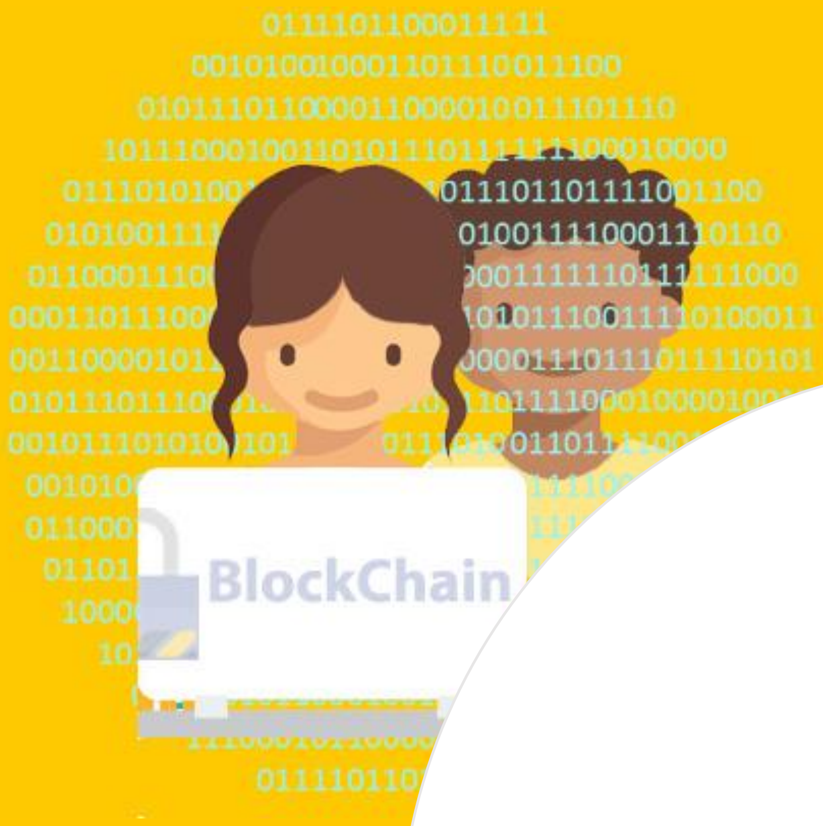
BeeBryte
Energy Intelligence

10 min

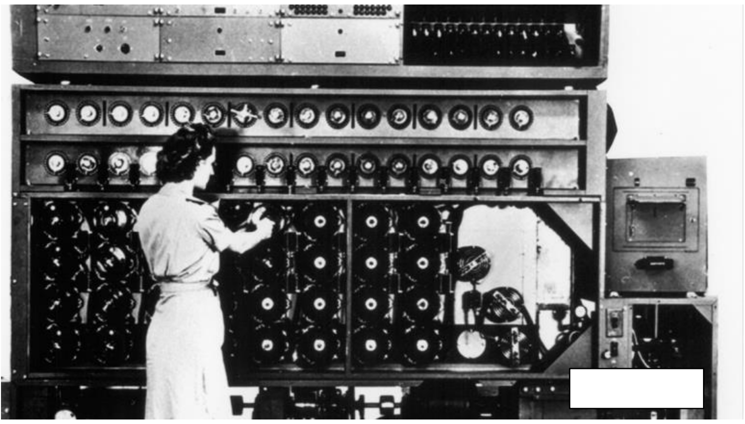


Cryptography Unchained

Maxime LAHARRAGUE, Pierre-Jean LARPIN, Johanna DREANO

May 30, 2018

In a world where the flow of information is increasing exponentially, it becomes essential to have powerful tools to ensure security features in terms of data exchange and storage (such as authenticity, integrity, confidentiality…). Among the available tools, the "BlockChain" focuses today all the attention and passions.



As a particularly complex mathematical construction, and a subject of many articles and books, BlockChain could deeply transform many sectors and appears as an essential component of the development of tomorrow's communication network. Its "basic bricks" are quite old and belong to the vast and fascinating field of cryptography. In this article, we propose to go back to the roots and explore this science of secrecy.

Very early in the development of its energy intelligence & automation solution, our company BeeBryte got interested in the cryptographic processes and technologies that lay the foundations of the security of any information system (confidentiality, authenticity and integrity). Used separately or in combination, these technologies are already largely integrated into our SaaS, especially to protect communications between our IoT gateway (Box) and our Cloud-based platform against cyber-attacks.

Cryptography is a discipline that aims at protecting the integrity and authenticity of a message as well as its privacy from third-party intrusions. Used in a vast range of areas, cryptography covers many features, on top of which the secure transmission of data (encryption), the generation of secrets between users, the authentication (or signature) of a message, and even its protection against any alteration.

Our wish here is to present some functional components, which are essential to the implementation of cryptographic protocols, such as those underlying the BlockChain.

We will not go into detail about the complex mathematical theories underpinning the exposed tools. Globally, they rely on operations, which are relatively simple to perform but extremely difficult to reverse, except resorting to "brute force", i.e. the systematic exploration of wide combinations, limited by the available computing power. We invite the reader interested in exploring the underlying mathematics to further his searches on his own.

KEY CONCEPTS

Symmetric Encryption – AES algorithm

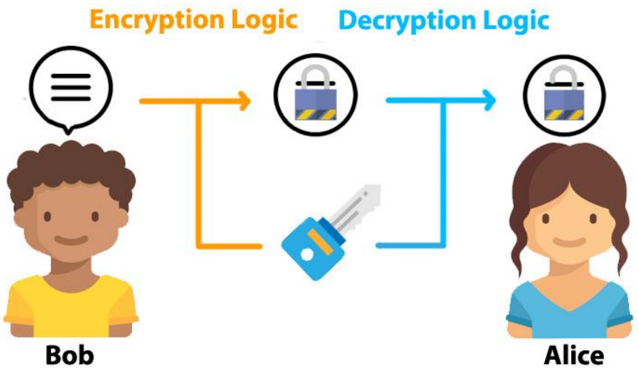
Symmetric encryption's basic concept relies on the sharing of a common key between interlocutors. This key is used by the sender to encrypt his message, which can only be decrypted by using this same key, that only the recipient and the sender of the message are familiar with.

The encryption algorithm makes it possible to transform plain text into blocks which remain incomprehensible without the key. However, these algorithms have the major disadvantage of requiring the prior – confidential – sharing of a common key. This drawback can however be overcome with the help of techniques such as the Diffie-Hellman key exchange protocol.

The most commonly used symmetric encryption algorithm is the Advanced Encryption Standard (AES), notably adopted by the NSA in the United States. This algorithm defines a number of transformations dedicated to be carried out on data stored in a table. The length of the key (128, 192 or 256-bit) determines the strength of the encryption.

Illustration of a typical use case:

- Bob wants to send a message to Alice, with whom he shares a secret that they will use as the encryption key. They take good care not to communicate this key to any other third party.



- Bob encrypts the message using the encryption key, and then sends the encrypted message to Alice.
- By also using the encryption key, Alice is able to decrypt the message she receives, and accesses successfully to the initial message from Bob.

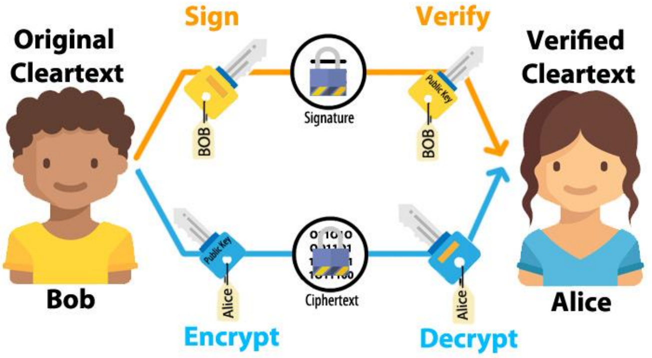
Asymmetric encryption - RSA Algorithm

In asymmetric encryption, each user has a pair of keys: a public key, broadcasted to all users, and a private key known only to him. Each message encrypted by one of the two keys can only be decrypted with the other key constituting the pair.

Because of the high computational complexity of symmetric encryption algorithms, they are typically used to encrypt small amounts of data. The asymmetric encryption is mainly used in 2 situations:

- **To certify the identity of the message sender**
  Let's consider the case where Bob wants to transmit a message and ensures to the recipient that he is the author. By encrypting the message with his private key, Bob creates a digital signature that is specific to him. Anyone can decipher his message using Bob's public key and thus confirm that the message is from Bob. Indeed, the use of another public key would return a wrong message.

- **To send protected content to a specific recipient**
  This time, Bob wants to send a message to Alice while ensuring that she will be the only one able to read it. By encrypting his message with the public key of Alice, Bob ensures that only Alice, the unique holder of the associated private key, will decrypt and read it.



Hash Functions

In cryptography, hash functions are mathematical functions that take as input any type of data and generate a code (the hash) of fixed length, representative of the input data.

A hash function has several unique properties that make it a useful tool for diverse applications in computer security:

- It is deterministic: for a given input value, the hash function will always return the same output. Therefore, the hash can be seen as a signature of fixed size of a message… but a non-exclusive signature.

- In fact, several different messages can have the same hash, but the probability of collisions (i.e. the probability that two different messages have the same hash) is extremely low.

  This property allows you to authenticate a message: if one knows the hash (or signature) of a message and that it receives independently a copy of this message, after calculation of its hash and positive comparison with the signature, the authenticity of the received message is quasi certain.

- A hash function is generally very sensitive to initial conditions: the slightest modification of the content input produces a totally different hash. This property allows – in addition to authentication – to ensure the integrity of a message, since the smallest alteration will dramatically invalid its hash.

- It is not directly invertible: calculating the hash corresponding to a given message is easy, but the opposite way, that is: to find a message that corresponds to a hash, is not possible by any direct calculation. The only solution is to 'chop' or hash, in a systematic way, different messages until falling on the expected hash, this method being physically limited by available computing capacity.

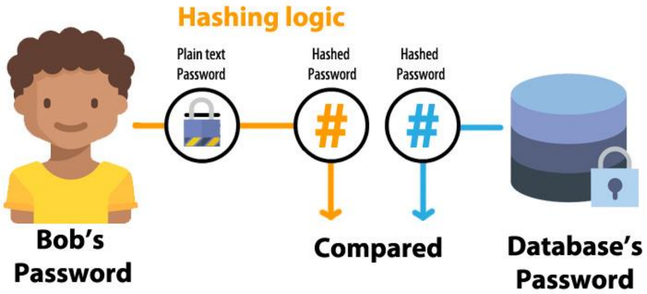


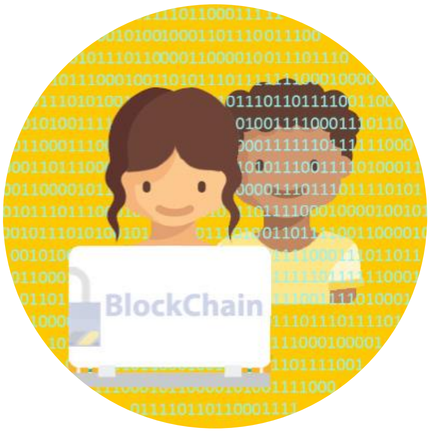
Illustration of a typical use case:

- Bob wants to send a message to Alice and wants to ensure that the content of his message is not altered during the transmission.
- Alice, on the other hand, wants to ensure that the message she receives is exactly what Bob wanted to send her.
- Bob hashes the message that he wants to send to Alice. He sends his message as well as the corresponding hash.
- When she receives the message, Alice reads the plain text message from Bob, and hashes it on her side. By comparing this hash to the hash Bob transmitted, she is able to determine whether or not the message she just read has been corrupted during the transmission.

Conclusion

The few elementary cryptographic techniques that we have outlined in this article constitute the basic components from which are developed more complex systems such as the BlockChain but can (and must) also be considered individually as means to address a large number of features for securing data.

This perspective appeared to us as essential to further dispassionately discuss the potential applications of the BlockChain to the energy sector, with their strengths and potentials but also their constraints and relevance limits.



BeeBryte is using IoT, AI and BlockChain to get commercial buildings, factories, EV charging stations or entire eco-suburbs to consume electricity in a smarter, more efficient and cheaper way while reducing carbon footprint! BeeBryte is based in France and Singapore, and is accelerated by Intel & TechFounders. Since its creation in 2015, BeeBryte's solutions have been awarded by prestigious organizations, such as EDF Pulse, DENA Start-up Energy Transition award & Hello Tomorrow Challenge.

If you want to participate in the energy revolution, please contact us at:

contact@beebryte.com
www.twitter.com/BeeBryteGroup
www.beebryte.com