Blockchain 2

This set of slides was developed based on the reference books as listed in the course document.

Previous discussion

In our previous discussion, the relation between

- trust,
- integrity,
- purely distributed peer-to-peer systems, and
- the blockchain

Are discussed. As a result, you now have a good understanding of *what the blockchain is*, *why it is needed*, and *what problem it solves*.

Now we start to discus **how the blockchain works**.

Planning the Blockchain

The basic concepts of managing ownership with the blockchain

7 Major Tasks

There are seven major tasks that need to be addressed when designing and developing a software system that manages ownership by using a purely distributed peer-to-peer system of ledgers in an open and untrustworthy environment:

1. Describing ownership
2. Protecting ownership
3. Storing transaction data
4. Preparing ledgers to be distributed in an untrustworthy environment
5. Distributing the ledgers
6. Adding new transaction to the ledgers
7. Deciding which ledgers represents the truth

Task 1: Documenting Ownership

7 Major Tasks

There are seven major tasks that need to be addressed when designing and developing a software system that manages ownership by using a purely distributed peer-to-peer system of ledgers in an open and untrustworthy environment:

1. Describing ownership
2. Protecting ownership
3. Storing transaction data
4. Preparing ledgers to be distributed in an untrustworthy environment
5. Distributing the ledgers
6. Adding new transaction to the ledgers
7. Deciding which ledgers represents the truth

Using the course of history as evidence for the current state of ownership

We are going to explain how the blockchain documents ownership and handles the transfer of ownership.

The Goal: The goal is the documentation of ownership in a transparent and comprehensible way. Anyone who reads that documentation should be able to make an unambiguous statement concerning the association of the goods to its owners.

The Challenge: The challenge is to find documentation of ownership that not just claims that someone is the owner of something, but also provides evidence of ownership and hence serves as proof of ownership.

The Idea: A list of all transfers of ownership is maintained in a ledger in an ongoing fashion. Every transfer of ownership is described by transaction data that clearly point out which owner hands off ownership of what item and to whom at what time. The whole history of transaction data stored in a ledger becomes an audit trail that provides evidence of how everyone achieved his or her possession.

Using the course of history as evidence for the current state of ownership

We are going to explain how the blockchain documents ownership and handles the transfer of ownership.

The Goal: The goal is the documentation of ownership in a transparent and comprehensible way. Anyone who reads that documentation should be able to make an unambiguous statement concerning the association of the goods to its owners.

The Challenge: The challenge is to find documentation of ownership that not just claims that someone is the owner of something, but also provides evidence of ownership and hence serves as proof of ownership.

The Idea: A list of all transfers of ownership is maintained in a ledger in an ongoing fashion. Every transfer of ownership is described by transaction data that clearly point out which owner hands off ownership of what item and to whom at what time. The whole history of transaction data stored in a ledger becomes an audit trail that provides evidence of how everyone achieved his or her possession.

Using the course of history as evidence for the current state of ownership

We are going to explain how the blockchain documents ownership and handles the transfer of ownership.

The Goal: The goal is the documentation of ownership in a transparent and comprehensible way. Anyone who reads that documentation should be able to make an unambiguous statement concerning the association of the goods to its owners.

The Challenge: The challenge is to find documentation of ownership that not just claims that someone is the owner of something, but also provides evidence of ownership and hence serves as proof of ownership.

The Idea: A list of all transfers of ownership is maintained in a ledger in an ongoing fashion. Every transfer of ownership is described by transaction data that clearly point out which owner hands off ownership of what item and to whom at what time. The whole history of transaction data stored in a ledger becomes an audit trail that provides evidence of how everyone achieved his or her possession.

How Documenting Ownership with the Blockchain Works

Documenting ownership with the blockchain involves the following aspects:

- Describing the transfer of ownership
- Maintaining the history of transfers

Describing the Transfer of Ownership

The act of transferring ownership relies on data that contains all information necessary to execute the transfer of ownership.

The information used by the blockchain to describe a transaction are:

- An identifier of the account that is to hand off ownership to another account
- An identifier of the account that is to receive ownership
- The amount of the goods to be transferred
- The time the transaction is to be done
- A fee to be paid to the system for executing the transaction
- A proof that the owner of the account that hands off ownership indeed agrees with that transfer

Maintaining the History of Transfers

The blockchain maintains the *whole history of all transactions* that have ever happened by storing their transaction data in the blockchain-data-structure *in the order* in which they occurred.

Any transaction not being part of that history is regarded as if it never happened.

Hence, *adding transaction data* to the blockchain-data-structure means making this transaction happen and *allowing it to influence the result of using the history in order to identify the current owner*.

Integrity of the Transaction History

The history of transaction data is the heart of any blockchain that manages ownership, therefore:

- it is the basis for reconstructing the state of ownership.
- it is necessary to keep that history of data *safe, complete, correct, and consistent* in order to maintain the integrity of the whole system
- the blockchain needs to provide *security measures* to ensure that only valid transaction data are added to the blockchain-data-structure.

Task 2: Protecting Ownership from Unauthorized Access

7 Major Tasks

There are seven major tasks that need to be addressed when designing and developing a software system that manages ownership by using a purely distributed peer-to-peer system of ledgers in an open and untrustworthy environment:

1. Describing ownership
2. Protecting ownership
3. Storing transaction data
4. Preparing ledgers to be distributed in an untrustworthy environment
5. Distributing the ledgers
6. Adding new transaction to the ledgers
7. Deciding which ledgers represents the truth

Hashing Data

How Hashing Works?

Hash functions transform any kind of data into a *fixed length hash*, regardless of the size of the input data.

There are many different hash functions that differ among others with respect to the length of the hash value they produce.

An important group of hash functions is called cryptographic hash functions, which create digital fingerprints for any kind of data.

How Hashing Works?

Generation of a cryptographic hash, or hashing, is one of the key activities performed at may different points throughout an examination:

Data —-> Hash Function —> fixed-size string

The resulting value is a hash of data

How Hashing Works?

Cryptographic hash functions have the following properties

- Providing hash values for any kind of data quickly
- Being deterministic: the hash function yields identical hash values for identical input data.
- Being pseudorandom: the hash value returned by a hash function changes unpredictably when the input data are changed
- Being one-way functions: there is no way to trace its input values by its outputs.
- Being collision resistant: it is very hard to find two or more distinct pieces of data for which it yields the identical hash value

Try it!

- A website that provides a tool for creating hash values of simple text data: http://www.blockchain-basics.com/HashFunctions.html

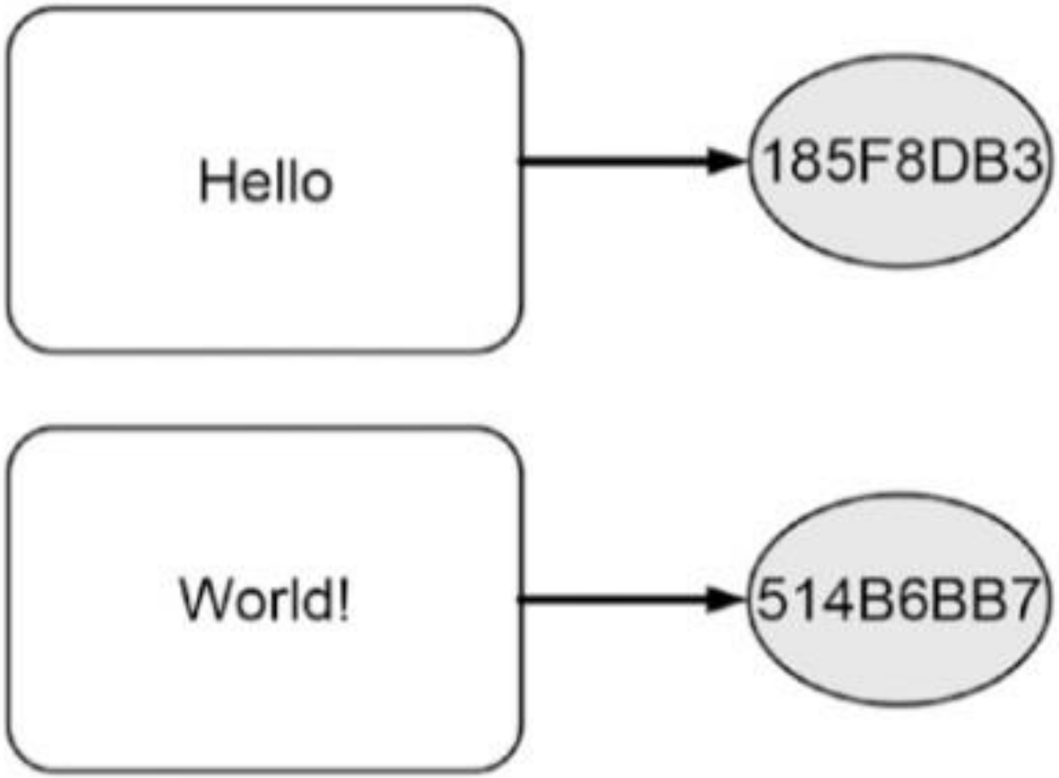
- A website that provides shortened hash value: www.blockchain-basics.com/Hashing.html

Patterns of Hashing Data

- There is no hash function that accepts a bunch of independent data at once.
- But, in reality, we often need one single hash value for a large collection of data. In particular, the blockchain-data- structure has to deal with many transaction data at once and requires one single hash value for all of them.
- To deal with this, we could use one of the following patterns in applying hash functions to data:
  - Independent hashing
  - Repeated hashing
  - Combined hashing
  - Sequential hashing
  - Hierarchical hashing

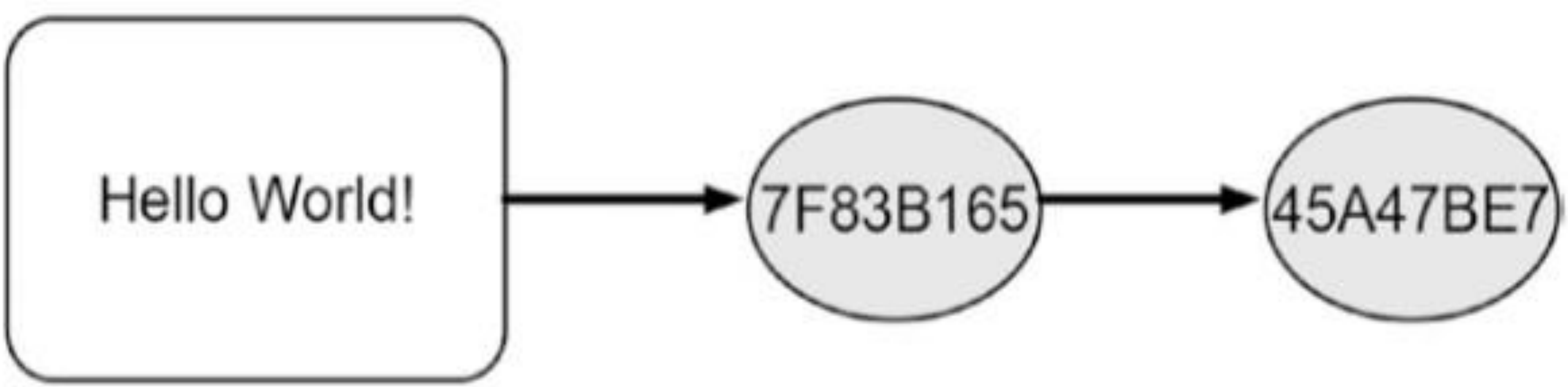
Independent Hashing

Independent hashing means applying the hash function to each piece of data independently:



Repeated Hashing

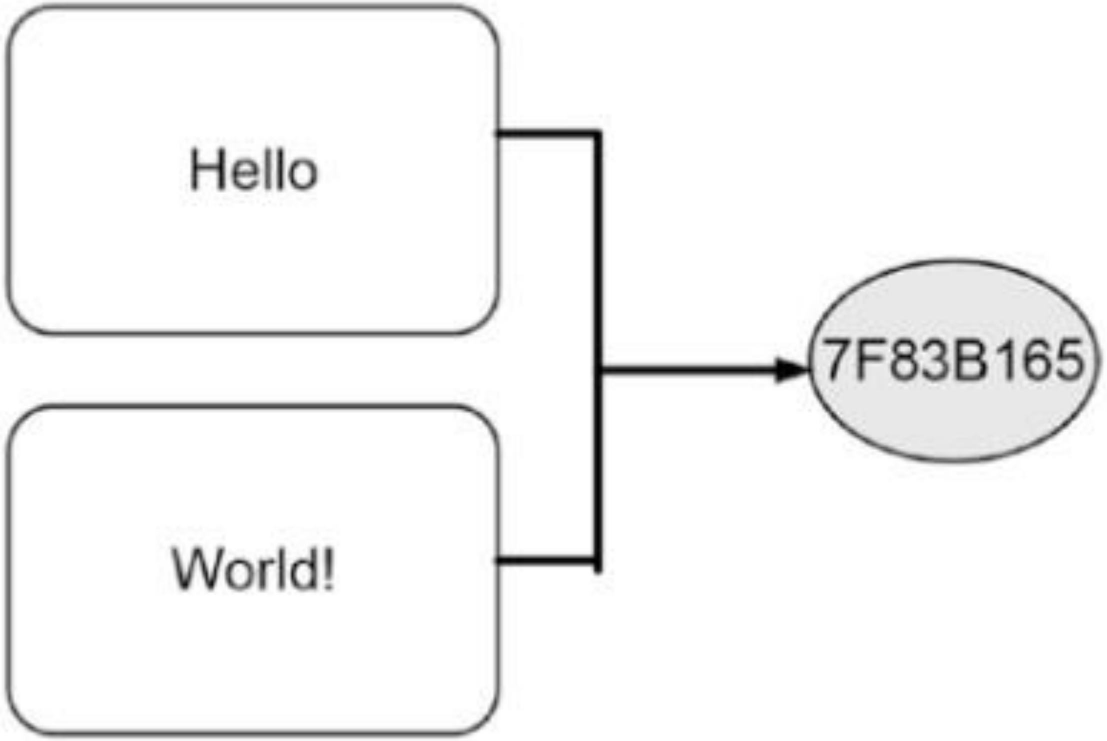
A hash value itself can be considered a piece of data. Hence, it should be possible to provide a hash value as input to a hash function and calculate its hash value too.



Combined Hashing

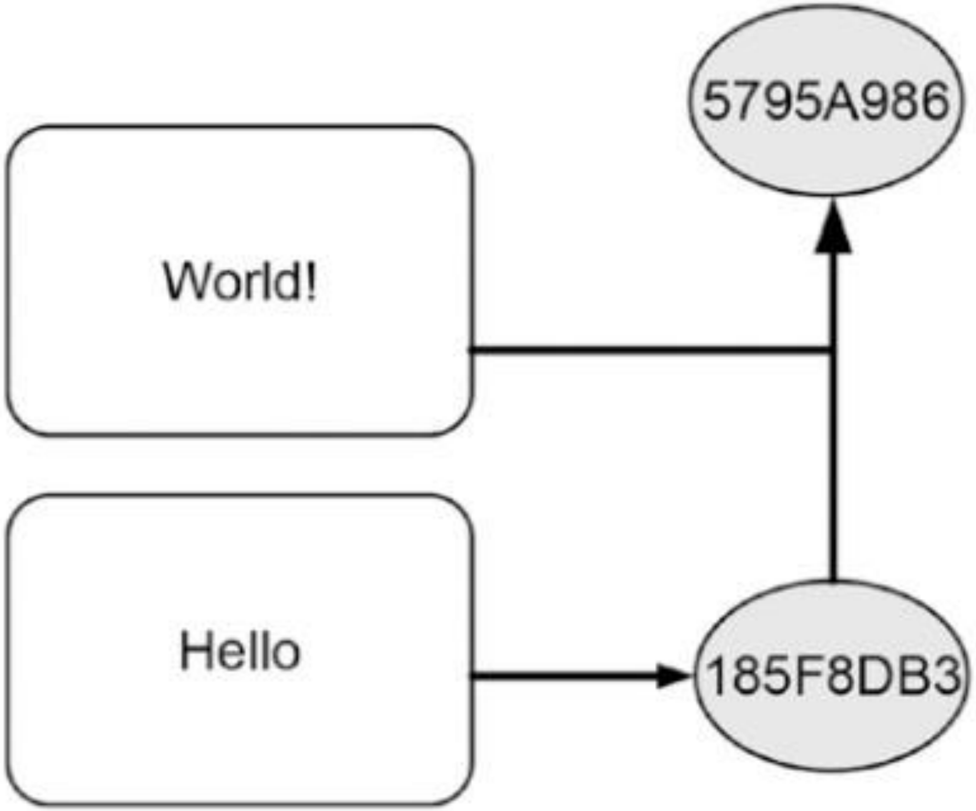
The individual words are first combined into one word with a letter space between them and the resulting phrase is hashed afterward.

Sometimes specific symbols such as the plus sign (+) or hashtag sign (#) are used to mark the point where the data are connected, which, as a result, influences the resulting hash value.



Sequential Hashing

The goal of sequential hashing is the incremental update of a hash value as new data arrive. The existing hash value is combined with new data and is then handed over to the hash function in order to get the updated hash value.

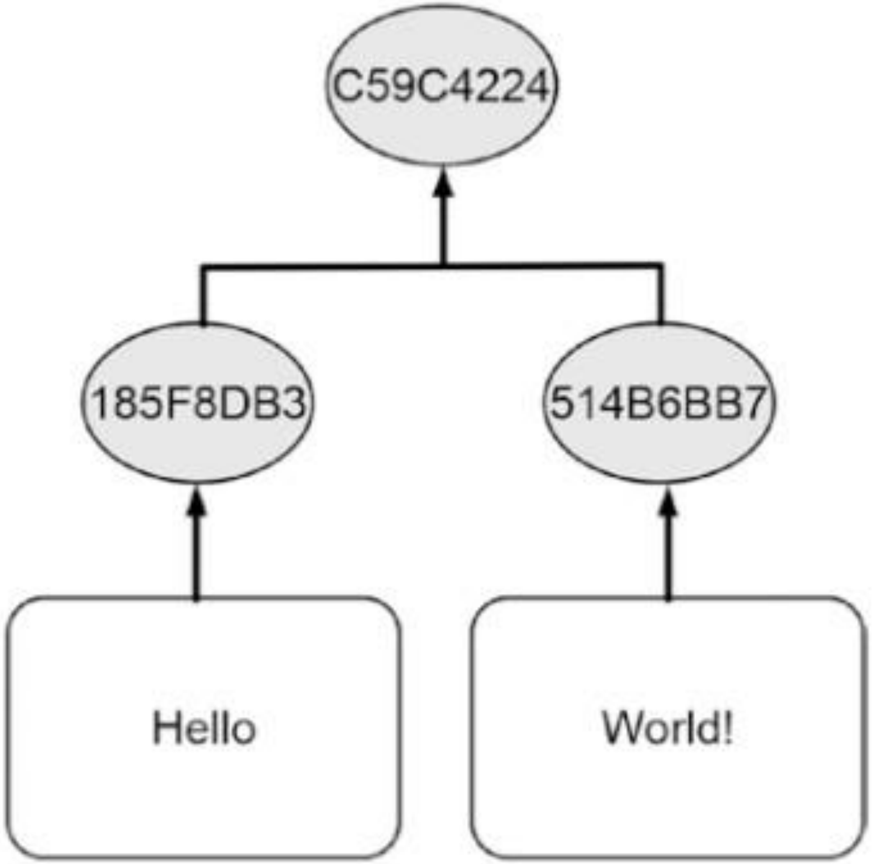


Sequential Hashing

Sequential hashing is in particular useful if you want to maintain *a single hash value* over time and update it as soon as *new data arrive*.

An advantage of this type of hashing is that at any given point in time you have a *hash value* whose evolution can be *traced back* to the *arrival of new data*.

Hierarchical Hashing



Hierarchical hashing is similar to combined hashing, but it only combines two hash values in every step, while combined hashing will combine as many data as you provide in one attempt.

Hashing in the Real World

The Application of Hash Functions in Real World

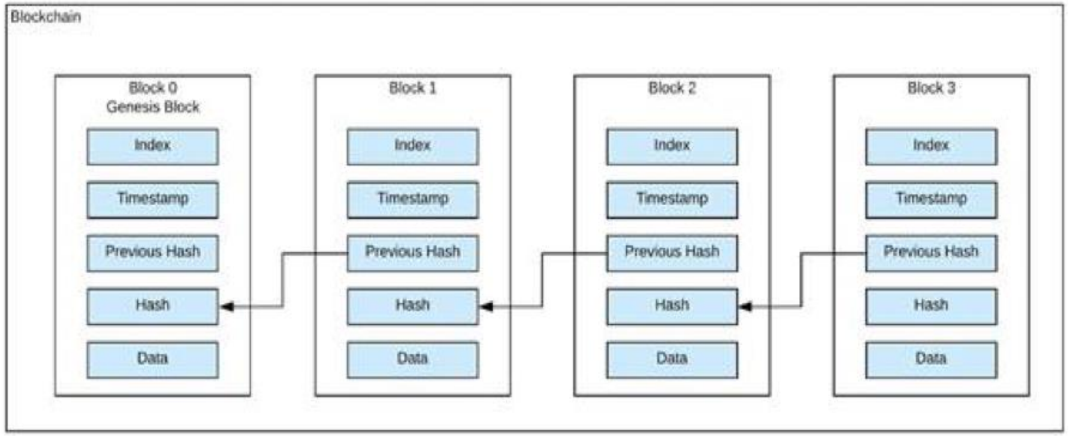
Major use cases of hash functions are:

1. Comparing data
2. Detecting changes in data (e.g. in forensics)
3. Hash references

Hash References

Hash references:

- The goal is to refer to data (e.g., transaction data) that are stored somewhere else (e.g., on a hard disk or in a database) and ensure that the data have remain unchanged.
- The idea is to combine the cryptographic hash value of the data being stored with information about the place where the data are located. If the data were changed, both pieces of information would no longer be consistent and hence the hash reference would become invalid.
- It is heavily used in blockchain.



Identifying and Protecting User Accounts

Asymmetric Cryptography

Using asymmetric cryptography in real life consists of two major steps:

- Creating and distributing the keys
- Using the keys

What is asymmetric cryptography and how it works?

- Asymmetric cryptography, also known as public-key cryptography, is a process that uses a pair of related keys -- one public key and one private key -- to encrypt and decrypt a message and protect it from unauthorized access or use.
- A public key is a cryptographic key that can be used by any person to encrypt a message so that it can only be deciphered by the intended recipient with their private key.
- A private key -- also known as a secret key -- is shared only with key's initiator.
- Video: https://www.youtube.com/watch?v=AQDCe585Lnc

Creating and Distributing the Keys

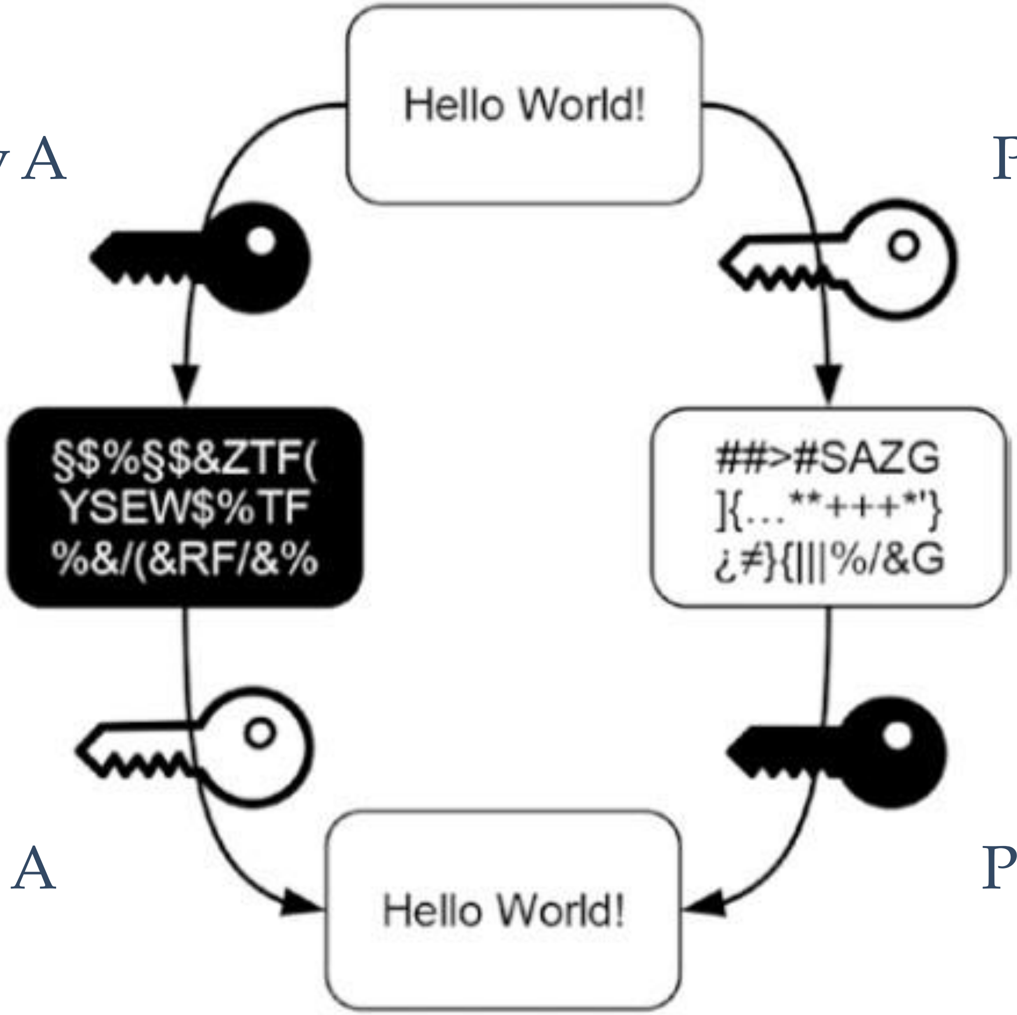
Hence, the first steps to be performed in any application of asymmetric cryptography are:

1. Create a pair of complementary keys by using cryptographic software
2. Give one key the name public key
3. Give the other key the name private key
4. Keep the private key for yourself
5. Give your public key to everyone else

Using the Keys

Private Key A

Public Key B



Public Key A

Private Key B

Asymmetric Cryptography in the Blockchain

The blockchain uses asymmetric cryptography in order to achieve two goals:

- Identifying accounts
- Authorizing transactions

Identifying Accounts

The blockchain uses the *public-to-private* approach of asymmetric cryptography for identifying user accounts and transferring ownership between them.

Account numbers in the blockchain are actually *public cryptographic keys*.

Hence, transaction data use the public cryptographic keys for identifying the accounts involved in the transfer of ownership.

Authorizing Transactions

Transaction data always have to include a piece of data that serves as proof that the owner of the account who hands off ownership indeed agrees with the described transfer of ownership.

The flow of information implied by this agreement:

- starts at the owner of the account who hands off ownership and is supposed to reach everyone who inspects the transaction data —> *private-to-public use case of asymmetric cryptography*
- The owner of the account who hands off ownership creates some cypher text with his or her private key.
- All others can *verify* this proof of agreement by *using the public cryptographic key* = the number of the account that hands off ownership.

The details of this procedure, which is called digital signature, will be explained in more detail later.

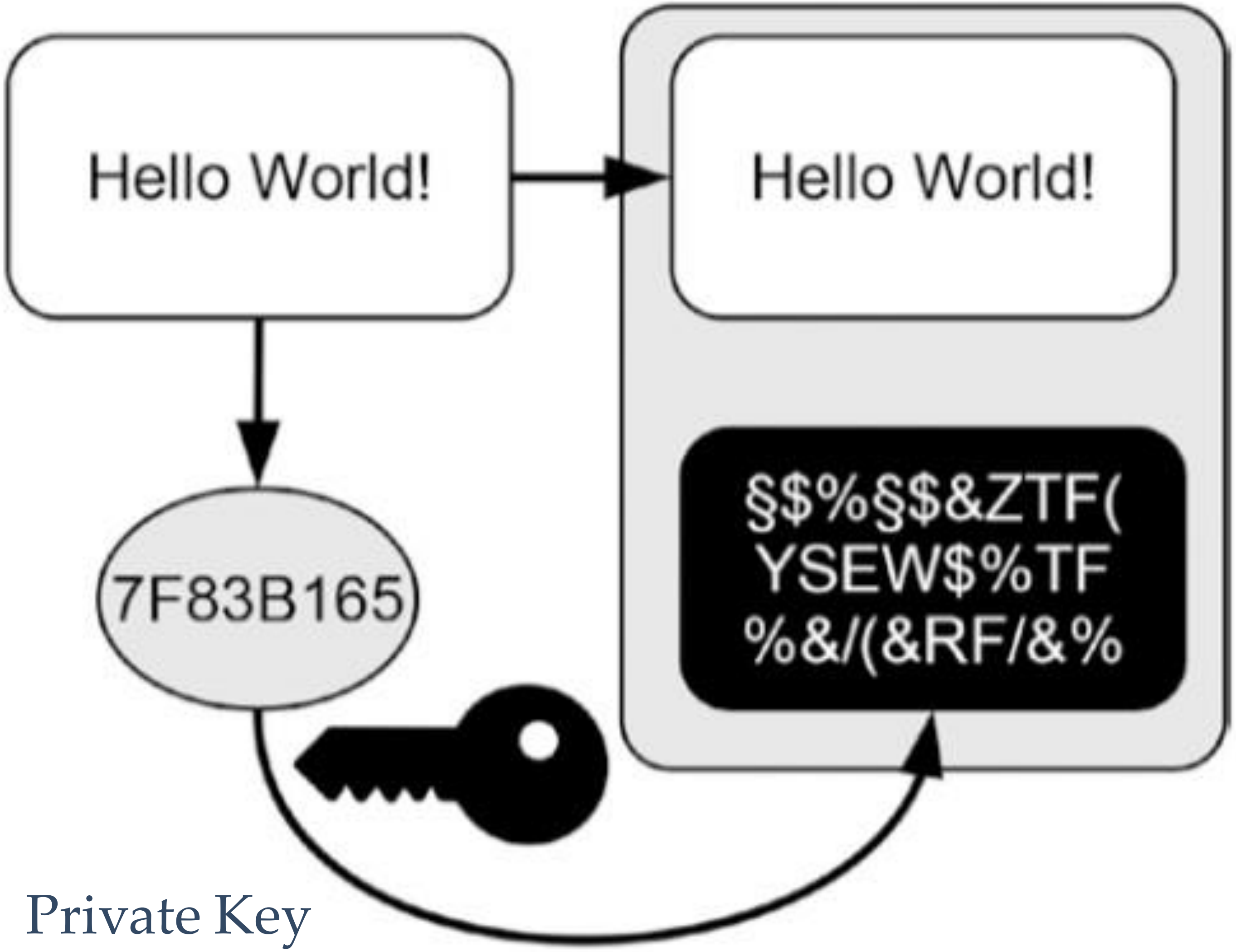
Authorizing Transactions

Digital Signatures

Three major elements of digital signatures:

1. Creating a signature
2. Verifying data by using the signature
3. Identifying fraud by using the signature

Creating a Signature



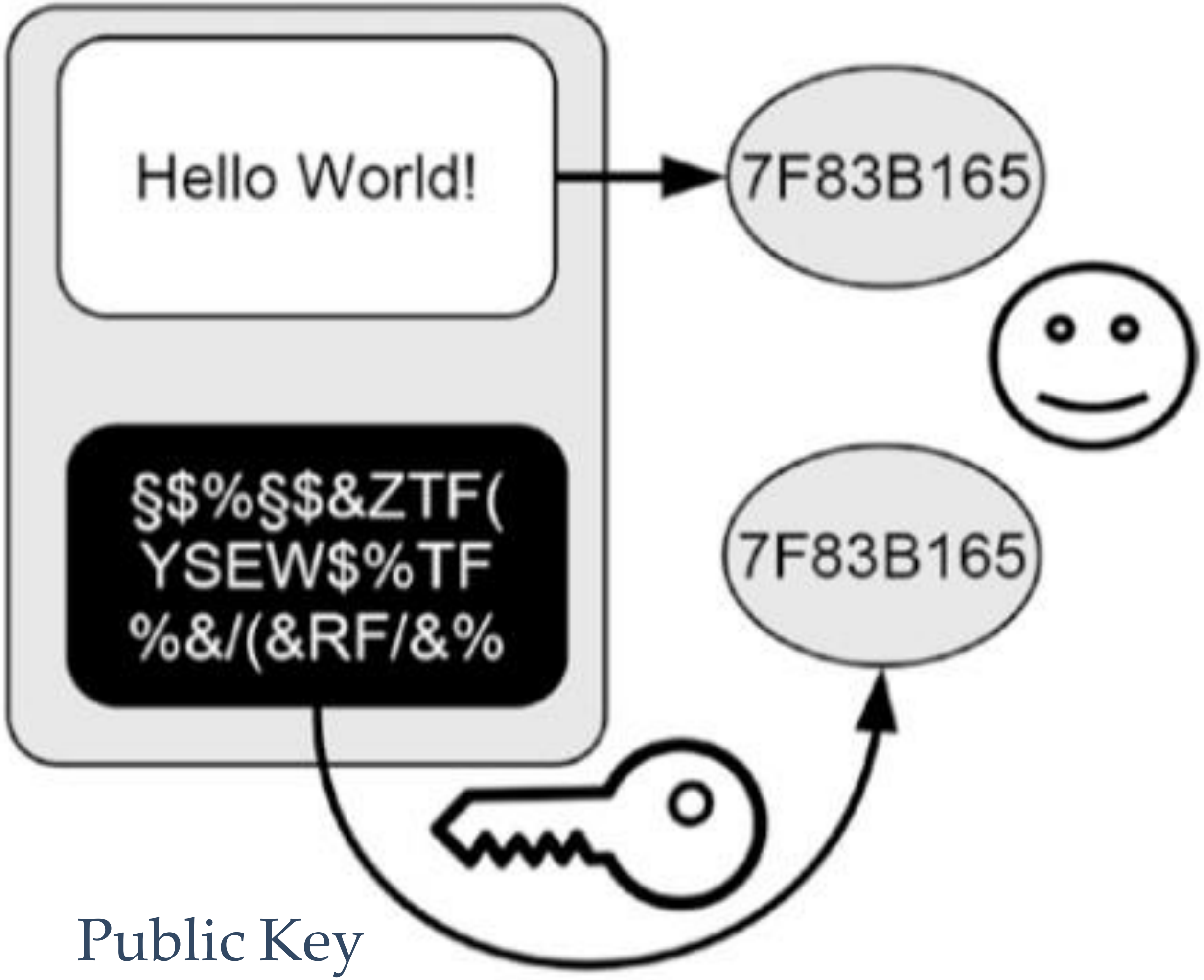
Private Key

Signing a Transaction

In order to create a digital signature for a transaction, the owner of the account who hands off ownership performs the following steps:

1. Describes the transaction with all necessary information such as the involved account numbers, amount being transferred, and so on except the signature itself as it is not yet available.
2. Create the cryptographic hash value of the transaction data.
3. Encrypt the hash value of the transaction with the private key of the account that hands off ownership.
4. Add the cypher text created in point 3 to the transaction as its digital signature.

Verifying Data by Using the Signature



Public Key

Verifying a Transaction

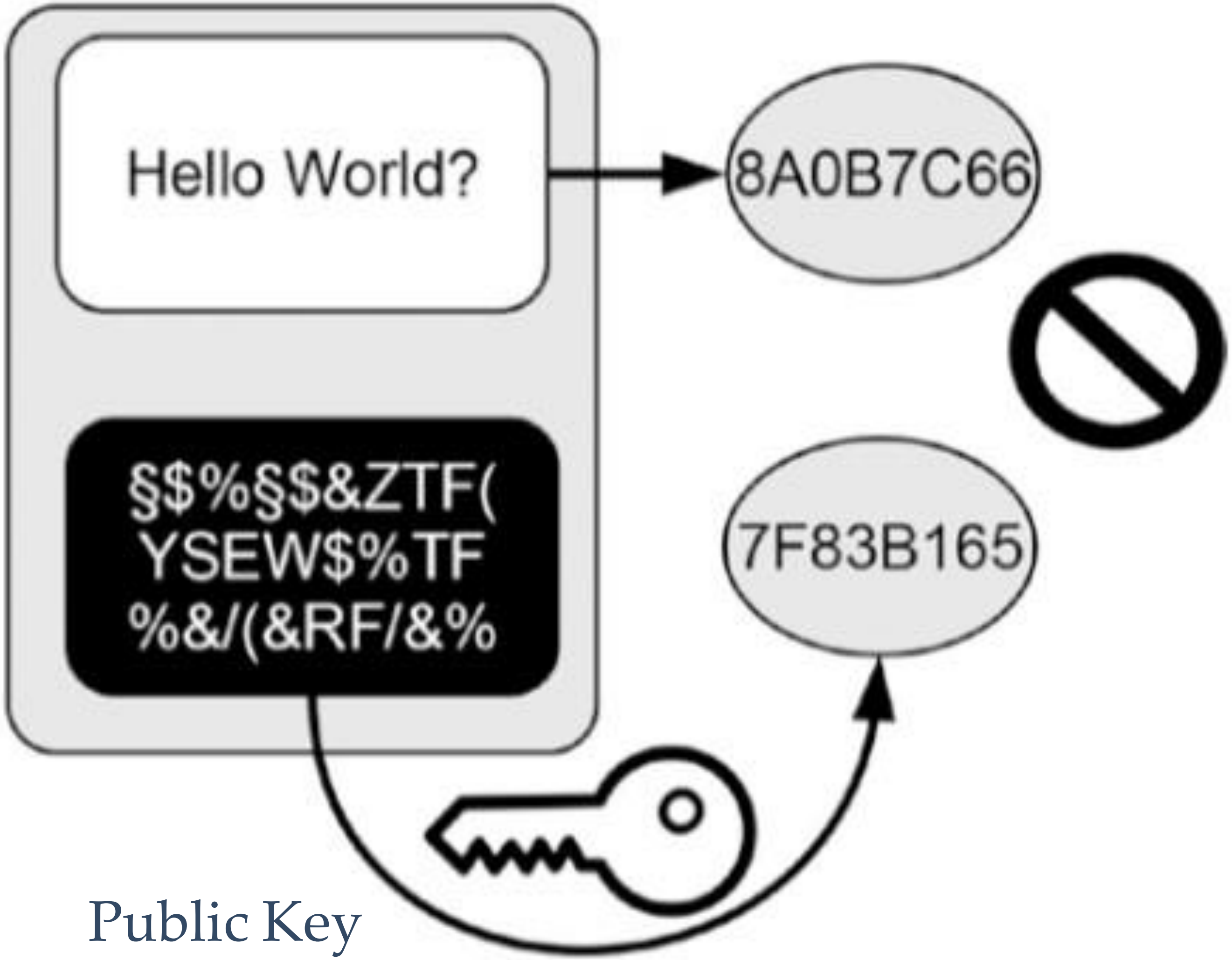
In order to verify a transaction, the following steps must be performed:

1. Create the hash value of the transaction data to be verified except the signature itself.

2. Decrypt the digital signature of the transaction under consideration with the account number that hands off ownership.

3. Compare the hash value of step 1 with the value gained in step 2. If both are identical, the transaction is authorized by the owner of the private key that corresponds to the account that hands off ownership, otherwise it is not.

Identifying Fraud by Using the Signature



Public Key