسب بیت (با فرض اینکه هر کاراکتر 8 بیت است) برابر است با 24 بیت. بیت '1' را به انتهای 24 بیت اضافه می‌کنیم و به تعداد 423 بیت '0' و در انتها عدد معادل  $L$  را به صورت یک عدد باینری 64 بیتی به حاصل قبلی می‌افزاییم. حال خروجی مرحله اول به صورت زیر است، عدد باینری حاصل را به بلاک‌های 512 بیتی ( $M^{(1)}, M^{(2)}, \dots, M^{(N)}$ ) تقسیم می‌کنیم. (در این مثال به دلیل کوتاه بودن طول پیام فقط یک بلاک ایجاد می‌شود ولی اگر طول پیام بزرگ باشد تعداد بلاک‌ها می‌تواند بیشتر شود).

$$L = 3 \times 8 = 24$$

$$L + 1 + k = 448 \Rightarrow k = 423 \text{ bit}$$

01100001	01100010	01100011	1	0000....0000	0000....11000
a, 8bit	b, 8bit	c, 8bit	1 bit pad	423 bit pad	L in binary

### مرحله دوم: گسترش پیام<sup>۸</sup>

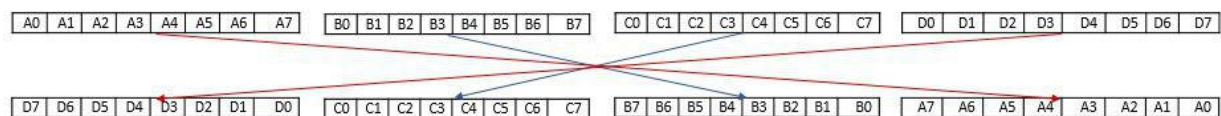
در این مرحله هر بلاک 512 بیتی حاصل از مرحله قبل به 16 بلاک 32 بیتی ( $M_t^{(i)}$  for  $0 < t < 15$ ) تقسیم می‌شود. در ادامه هر بلاک 512 بیتی به 64 بلاک 32 بیتی ( $W_t$ ) به صورت زیر بسط می‌یابد:

$$W_t = \begin{cases} M_t^{(i)} & 0 < t < 15 \\ \sigma_1(W_{t-1}) + W_{t-6} + \sigma_0(W_{t-12}) + W_{t-18} & 16 < t < 63 \end{cases}$$

$$\sigma_0(x) = ROT_{17}(x) \oplus ROT_{14}(x) \oplus SHF_{12}(x)$$

$$\sigma_1(x) = ROT_9(x) \oplus ROT_{19}(x) \oplus SHF_9(x)$$

در فرمول‌های فوق  $ROT_n(x)$  عملیات  $n$  مرتبه شیفت دورانی به راست است و  $SHF_n(x)$  عملیات  $n$  مرتبه شیفت منطقی به راست  $x$  است و عملیات  $+$  نشان دهنده جمع باینری 32 بیتی است. در مرحله بعد هر بلاک 32 بیتی از یک تابع جایگشت<sup>۹</sup> عبور داده می‌شود. دیاگرام زیر نشان‌دهنده این تابع است.



### مرحله سوم: فشرده‌سازی<sup>۱۰</sup>

در این مرحله  $W_i$ ‌های حاصل از مرحله قبل به عنوان ورودی این مرحله محسوب می‌شوند. تابع فشرده‌سازی دارای 8 متغیر 32 بیتی  $A \dots H$  است که با 32 بیت اول بخش اعشاری ریشه دوم 8 پرایم اول ( $H_0^{(0)} - H_7^{(0)}$ )

<sup>8</sup> Expansion

<sup>9</sup> Permutation Box

<sup>10</sup> Compression



( در شروع هر بار فراخوانی تابع فشرده‌سازی مقداردهی اولیه می‌شوند. در ادامه تابع فشرده‌سازی 64 بار تکرار می‌شود و به صورت زیر مشخص می‌شود (نمایش مقادیر اولیه A...H بصورت هگزادسیمال است):

$$H_0^{(0)} = 6a09e667 - H_1^{(0)} = bb67ae85 - H_2^{(0)} = 3c6ef372 - H_3^{(0)} = a54ff53a$$

$$H_4^{(0)} = 510e527f - H_5^{(0)} = 9b05688c - H_6^{(0)} = 1f83d9ab - H_7^{(0)} = 5be0cd19$$

$$T_2 = H + \sum_1 (E) + Ch(E, F, G) + K_t + W_t$$

$$T_1 = \sum_0 (A) + Maj(A, B, C) + \sum_2 (C + D)$$

$$H = G$$

$$F = E$$

$$D = C$$

$$B = A$$

$$G = F$$

$$E = D + T_1$$

$$C = B$$

$$A = 3T_1 - T_2$$

که در این روابط داریم:

$$Ch(x, y, z) = (x \text{ AND } y) \oplus (\overline{y} \text{ AND } z) \oplus (\overline{x} \text{ AND } z)$$

$$Maj(x, y, z) = (x \text{ AND } z) \oplus (x \text{ AND } y) \oplus (y \text{ AND } z)$$

$$\sum_0(x) = ROT_2(x) \oplus ROT_{13}(x) \oplus ROT_{22}(x) \oplus SHF_7(x)$$

$$\sum_1(x) = ROT_6(x) \oplus ROT_{11}(x) \oplus ROT_{25}(x)$$

$$\sum_2(x) = ROT_2(x) \oplus ROT_3(x) \oplus ROT_{15}(x) \oplus SHF_5(x)$$

ورودی‌های  $K_t$ ، 64 ثابت 32 بیتی هستند که با 32 بیت اول بخش کسری ریشه سوم 64 عدد اول مقداردهی اولیه می‌کنند. بعد از 64 مرتبه تکرار تابع فشرده‌سازی مقادیر میانی  $H^{(i)}$  بصورت زیر محاسبه می‌شوند:

$$H_0^{(i)} = A + H_0^{(i-1)} - H_1^{(i)} = B + H_1^{(i-1)} - H_2^{(i)} = C + H_2^{(i-1)} - H_3^{(i)} = D + H_3^{(i-1)}$$

$$H_4^{(i)} = E + H_4^{(i-1)} - H_5^{(i)} = F + H_5^{(i-1)} - H_6^{(i)} = G + H_6^{(i-1)} - H_7^{(i)} = H + H_7^{(i-1)}$$

و مقادیر  $K_t$  (با نمایش هگزادسیمال) بصورت زیر مقدار دهی اولیه می‌شوند:



تحويل مطابق جدول زمانبندی

428a2f98-71374491-b5c0fbcf-e9b5dba5-3956c25b-59f111f1-923f82a4-ab1c5ed5  
d807aa98-12835b01-243185be-550c7dc3-72be5d74-80deb1fe-9bdc06a7-c19bf174  
e49b69c1-efbe4786-0fc19dc6-240ca1cc-2de92c6f-4a7484aa-5cb0a9dc-76f988da  
983e5152-a831c66d-b00327c8-bf597fc7-c6e00bf3-d5a79147-06ca6351-14292967  
27b70a85-2e1b2138-4d2c6dfc-53380d13-650a7354-766a0abb-81c2c92e-92722c85  
a2bfe8a1-a81a664b-c24b8b70-c76c51a3-d192e819-d6990624-f40e3585-106aa070  
19a4c116-1e376c08-2748774c-34b0bcb5-391c0cb3-4ed8aa4a-5b9cca4f-682e6ff3  
748f82ee-78a5636f-84c87814-8cc70208-90befffa-a4506ceb-be49a3f7-c67178f2

الگوریتم فشرده سازی sudo-SHA-256 بر روی تمامی بلاک های 512 بیتی دیگر تکرار می شود. 256 بیت خروجی ( $H^{(N)}$ ) با پشت سرهم قرار دادن (Concatenate) مقادیر زیر حاصل می شود:

$$H^{(N)} = H_0^{(N)} \& H_1^{(N)} \& H_2^{(N)} \& H_3^{(N)} \& H_4^{(N)} \& H_5^{(N)} \& H_6^{(N)} \& H_7^{(N)}$$

هر بلاک در زنجیره دارای سرآیند است که اطلاعاتی را در مورد بلاک می دهد. بخش های مختلف سرآیند در شکل 1 نشان داده شده است و در جدول نیز بخش های مختلف به تفکیک معرفی شده است.

Version	hashPrevBlock	hashMerkleRoot	Time	Difficulty	nonce
4 byte	32 byte	32 byte	4 byte	4 byte	4 byte

شکل 1. ساختار بلاک سرآیند

بخش	هدف	زمان بروزرسانی	اندازه (بایت)
Version	شماره نسخه هر بلاک	زمان ارتقاء نرم افزار	4
hashPrevBlock	256 بیت hash مربوط به بلاک قبلی	وقتی که بلاک جدید وارد می شود	32
hashMerkleRoot	256 بیت hash بر اساس همه تراکنش های درون بلاک	وقتی که تراکنش پذیرفته می شود	32
Time	زمان حال بر حسب ثانیه از سال 1970	هر چند ثانیه	4
Difficulty	هدف فعلی <sup>11</sup> در فرمت فشرده	زمان تعیین difficulty	4
Nonce	عدد 32 بیتی با شروع از مقدار صفر	زمان اجرای الگوریتم	4

شبه کد مربوط به الگوریتم فشرده سازی به صورت زیر است:

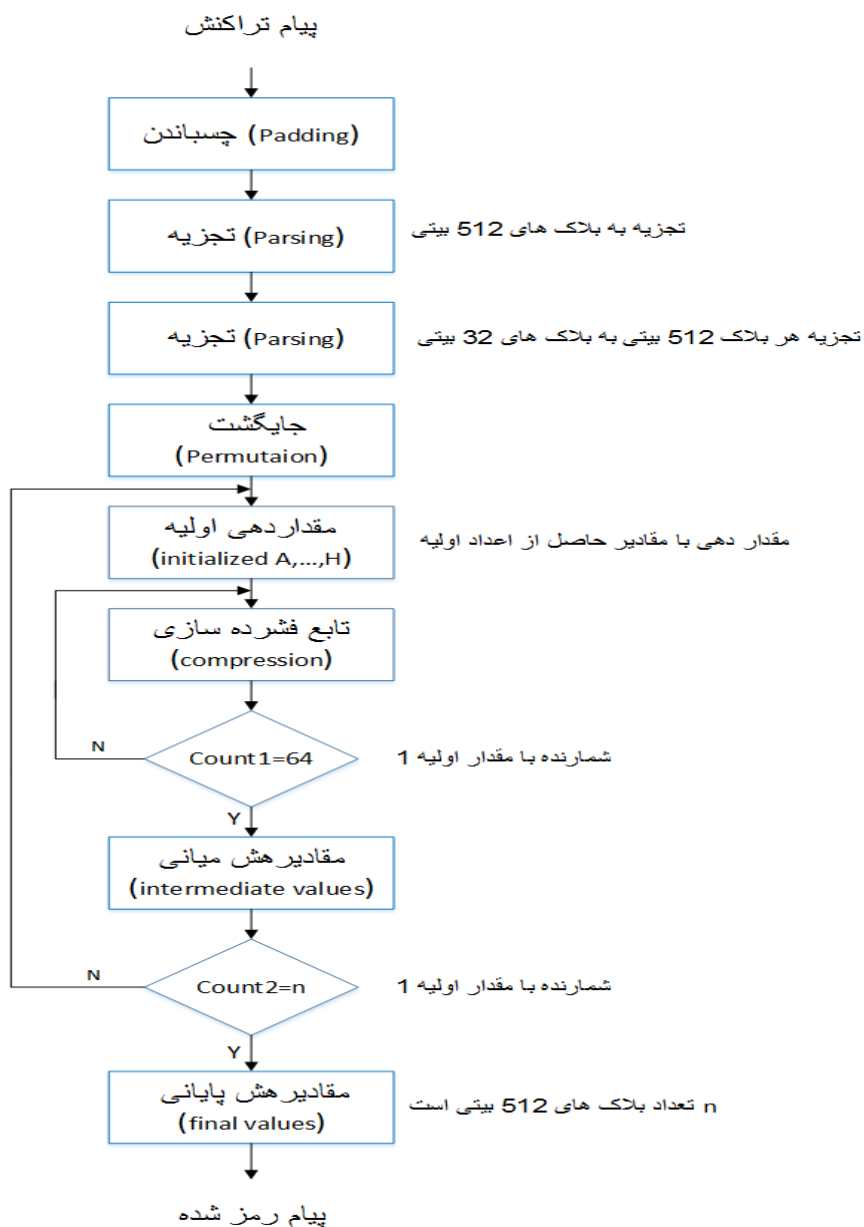
<sup>11</sup> Current Target



Nonce: increment sequentially



تحويل مطابق جدول زمانبندی



شکل 2. مراحل اجرای الگوریتم sudo-SHA-256











## کلیات فاز چهارم (طراحی توأمان)

هنگامی که بیت‌کوین برای اولین بار معرفی شد، واحد پردازش مرکزی (CPU) از اینتل و AMD به عنوان استخراج‌گر استفاده شدند، اما آن‌ها به سرعت توسط واحد پردازش گرافیکی (GPU) از NVIDIA و AMD جایگزین شدند. CPUها تعداد نسبتاً کمی از واحد منطقی محاسباتی (ALU) دارند و برای اجرای نرم‌افزارها و تصمیم‌گیری‌های عمومی طراحی شده‌اند. GPUها توانایی انجام بسیاری از کارهای تکراری را دارند، زیرا تعداد زیادی ALU برای انجام عملیات‌های ریاضی در آن‌ها در نظر گرفته شده است. این ALUهای مشابه را می‌توان بارها و بارها برای اجرای Hashهای مختلف استفاده کرد؛ زیرا تعداد ALUها تأثیر مستقیم بر خروجی Hash دارد. پیکربندی FPGAها جهت اجرای با بهره‌وری بیشتر الگوریتم sudo-SHA256 می‌شود.

در این فاز نیاز است تا وظایف مربوط به Checker که شرط حلقه را بررسی می‌کند و بخش مربوط به تعیین Block Header همچنان بر عهده نرم‌افزار باقی بماند و تنها پیاده‌سازی تابع SUDO-SHA-256 را به سخت‌افزار واگذار کنیم.





تحويل مطابق جدول زمانبندی

شرایط و زمانبندی تحويل پروژه به شرح جدول زیر است:

تاریخ	فاز
1397/3/13	فاز اول
1397/3/21	فاز دوم
1397/4/17	فاز سوم
1397/4/17	تحويل حضوری

موارد تحويلی در تمامی فازها:

1. فایل کامل پروژه شامل تمامی کدهای VHDL
2. شکل موجهای شبیهسازی
3. برنامه محک جهت تست طراحی
4. گزارش سنتز شامل منابع مصرفی، توان مصرفی و فرکانس کاری مدار طراحی شده
5. انجام پروژه در گروههای حداکثر دو نفره مجاز است. انجام پروژه به صورت تک نفره نمره اضافه نخواهد داشت.
- نکته: در صورتی که حجم فایلهای شما بیشتر از مقدار مجاز سایت درس می باشد (20 مگابایت)، پروژه خود را در گوگل درایو، دراپ باکس یا وان درایو آپلود کنید و لینک اشتراک آن را در سایت درس بارگذاری کنید. پس از ارسال لینک فایل مذکور را تحت هیچ شرایطی ویرایش نکنید. در غیر این صورت نمره آن فاز صفر در نظر گرفته می شود.
6. زمانبندی ارائههای حضوری متعاقباً اعلام خواهد شد. تحويل حضوری فقط در روز اعلام شده خواهد بود و پس از آن به هیچ عنوان پروژه تحويل گرفته نخواهد شد.
7. در صورت عدم حضور در تحويل حضوری هیچ نمره ای به پروژه شما تعلق نخواهد گرفت.

[j.talafy@aut.ac.ir](mailto:j.talafy@aut.ac.ir)

[hanie.ghasemy@gmail.com](mailto:hanie.ghasemy@gmail.com)

موفق باشید