COTW: Cryptography Decrypted

Cryptographic algorithms perform a variety of essential functions, including as a key component of cryptocurrencies. From the Caesar Cipher to ECDSA, we review the basics of cryptography in this week's Chart of the Week.

Cryptology Defined: From secure network communication to user authentication, data integrity verification, and disk encryption, cryptography performs a variety of functions and helps power many real-world use cases like online banking, secure messaging, and cryptocurrencies. Cryptography is the study of the techniques for securing communication and data in the presence of adversaries, and together with cryptanalysis, or the study of decryption without possession of the secret key, forms the larger discipline of cryptology. Cryptography uses various encryption algorithms referred to as a cipher to secure communication, where a cipher and a secret key transform input data known as plaintext into encrypted output known as ciphertext. Strong cryptographic algorithms will adhere to the key principles of confidentiality, integrity, non-repudiation, and authentication, and will utilize entropy and computation to protect the plaintext and secret key even when an adversary has accessed the ciphertext and understands the inner workings of the cryptographic algorithm employed. Given its central role in cryptocurrencies and beyond, we review the basics of cryptography, split out into symmetric cryptography, asymmetric cryptography, and cryptographic hashing functions.

Symmetric Cryptography: Symmetric cryptography uses the same secret key for encryption of the plaintext and decryption of the ciphertext, thus requiring both the sender and receiver of the message to possess the secret key. Symmetric cryptography is generally more simple and less computationally intensive (read faster) than more advanced forms of cryptography but requires a way to securely share the secret key and also requires a separate secret key for each pair of users in a network. Symmetric key cryptography is often therefore used for bulk encryption of data at rest since the need to share the secret key is removed when only one person is accessing the data. To illustrate how symmetric cryptography works, we provide the following examples:

- Monoalphabetic Ciphers: A monoalphabetic cipher is a cipher in which the letters of the plaintext are mapped to ciphertext letters based on a single alphabetic key. A simple example is the Caesar Cipher, where each letter in the plaintext is shifted a fixed number of positions down the alphabet to create the ciphertext, with the fixed number being the secret

key. For example, if the secret key is two, A is shifted two positions to become C, B becomes D, C becomes E, and so on. The sender may then encrypt the message using this shift, and the recipient can decrypt it by reversing the shift - plaintext of “GSR” is encrypted to ciphertext of “IUT”, and the recipient knowing the secret key of two can simply shift each letter back two spots to decrypt to plaintext. One issue with the Caesar Cipher is that all one needs to do to decrypt it is to try all possible shifts, equating to 25 shifts for the English language and making it a poor encryption algorithm.

- Keyword Cipher: A keyword cipher is another type of monoalphabetic cipher, but rather than using a simple alphabetic shift, it inserts any keyword, for example, “cipher,” at the beginning of the mapped substitution alphabet before continuing on at the end of they keyword using the traditional alphabet and skipping over any letters already used in the keyword. For example, using our keyword of “cipher”, the substitution alphabet would be “CIPHERABDFGJKL…” so A would be mapped to C, B to I, C to P, etc., and our plaintext of “GSR” would map to “ASQ”. While more secure than the Ceasar Cipher, a keyword cipher is prone to frequency analysis, where a frequently occurring letting in the ciphertext is more likely to map to a commonly used letter like e, t, or a, and this information can ultimately be used to decrypt the message even without knowing the keyword.
- Polyalphabetic Cipher: Rather than use a single, constant alphabetic key as in a monoalphabetic cipher, a polyalphabetic cipher uses multiple substitution alphabets, therefore varying the mapping of the plaintext letters to the substitution alphabet (ie. a specific letter will map to various letters rather than a specific, static one). The Vigenère cipher is perhaps the most well-known polyalphabetic cipher and uses a table of alphabets where the alphabet is written out 26 times in different rows, with each row shifting the alphabet one letter to the left compared to the prior row. The message sender then uses a repeating keyword equal in length to the plaintext message to encrypt the message using the table of alphabets and the message can similarly be decrypted using this table and the keyword. The Vigenère cipher was invented in 1553 and was not decrypted for over three centuries, earning it the title of “le chiffrage indéchiffrable.”
- Data Encryption Standard (DES): DES was the result of research conducted by IBM in the late 1960s, eventually being commercialized with input from the NSA in the early 1970s. In 1977, DES was adopted by the National Institute of Standards and Technology as an official Federal Information Processing Standard for the encryption of sensitive but unclassified government information. DES is a type of block cipher, where a key and algorithm is applied to a block of data, in this case, 64 bits in size, rather than to each binary digit in a data stream one bit at a time as in a stream cipher. DES consists of 16 rounds of encryption utilizing substitution and transposition. A different key is used for each round of encryption and the order of the 16 keys is reversed to decrypt the ciphertext. DES’s short key size makes it relatively insecure, and it was eventually broken by an exhaustive search attack in the late 1990s before it was replaced with the Advanced Encryption Standard (AES).

Asymmetric Cryptography: In contrast to symmetric cryptography, asymmetric cryptography or public-key cryptography utilizes mathematically linked public and private keys to eliminate the need to share a secret key. As such, it is more appropriate for large and expanding networks with frequent message sharing between different parties. Public keys, which are freely shareable, are created from private keys that serve as one-factor authentication mechanisms and should be kept strictly

confidential. Importantly, the public and private keys are mathematically related, and while it is easy to calculate the public key from the private key, it is mathematically infeasible to go the other way. The mathematical link between the public and private keys also provides the ability to prove that one knows the private key without revealing it, enabling the creation of digital secrets and signatures. One can simply encrypt messages to a recipient's public key that can then only be decrypted by the recipient's private key (providing encryption), and the sender's public key can be used to verify that the sender is the holder of the private key without the need to reveal the private key (providing authentication). Keypair generation should be computationally economical to be practical, though the security of the algorithm will rely on the amount of computational effort required to find the private key from the public key.

- Diffie-Hellman (DH): The Diffie-Hellman Algorithm enables two parties with no prior knowledge of each other to establish a mutual secret over a public communications channel. DH uses large prime numbers and modulo arithmetic to do so but is often conceptually described using paint. For example, Alice and Bob will start with an arbitrary, publicly-known common paint color, say yellow. They will then each select their own secret color that they will keep to themselves. They each mix the common color with their secret color and exchange these mixtures with each other over the public network. Finally, they then mix their own secret color with the received mixture to now both have the same shared resulting secret color that cannot be determined by any of the information that was shared over the public network. DH is one of the most important developments in cryptography and forms the basis for many security protocols such as SSL and TSL.
- Rivest-Shamir-Adleman (RSA): RSA is the most widely accepted approach to public-key cryptography. It relies on the properties underlying the fundamental theorem of arithmetic which states that every number greater than 1 can be represented uniquely as the product of prime numbers or its unique prime factorization. As an example, the number 60, like all numbers, can be uniquely expressed as a product of its prime factors: $2^2*3*5 = 60$. RSA is based on the idea that it's easy to multiply large prime numbers together, but given a large prime number, it's incredibly difficult to determine that number's unique prime factorization. RSA, however, is susceptible to a brute force attack, where an adversary tries all possible keys until the message is decrypted, and will continue to be more susceptible to such attacks as computational advances occur. The recommended key size for RSA currently is 2048-bit, but the larger key size slows the encryption/decryption process. Given such resource intensity, RSA typically isn't used to encrypt messages or files and is more frequently used to encrypt a symmetric key that is able to operate at a much faster speed. RSA has had tremendous staying power as it's often used in combination with other encryption schemes.
- Elliptic Curve Cryptography (ECC) / Elliptic Curve Digital Signature Algorithm (ECDSA): ECC is a method of public-key cryptography based on the use of elliptic curves over finite fields and is used by ECDSA to generate particularly efficient keys with a high level of cryptographic strength. In ECDSA, a user selects a private key, usually at random, and runs elliptic curve operations on it to generate a mathematically linked public key that can't be used to infer the private key that created it. Elliptic curves follow the formula $y^2 = x^3+ax+b$, are symmetric about the x-axis, and any line drawn between two points will always intersect a third point. An elliptic curve cryptographic algorithm takes a starting point P, draws a line

tangent to it, and takes the intersection point of that tangent line and the elliptic curve before flipping across the x-axis to generate a point 2*P (this set of operations is adding point P to itself). This is repeated n number of times, cycling around the curve to end up at a point Q, as defined by Q=n*P. Q will seemingly have no relationship to the starting point P, and it is computationally infeasible for someone to know n (ie. how many times you cycled around the curve) even when one knows the curve, Q, and P. n may therefore be used as the private key and Q as the public key. ECDSA is the main signature scheme used by Bitcoin, which uses a specific elliptic curve called secp256k1, and is also used for TLS to encrypt connections between web browsers and applications.

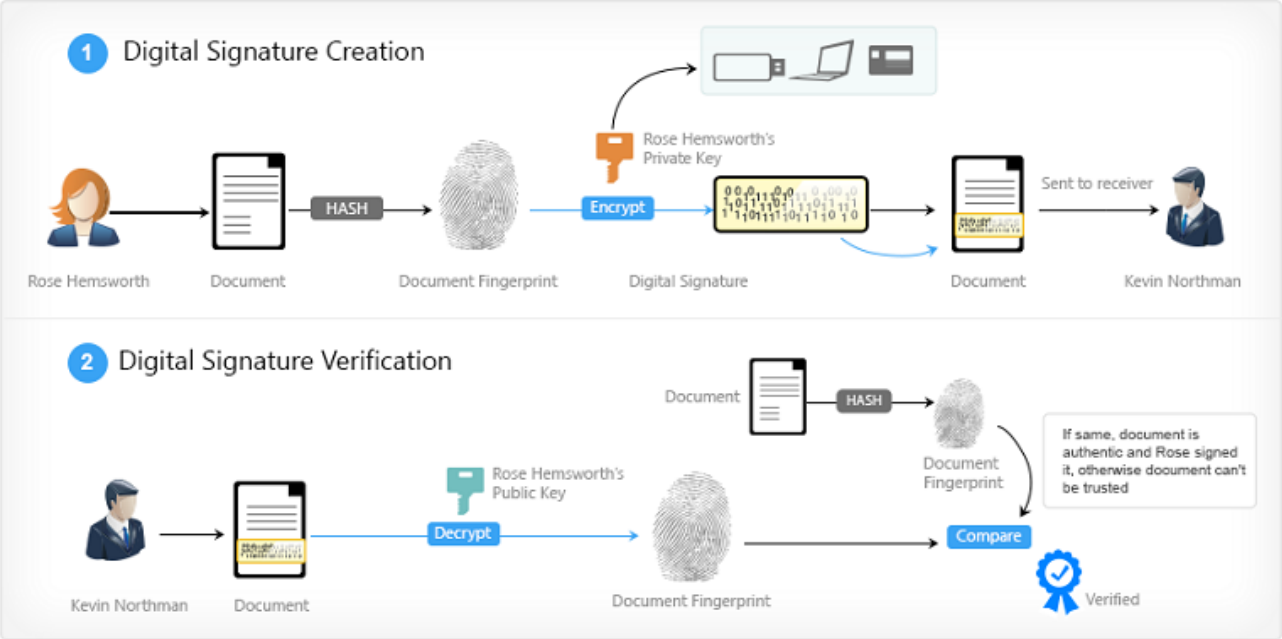
Cryptographic Hashing Functions: In contrast to symmetric and asymmetric cryptography, hashing functions do not use keys, but instead create a digital fingerprint of any arbitrary amount of data. To do so, the hashing algorithm splits the data into pieces and runs many rounds of local operations on them like AND, OR, and XOR, losing information as it goes and ultimately converting the data into a numeric string of fixed length such as 256 ones and zeros or 64 hexadecimal characters. Hashing functions should be one-way (the only way to know the input from a given output is to try all possible inputs), deterministic (returns the same output for a given input), easy to compute (but not so easy that one can quickly cycle through all potential inputs to solve), and produce few collisions (two different inputs should not produce the same output). Hashes have several benefits, such as improving efficiency and allowing for data verification without revealing the contents of the data. For example, rather than store passwords in a database that could potentially be hacked, a website can store hashed passwords. Then, when a user enters his or her password upon log-in, the website can simply take a hash of the entered password and compare it to its database of hashed passwords, materially enhancing security by not storing the passwords themselves (most websites modify this by adding a unique, user-specific random number to a user's password prior to hashing in what's called a salted hash. That way, if a hacker does get a hold of hashed passwords, the hacker can't simply use a dictionary of hashes of common passwords to figure out some of the simpler passwords).

- SHA-256: Bitcoin uses a specific hashing algorithm called SHA-256, which can be explored in this online SHA-256 hash calculator. Notice the high avalanche effect, where making one small change to the input data completely and unpredictably changes the resulting hash.

Cryptography & Bitcoin: Cryptographic algorithms are a key component of cryptocurrencies. Bitcoin, for example, uses cryptographic signatures to verify transaction authenticity as well as cryptographic hashing functions to improve data efficiency, expose tampering, and secure the network as part of its consensus mechanism. A user wishing to send funds to another would take a hash of the transaction and sign it using ECDSA with transaction information, a random number called a nonce, and his or her private key as inputs to generate the digital signature. This digital signature can then be cryptographically verified (ie. proven that it came only from the person holding the private key) using only the digital signature, the transaction, and the sender's public key. One does not need the private key to verify the transaction, and, since the digital signature depends on a nonce and the transaction itself, one's digital signature will be different for every transaction, preventing malicious actors from simply copying prior valid transactions. The Bitcoin blockchain also links blocks together by including a hash of the previous block header, which time-orders the blocks, improves searchability, and makes them tamper-evident. And Bitcoin's proof-of-work consensus mechanism relies heavily on hashing, where miners repeatedly hash their proposed block plus a nonce to be the first to solve the mining

puzzle. Because the output of a hash function cannot easily be guessed, this ensures that miners are expending significant energy and computational resources to post a block, erecting a barrier for potential malicious actors and securing the network. For a much more detailed explanation of how bitcoin uses cryptography, please see our in-depth primer, How Bitcoin Works.

Exhibit 1: A Simple Digital Signature Algorithm



Source: BitcoinClassroom.org, GSR

Authors
Brian Rudick, Senior Strategist
Matt Kunke, Junior Strategist

Sources

- Qvault: What is Cryptography? A Complete Overview
- Coding Tech: Cryptography for Beginners
- Edureka!: What is Cryptography?
- InfoSec Insights: Cryptology vs Cryptography: What's the Difference?
- SciShow: The Science of Making and Breaking Codes
- Harvey: Blockchain Business Models

About GSR

GSR is a global leader in digital asset trading, market making, OTC derivatives, and investments. We operate in a culture of excellence and leverage our first-rate reputation, deep relationships and proprietary trading technology to move swiftly and capitalize on market opportunities.

GSR's experienced team brings together decades of institutional trading expertise, while our industry-leading proprietary technology stack anchors everything we do.

Our main service areas are: market making; proprietary and algorithmic trading; client execution; structured products; risk management solutions; and portfolio investments.

For more information or if we can help with anything, please see gsr.io or contact us at gsr@gsr.io.

Required Disclosures

This material is a product of the GSR Sales and Trading Department. It is not a product of a Research Department, not a research report, and not subject to all of the independence and disclosure standards applicable to research reports prepared pursuant to FINRA or CFTC research rules. This material is not independent of the Firm's proprietary interests, which may conflict with your interests. The Firm trades instruments discussed in this material for its own account. The author may have consulted with the Firm's traders and other personnel, who may have already traded based on the views expressed in this material, may trade contrary to the views expressed in this material, and may have positions in other instruments discussed herein. This material is intended only for institutional investors. Solely for purposes of the CFTC's rules and to the extent this material discusses derivatives, this material is a solicitation for entering into a derivatives transaction and should not be considered to be a derivatives research report.

This material is provided solely for informational purposes, is intended for your use only and does not constitute an offer or commitment, a solicitation of an offer or comment (except as noted for CFTC purposes), or any advice or recommendation, to enter into or conclude any transaction (whether on the indicative terms shown or otherwise), or to provide investment services in any state or country where such an offer or solicitation or provision would be illegal.

Information is based on sources considered to be reliable, but not guaranteed to be accurate or complete. Any opinions or estimates expressed herein reflect a judgment made as of the date of publication, and are subject to change without notice. Trading and investing in digital assets involves significant risks including price volatility and illiquidity and may not be suitable for all investors. GSR will not be liable whatsoever for any direct or consequential loss arising from the use of this Information. Copyright of this Information belongs to GSR. Neither this Information nor any copy thereof may be taken or rented or redistributed, directly or indirectly, without prior written permission of GSR. Not a solicitation to U.S. Entities or individuals for securities in any form. If you are such an entity, you must close this page.