D5.1 BLOCKCHAIN PRIVACY REPORT

Grant Agreement nr	856879
Project acronym	PRESENT
Project start date (duration)	September 1st 2019 (36 months)
Document due:	30/08/2020
Actual delivery date	05/08/2020
Leader	INFOCERT
Reply to	roberto.deprisco@etutitus.com
Document status	version for submission

Project funded by H2020 from the European Commission

Project ref. no.	856879
Project acronym	PRESENT
Project full title	**P**hotoreal **RE**altime **S**entient **ENT**ity
Document name	PRESENT Project Handbook and Quality Plan
Security (distribution level)	CO - Confidential
Contractual date of delivery	30/08/2020
Actual date of delivery	05/08/2020
Deliverable name	D5.1 Blockchain Privacy Report
Type	Report
Status & version	Submission Version
Number of pages	23
WP / Task responsible	INFOCERT
Other contributors	-
Author(s)	Roberto De Prisco
EC Project Officer	Ms. Adelina Cornelia DINU - Adelina-Cornelia.DINU@ec.europa.eu
Abstract	A formal study on the needs regarding specific statements to provide privacy in the blockchain setting.
Keywords	Blockchain, Distributed ledger, Digital identities, Privacy, GDPR
Sent to peer reviewer	Yes
Peer review completed	Yes
Circulated to partners	No
Read by partners	No
Mgt. Board approval	No

Document History

Version and date	Reason for Change
Initial draft 13-01-2020	Outline, initial draft
Initial version 30-06-2020	For internal peer review
Revised version 24-07-2020	Version revised after internal peer review. Minor modifications needed.

Table of Contents

1. Abstract

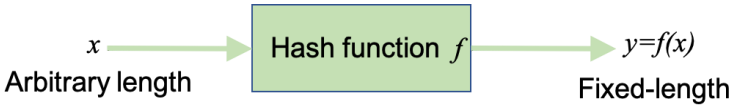
This deliverable presents an analysis of the use of blockchain technologies with particular attention to the privacy issues that arise from their usage. The document begins with a description of the cryptographic tools that are used in blockchain approaches and a description of the blockchain technology itself. Then a discussion of privacy issues, with specific considerations about the GDPR requirements, is presented. Finally, a specific blockchain technology which can be used to deal with the needs of the PRESENT project is described in detail.

2. Preliminaries

In this section, we introduce needed preliminaries to understand blockchains and distributed ledgers.

2.1. Hash functions

An important cryptographic tool that we need to understand in order to talk about blokchains is a *hash function*. A hash function is a function that takes an input of arbitrary length and produces an output of some, short, fixed-length. Assuming that data is represented with bits, we have that the input x of a hash function f is a sequence of bits of any, usually large, length and the output $y=f(x)$ consists of a fixed number N of bits. Typical values for N are 256 or 512.



Hence a hash function can be thought of as a compressor that is able to transform a very long input into a small, fixed-length, output. In this sense, a hash value can be seen as a *message digest* (footprint) of the input. Moreover, hash functions should be easy to compute; technically, the hash value of an n-bit input, should be computable in $O(n)$ time.

Given that the output of a hash function is a string of N bits, there are exactly 2^N possible outputs. Moreover, since we allow any input, the number of possible inputs is far greater than the 2^N possible outputs (it is actually infinite). Thus, it goes without saying that it is possible that two different inputs x_1 and x_2 produce the same output $y=f(x_1)=f(x_2)$. Such a situation is called a *collision*.

It should not be difficult to see that collisions are not good. For example, if we plan to use hash values as footprints, then a collision creates an ambiguity. However, even if the theoretical number of inputs is infinite, in real situations we do not have infinite inputs. For example, imagine that the possible inputs are all the books ever written. On August 5, 2010, a Google software engineer, Leonid Taycher, posted a blog in which he estimated the number of books in the world, coming up with something close to 130 million (more precisely 129,864,880). Let us assume that the estimate is correct. If we use a hash function with $N=256$, the number of possible outputs, 2^{256}, is way bigger than 130 million: