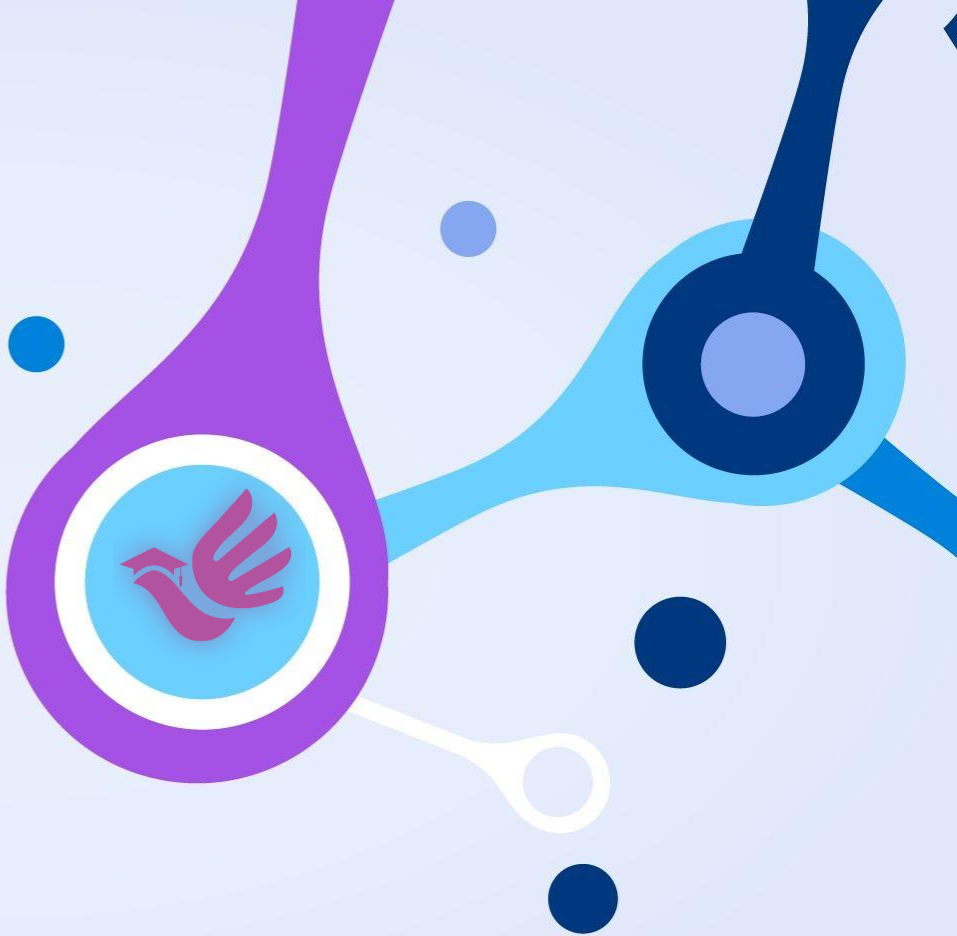
مفاهیم قراردادهوشمندو
زبان برنامه نویسی Solidity

مقدمات بلاکچین:

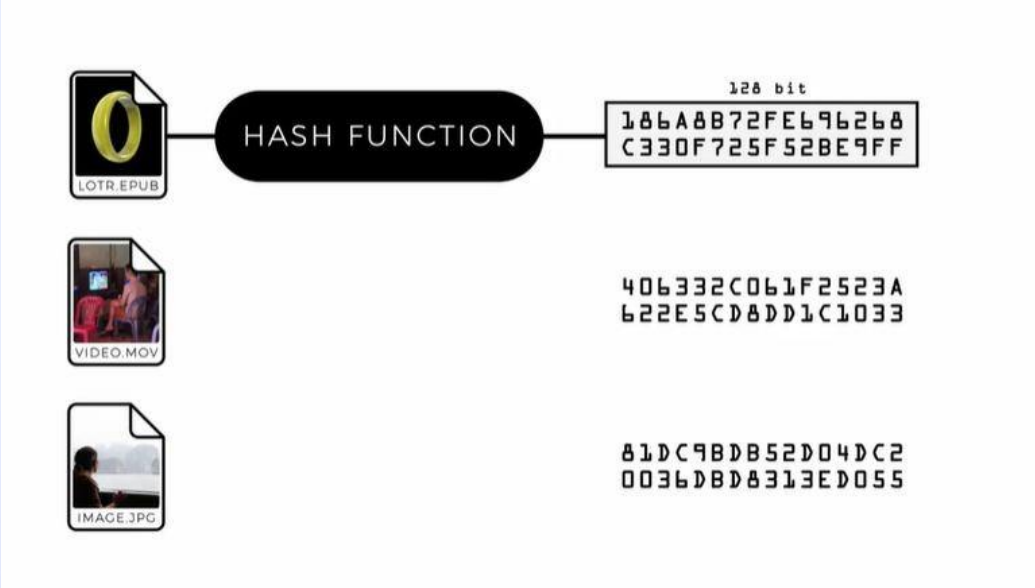
- توابع hash و علت استفاده از آنها
- رمزنگاری کلید عمومی
- درخت مرکل
- بلاکچین به عنوان یک ساختار داده
- قراردادهای هوشمند
- گره ها در شبکه غیرمتمرکز
- فورک ها و علت وقوع آنها



Hash Function

تابع هش تابعی است که داده های ورودی با هر اندازه ای را دریافت می کند و آنها را به خروجی با اندازه ثابت نگاشت می دهد.

سایت دموی بلاک چین



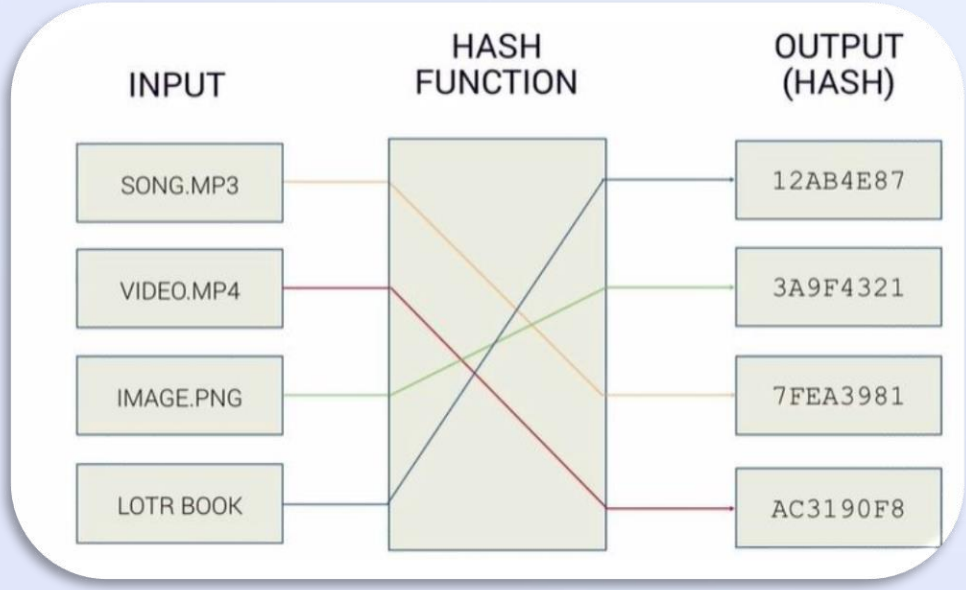
Hash Function

Hash خروجی

- مهم نیست ورودی یک قطعه موسیقی، یک فایل ویدئویی، یک تصویر، یا یک کتاب باشد، خروجی برای همه یک **فرمت و اندازه یکسان** دارد.
- تابع هش همیشه برای یک **ورودی ثابت**، یک خروجی ثابت به ما می دهد.

مشاهده تعریف هش در سایت

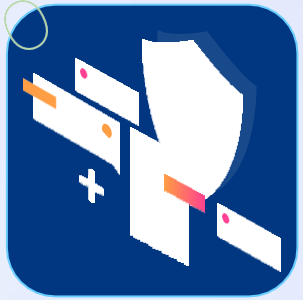
Hashing فرآیند



ویژگی های توابع Hash

1. قطعی هستند.
2. سریع هستند.
3. غیر قابل بازگشت هستند.
4. تصادم ندارند.
5. تغییرات کوچک در ورودی منجر به تغییرات بزرگ در خروجی می شود.

کاربردهای توابع Hash

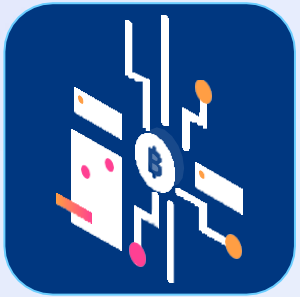


تایید صحت داده ها:

- Checksum
- هش پسورده

اثبات کار:

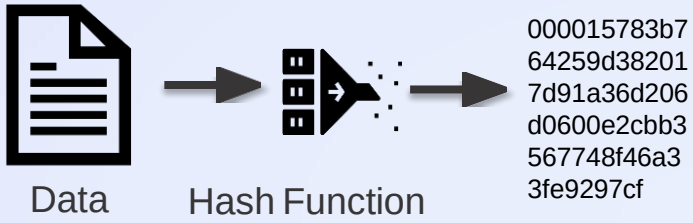
- پیدا کردن یک هش خاص سخت است.
- اثبات کار انجام شده برای پیدا کردن هش آسان است.



شناسایی داده:

- Commit-Reveal
- تولید اعداد شبه تصادفی

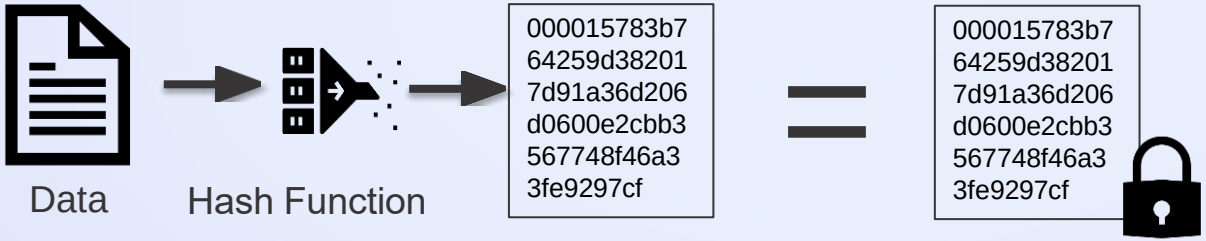
کاربردهای توابع Hash

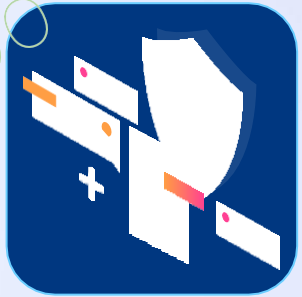


تایید صحت داده ها:

- Checksum

کاربردهای توابع Hash

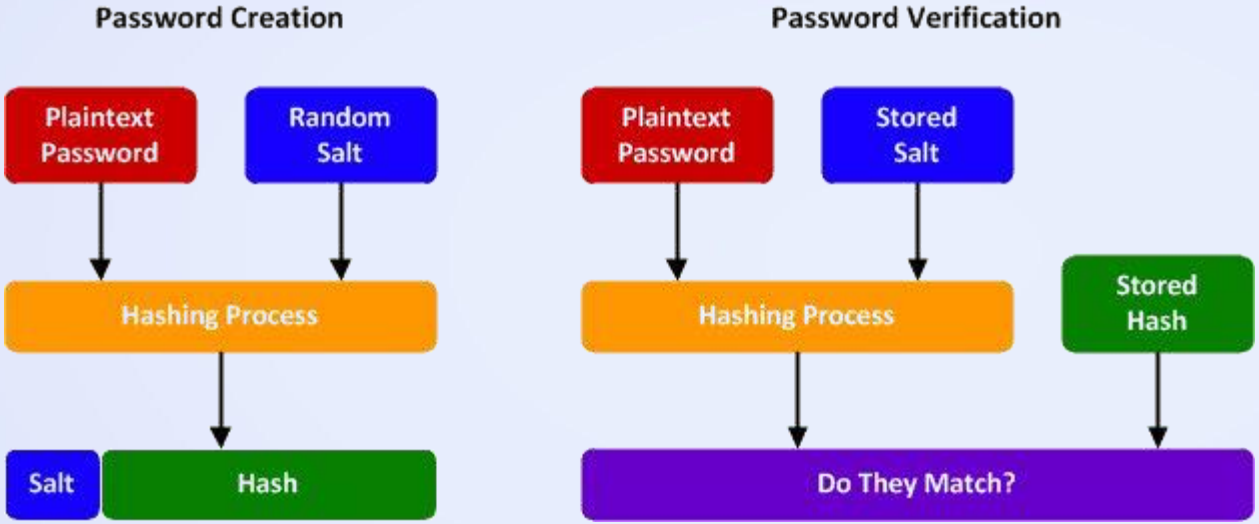




تایید صحت داده ها:

- Checksum

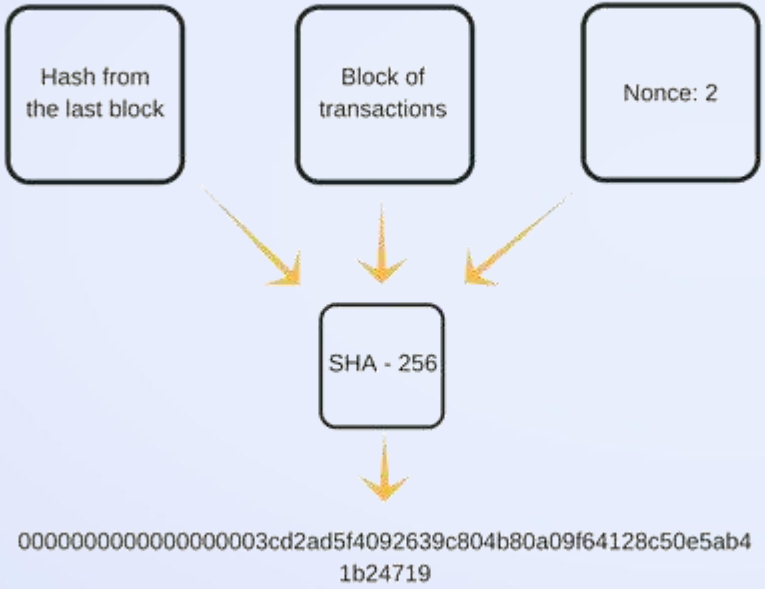
کاربردهای توابع Hash



تایید صحت داده ها:

- هش پسوردها

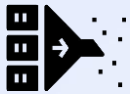
کاربردهای توابع Hash



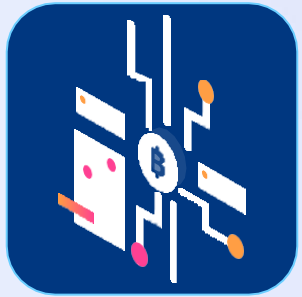
اثبات کار:

- پیدا کردن یک هش خاص سخت است.
- اثبات کار انجام شده برای پیدا کردن آن هش آسان است.

کاربردهای توابع Hash



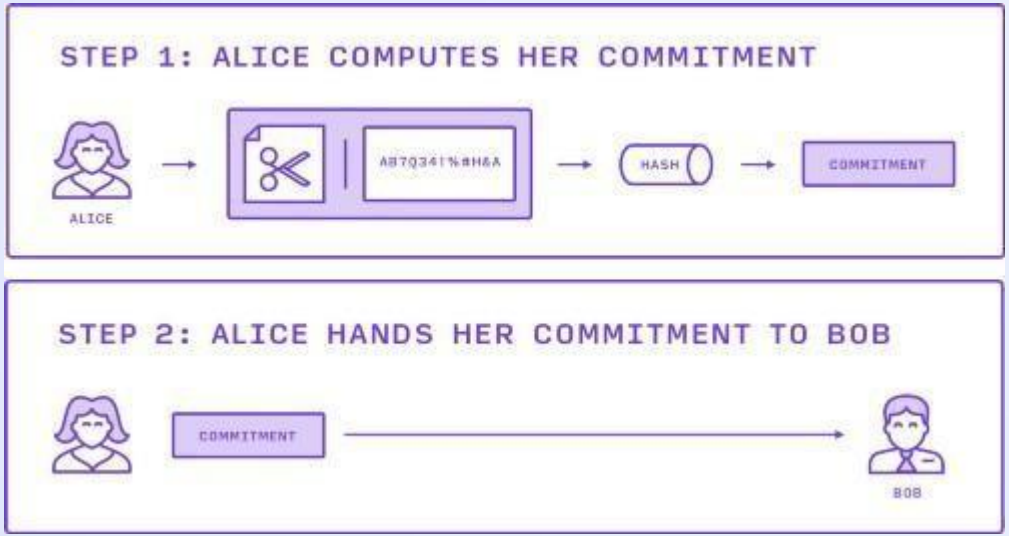
```
000015783b7
64259d38201
7d91a36d206
d0600e2cbb3
567748f46a3
3fe9297cf
```

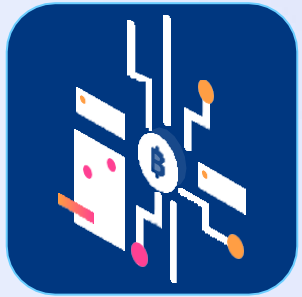


شناسایی داده:

- Commit-Reveal

کاربردهای توابع Hash

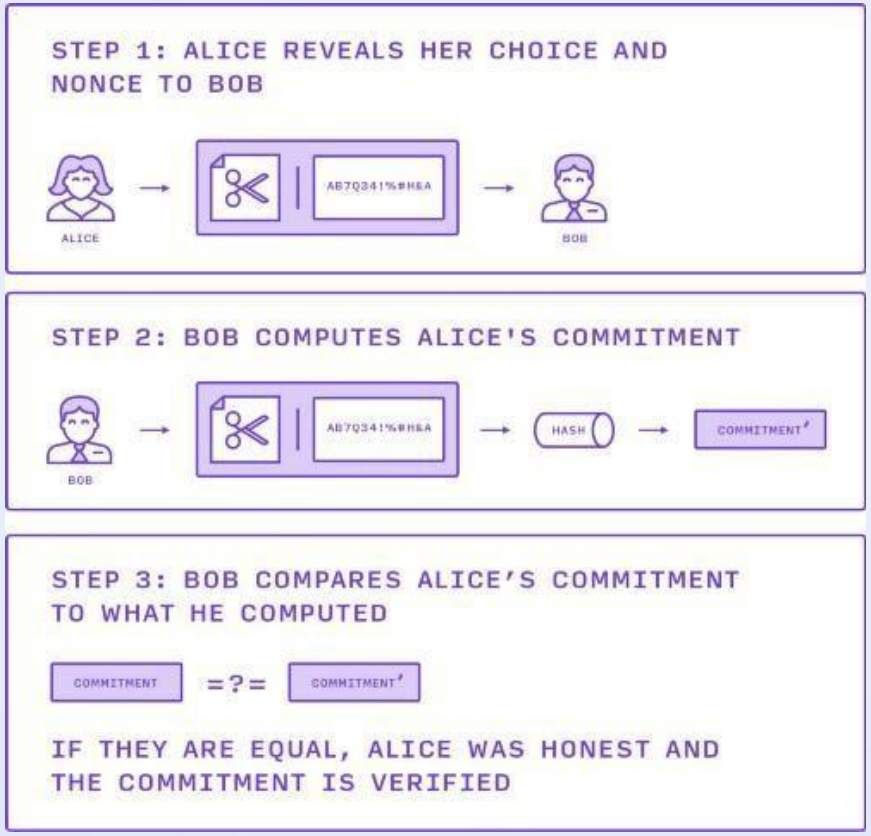


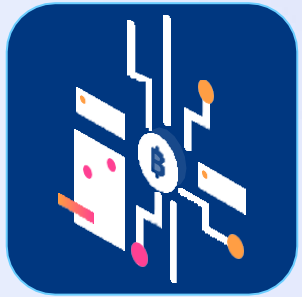


شناسایی داده:

- Commit-Reveal

کاربردهای توابع Hash



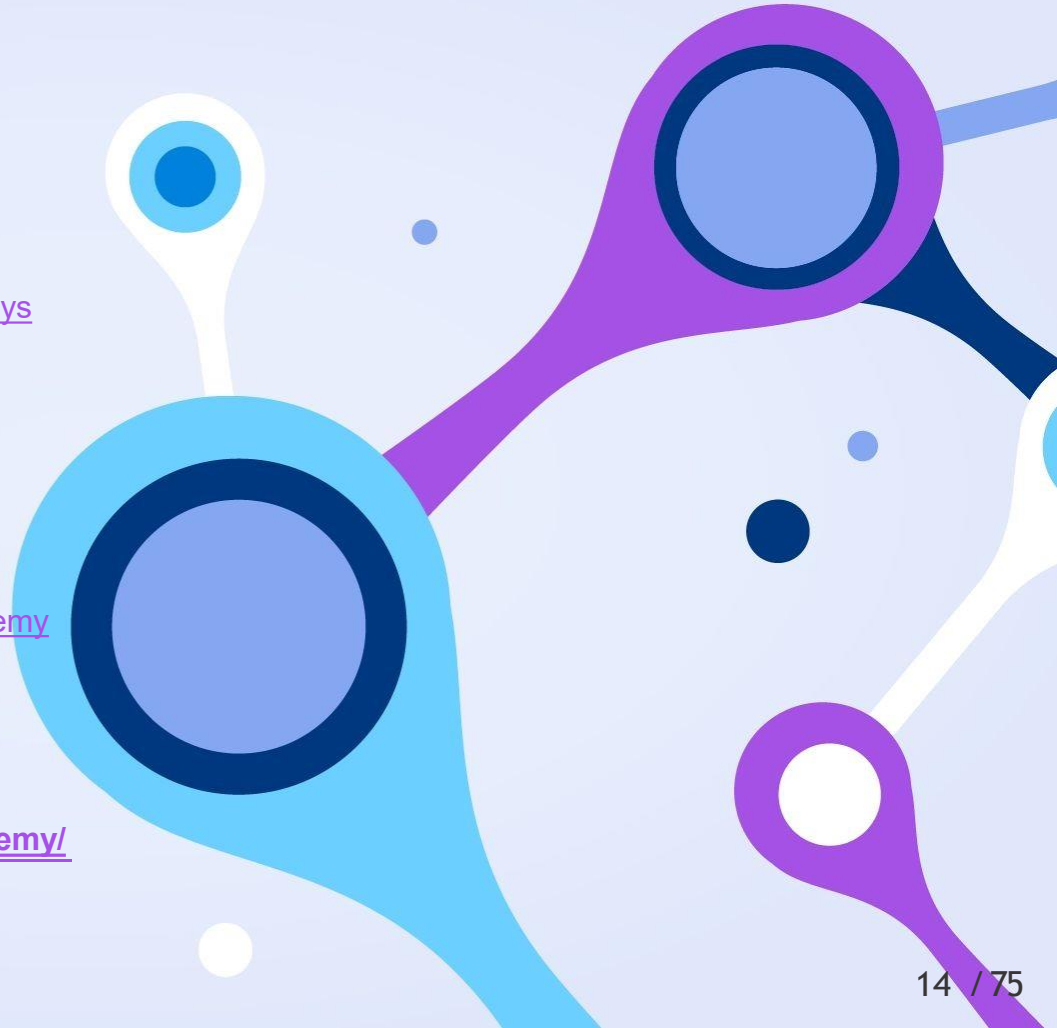


شناسایی داده:

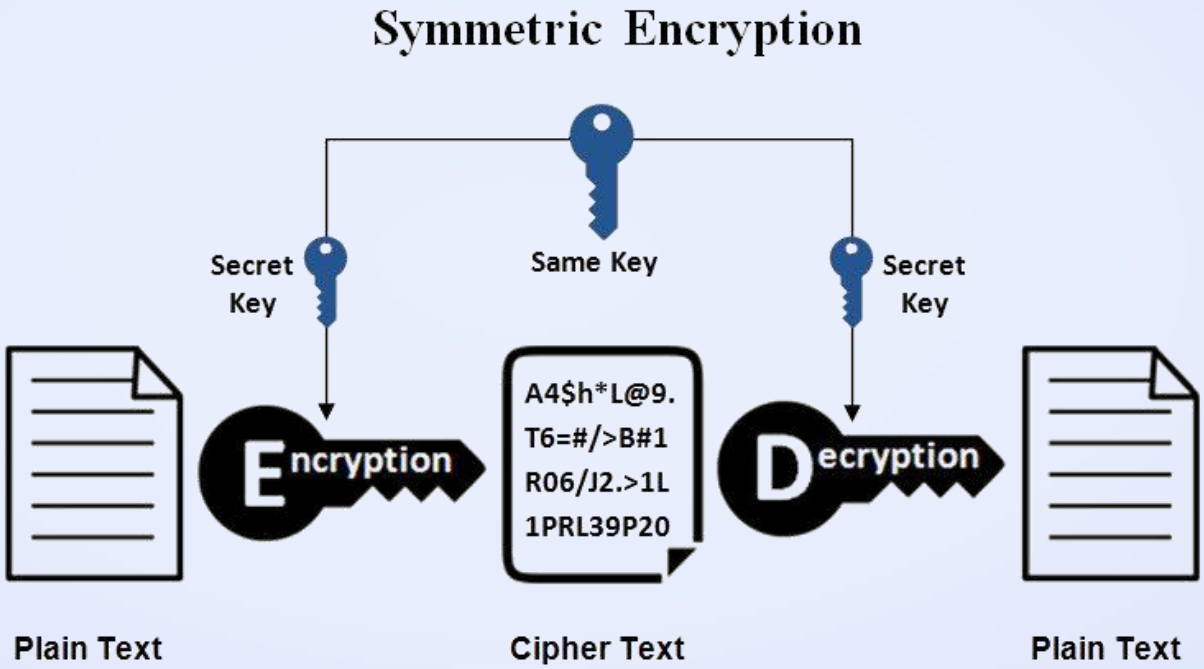
- Commit-Reveal

Additional Reading

1. Blockchain Underpinnings: Hashing by ConsenSys Media
2. How Hash Algorithms Work
3. Cryptographic hash functions on Wikipedia
4. Cryptographic hash functions by the Khan Academy
5. Video: Hashing Algorithms and Security by Computerphile
6. **https://observablehq.com/@consensys-academy/cryptographic-hash-functions**

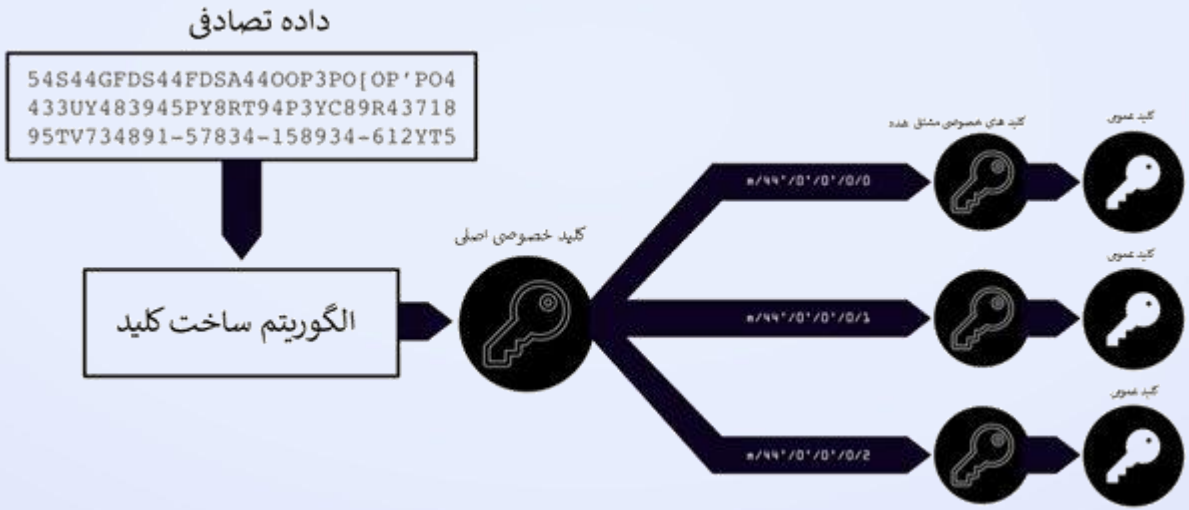


رمزنگاری متقارن



رمزنگاری کلید عمومی

رمزنگاری کلید عمومی که به نام رمزنگاری نامتقارن هم شناخته می شود، سیستمی است که در آن از جفت کلید های عمومی و خصوصی استفاده می شود.



رمزنگاری کلید عمومی

رمزنگاری کلید عمومی چه ویژگی هایی در اختیار ما قرار می دهد:

رمز نگاری:

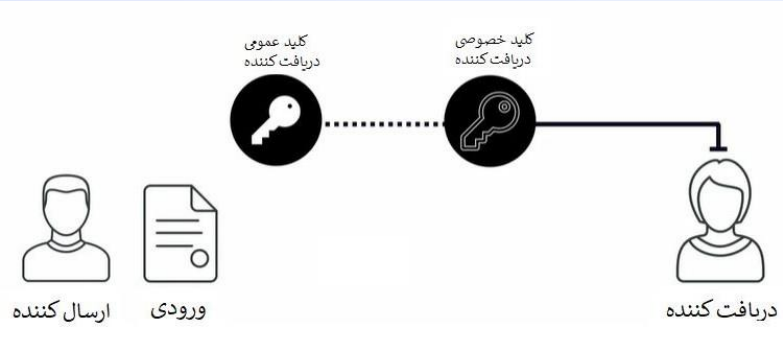
- تنها دارنده کلید خصوصی می تواند یک پیام رمزنگاری شده با کلید عمومی مرتبط را رمزگشایی کند.

احراز هویت:

- یک کلید عمومی می تواند مالکیت کلید خصوصی را تایید کند.
- امضای دیجیتال
- امضای تراکنش ها روی بلاکچین

رمزنگاری کلید عمومی

ارسال یک پیام خصوصی برای یک نفر با اطمینان از اینکه تنها آن فرد قادر به خواندن آن است:

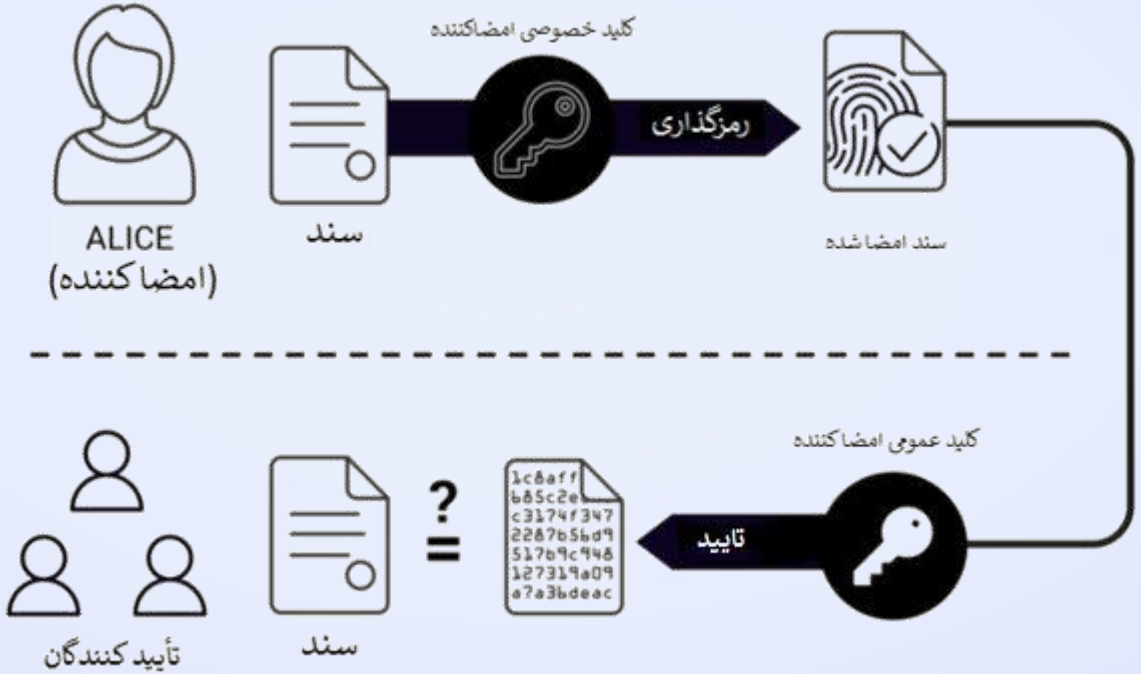


پیام رمزنگاری شده ارسال می شود.

ارسال کننده پیام را با کلید عمومی دریافت کننده رمز می کند.

دریافت کننده با استفاده از کلید خصوصی خود پیام را رمزگشایی می کند.

امضای دیجیتال



Additional Reading

1. How are Ethereum addresses generated?
2. Wikipedia: Public Key Cryptography

سایت دموی رمزنگاری

درخت مرکل

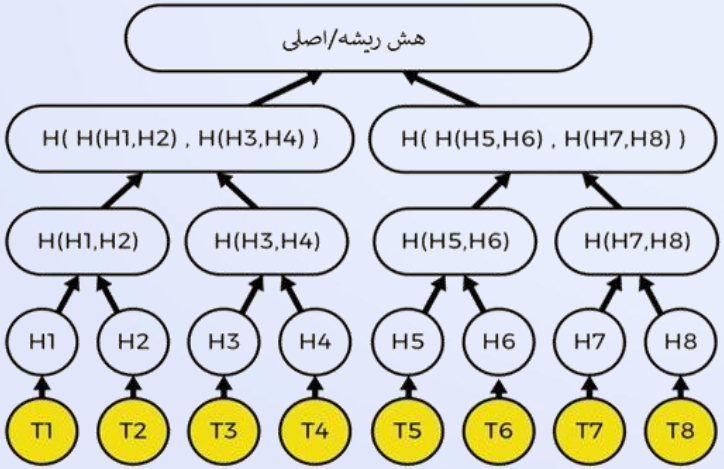
درخت مرکل یک ساختمان داده دودویی مشتق شده از مجموعه ای از داده ها است که در آن:

- برگ ها حاوی هش داده ها هستند.
- گره ها حاوی هش فرزندانشان هستند.

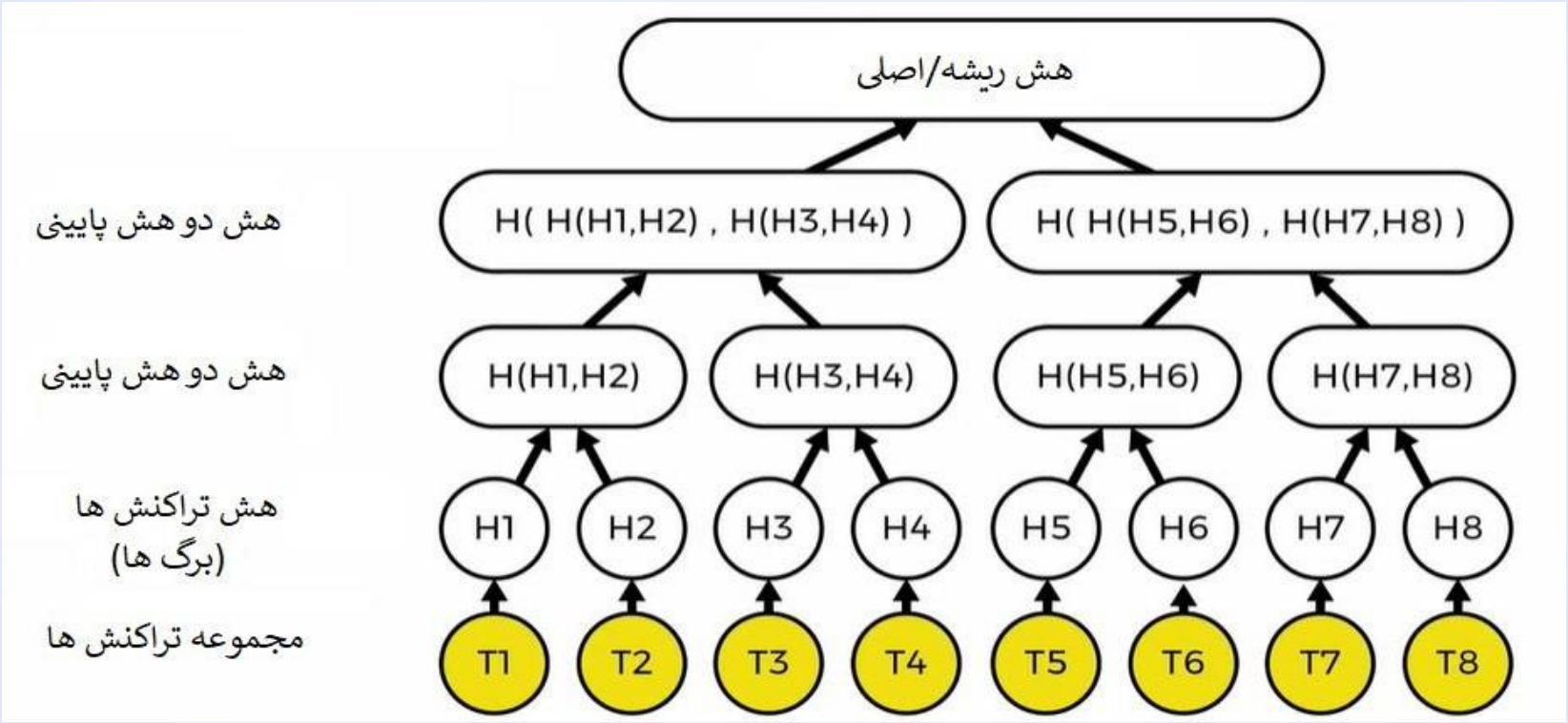
ویژگی ها:

★ قابلیت تایید رمزنگاری شده

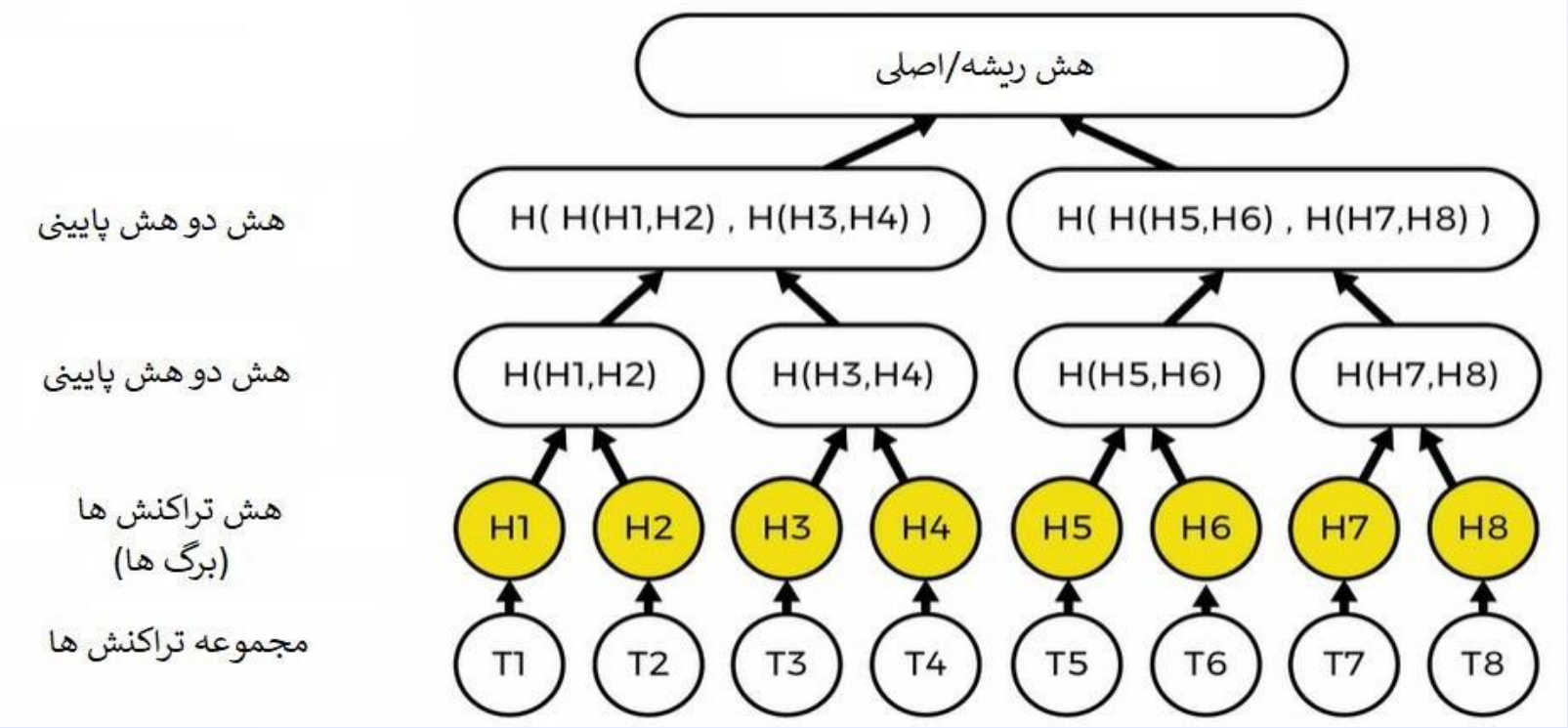
○ تغییر هر داده منجر به تغییر ریشه درخت می شود



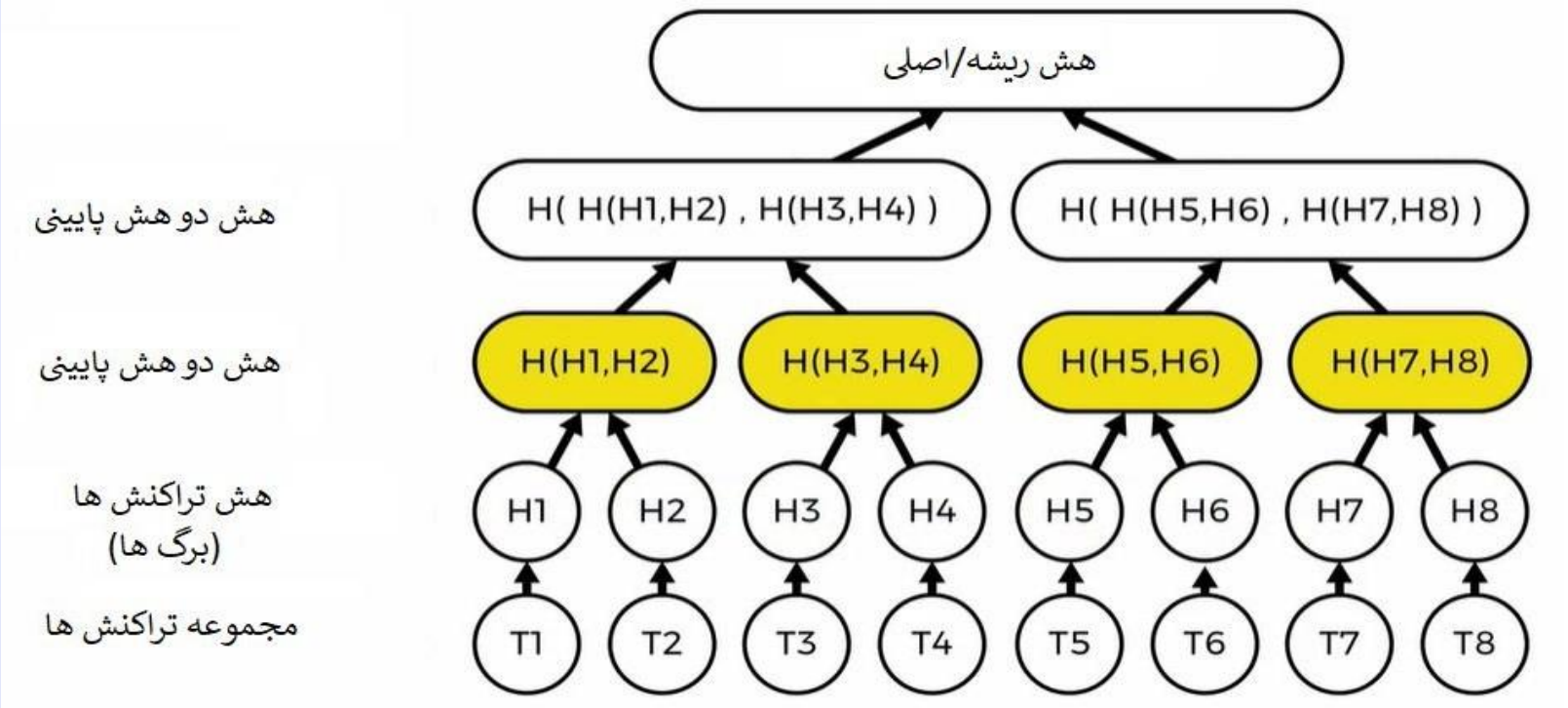
درخت مرکل


هش ریشه/اصلی
هش دو هش پایینی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
هش دو هش پایینی
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
هش تراکنش ها
(برگ ها)
H1
H2
H3
H4
H5
H6
H7
H8
مجموعه تراکنش ها
T1
T2
T3
T4
T5
T6
T7
T8

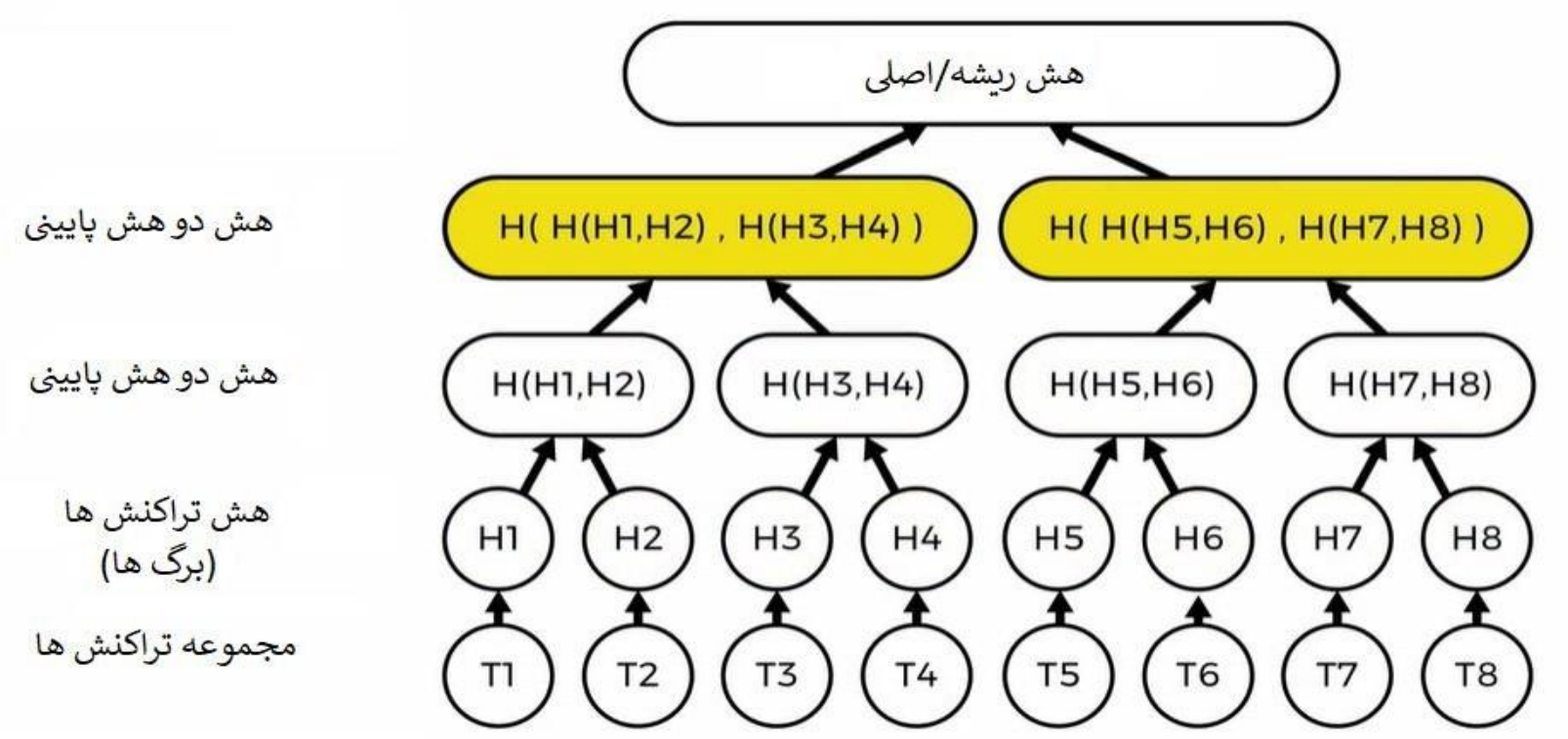
درخت مرکل


هش ریشه/اصلی
هش دو هش پایینی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
هش دو هش پایینی
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
هش تراکنش ها
(برگ ها)
H1
H2
H3
H4
H5
H6
H7
H8
مجموعه تراکنش ها
T1
T2
T3
T4
T5
T6
T7
T8

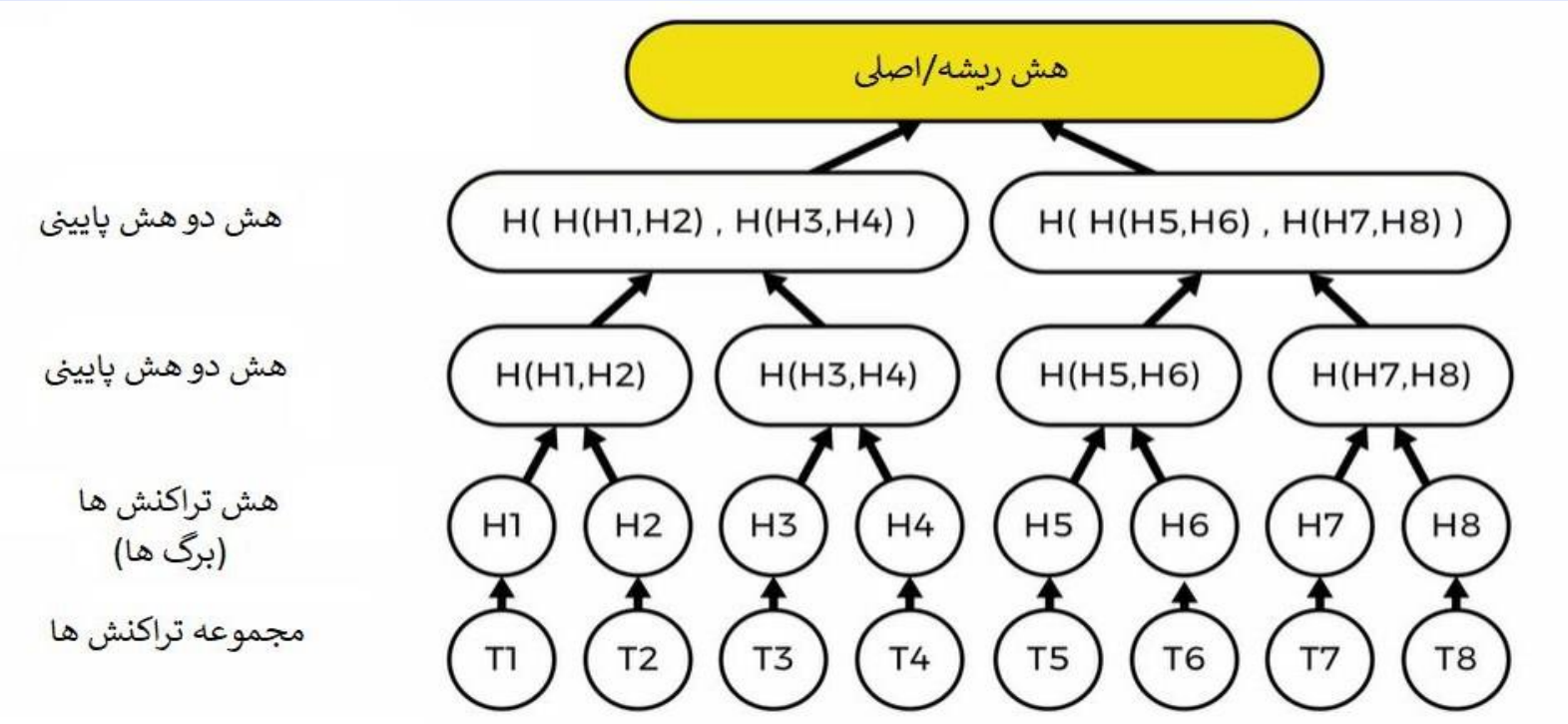
درخت مرکل


هش ریشه/اصلی
هش دو هش پایینی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
هش دو هش پایینی
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
هش تراکنش ها
(برگ ها)
H1
H2
H3
H4
H5
H6
H7
H8
مجموعه تراکنش ها
T1
T2
T3
T4
T5
T6
T7
T8

درخت مرکل

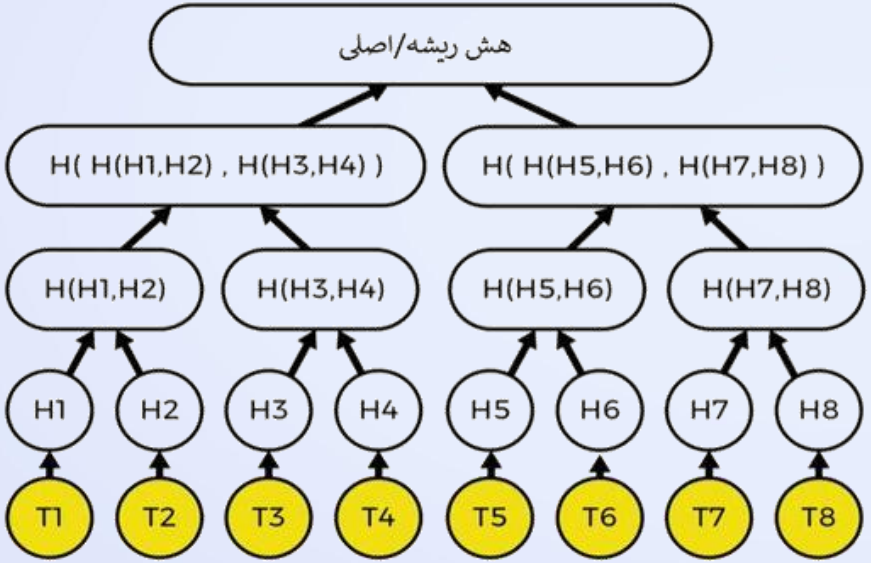

هش ریشه/اصلی
هش دو هش پایینی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
هش دو هش پایینی
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
هش تراکنش ها
(برگ ها)
H1
H2
H3
H4
H5
H6
H7
H8
مجموعه تراکنش ها
T1
T2
T3
T4
T5
T6
T7
T8

درخت مرکل


هش ریشه/اصلی
هش دو هش پایینی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
هش دو هش پایینی
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
هش تراکنش ها
(برگ ها)
H1
H2
H3
H4
H5
H6
H7
H8
مجموعه تراکنش ها
T1
T2
T3
T4
T5
T6
T7
T8

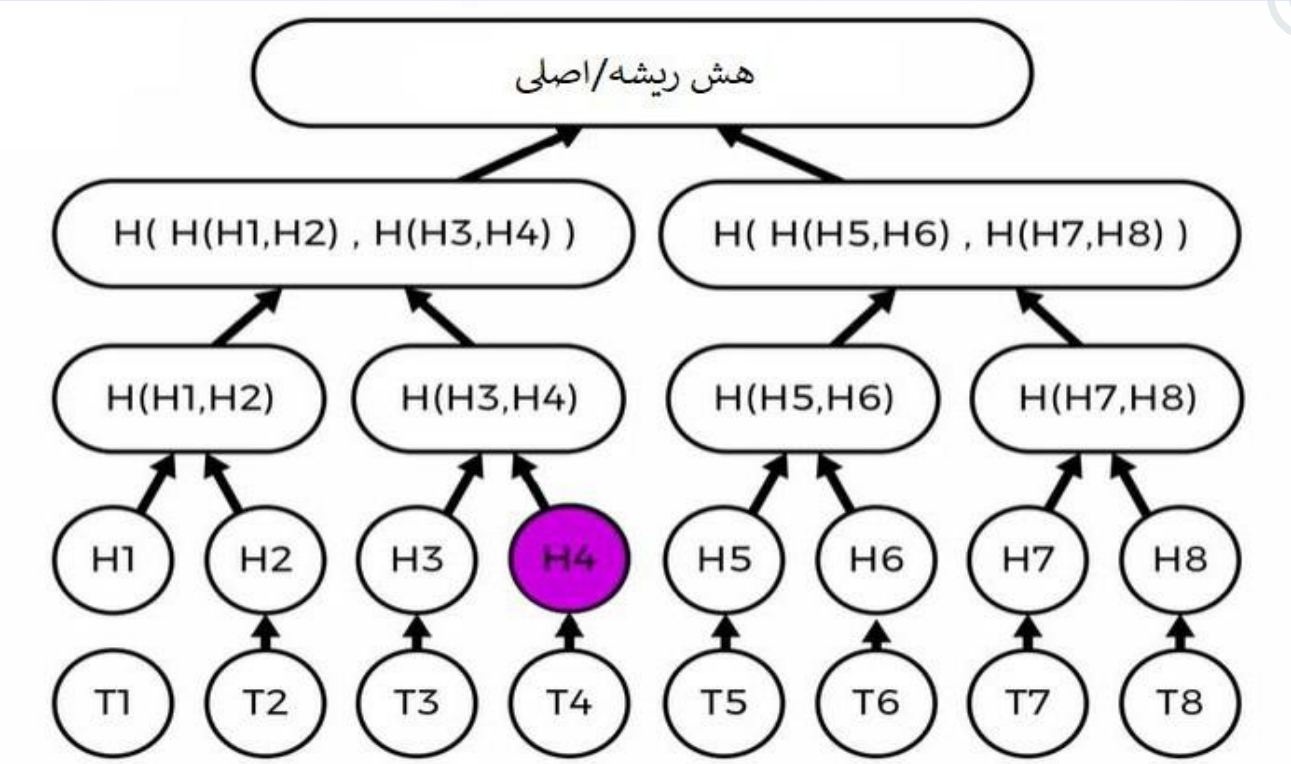
درخت مرکل

★ قابلیت تایید محتوا به صورت کارآمد

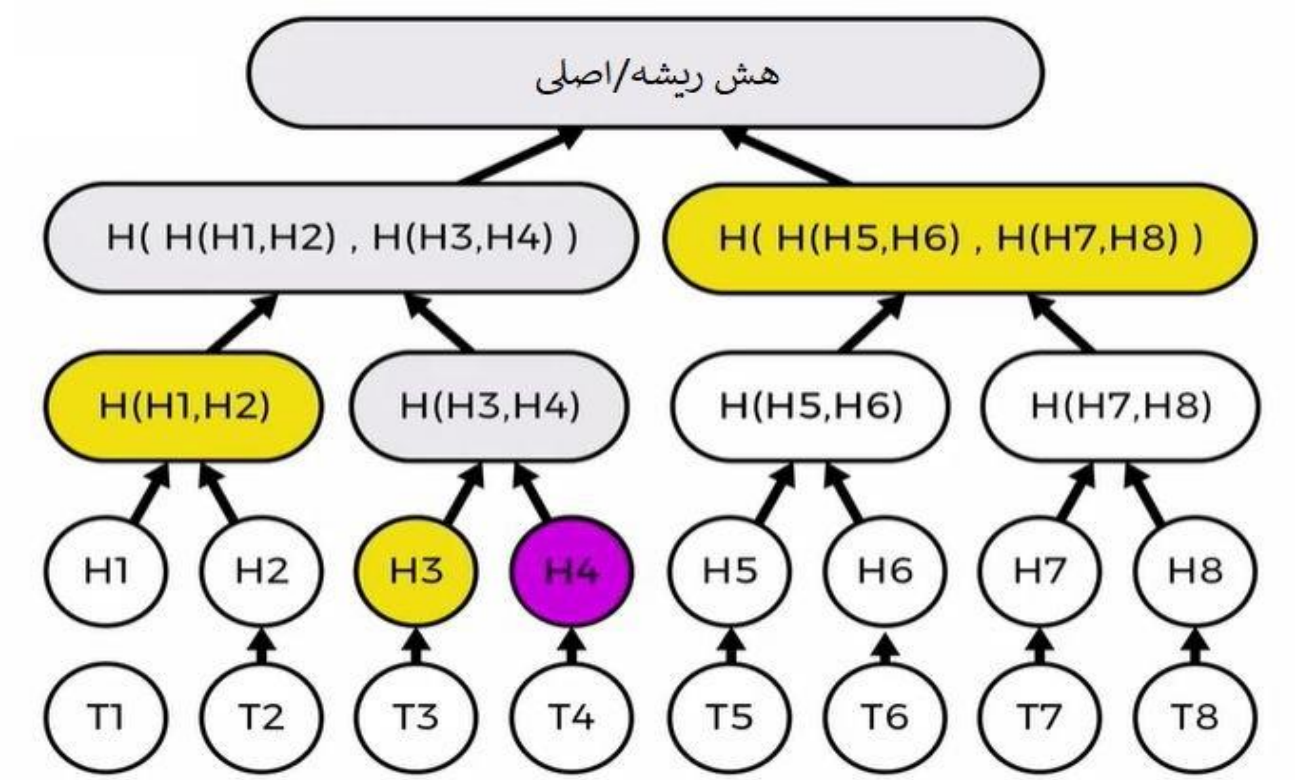


- اثبات آسان وجود یک داده در مجموعه
- نیاز به زیرمجموعه کوچکی از داده ها
- مقیاس پذیری
- عدم نمایش محتو

درخت مرکل

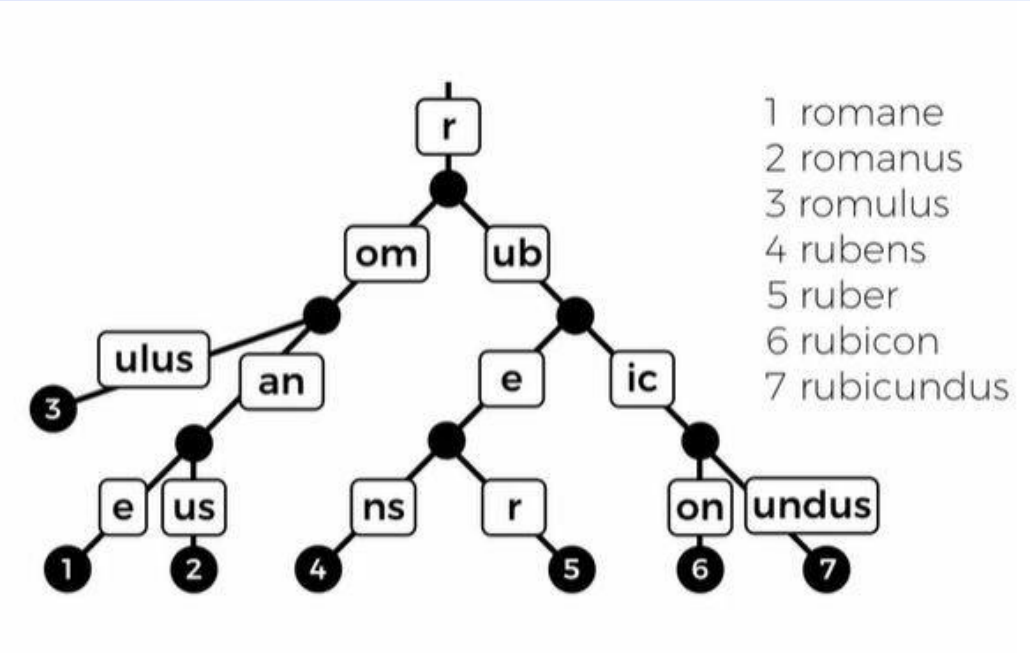

هش ریشه/اصلی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
H1
H2
H3
H4
H5
H6
H7
H8
T1
T2
T3
T4
T5
T6
T7
T8

درخت مرکل


هش ریشه/اصلی
H(H(H1,H2) , H(H3,H4))
H(H(H5,H6) , H(H7,H8))
H(H1,H2)
H(H3,H4)
H(H5,H6)
H(H7,H8)
H1
H2
H3
H4
H5
H6
H7
H8
T1
T2
T3
T4
T5
T6
T7
T8

درخت پاتریشیا

درخت پاتریشیا یک درخت رادیکس باینری درخت مرکلی با فضای ذخیره سازی بهینه است.

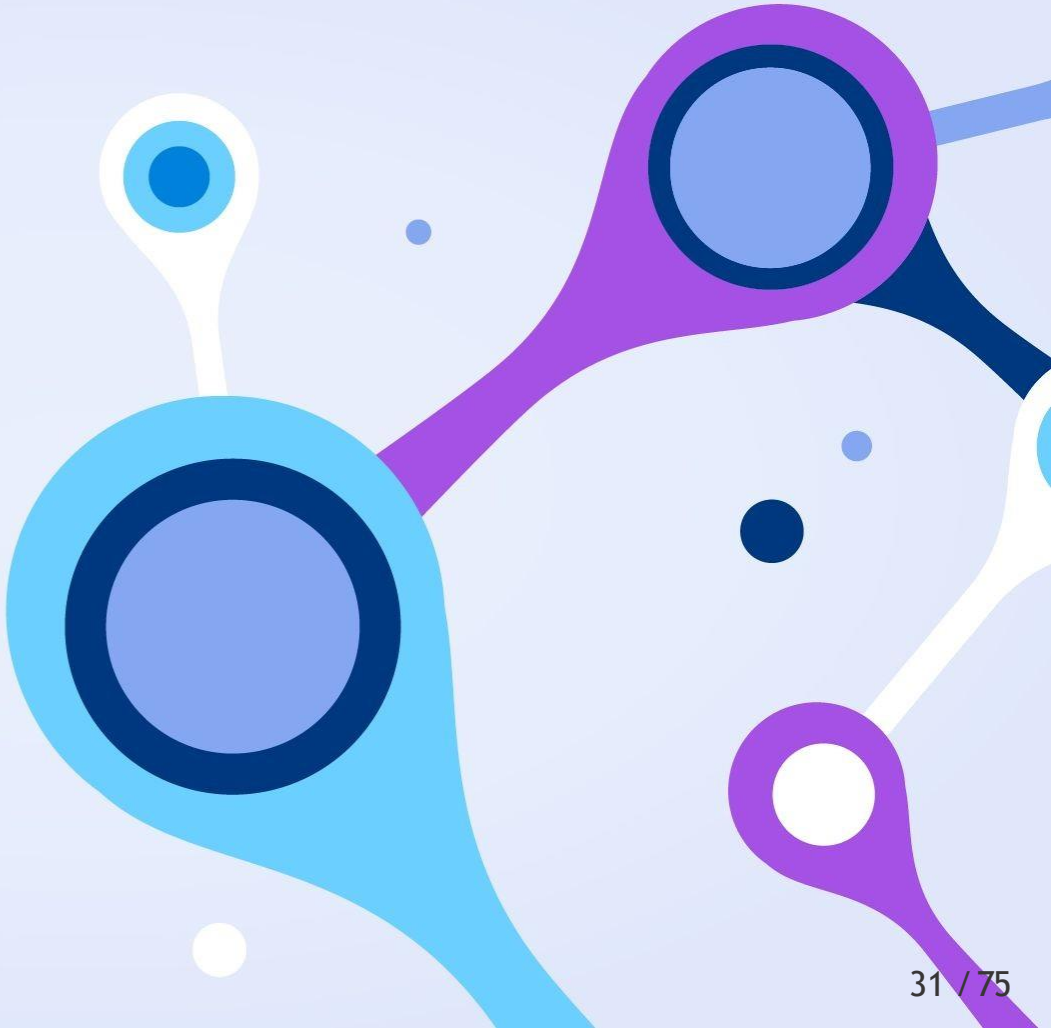


ویژگی ها:

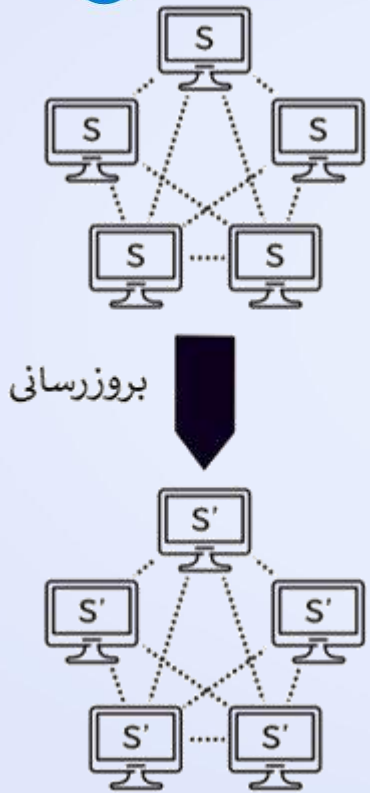
- ذخیره کلید/مقدار
- قابلیت جستجو
- امکان حذف و اضافه به صورت کارآمد
- امکان نگاشت مجموعه های بزرگ داده

Additional Reading

1. Merkling in Ethereum by Vitalik Buterin
2. Ever Wonder How Merkle Trees Work? by ConsenSys Media
3. Patricia Merkle Trees
4. **https://observablehq.com/@consensys-academy/merkle-trees**



ساختار بلاکچین

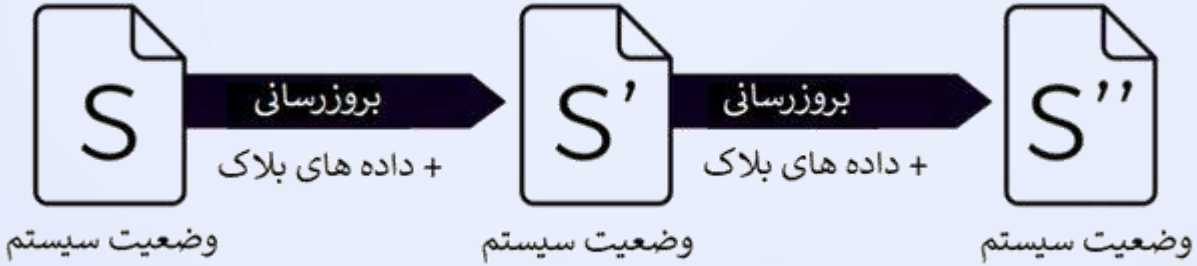


ویژگی ها:

- نگهداری وضعیت در یک شبکهP2P
- مدیریت بروزرسانی های وضعیت
  - همه شرکت کنندگان روی معتبر بودن به روزرسانی ها توافق دارند
  - توافق عمومی Consensus روی وضعیت جدید سیستم

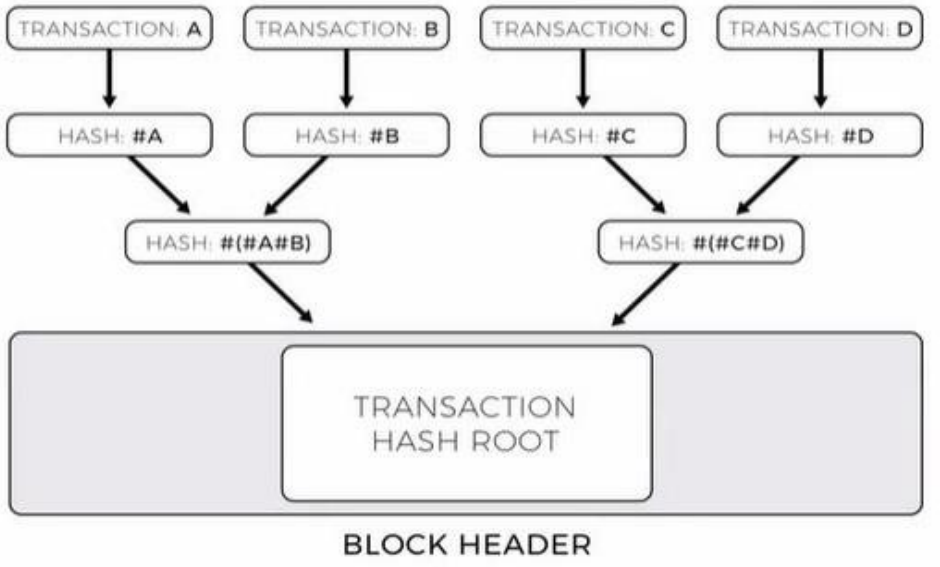
ساختار بلاکچین

بروزرسانی ها در بلاکچین به مجموعه هایی (به نام بلاک) تقسیم می شوند که توسط یک درخت مرکل خلاصه شده اند.



بلاک ها

یک نمونه بلاک ساده

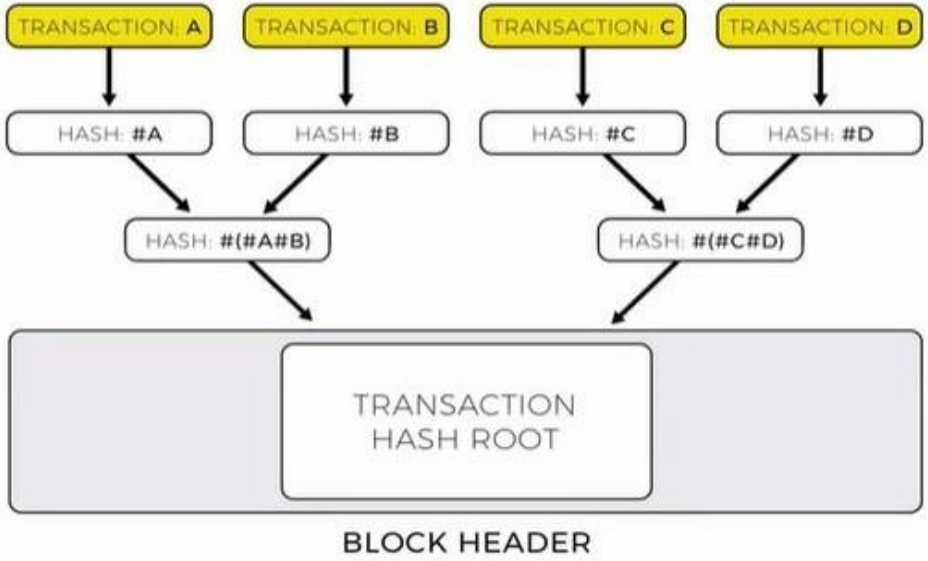


بلاک ها

یک نمونه بلاک ساده

یک بلاک شامل است بر:

- لیستی از تراکنش ها
  - یک درخت مرکل بر اساس این لیست

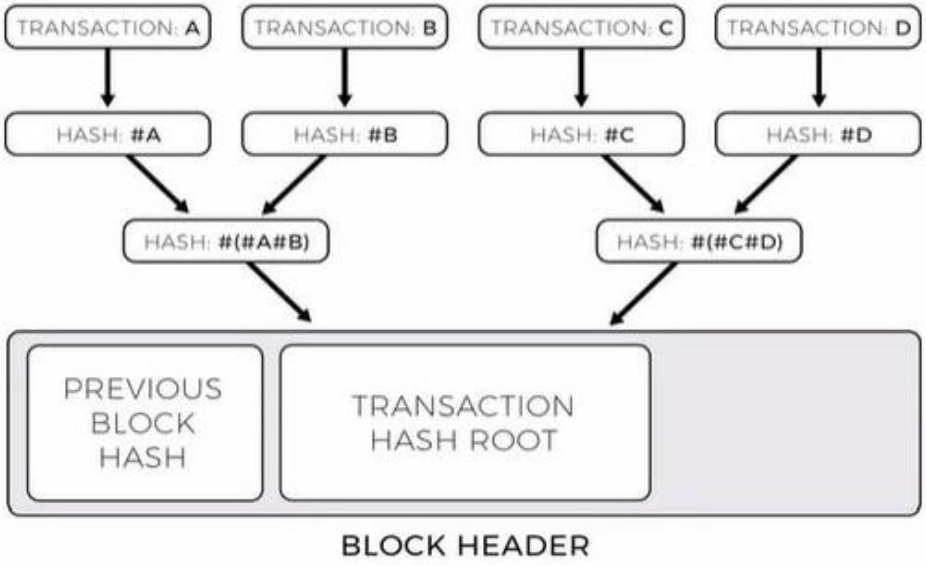


بلاک ها

یک نمونه بلاک ساده

یک بلاک شامل است بر:

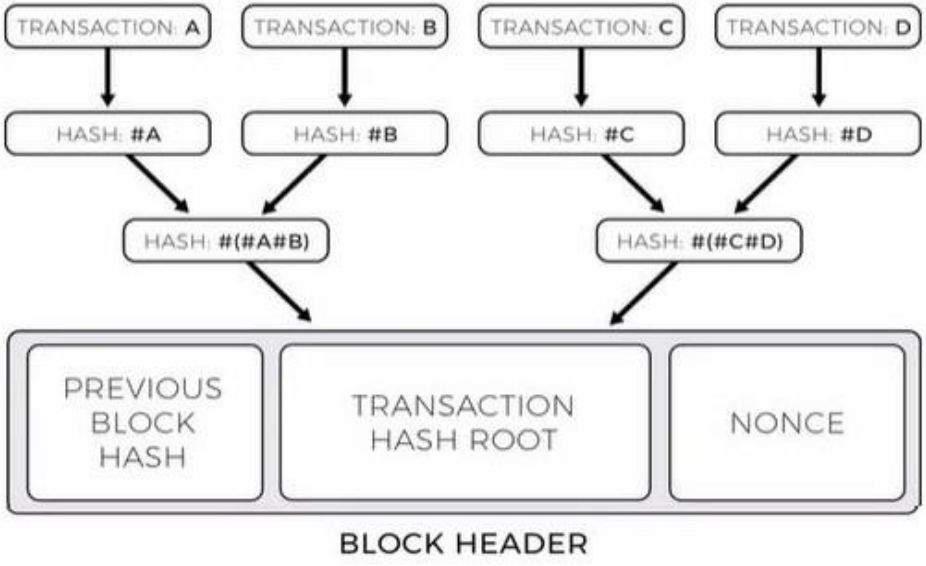
- لیستی از تراکنش ها
  - یک درخت مرکل بر اساس این لیست
- هش بلاک قبلی



بلاک ها

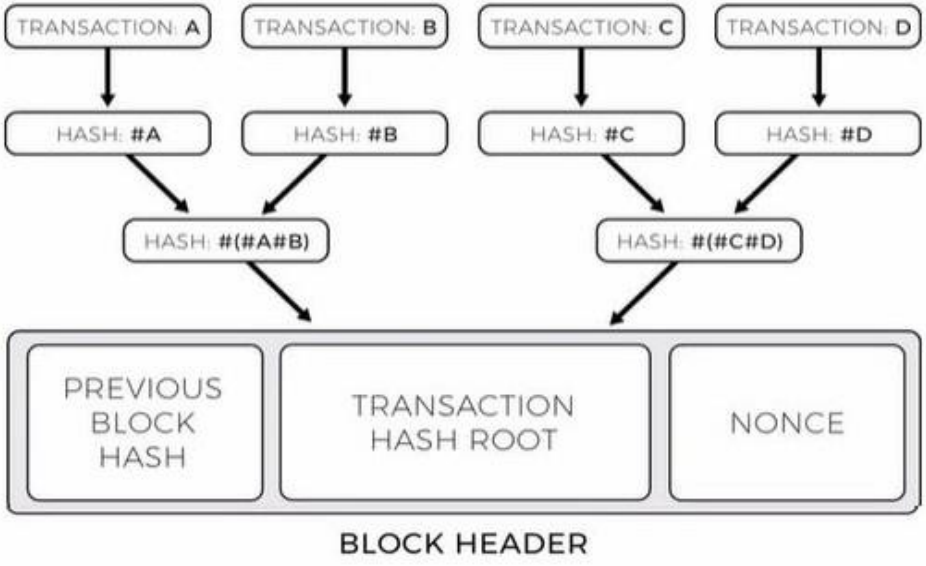
- Nonce

  - در اثبات کار استفاده می شود
  - برای بدست آوردن هش معتبر بلاک تغییر می کند
  - توافق عمومی در مورد وضعیت جدید سیستم



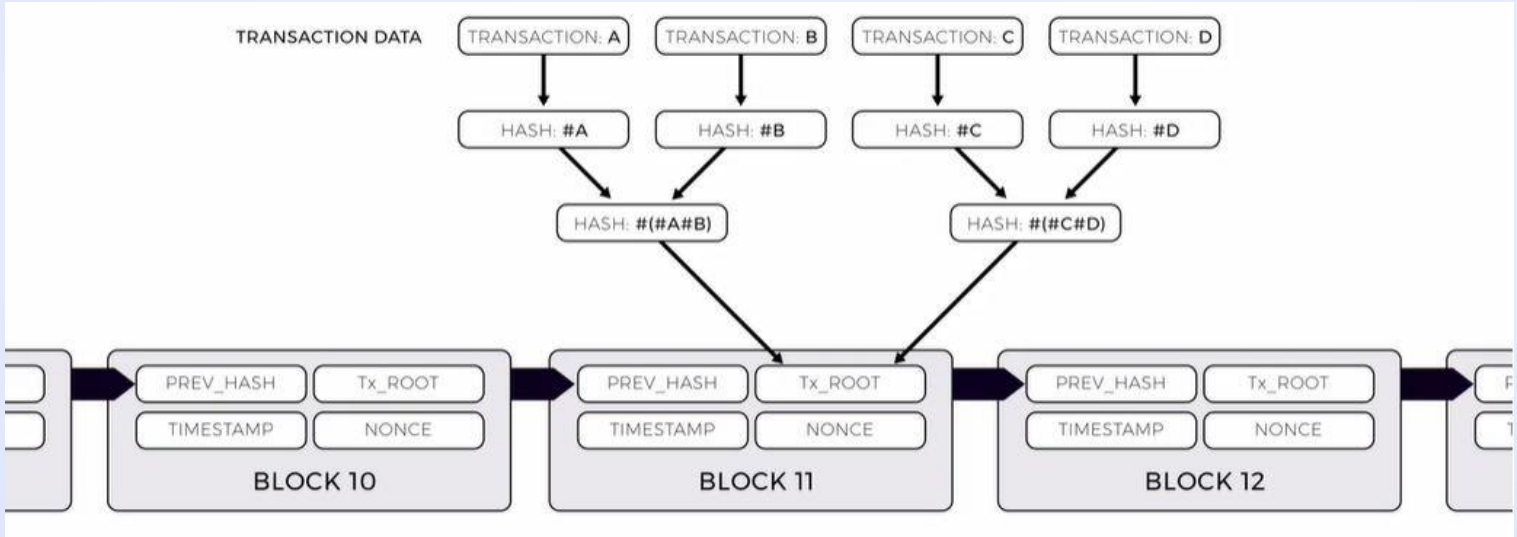
بلاک ها

یک نمونه بلاک ساده



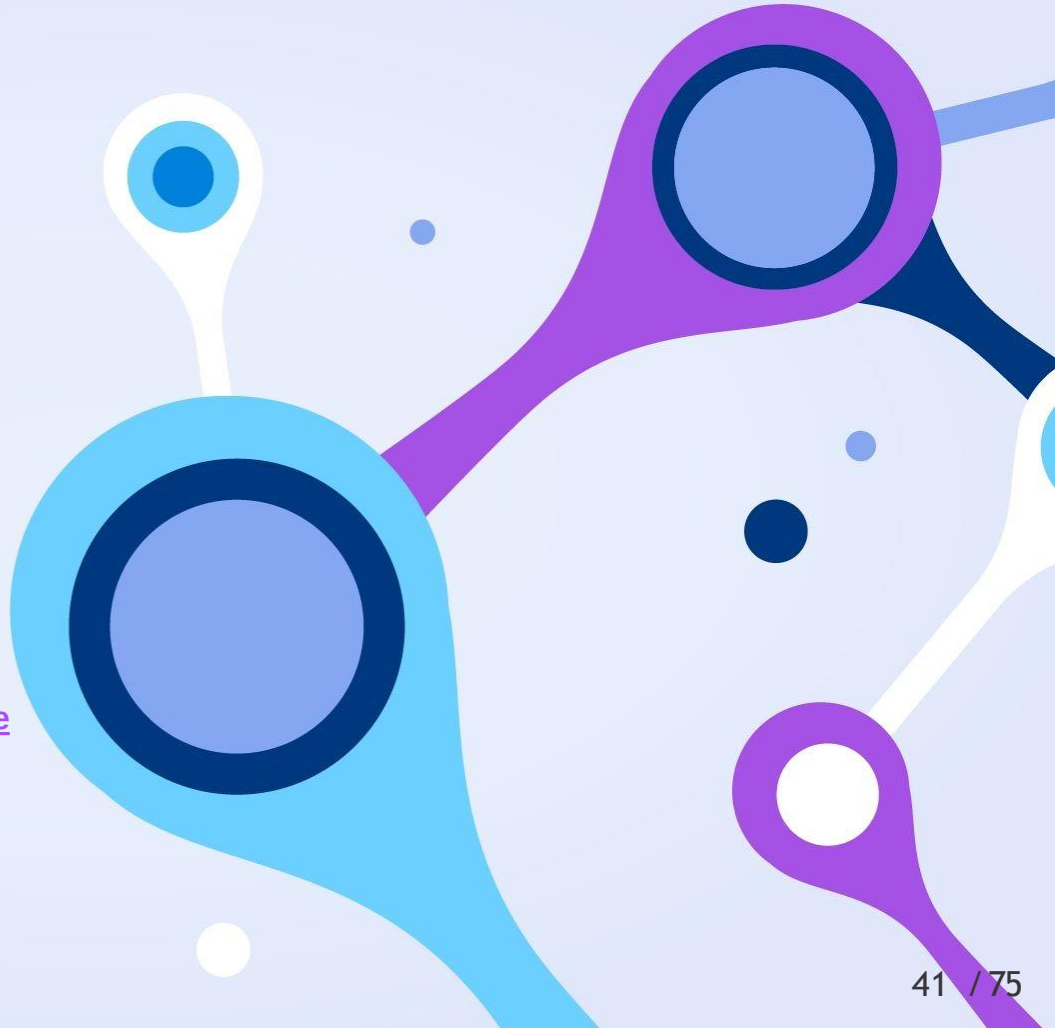
- یک بلاک معتبر پس از ایجاد در تمام شبکه پخش می شود.
- کار بر روی بلاک بعدی آغاز می شود.
- بلاک های ایجاد شده تغییرناپذیر هستند.
- هش بلاک در بلاک بعدی قرار می گیرد.

زنجیره ای از بلاک ها



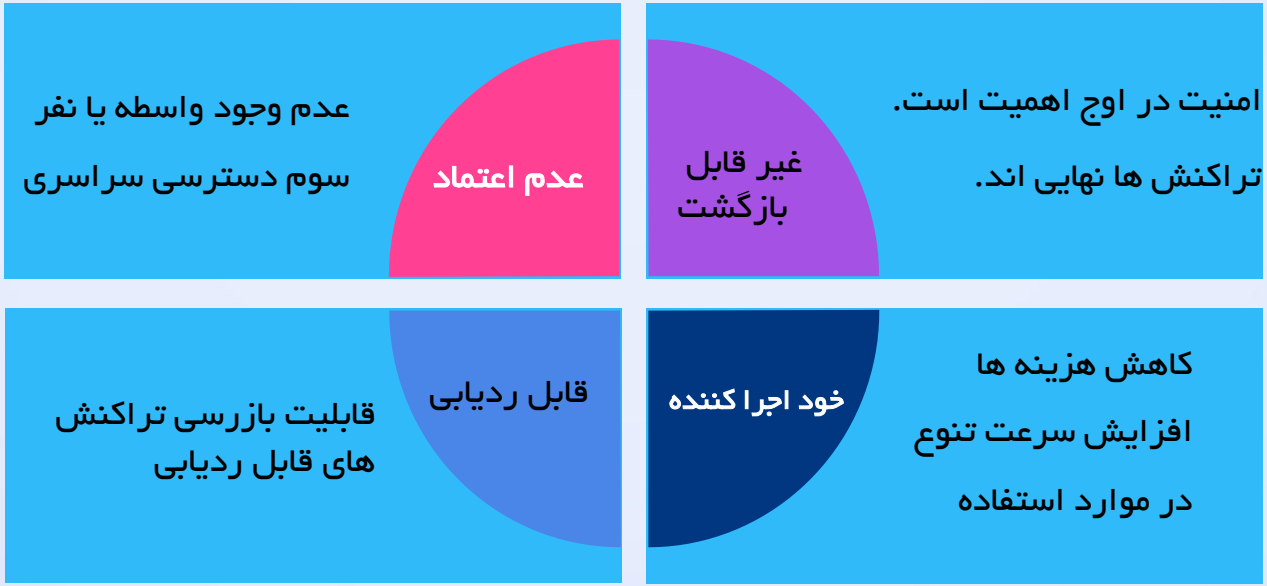
Additional Reading

1. A visual demonstration of a blockchain data structure
2. Bitcoin Wiki - Blockchain
3. Solidity Documentation Blockchain Basics
4. Directed Acyclic Graphs - Wikipedia
5. **https://observablehq.com/@consensys-academy/building-a-blockchain**

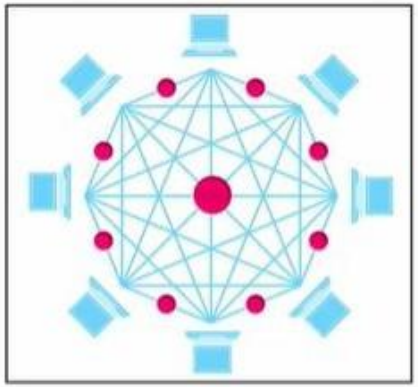


قراردادهای هوشمند

یک پروتکل کامپیوتری که به صورت دیجیتال مفاد یک قرارداد را اجرا می کند.



قراردادهای هوشمند



قرارداد های هوشمند لزوما روی بلاکچین نیستند، اما بلاکچین قابلیت های زیر را ارائه می دهد:

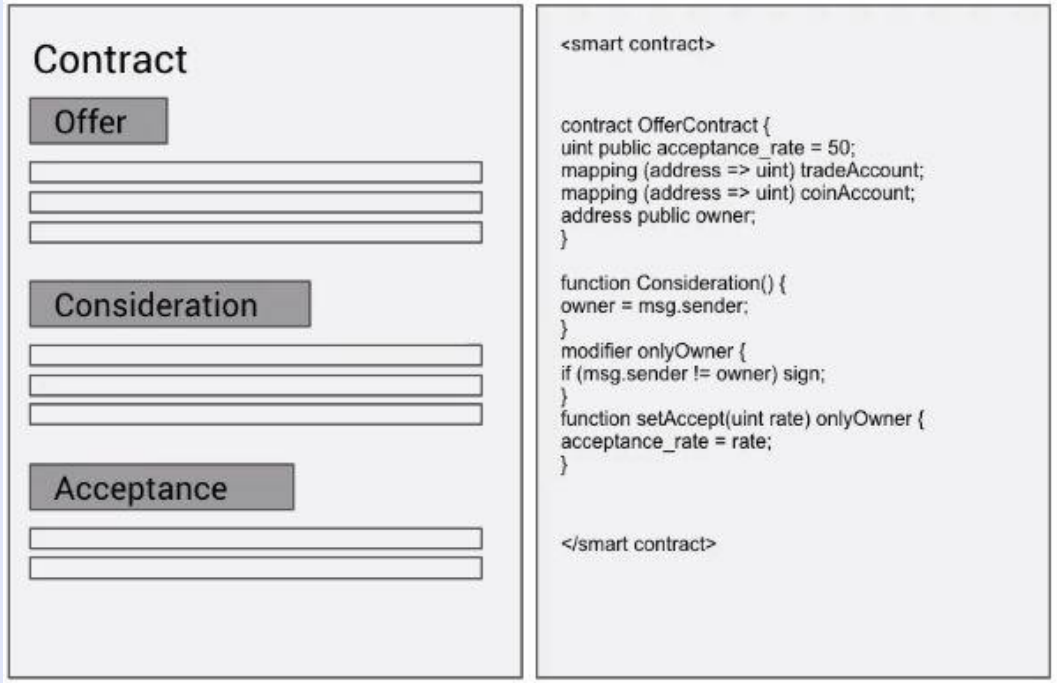
- یک سیستم بر پایه عدم اعتماد و امکان دسترسی سراسری
- تراکنش های برگشت ناپذیر و قابل ردیابی
- مکانیزم های برای امکان اجرای خودکار

تعاریف قرارداد هوشمند

۱ .کد ذخیره شده بر روی بلاکچین با قابلیت اجرای خودکار با استفاده از اعتماد و امنیت شبکهبلاکچین

۲ .منطق اپلیکیشن که به صورت غیرمتمرکز روی بلاکچین اتریوم و با استفاده از ماشین مجازی اتریوم اجرا می شود.

EVM

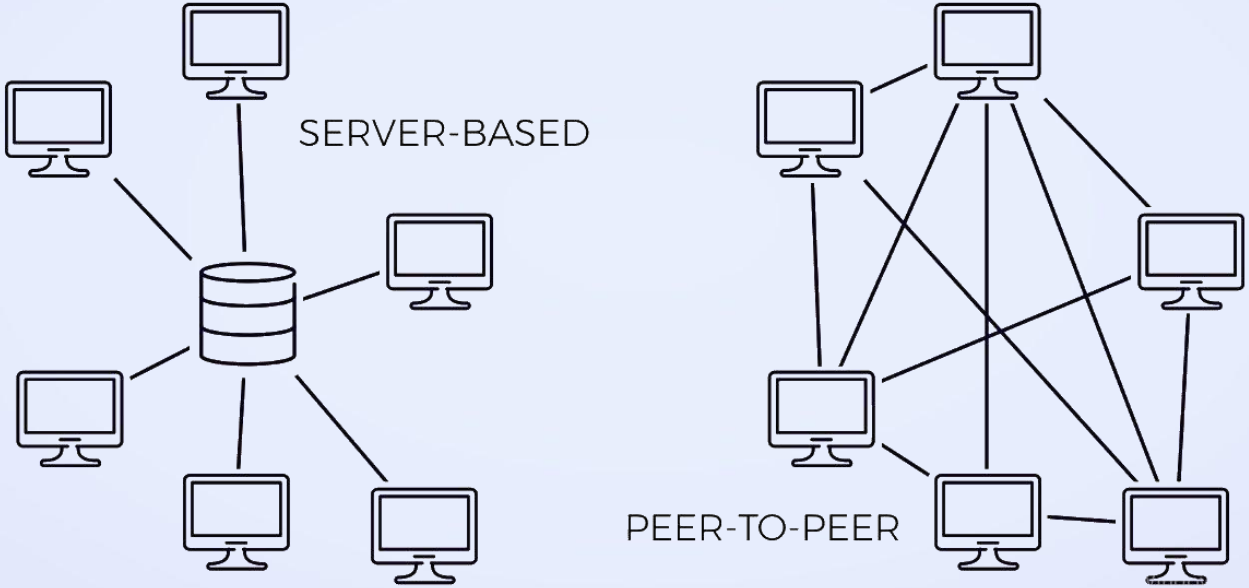


زبان های برنامه نویسی قرارداد های هوشمند

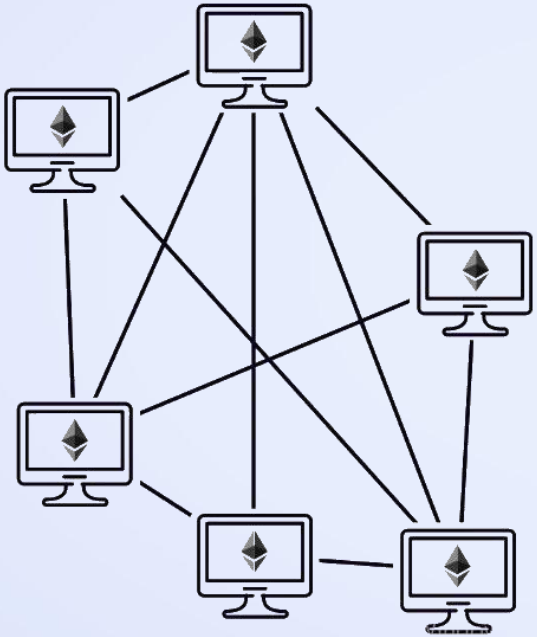
Solidity

Vyper

گره ها در شبکه



شبکه Peer-to-Peer



- گره ها با اجرای نرم افزار مناسب قادر به مشارکت خواهند بود.

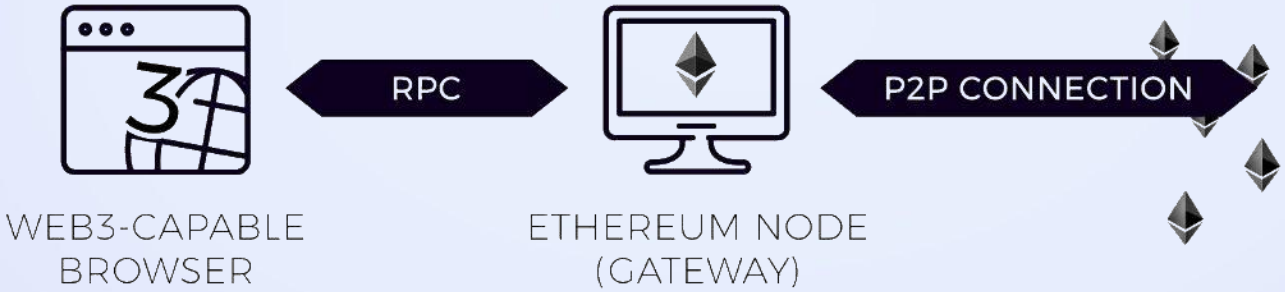
- این شبکه می تواند یک شبکه بلاکچین یا شبکه به اشتراک گذاری فایل دیگری باشد.

Bitcoin

Ethereum

BitTorrent

گره ها در شبکه



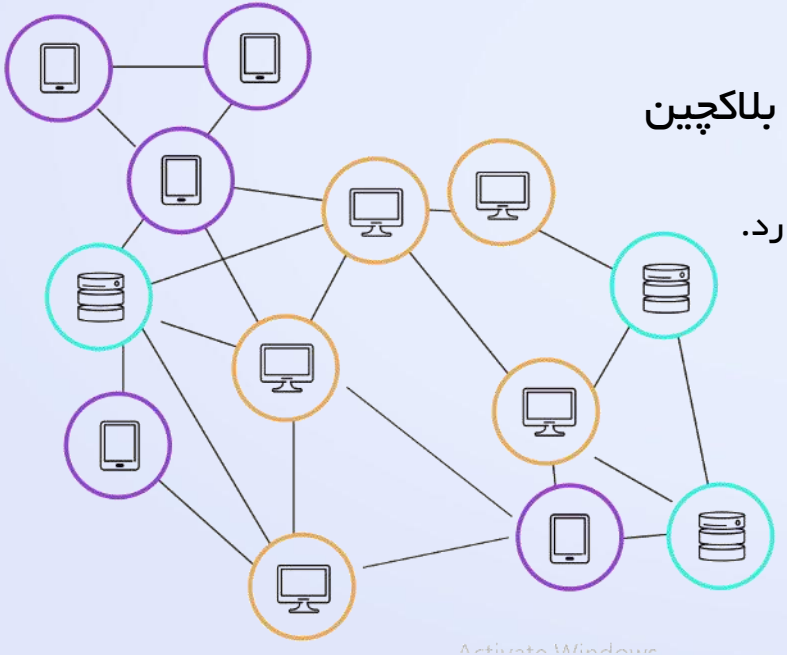
انواع گره در شبکه

گره ها به سه روش می توانند در شبکه مشارکت داشته باشند:

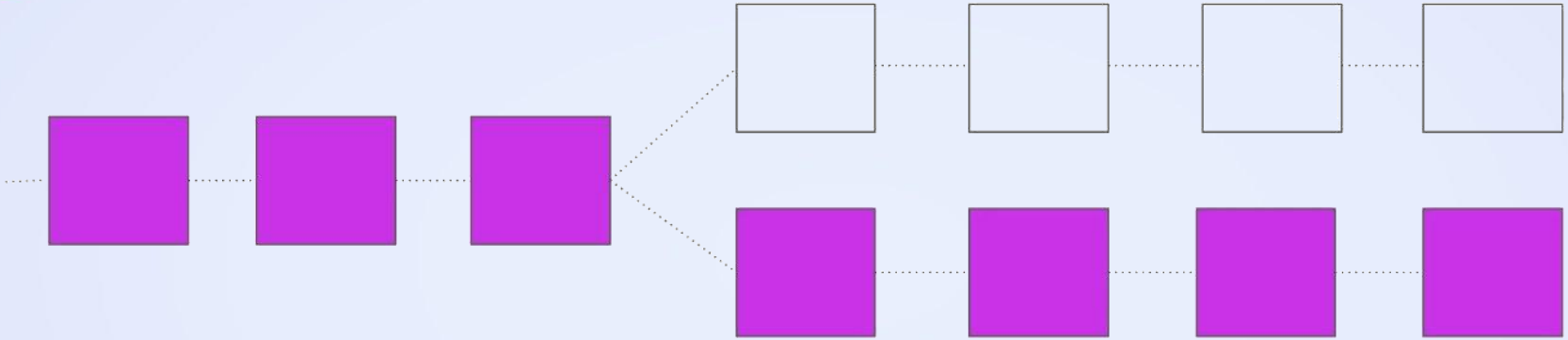
Full Node: کپی کامل بلاکچین

Run a Light Client: یک کپی کم عمق از بلاکچین

Mining: یک فرمول نود که امکان mineکردن هم دارد.



Forks



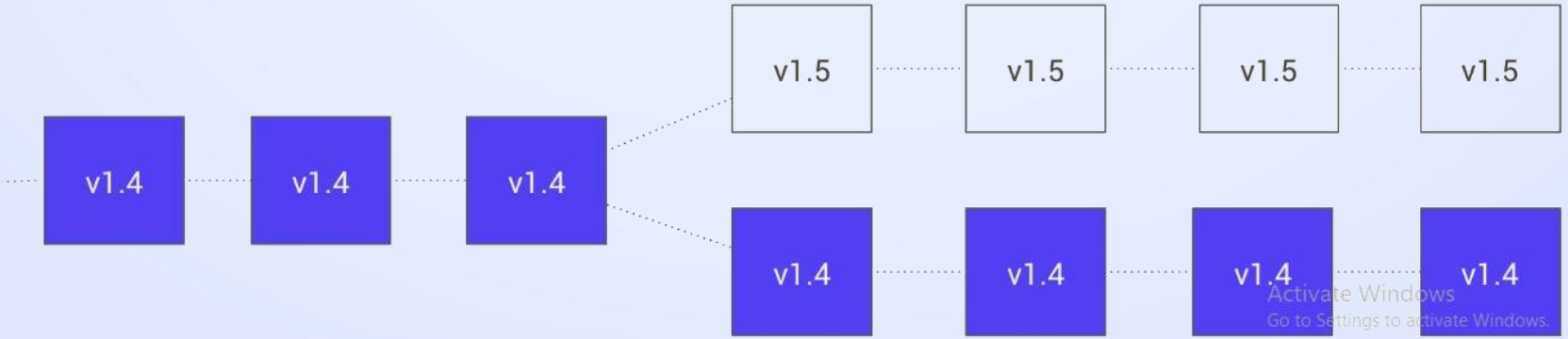
انشعاب در بلاکچین:

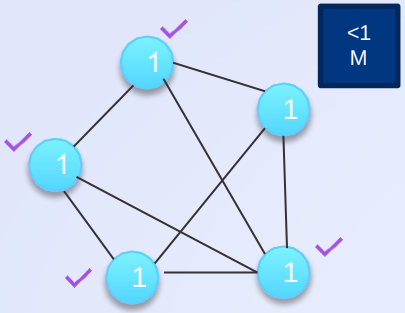
- عمدی یا غیرعمدی
- تغییرات در شبکه
- عدم توافق در کامیونیتی

- Hard Fork
- Soft Fork

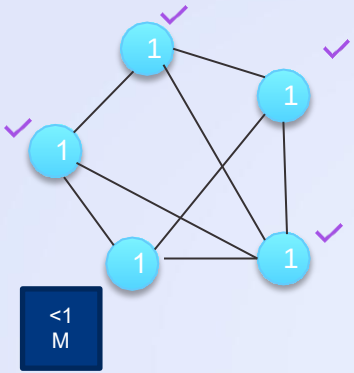
Hard Forks

زمانی اتفاق می افتد که در مورد تغییر پروتکل در شبکه توافق صورت گیرد. در این وضعیت بلاک های ایجاد شده توسط نود های به روز رسانی نشده توسط نود های به روز رسانی نشده رد خواهند شد.

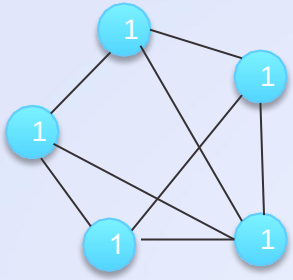




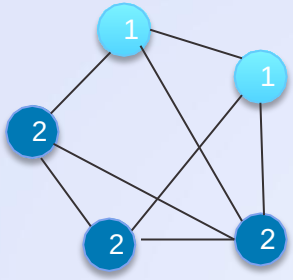
1001

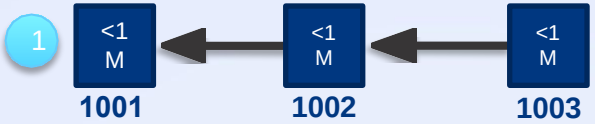

1
1
1
1
1
<1
M

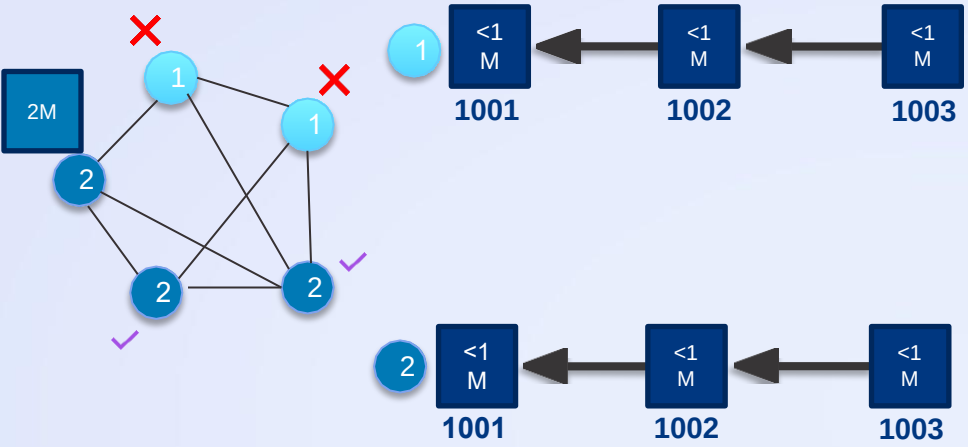

1
<1
M
<1
M
1001
1002

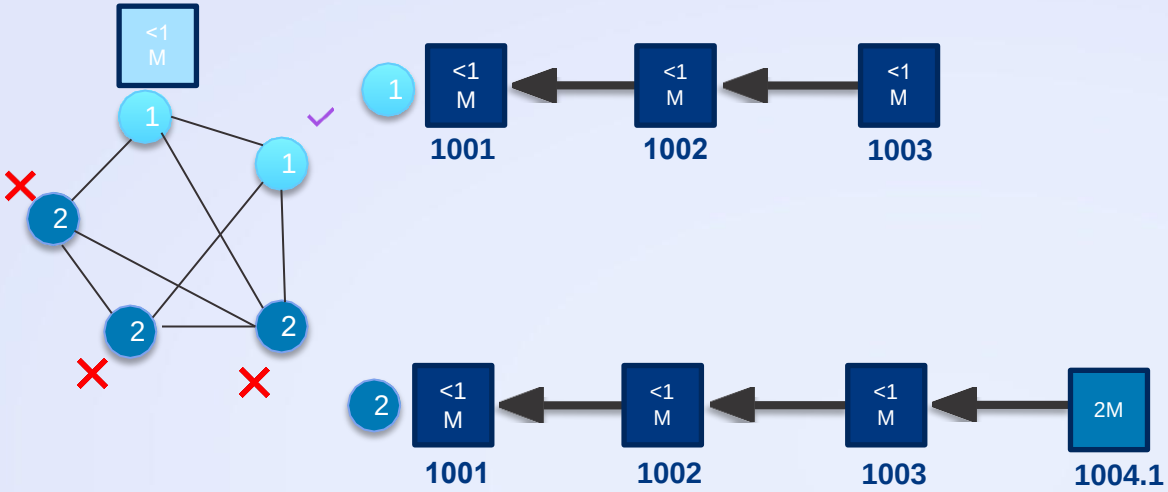

1
1
1
1
1

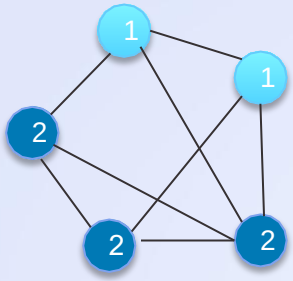

1
<1
M
1001
<1
M
1002
<1
M
1003


1
1
2
2
2

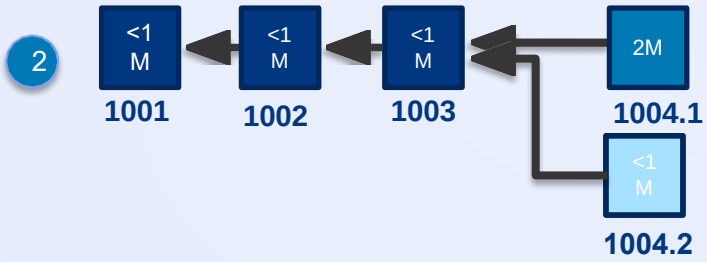

1
<1
M
<1
M
<1
M
1001
1002
1003

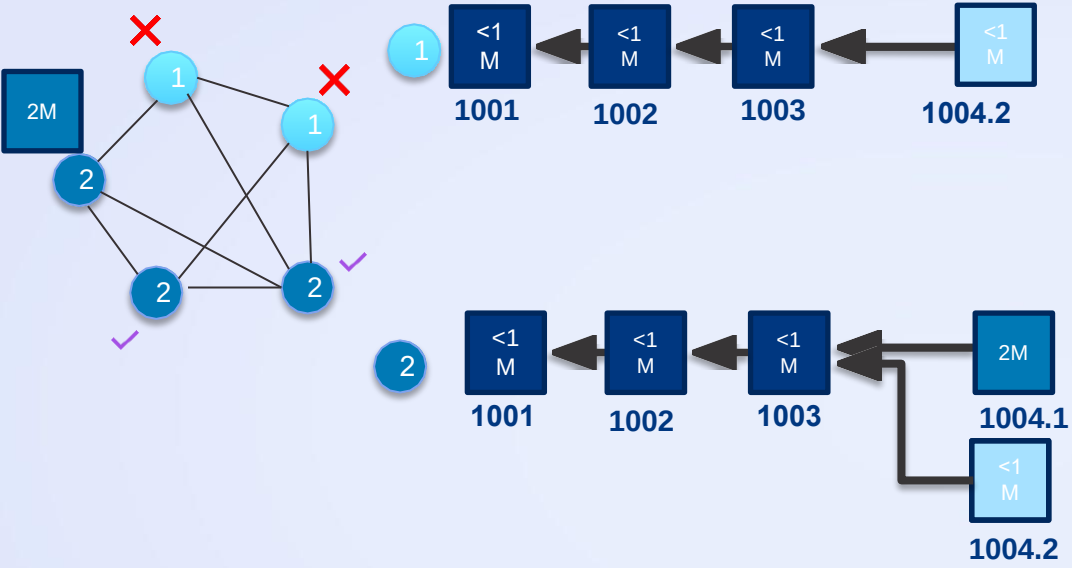

2M
1
1
2
2
2
1
<1
M
<1
M
<1
M
1001
1002
1003
2
<1
M
<1
M
<1
M
1001
1002
1003

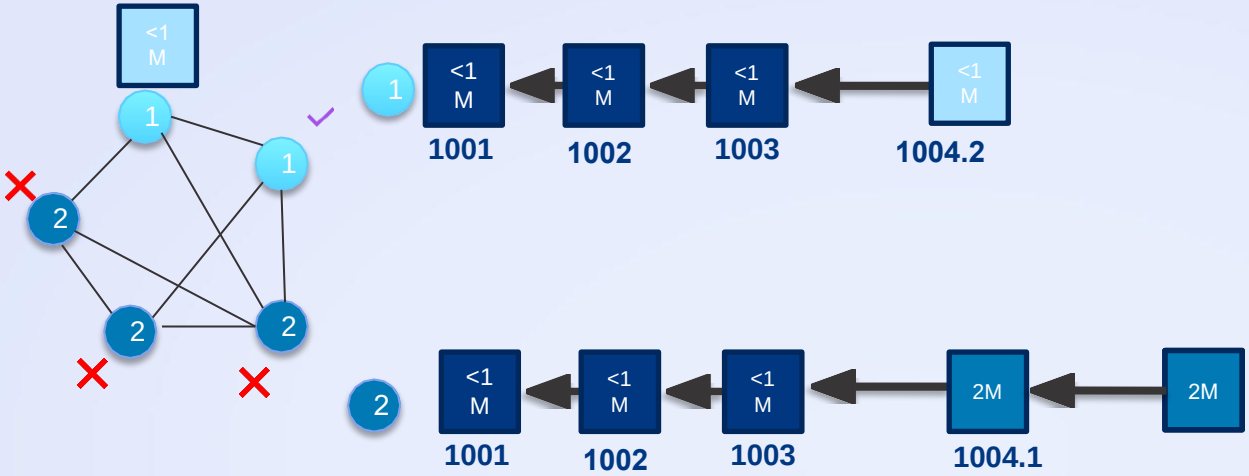

<1 M
1
1
2
2
2
1
<1 M
<1 M
<1 M
1001
1002
1003
2
<1 M
<1 M
<1 M
2M
1001
1002
1003
1004.1

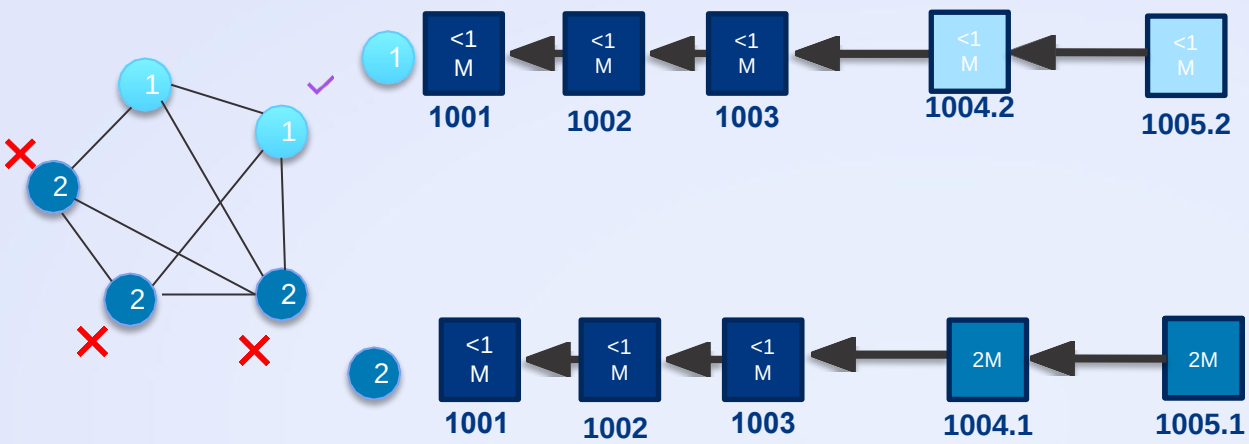

1
1
2
2
2


1
<1 M
<1 M
<1 M
<1 M
1001
1002
1003
1004.2


2
<1 M
<1 M
<1 M
2M
<1 M
1001
1002
1003
1004.1
1004.2

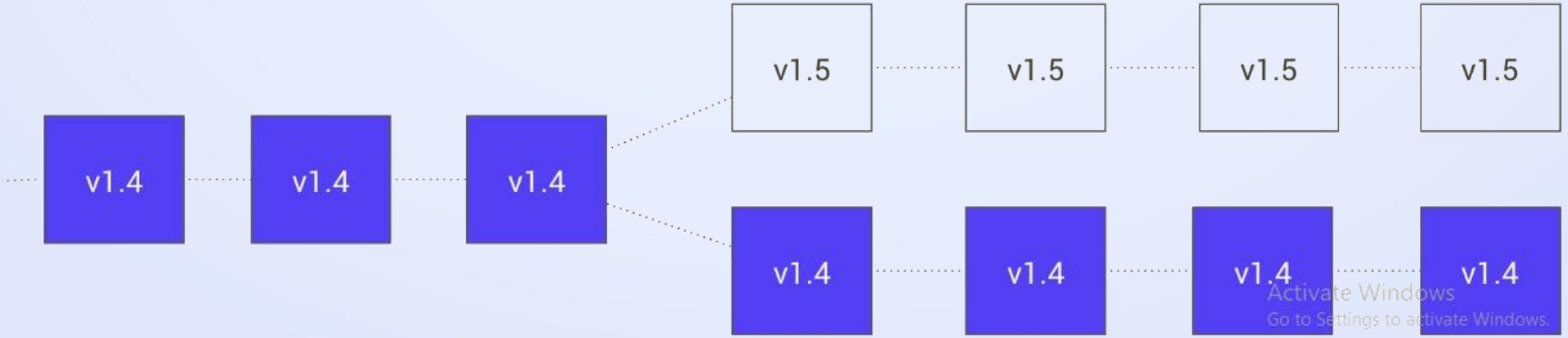

2M
1
1
2
2
2
1
<1 M
<1 M
<1 M
<1 M
1001
1002
1003
1004.2
2
<1 M
<1 M
<1 M
2M
<1 M
1001
1002
1003
1004.1
1004.2


<1 M
1
1
2
2
2
1
<1 M
<1 M
<1 M
<1 M
1001
1002
1003
1004.2
2
<1 M
<1 M
<1 M
2M
2M
1001
1002
1003
1004.1


1
1
2
2
2
1
<1 M
<1 M
<1 M
<1 M
<1 M
1001
1002
1003
1004.2
1005.2
2
<1 M
<1 M
<1 M
2M
2M
1001
1002
1003
1004.1
1005.1

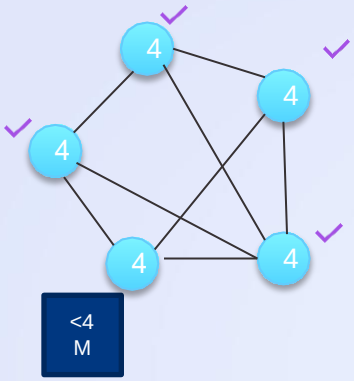
Hard Forks

- Chain جدید با Chain قبلی سازگار نیست
- این دو Chain دیگر هرگز با هم تلاقی نخواهند داشت
- در وضعیتی که یک Consensus مشخص در مورد یک تغییر شکل گرفته باشد استفاده می شود.
- اگر یک Chain برنده مشخص به ثبات نرسد پیامدهای آن می تواند پروژه را نابود کند

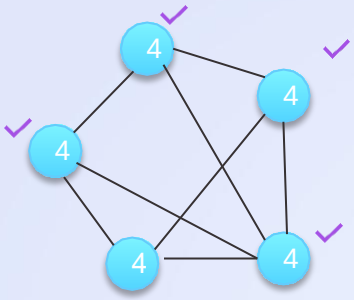


Soft Forks

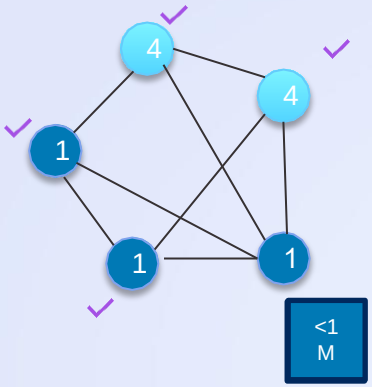
تغییرات با وضعیت اصلی نرم افزار قبل از وقوع fork سازگار هستند


4
4
4
4
4
<4
M

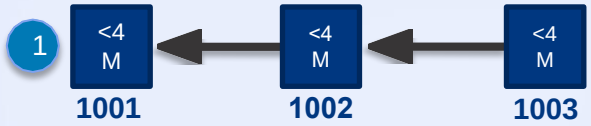

4
<4
M
1001
<4
M
1002

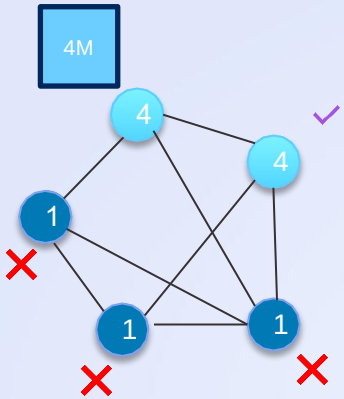

4
4
4
4
4


4
<4
M
1001
<4
M
1002
<4
M


4
4
1
1
1
<1
M

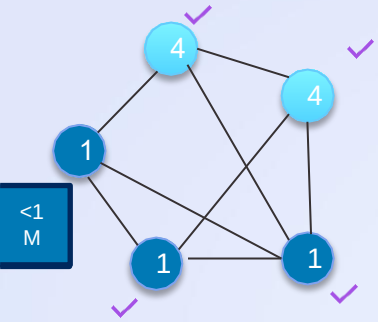

4
<4
M
<4
M
<4
M
1001
1002
1003

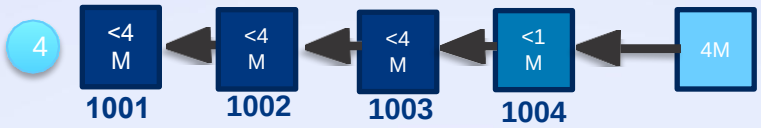

1
<4
M
<4
M
<4
M
1001
1002
1003


4M
4
4
1
1
1

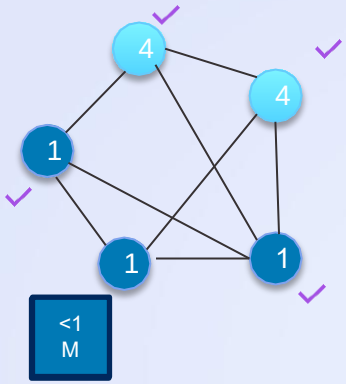

4
<4 M
1001
<4 M
1002
<4 M
1003
<1 M
1004

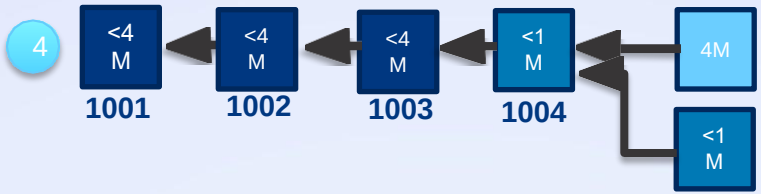

1
<4 M
1001
<4 M
1002
<4 M
1003
<1 M
1004


4
4
1
<1
M
1
1

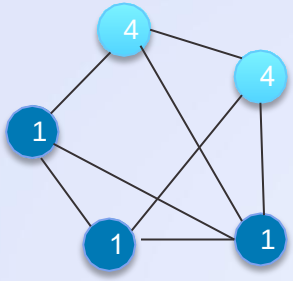

4
<4
M
<4
M
<4
M
<1
M
4M
1001
1002
1003
1004

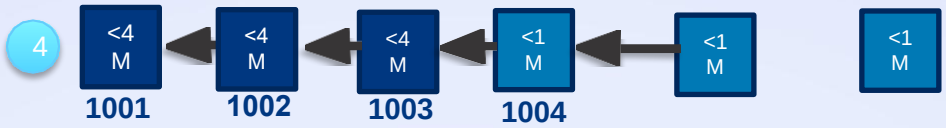

1
<4
M
<4
M
<4
M
<1
M
1001
1002
1003
1004



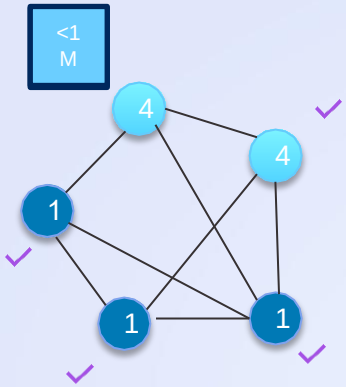


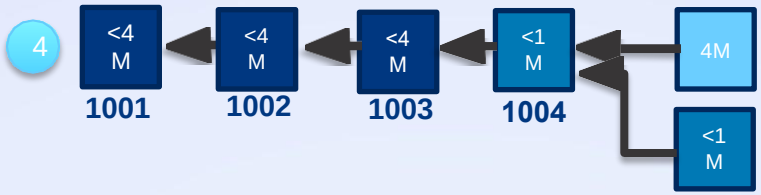
احتمال ۱ :ساخت بلاک ۱ مگابایتی توسط گره های به روز رسانی شده



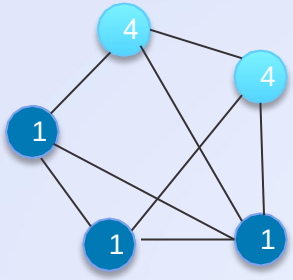


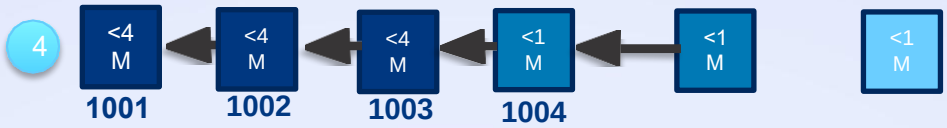
احتمال ۱ :ساخت بلاک ۱ مگابایتی توسط گره های به روز رسانی شده





احتمال ۲ :ساخت بلاک ۱ مگابایتی توسط گره های به روز رسانی شده





احتمال ۲ :ساخت بلاک ۱ مگابایتی توسط گره های به روز رسانی شده

- تغییرات با وضعیت اصلی نرم افزار قبل از وقوع fork سازگار هستند
- هر دو مجموعه قوانین در یک chain وجود دارند
- اگر ۵۱ %شبکه با قوانین جدید تطبیق پیدا کنند، قوانین قبلی دیگر پذیرفته نخواهد شد.
- معمولا برای تغییرات جزیی تر در شبکه استفاده می شود.

بلاک های تابع قوانین قبلی

بلاک های تابع قوانین جدید

Fork غیر عمدی

انشعاب در بلاکچین

- انشعاب در بلاکچین
- غیرعمدی
- اثبات کار
- ماینر ها به صورت همزمان یک بلاک جدید پیدا می کند