Mark Lucernas
MATH 254
Mohamed Benbourenane
June 22, 2021

Secure Hash Algorithms (SHA)

These days and age, technology has made quite an extreme advancement in how things are done since the birth of computers. Modern-day people consider mailing messages through the old-fashioned snail mail an excruciatingly long process: Construct a structured message on a piece of paper neatly, seal the paper tight, put a stamp on the envelope with the addresses, drop the mail to the nearest post office, and hope for the best that the mailman secure and deliver your mail timely. Knowing that there is no guarantee that your message will be delivered on time, if at all, is up to the mailman's ability to successfully drop hundreds of mails in mailboxes per day. Although rare, about 3% of the 130 billion postal mails per year are lost, according to Wikipedia. What if those mails contain sensitive information such as your tax report or a 1 in 13,983,816 chance you are supposed to receive a 100 million dollar check from a winning lottery ticket via mail. How can we protect our sensitive data in the event the mails are misdelivered? Maybe encrypt your message using Caesar cipher? Or maybe invent a new esoteric language only you and the receiver can comprehend? In any case, delivering messages on foot is hands down unreliable.

The technology of today made these massage delivery process trivial with the power of the internet; a vast network that connects computers all around the world in a web-like network. The internet uses communications protocols such as TCP/IP to follow instructions on how to send or receive messages, much like how we package a message in an envelope and put a stamp on it as the postal services require. Mails are sent via Simple Mail Transfer Protocol (SMPT) as the universal standard. But does this mean a guaranteed delivery? Not necessarily. According to a blog post "Email Delivery is Never Guaranteed" by Carly Brantz, emails goes through all sorts of checks including checks from your Internet Service Provider (ISP) to protect you from malicious emails, and 20% of the time it goes astray into cyberspace or someone else's spam folder. Hence, bringing us to the main point, the extra-thick layer of security using Secure Hash Algorithms or SHA.

SHA is a family of cryptographic functions published by the National Institute of Standards and Technology (NIST). These algorithms map data of arbitrary size into a bit-array of a fixed size (the hash value). In other words, it creates a unique id to any data passed into the algorithm that is practically impossible to reverse. A one-way function. SHA-0 and SHA-1 are the earliest versions of the SHA family that stores the hash value into a 160 or (5*32) bit-array. If you think that's enough to uniquely identify every data you pass in the function, you guessed wrong. For a long time, SHA-1 has been the standard until 2010 when it was officially replaced by SHA-256 and SHA-512 or simply SHA-2 designed by the National Security Agency (NSA), with significant improvement on hash value "collision", better randomization algorithm, and significantly more bit size to store the message digest or the hash value in. Below is a sample application of SHA-256:

```
echo "This is a secret message" > message.txt

sha256sum message.txt
```

The code snippet above stores a message "This is a secret message" into a file called message.txt in the file system. Passing the file into SHA-256 produces a unique 64-long hash value:

```
47985b79593f1df4031e333bc1d1f83906d9ff5493ece9549e5020b26642e46b
```

Now, I will append a period into the message.txt file at the end of the message then re-hash the file which gives a hash value of

```
6432f513cfd40d47c8584494c0524468257e50dc1a0422f73becac85189543f8
```

that is an entirely different message digest. Nevertheless, no matter how random the results may seem to be, the function produces the same hash value for the same data, given the state of the data is never altered.

Going back to the email story, SHA is used as a Digital Signature in which only the user(s) who knows the hash value of the data packet can access the contents of the data. Don't even think about decrypting a one-way hash function as it would probably take more than a normal human life-span on earth even with the all the computing power you can muster. So, even when your tax report ends up in a malicious user's junk mail, it would be close to impossible to decrypt, for now.

Another popular use case outside of messaging services is the new kid in the block, Cryptocurrency. Bitcoin, the first cryptocurrency invented by an anonymous person that goes by the name of Satoshi Nakamoto, is a decentralized digital asset designed to work as another medium of exchange. Coin ownership records are stored in a database called the blockchain using strong cryptography to secure transaction records, control the creation of new coins and verify the transfer of ownership. Bitcoin uses the SHA-256 hash algorithm as the basis of creating a coin in the process called "Bitcoin mining", where miners try to solve a mathematical problem and puzzles in return for bitcoins. Because Bitcoin, in contrast to fiat currency (government-issued), in essence, is decentralized, where no federal reserve to validate the coins and their transactions. Therefore, Satoshi Nakamoto proposed a solution to incentivize this validation problem for those who are willing to expend computational power, while promoting competition among other miners. The miner who solves the problem the fastest is rewarded a new 12.5 BTC (currently) created out of thin air. Of course, this involves an extremely convoluted process and a lot of hashing power to even have the slightest chance to go against the mining giants. There are other ways to get around the competition and to be rewarded only for what you have contributed by joining a mining pool. But that's for another story.

In conclusion, SHA has been one of the most important algorithms in terms of software authentication and data encryption protocols. It offers fast, unique, and irreversible hash functions that have an infinitesimally small margin for error. Although SHA-2 will still be around for many years, the state-of-the-art version of SHA is the SHA-3 released on August 5, 2015, by NIST that offers faster hashing and overall improved security. There are other use cases for SHA such as password verification, data randomization, and more. Technology is ever-growing and these software programs can only get better and even more powerful over time.

Works Cited

Email Delivery is Never Guaranteed by Carly Brantz

- https://sendgrid.com/blog/email-delivery-is-never-guaranteed/

Considering the Percentage of Lost Mail in US Postal System — Doculivery Solves Many Problems

- https://natpay.com/blog/considering-the-percentage-of-lost-mail-in-us-postal-system-doculivery-solves-many-problems/

SMTP Email Delivery 101: When Is An Email Actually Considered Delivered? by John Porrini

- https://www.socketlabs.com/blog/smtp-email-delivery/

SHA: Secure Hash Algorithm by Computerphile

- https://www.youtube.com/watch?v=DMtFhACPnTY

Secure Hash Algorithms by Wikipedia

- https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

Cryptocurrency by Wikipedia

- https://en.wikipedia.org/wiki/Cryptocurrency

How Bitcoin mining really works by Subhan Nadeem

- https://www.freecodecamp.org/news/how-bitcoin-mining-really-works-38563ec38c87/

Bitcoin Mining Explained by Shivam Arora

- https://www.simplilearn.com/bitcoin-mining-explained-article