**Hashing**

آیا Hashing نوعی Encryption می‌باشد؟

آیا زمانی که چیزی را Hash می‌کنیم، می‌توانیم از Hash مربوطه به اصل آن چیز برسیم؟

آیا این‌که نمی‌توانیم از Hash یک چیز، به خود آن چیز برسیم، بدان معنا است که الگورتیم Hashing، یک الگوریتم محرمانه است که هیچ شخصی آن را ندارد؟

در واقع Hashing بر اساس سری فوریه بوده و تابعی (الگوریتمی) است که تابع معکوس ندارد! یعنی برگشت ناپذیر است!

f-1

For Each x, y

X <> Y 🡺 f(x) <> f(y)

P 🡺 q معادل است با ~q 🡺 ~p

f(x) = f(y) 🡺 x = y

چند الگورتیم مهم Hashing

MD5 (32 Hexadecimal Digits)

SHA-1 (40 Hexadecimal Digits)(160 Bits)

SHA-2

SHA-224

SHA-256 (64 Hexadecimal Digits)(256 Bits)

SHA-384

SHA-512

**کاربردهای مهم Hashing**

* ذخیره سازی گذرواژه در بانک اطلاعاتی و فرآیند Login

مشکل MD5 چیست که دیگر استفاده نمی‌شود؟

برای بدست آوردن گذرواژه، باید فرآیند (زمان) زیادی صرف کرد!

Password: 8 (0-9)(a-z)(A-Z)

10 + 32 + 32 = 74 حالت

74 \* … \* 74 = 748 = ۸۹۹,۱۹۴,۷۴۰,۲۰۳,۷۷۶ حالت

با توجه به این حجم حالت، پس چرا بعضی از سایت‌ها زمانی که Hash چیزی را می‌دهیم در کسری از ثانیه آن را به ما می‌دهند؟

74 \* ۱۲,۱۵۱,۲۸۰,۲۷۳,۰۲۴

سالت هش (Salt Hash) چیست؟

* یک آنتی‌ویروس بسیار قوی برای سازمان‌های امنیتی و خاص
* کنترل فایل‌هایی که از اینترنت دانلود می‌کنیم، چیزی شبیه Checksum
* پیدا کردن فایل‌های مشابه بر روی یک Hard Disk (موزیک‌ها، فیلم‌ها، مستندات و غیره)
* برای قفل‌گذاری بر روی برنامه‌ها با استفاده از شماره سریال HDD و یا CPU
* برای Data Integrity یک Record در بانک اطلاعاتی.
* برای Sign کردن یک محتوا (Content) که البته باید با Asymmetric Encryption ترکیب شود.
* برای Blockchain

Diagram

Description automatically generated

<https://ethereum.org/en/developers/docs/blocks/>

* <https://ethereum.org/en/developers/docs/gas/>
* <https://ethereum.org/en/developers/docs/transactions/>
* <https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/mining/>

**WHAT'S IN A BLOCK?**

**timestamp** – the time when the block was mined.

**blockNumber** – the length of the blockchain in blocks.

**baseFeePerGas** - the minimum fee per gas required for a transaction to be included in the block.

**difficulty** – the effort required to mine the block.

**mixHash** – a unique identifier for that block.

**parentHash** – the unique identifier for the block that came before (this is how blocks are linked in a chain).

**transactions** – the transactions included in the block.

**stateRoot** – the entire state of the system: account balances, contract storage, contract code and account nonces are inside.

**nonce** – a hash that, when combined with the mixHash, proves that the block has gone through proof-of-work.