اختلال در بلاک چین و قرارداد هوشمند

الهام اعزی و مهدی علائیان

دانشکده ریاضی و علوم کامپیوتر، دانشگاه علم و صنعت ایران، نارمک، تهران ، ایران.

چکیده

اختلال در بلاکچین و قرارداد هوشمند از مشکلاتی است که در حوزه فناوری بلاکچین و ارز های دیجیتالی گاهی به وجود می آید. این اختلالات می توانند به دلیل خطاهای برنامه نویسی، نقص های امنیتی، نوسانات بازار و یا عوامل دیگر رخ دهند. در صورتی که این اختلالات در قرارداد های هوشمندی که بر اساس بلاکچین اجرا می شوند، به وجود آیند، ممکن است منجر به خسارات مالی جدی برای طرفین قرارداد شود. ایجاد یک سیستم پشتیبانی حل مسائل قابل اعتماد نیز میتواند در کاهش خطرات احتمالی مرتبط با قراردادهای هوشمند و بلاکچین مفید باشد. این امر بهترین راه برای جلوگیری از اختلال در بلاکچین و حفظ اعتماد طرفین به قرارداد هوشمند است

کلمات کلیدی: اختلال،کد، اثبات سهام، بلاکچین، قرارداد هوشمند،

۱. مقدمه

بلاکچین و قرارداد هوشمند دو تکنولوژی نوین هستند که در حال حاضر در بسیاری از صنایع به کار گرفته می‌شوند. بلاکچین به عنوان یک سیستم توزیع شده، امکان انتقال امن و شفاف اطلاعات را بین افراد و سازمان‌ها فراهم می‌کند، در حالی که قراردادهای هوشمند، قراردادهایی هستند که با استفاده از بلاکچین ساخته شده‌اند و توسط کد کامپیوتری اجرا می‌شوند. با این حال، همانند هر فناوری دیگری، اختلالاتی نیز در بلاکچین و قراردادهای هوشمند ممکن است رخ دهند. در این مقاله، به بررسی این اختلالات و راه حل‌های ممکن برای آن‌ها پرداخته خواهد شد.

۲. اختلال در بلاکچین

اختلال در بلاکچین ممکن است به دلیل بسیاری از عوامل رخ دهد، از جمله اشتباهات در کد نوشته شده، حملات کامپیوتری، مشکلات سخت‌افزاری و نرم‌افزاری و غیره. این اختلالات می‌تواند باعث تغییر در اطلاعات ثبت شده در بلاکچین شود و در نتیجه، امنیت و شفافیت آن را تهدید کند.

یکی از راه حل‌های ممکن برای این اختلالات، استفاده از الگوریتم‌های تأیید اثبات کار (Proof of Work) یا تأیید اثبات سهام (Proof of Stake) است. این الگوریتم‌ها به طور مداوم برای تأیید صحت اطلاعات در بلاکچین استفاده می‌شوند و به این ترتیب، از اختلالات و حملاتی که ممکن است به بلاکچین وارد شوند، الگوریتم‌های قابل مقایسه می‌سازند. با این حال، این الگوریتم‌ها نیز دارای مشکلاتی هستند، از جمله مصرف بالای انرژی و هزینه بالای سیستم‌های پردازشی برای اجرای آن‌ها.

علاوه بر این، اختلال در بلاکچین می‌تواند ناشی از خطاهای انسانی باشد، که می‌تواند در هر مرحله از فرایند ثبت اطلاعات در بلاکچین رخ دهد. برای مثال، یک کاربر ممکن است به اشتباه اطلاعات نادرستی را در ورودی برای یک تراکنش

وارد کند، که باعث تغییر در اطلاعات ثبت شده در بلاکچین شود. برای جلوگیری از این نوع اختلالات، استفاده از قراردادهای هوشمند می‌تواند مفید باشد[1].

۳. اختلال در قرارداد هوشمند

قراردادهای هوشمند، یکی از کاربردهای مهم بلاکچین هستند که به وسیله آن‌ها می‌توان قراردادهایی را که نیاز به اجرای خودکار دارند، به صورت کاملاً شفاف و امن ایجاد کرد. با این حال، اختلالاتی نیز ممکن است در این قراردادهای هوشمند رخ دهند، که می‌تواند به دلیل اشتباهات در کد نوشته شده، مشکلات سخت افزاری و نرم افزاری و غیره باشد.
برای قراردادهای هوشمند، از زبان برنامه‌نویسی Solidity استفاده می‌شود که برای بسیاری از توسعه‌دهندگان، زبانی جدید و بسیار پیچیده است. همچنین، مشکلاتی همچون اشتباهات در کد و عدم صحت لازم در ورودی‌ها می‌تواند منجر به اختلال در قرارداد هوشمند شود.
برای جلوگیری از این نوع اختلالات، می‌توان از ابزارهای تحلیلی و تستی مانند تست مدل (Model Testing) و تحلیل‌های فرمال (Formal Analysis) استفاده کرد. این ابزارها می‌توانند با استفاده از روش‌های ریاضی، اشکالات و اختلالات ممکن در قرارداد هوشمند را تشخیص دهند و باعث افزایش امنیت و صحت قراردادهای هوشمند شوند.

۴. فرمول های ریاضی

برای تحلیل اختلالات در بلاکچین و قراردادهای هوشمند، از فرمول‌های ریاضی مختلف استفاده می‌شود. برای مثال، می‌توان از فرمول‌های ریاضی برای تحلیل کارایی الگوریتم‌های تأیید اثبات کار و تأیید اثبات سهام استفاده کرد.
فرمول‌های ریاضی در تحلیل اختلالات در بلاکچین و قراردادهای هوشمند بسیار مهم هستند و به ما کمک می‌کنند تا با دقت بیشتری به مسائل بپردازیم. در ادامه، به برخی از این فرمول‌ها اشاره خواهیم کرد:

I. فرمول بلاکچین:

در بلاکچین، هر بلوک شامل داده‌هایی است که به صورت رمزگذاری شده در آن ذخیره شده‌اند. فرمول بلاکچین، برای تحقق این مسئله که هر بلوک باید به صورت منحصر به فرد باشد، به کار می‌رود. این فرمول به شکل زیر است:

$$hash(block) = SHA\text{-}256(S+ prev\_hash + nonce)$$

در این فرمول، SHA-256 یک الگوریتم رمزنگاری است که برای تولید مقدار هش برای داده‌های ورودی استفاده می‌شود. S به معنای داده‌هایی است که در بلوک جاری ذخیره شده‌اند، prev_hash به معنای مقدار هش بلوک قبلی است و nonce یک عدد تصادفی است که توسط ماینرها به صورت تکرار و خطا انتخاب می‌شود تا مقدار هش بلوک، شرایط مورد نظر را برآورده کند.

II. فرمول تایید اثبات کار

الگوریتم تأیید اثبات کار (PoW) یکی از الگوریتم‌های استفاده شده در بلاکچین است که برای تأیید صحت تراکنش‌ها و ایجاد بلوک جدید، به کار می‌رود. این الگوریتم برای تولید مقدار هش بلوک، از فرمول زیر استفاده می‌کند:

hash(block) = SHA-256(S+ prev_hash + nonce) [2]

در این فرمول، S به معنای داده‌هایی است که در بلوک جاری ذخیره شده‌اند، prev_hash به معنای مقدار هش بلوک قبلی است و nonce یک عدد تصادفی است که توسط ماینرها به صورت تلاش و خطا انتخاب می‌شود تا مقدار هش بلوک، شرایط مورد نظر را برآورده کند.

III. فرمول تایید اثبات سهام

الگوریتم تأیید اثبات سهام (PoS) دیگر الگوریتمی است که در بلاکچین استفاده می‌شود. در این الگوریتم، برای ایجاد بلوک جدید، به جای استفاده از پردازش رایانه‌ای و مصرف انرژی بالا، از تعداد سکه‌های نگهداشته شده توسط کاربران استفاده می‌شود. برای تولید مقدار هش بلوک در الگوریتمPoS ، از فرمول زیر استفاده می‌شود:

hash(block) = SHA-256(S+ prev_hash + timestamp + validator_address)

در این فرمول، S به معنای داده‌هایی است که در بلوک جاری ذخیره شده‌اند، prev_hash به معنای مقدار هش بلوک قبلی است، timestamp زمان ایجاد بلوک و validator_address آدرس کاربری است که برای ایجاد بلوک انتخاب شده است.

IV. فرمول تحلیل قرارداد هوشمند

از فرمول‌های مهم برای تحلیل قراردادهای هوشمند، فرمول Solidity Smart Contract است. این فرمول برای تعریف و پیاده‌سازی قراردادهای هوشمند در بلاکچین Ethereum استفاده می‌شود. این فرمول به زبان Solidity نوشته می‌شود و شامل متغیرها، توابع، رویدادها و مدیریت ورودی و خروجی قرارداد است.

یک نمونه ساده از فرمول Solidity Smart Contract به شکل زیر است:

```
pragma solidity ^0.8.0;

contract SimpleStorage
uint storedData;

function set(uint x) public
storedData = x;

function get() public view returns (uint)
return storedData;
```

در این فرمول، قرارداد هوشمند SimpleStorage تعریف شده است که یک متغیر به نام storedData را درخود ذخیره می‌کند و دو تابع set و get برای تنظیم و بازیابی مقدار storedData تعریف شده‌اند.

در کل، فرمول‌های ریاضی برای تحلیل اختلالات در بلاکچین و قراردادهای هوشمند بسیار مهم هستند و با کمک آن‌ها، می‌توان به صورت دقیق‌تری به مسائل و مشکلات پیش آمده در این فناوری‌ها پرداخت.

اما در اینجا سوالی به وجود می آید که: آیا فرمول‌های ریاضی در بلاکچین و قراردادهای هوشمند تنها برای تحلیل اختلالات استفاده می‌شوند؟
در پاسخ باید بگوییم خیر، فرمول‌های ریاضی در بلاکچین و قراردادهای هوشمند به عنوان ابزار اصلی برای ایجاد و پیاده‌سازی این فناوری‌ها نیز به کار می‌روند. به‌طور کلی، بلاکچین‌ها و قراردادهای هوشمند برای ایجاد امنیت و شفافیت در تراکنش‌ها و انتقال داده‌ها در شبکه‌های اینترنتی به کار می‌روند و فرمول‌های ریاضی به عنوان ابزار اصلی برای تضمین امنیت و صحت این تراکنش‌ها به کار می‌روند.
به عنوان مثال، در بلاکچین‌هایی که از الگوریتم تأیید اثبات کار (PoW) استفاده می‌کنند، فرمول‌های ریاضی برای محاسبه مقدار هش بلوک و تأیید صحت آن‌ها به کار می‌روند. همچنین در الگوریتم تأیید اثبات سهام (PoS) نیز، برای محاسبه مقدار هش بلوک از فرمول‌های ریاضی استفاده می‌شود. در قراردادهای هوشمند نیز، فرمول‌های ریاضی برای تعریف شرایط و قوانین قرارداد و همچنین برای محاسبه مقدار تراکنش‌ها و تأیید صحت آن‌ها به کار می‌روند.

و در انتها، یک نمونه کد قرارداد هوشمند برای بلاکچین اتریوم (Ethereum) آورده شده است:

مثالی از یک قرارداد هوشمند ساده برای ساختار سیم کارت

```
pragma solidity ^0.8.0;

contract SimCard
```

متغیرهای قرارداد

```
address owner;
uint phone_number;
bool activated;
```

مدیر قرارداد را با آدرس مالک قرارداد مطابقت دهید

```
constructor()
owner = msg.sender;
```

تابع برای خرید سیم کارت

```
function buySimCard(uint _phone_number) public payable
```

بررسی مبلغ پرداخت شده

```
require(msg.value == 0.1 ether);
```

بررسی اینکه سیم کارت فعال نشده باشد

```
require(!activated);
```

تنظیم شماره تلفن و فعال کردای سیم کارت

```
phone_number = _phone_number;
activated = true;
```

ارسال مبلغ به مالک قرارداد

```
payable(owner).transfer(msg.value);
```

تابع برای مشاهده شماره تلفن وضعیت فعال بودن سیم کارت

```
function getSimCardDetails() public view returns (uint, bool)
return (phone_number, activated);
```

این قرارداد هوشمند یک سیم کارت را نمایش می‌دهد که می‌توان آن را با پرداخت مبلغی خریداری کرد و سپس شماره تلفن آن فعال شده و می‌توان آن را استفاده کرد. همچنین، با استفاده از تابع`getSimCardDetails` ، می‌توان شماره تلفن و وضعیت فعال بودن سیم کارت را بررسی کرد.
این قرارداد هوشمند در بلاکچین اتریوم (Ethereum) با استفاده از زبان برنامه‌نویسی Solidity نوشته شده است. در قسمت`constructor` ، آدرس مالک قرارداد تعیین شده و در تابع`buySimCard` ، مبلغ پرداختی بررسی شده و در صورت موفقیت آمیز بودن، شماره تلفن تنظیم و سیم کارت فعال می‌شود. در انتها، مبلغ پرداختی به مالک قرارداد ارسال می‌شود.
با استفاده از این قرارداد هوشمند، می‌توان یک سیستم خرید و فروش سیم کارت را روی بلاکچین پیاده کرد که به دلیل امنیت و شفافیت بلاکچین، امکان تقلب و تغییر دادن اطلاعات را به حداقل می‌رساند. همچنین، با استفاده از امکانات قراردادهای هوشمند در بلاکچین، می‌توان قوانین و شرایط خرید و فروش سیم کارت را به صورت دقیق و شفاف تعریف کرد و از طریق تابع‌های دیگری مثل `transfer`، مبالغ پرداخت شده را به صورت خودکار به حساب مالک قرارداد انتقال داد.
برای اجرای این قرارداد هوشمند در بلاکچین اتریوم، می‌توان از یک محیط توسعه‌ی اتریوم مانند Remix استفاده کرد. محیط Remix یک محیط آنلاین است که اجازه می‌دهد قراردادهای هوشمند Solidity را توسعه، تست و اجرا کرد.
برای اجرای این قرارداد هوشمند، می‌توان به صورت زیر عمل کرد:

۱. وارد محیط Remix شوید.
۲. یک فایل جدید با پسوند sol. بسازید.
۳. کد قرارداد هوشمند را در فایل ساخته شده کپی و پیست کنید.
٤. فایل را ذخیره کنید.
٥. از بالای صفحه، به بخش "Compile" بروید
٦. در بخش"Compile" ، روی دکمه "Compile SimCard.sol" کلیک کنید تا کد قرارداد هوشمند اجرا و کامپایل شود.
۷. سپس، به بخش "Run" بروید و در قسمت"Deploy & Run Transactions" ، قرارداد را انتخاب کنید و بر روی دکمه "deploy"کلیک کنید.
۸. در پنجره pop-up جدید، می‌توانید مشخصات قرارداد را تعیین کرده و بر روی دکمه "Confirm" کلیک کنید تا قرارداد روی بلاکچین اتریوم اجرا شود.
۹. پس از اجرای قرارداد، در پنجره"Deployed Contracts" ، می‌توانید قرارداد را مشاهده کنید.

با استفاده از تابع‌های موجود در قرارداد، می‌توانید تراکسیون‌های مختلف را اجرا کنید. به عنوان مثال، با استفاده از تابع `buySimCard`، می‌توانید یک سیم کارت را خریداری کنید و با استفاده از تابع `getSimCardDetails` ، می‌توانید اطلاعات سیم کارت را بررسی کنید.

۵. نتیجه‌گیری

فرمول‌های ریاضی در بلاکچین و قراردادهای هوشمند علاوه بر تحلیل اختلالات، برای ایجاد و پیاده‌سازی این فناوری‌ها نیز به کار می‌روند. به طور کلی، فرمول‌های ریاضی در بلاکچین و قراردادهای هوشمند به عنوان ابزار اصلی برای ایجاد امنیت و شفافیت در تراکنش‌ها و انتقال داده‌ها در شبکه‌های اینترنتی به کار می‌روند. همچنین، فرمول‌های ریاضی در بلاکچین و قراردادهای هوشمند برای ایجاد قوانین و شروط قراردادها و همچنین برای محاسبه مقدار تراکنش‌ها و تأیید صحت آن‌ها نیز به کار می‌روند. بنابراین، فرمول‌های ریاضی در این فناوری‌ها به عنوان یکی از ابزارهای اصلی برای پیاده‌سازی و ایجاد امنیت و شفافیت در تراکنش‌ها و قراردادهای هوشمند به کار می‌روند.

مراجع

[1] Lin William Cong, Zhiguo He. Blockchain Disruption and Smart Contracts. https://academic.oup.com/rfs/article/32/5/1754/5427778

[2] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." *Decentralized business review* (2008): 21260.

[3] *Sharma, Pratima, Rajni Jindal, and Malaya Dutta Borah. "A review of smart contract-based platforms, applications, and challenges." Cluster Computing 26.1 (2023): 395-421.*