Hash

هش چیست؟

شا ها خانواده ای از توابع هش هستند که توسط آژانس امنیت ملی در ایالات متحده توسعه یافته است. یک تابع هش به عبارت ساده، یک الگوریتم ریاضی است که یک ورودی (هر داده‌ای مانند یک فایل یا یک رمز عبور) را دریافت میکند و یک رشته کاراکتر با اندازه ثابت تولید می‌کند که مقدار هش یا خلاصه آن داده ورودی است. اما بخاطر داشته باشید که این فرایند شامل رمزگذاری نمی‌شود.

یکی از معروف ترین هش‌هایی که امروزه به کار می‌رود شا256 است این هش از نسخه قبلی خود یعنی شا1 امنیت بیشتری دارد و اشکالات قبلی را پوشش داده است و امروزه در جاهای مختلفی به چشم می‌خورد.

ویژگی های شا256:

منحصر به فرد بودن

هنگامی که از تابع هش SHA 256 استفاده می‌شود، ورودی‌های مجزا همیشه مقادیر هش منحصربه‌فردی تولید می‌کنند؛ حتی یک تغییر کوچک در ورودی منجر به یک مقدار هش بسیار متفاوت می‌شود. این به عنوان اثر بهمن (Avalanche Effect) شناخته می‌شود. علاوه‌براین، فارغ از اندازه ورودی، مقدار هش همیشه ۲۵۶ بیت خواهد بود.

برگشت ناپذیری

مقادیر هش ایجاد شده با استفاده از الگوریتم SHA 256 از نظر محاسباتی برای مهندسی معکوس غیرممکن است؛ به این معنی که شما نمی‌توانید داده‌های ورودی اصلی را از مقدار هش بدست آورید. این تضمین می‌کند که حتی اگر مقدار هش در دسترس عموم باشد، داده‌ها محافظت می‌شوند که اشتراک‌گذاری فایل‌ها را به صورت عمومی آسان می‌کند.

قطعیت

SHA 256همیشه مقدار هش یکسانی را برای یک ورودی خاص تولید می‌کند .*یعنی اگر یک ورودی خاص به این الگوریتم بدهید، همواره مقدار هش یکسانی دریافت خواهید کرد* .این ویژگی ثبات در فرآیند هش را تضمین می‌کند و امکان تایید داده‌ها را در سیستم‌ها فراهم می‌کند.

کاربرد های شا256:

امضای دیجیتال

در امضاهای دیجیتال، الگوریتم SHA 256 می‌تواند یکپارچگی و صحت اسناد و پیام‌ها را تضمین کند. به عنوان مثال، SHA 256یک مقدار هش از محتوای امضا شده تولید می‌کند که به عنوان **اثر انگشت دیجیتال** منحصر به فرد عمل می‌کند. سپس از کلید خصوصی امضا کننده برای رمزگذاری مقدار هش و ایجاد امضای دیجیتال استفاده می‌شود.

در انتها، گیرنده یک برنامه، می‌تواند امضا را با استفاده از کلید عمومی مربوطه، رمزگشایی کرده و مقدار هش سند را محاسبه کند.

هش رمز عبور

یکی از محبوب‌ترین کاربردهای الگوریتم SHA 256 هش کردن رمز عبور است. شرکت‌ها به جای ذخیره رمزهای عبور واقعی، مقادیر هش خود را استخراج می‌کنند. این فرایند برای کاربر بسیار ایمن‌تر است. هر بار که رمز عبور خود را وارد می‌کنید، سیستم یک مقدار هش جدید دریافت کرده و بررسی می‌کند که آیا با رمز ذخیره شده در پایگاه داده مطابقت دارد یا خیر.

فناوری بلاکچین

فناوری بلاکچین نیز از الگوریتم SHA 256 برای ایمن‌سازی، یکپارچگی و تغییرناپذیری داده‌های ذخیره شده در بلوک‌ها استفاده می‌کند. از آنجایی که *هر بلوک در یک بلاکچین دارای یک اثر انگشت دیجیتال منحصر به فرد است، هیچ‌کس نمی‌تواند محتوای بلاک را بدون تغییر دادن هش تغییر دهد* .به عبارت دیگر، با پیوند دادن بلوک‌ها با استفاده از مقادیر هش، بلاکچین یک دفتر کل شفاف و ضد دستکاری ایجاد می‌کند که هر کسی می‌تواند آن را تایید کند.

یکپارچگی فایل

هش کردن می‌تواند به محافظت از یکپارچگی هر فایلی کمک کند .*اسناد، فیلم‌ها، فایل‌های اجرایی نرم‌افزار و هرگونه فایل دیگری می‌تواند به کمک این فرایند، یکپارچه و ایمن شود* .این موضوع بسیار مهم است؛ زیرا در حین استفاده از امضای دیجیتال یا به‌روزرسانی یک نرم‌افزار، می‌توانید تایید کنید که هیچ‌یک از این فایل‌ها دستکاری نشده است.

گواهینامه‌هایSSL/TLS

توابع هش مانند SHA به بهتر شدن مرور وب کمک می‌کنند SHA 256 .می‌تواند با ایجاد امضای دیجیتالی که دستگاه شما توانایی تایید آن را دارد، به ایمن کردن گواهینامه‌های SSL/TLS (امنیت لایه حمل و نقل داده) کمک می‌کند. به عنوان مثال، هنگامی که یک سرور گواهی TLS خود را به مشتریانی مانند مرورگرهای وب ارائه می‌دهد، مشتری می‌تواند از کلید عمومی مربوطه برای رمزگشایی و تایید امضا استفاده کند. اگر گواهی SSL توسط یک مرجع گواهی معتبر صادر نشده یا دستکاری شده باشد، مقادیر هش با هم مطابقت ندارند.

تبدیل متن به هش در پایتون

```
import hashlib # import hash library

text = "mr"
byte_text = bytes(text, "utf-8")
Hash_text = hashlib.sha256(byte_text).hexdigest()
Hash_text
```

```
'79032b8a85663acddd601fd25371b7cb91f3ee6fbe68215f3cf6d4736bcbeea9'
```

نمونه هایی دیگر از هش ها

```
from hashlib import sha1
from hashlib import sha512
from hashlib import md5

text = "Hello word"
byte_text = bytes(text, "utf-8")

sha1_text = sha1(byte_text).hexdigest()
sha512_text = sha512(byte_text).hexdigest()
md5_text = md5(byte_text).hexdigest()

print("The text in sha1 hash: ", sha1_text)
print("The text in sha512 hash: ", sha512_text)
print("The text in md5 hash: ", md5_text)
```

```
The text in sha1 hash:  739921b9bee642f0c9466d88e6a9de77be52d91f
The text in sha512 hash:  e4873cf35753072cee0da8f8f287d8472ab47ec195eb68971c449c7039232ace134f3369c83e64c005a1609a9d8d2c84908d9f8d3c12281e8ce4db5d36fe0222
The text in md5 hash:  139b4974af77ddb8e72bf344f0e658ba
```

```python
import streamlit as st
import hashlib

def check_text(text, person):
    targets = ['8f434346648f6b96df89dda901c5176b10a6d83961dd3c1ac88b59b2dc327aa4']
    text = text.split()
    text_hash = list()
    for word in text:
        bytes_word = bytes(word, "utf-8")
        word_hash = hashlib.sha256(bytes_word).hexdigest()
        text_hash.append(word_hash)

    for word in text_hash:
        if word in targets:
            st.session_state.Violation = person
            return False
    st.session_state.Violation = ''
    return True

def UI(chat):
    if "Violation" not in st.session_state:
        st.session_state.Violation = ''
        st.session_state.chat = []

    st.set_page_config(page_title="کاربرد هش ها")
    st.write("<div style='text-align: center'> <h1>کاربرد هش<h1> </div>", unsafe_allow_html=True)


    col1, col2, col3 = st.columns(3)
    with col1:
        p1_text = st.text_input(label="علی:")
        if st.button(label="ارسال", key="p1"):
            security = check_text(p1_text, "علی")
            if security:
                st.session_state.chat.append("علی:" + p1_text)
        if st.session_state.Violation == 'علی':
            st.write(F"""تخلف نوشتاری از سمت {st.session_state.Violation}""")

    with col3:
        p2_text = st.text_input(label="امیر محمد:")
        if st.button(label="ارسال", key="p2"):
            security = check_text(p2_text, "امیر محمد")
            if security:
                st.session_state.chat.append("امیر محمد:" + p2_text)
        if st.session_state.Violation == 'امیر محمد':
            st.write(F"""تخلف نوشتاری از سمت {st.session_state.Violation}""")

    with col2:
        st.write(st.session_state.chat)
        # text = ""
        # text += chat
        # st.write(text)



def main():
    UI(chat)

chat = """"""
main()
```


کاربرد هش
علی:
[]
امیر محمد:
ارسال
ارسال


کاربرد هش
علی:
salam
[
0 : "علی:salam"
]
امیر محمد:
ارسال
ارسال


کاربرد هش
علی:
mr
ارسال
[
0 : "علی:salam"
]
امیر محمد:
ارسال
تخلف نوشتاری از سمت علی

منابع:

/https://wallex.ir/blog/sha256-algorithm

https://en.wikipedia.org/wiki/SHA-2

آدرس گیت‌هاب:

https://github.com/MrShotgun1182/Hash