پروژه ی نرم افزار دوره کارشناسی

گزارش شماره ۲ - مفاهیم (بلاک ها)

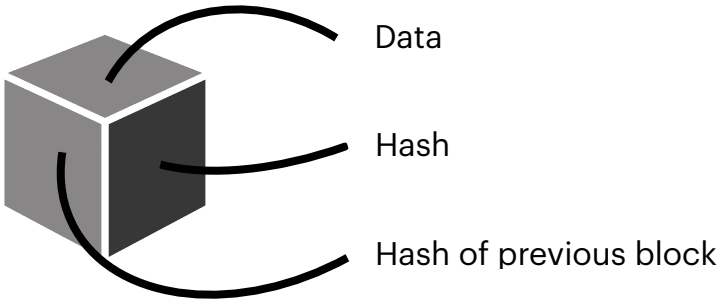
آریا رادمهر - ۹۷۴۶۳۱۲۵

دکتر سجاد حق زاد کلیدبری

November 9, 2021

مفاهیم بلاک ها

حال که با تعریف بلاکچین آشنا شدیم میتوانیم به صورت دقیق تر به اجزای تشکیل دهنده ی بلاکچین نگاه کنیم. به صورت ساده هر بلاک از ۳ بخش اصلی ۱- داده ۲- هش بلاک و ۳- هش بلاک قبل تشکیل شده است.



داده

داده ها در بلاکچین ها همواره با توجه به نوع و ساختار آن بلاکچین متفاوت هستند. برای مثال؛ بلاکچین های رمز ارز ها اطلاعات تراکنش ها را درخود به عنوان داده ذخیره میکنند. این اطلاعات شامل اطلاعات فرستنده ی رمز ارز، گیرنده ی رمز ارز و مقدار سکه ی مبادله شده میباشد. این داده ها در نهایت توسط فرآیند هش ، هش بلاک را بوجود می آورند.

هش بلاک

فرآیند هش از همان ابتدا جوابی برای سوالات امنیتی بلاک چین ها بود. برای مثال فرآیند تغییر ناپذیری بلاک ها و دستکاری نکردن اطلاعات بلاک ها در بلاکچین توسط هش محقق میشود. هش برای بلاک ها به مانند اثر انگشت برای آدمهاست. این جزء همواره هویت بلاک و کل محتویات آن را تعیین میکند و با توجه به الگوریتمی که دارد همواره منحصر به فرد میباشد. و اما خود هش چیست و چگونه تولید میشود.

هش به طور کلی یک تابع ریاضی است که یک ورودی میگیرد و در ازای آن ورودی خاص یک خروجی خاص را برمیگرداند. یعنی در تابع f به صورت $f(x) = y$ ، همواره در ازای ورودی x به تابع هش f، y بازگردانده میشود. به عنوان مثال، فرض کنید تابع هش f به صورتی است که به ازای دریافت هر عدد، مقادیر آن عدد را با همدیگر جمع زده و یکان حاصل جمع را بازمیگرداند.

f(142) = 1+4+2 = 7 -> hash(142) by hash function f = 7

f(237) = 2+3+7 = 12 -> hash(237) by hash function f = 2

در مثال فوق f یک تابع هش به صورت ساده میباشد، اما در دنیای کامپیوتر ها و بلاکچین ها توابع هش میبایست چند ویژگی خاص را دارا باشند.

۱- توابع هش باید ورودی با هر اندازه و طولی را دریافت کنند.

۲- توابع هش باید یک خروجی با اندازه و طول ثابت داشته باشند.

۳- خروجی توابع هش باید توسط کامپیوتر ها در زمان نسبتا سریع و قابل قبولی (برای مثال O(n)) حساب شوند.

حال بنظر میرسد که تابع هش f که در بالا ساختیم تمام ویژگی های نام برده شده را داراست اما، نکات و نواقصی در تابع هش f جامانده که با ذکر مثال هویدا میشوند؛ برای مثال بگذارید تا عدد 601 را به تابع f بدهیم تا هش آن محاسبه شود. با توجه به ساختار تابه هش f، جواب عدد 7 میباشد. حال دوباره این سوال را از خود میپرسیم که مگر هش در بلاک ها همواره منحصر به فرد نبودند؟ پس تابع هش f به قدر کافی پاسخگوی نیاز های ما نمیباشد.

نکته ی دیگری که حائز اهمیت است که عدد 7 به طرز غیر قابل چشم پوشی جواب بسیاری از اعدادی است که ما حتی از آنها مطلع نیستیم. این باعث میشود تا متوجه نقض دیگر تابع هش f بشویم.

آخرین نکته ی مهم نیز در تابع هش f این است که ما براحتی میتوانیم با داشتن عدد 7 به عنوان هش، به سرعت اعدادی را تست کنیم تا در نهایت مثالی را به عنوان ورودی به تابع هش f در نظر بگیریم که با این جواب مطابقت دارد.

از سه نکته ی بالا به این نتیجه دست می یابیم که توابع هش به خودی خود پاسخگوی امنیت داده های ما در بلاک ها نمیباشند. در اینجا میتوانیم به سراغ هش های کریپتوگرافیک برویم که دقیقا سه نقص بالا را برطرف کرده و ویژگی زیر را دارند.

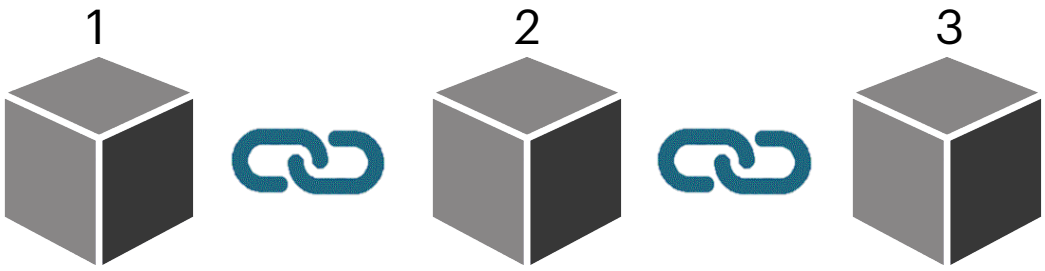
۱- هش های کریپتوگرافیک در برابر برخورد مقاوم هستند و collision resistance دارند. این بدان معناست که ورودی های متفاوت دارای خروجی های یکسان نیستند و یا این مقدار از برخورد بقدری کم و نادر است که قابل چشم پوشیست.

۲- هش های کریپتوگرافیک یک طرفه میباشند و one way هستند. بدان معنا که هیچوقت با دانستن هش یک ورودی، نمیتوان خود ورودی را حدس زد یا بدست آورد.

۳- هش های کریپتوگرافیک هش هایی هستند که امکان bruteforce و تست کردن مقادیر ورودی با سرعت بالا بر روی آنها وجود ندارد و این قابلیت puzzle friendliness هش های کریپتوگرافیک میباشد تا عملا کامپیوتر ها با تست کردن سریع تعداد بسیار زیادی از ورودی ها به خروجی هش مورد نظر برسند.

هش بلاک قبل

عنصر سومی که در بلاک ذخیره میشود هش بلاک قبل است. بگذارید تا با مثالی آنرا شرح دهیم.



Genesis block

Hash:	3Z8F	Hash:	6BQ1	Hash:	3H4Q
Previous hash:	0000	Previous hash:	3Z8F	Previous hash:	6BQ1

در این شکل یک زنجیره از سه بلاک وجود دارد. همانطور که در تصویر مشاهده میکنیم هر بلاک شامل یک هش و هش بلاک قبلی میباشد. از این رو بلاک شماره ۳ به بلاک شماره ۲، و بلاک شماره ۲ به بلاک شماره ۱ مربوط است. حال بلاک شماره ۱ کمی با دیگر بلاک ها متفاوت است چون بلاکی قبل از آن وجود ندارد و به اصطلاح به آن بلاک اولیه یا genesis گفته میشود.

حال فرض کنیم که بلاک دوم دستکاری بشود. با ایجاد تغییرات در بلاک شماره ۲، هش آن بلاک نیز متقابلا تغییر میکند و هیچکدام از بلاک های بعدی معتبر نیستند، زیرا دیگر هشی که از بلاک قبل خود به دست دارند معتبر نیست. از این رو اگر بلاکی کوچکترین تغییری بکند، تمام بلاک های بعد از آن نیز نامعتبر میشوند. این سیستم است که باعث شکل گیری یک زنجیره در بلاک ها میشود.

امضا ها و امضا های دیجیتال

استفاده از هش تنها ایده ی نبوغ آمیز در ایجاد و بهبود امنیت و اعتبار بخشیدن به بلاک ها نبود. استفاده از سیستم امضای دیجیتال ایده ی دیگری که به سراغ بلاکچین آمد تا در کنار فرآیند هش راهی برای تشخیص صحت و درستی داده های بلاک ها باشد.

فرض کنید که در بخش نظارت بر کیفیت یک شرکت تولید حافظه رم هستید. شما حافظه رم های تولید شده را میگیرید و در صورت کارکرد صحیح هر کدام از آنها، بارکدی را در جعبه آن میزنید که آن بارکد نمایانگر تایید کیفیت آن حافظه رم توسط شرکت شما میباشد. درج بارکد به صورتی است که تنها شما قادر به تولید آن هستید اما هر کسی میتواند با اسکن بارکد از صحت اصل بودن و کارکرد درست آن مطلع شود. در حقیقت این سیستمی است که باعث میشود کالا های شما از دیگر کالاهای نامعتبر مجزا شود.

این سیستم در دنیای دیجیتال به گونه ای توسعه یافت که با یک جفت کلید خصوصی و عمومی میتوان مدارک، اسناد یا داده ها را امضا کرد. کلید خصوصی یا secret key و کلید عمومی یا public key به این صورت زیر عمل میکنند.

۱- کلید خصوصی همانطور که از اسمش پیداست تنها برای شماست و کسی نباید اطلاعات آن را داشته باشد.

۲- کلید عمومی نیز بر خلاف کلید خصوصی برای عموم قابل دسترسی است.

فرض کنید که شما یک پیام دارید که میخواهید آن را به فرد دیگری انتقال دهید به طوری که فرد از ارسال آن پیام توسط خود شما اطمینان یابد، برای این کار فرآیند زیر انجام میشود.

۱- شما پیام خود را توسط کلید خصوصی تبدیل به یک امضا میکنید.

Sig = (Secret Key, Msg)

۲- سپس امضا یا sig خود را به همراه کلید عمومی و پیام در دسترس آن فرد قرار میدهید.

حال چنانچه آن فرد امضا، کلید عمومی و پیام شما را در کنار هم قرار داد و مقدار بازگشتی معتبر شد، میتوان ثابت کرد که آن پیام از طرف شخص خود شما به او مخابره شده است.

IsValid = (Public key, msg, sig)

در حقیقت میتوان گفت که آن فرد اطمینان می یابد که آن پیام از طرف کسی آمده است که کلید خصوصی را به همراه خود دارد.

کلید های خصوصی و عمومی به ترتیب به نوعی تداعی کننده ی قفل کردن و باز کردن یک داده هستند. شما با کلید خصوصی داده ای را قفل میکنید و چون دیگران با کلید عمومی شما میتوانند آن را باز کنند مطمئن میشوند که شما آن را قفل کرده بودید.

علاوه بر تایید اطلاعات بلاک ها از جفت کلید خصوصی و عمومی در دیگر اجزای وابسته به بلاکچین مثل ساخت کیف پول ها برای رمز ارز های مبتنی بر بلاکچین استفاده میشود که در آینده به آن میپردازیم.