



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0073217
(43) 공개일자 2023년05월25일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) H04L 9/00 (2022.01)
H04L 9/06 (2006.01) H04L 9/08 (2006.01)
(52) CPC특허분류
H04L 9/3278 (2013.01)
H04L 9/0637 (2013.01)
(21) 출원번호 10-2023-7010453
(22) 출원일자(국제) 2021년08월31일
심사청구일자 없음
(85) 번역문제출일자 2023년03월27일
(86) 국제출원번호 PCT/EP2021/073986
(87) 국제공개번호 WO 2022/069135
국제공개일자 2022년04월07일
(30) 우선권주장
2015477.9 2020년09월30일 영국(GB)

(71) 출원인
엔체인 라이선싱 아게
스위스 주크 6300 그라펜아우베크 6
(72) 발명자
데이비스, 잭, 오웬
영국 런던 더블유1더블유 8에이피 마켓 플레이스
30 엔체인 라이선싱 아게 내
라이트, 크레이그, 스티븐
영국 런던 더블유1더블유 8에이피 마켓 플레이스
30 엔체인 라이선싱 아게 내
(74) 대리인
특허법인 광장리앤코

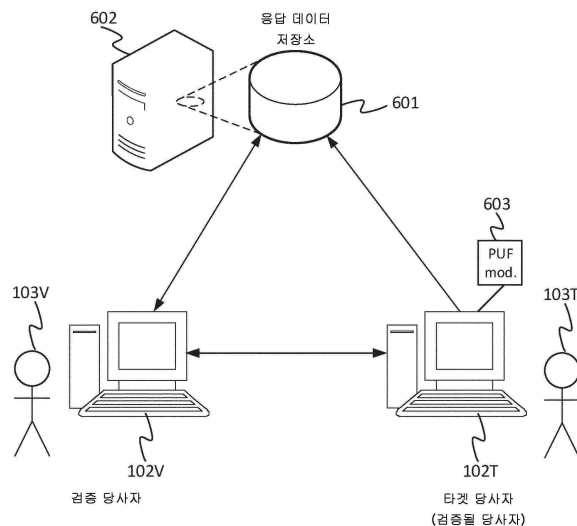
전체 청구항 수 : 총 40 항

(54) 발명의 명칭 물리적 복제 방지 함수

(57) 요약

본 발명은 검증 당사자가 타겟 당사자 또는 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하는 방법에 관한 것이다. 방법은, 셋-업 국면에서, 검증 당사자 이외의 당사자에 의해: PUF(physically unclonable function)에 기초하여 각각의 하나 이상의 응답 세트를 생성하기 위해 PUF를 포함하는 PUF 모듈에 하나 이상의 챌린지 세트를 입력하는 단계; 및 응답 세트 각각에 대한 개개의 응답 데이터 레코드를 타겟 당사자 또는 검증 당사자의 임의의 장비 외부에 있는 데이터 저장소에 저장하는 단계를 포함하고, 데이터 저장소는 제3자 컴퓨터 장비의 일부이거나 공개 피어-투-피어 공개 매체 중 어느 하나이다. 따라서, 응답 데이터 레코드는 후속 검증 국면에서 타겟의 아이덴티티를 검증하기 위해 검증 당사자에게 이용 가능하게 된다.

대표도 - 도6



(52) CPC특허분류

H04L 9/0643 (2013.01)

H04L 9/0825 (2013.01)

H04L 9/0866 (2013.01)

H04L 9/321 (2013.01)

H04L 9/3236 (2013.01)

H04L 9/3247 (2013.01)

H04L 9/50 (2022.05)

명세서

청구범위

청구항 1

하나 이상의 검증 당사자가 타겟 당사자 또는 상기 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위한 컴퓨터 구현 방법으로서,

상기 방법은, 셋-업 국면에서, 상기 하나 이상의 검증 당사자 중 임의의 검증 당사자 이외의 셋-업 당사자에 의해;

PUF(physically unclonable function)에 기초하여 개개의 하나 이상의 응답 세트를 생성하기 위해 상기 PUF를 포함하는 PUF 모듈에 하나 이상의 챌린지 세트를 입력하는 단계; 및

상기 하나 이상의 검증 당사자 중 임의의 검증 당사자 또는 상기 타겟 당사자의 임의의 장비 외부에 있는 데이터 저장소에 상기 응답 세트의 각각의 응답에 대한 개개의 응답 데이터 레코드를 저장하는 단계를 포함하고, 상기 데이터 저장소는 제3자 컴퓨터 장비의 일부이거나 공개 피어-투-피어 공개 매체이고, 이로써 후속 검증 국면에서 상기 타겟의 아이덴티티를 검증하도록 상기 하나 이상의 검증 당사자에게 상기 응답 데이터 레코드를 이용 가능하게 만들고, 각각의 응답에 대한 상기 응답 데이터 레코드는 개개의 응답 또는 상기 개개의 응답으로부터 도출된 데이터를 포함하는,

컴퓨터 구현 방법.

청구항 2

제1항에 있어서,

상기 셋-업 당사자는 상기 타겟 당사자이고, 상기 방법은 상기 타겟 당사자의 컴퓨터 장비를 이용하여 상기 타겟 당사자에 의해 수행되는,

컴퓨터 구현 방법.

청구항 3

제2항에 있어서,

상기 PUF 모듈은 상기 타겟 당사자의 컴퓨터 장비에 통합되는,

컴퓨터 구현 방법.

청구항 4

제2항에 있어서,

상기 PUF 모듈은 상기 타겟 당사자의 컴퓨터 장비에 대한 외부 주변기기인,

컴퓨터 구현 방법.

청구항 5

제2항 내지 제4항 중 어느 한 항에 있어서,

상기 데이터 저장소는 상기 제3자 컴퓨터 장비의 일부이고, 상기 타겟 당사자는 상기 제3자로부터 상기 챌린지 세트를 수신하고, 상기 PUF 모듈에 입력된 상기 챌린지 세트는 상기 제3자로부터 수신된 세트인,

컴퓨터 구현 방법.

청구항 6

제1항에 있어서,

상기 셋-업 당사자는 신뢰할 수 있는 제3자인,
컴퓨터 구현 방법.

청구항 7

제6항에 있어서,
상기 데이터 저장소는 상기 제3자 컴퓨터 장비의 일부이고, 상기 방법은 상기 제3자 컴퓨터 장비를 사용하여 상기 신뢰할 수 있는 제3자에 의해 수행되는,
컴퓨터 구현 방법.

청구항 8

제7항에 있어서,
상기 제3자 컴퓨터 장비는 하나 이상의 지리적 사이트에서 하나 이상의 서버 유닛을 포함하는 서버 장비를 포함하는,
컴퓨터 구현 방법.

청구항 9

제7항 또는 제8항에 있어서,
상기 PUF 모듈은 상기 제3자 컴퓨터 장비의 외부 주변기기 또는 착탈식 구성요소이고, 상기 방법은, 상기 타겟 당사자가 상기 검증 국면에서 상기 검증 당사자로부터의 챌린지에 응답하는 것을 가능하게 하도록, 상기 셋-업 국면 후에, 상기 PUF 모듈을 상기 타겟 당사자에게 물리적으로 보내는 단계를 포함하는,
컴퓨터 구현 방법.

청구항 10

제1항 내지 제6항 중 어느 한 항에 있어서,
상기 데이터 저장소는 상기 피어-투-피어 공개 매체인,
컴퓨터 구현 방법.

청구항 11

제10항에 있어서,
상기 피어-투-피어 공개 매체는 블록체인인,
컴퓨터 구현 방법.

청구항 12

제11항에 있어서,
상기 블록체인은 출력 기반 트랜잭션 모델을 사용하고, 상기 블록체인은 각각 적어도 하나의 입력 및 적어도 하나의 출력을 갖는 복수의 트랜잭션을 포함하고, 각각의 트랜잭션의 입력은 다른 트랜잭션의 출력을 지시하고, 상기 응답 데이터 레코드는 상기 블록체인에 레코딩된 하나 이상의 트랜잭션의 하나 이상의 출력에 저장되는,
컴퓨터 구현 방법.

청구항 13

제12항에 있어서,
상기 방법은, 상기 응답 데이터 레코드들 중 만료된 응답 데이터 레코드를 저장하는 트랜잭션의 출력을 지시하는 입력을 갖는 새로운 트랜잭션을 상기블록체인 상에 레코딩함으로써 상기 만료된 응답 데이터 레코드를 업데이트 또는 폐지하는 단계를 더 포함하는,

컴퓨터 구현 방법.

청구항 14

제1항 내지 제13항 중 어느 한 항에 있어서,

상기 저장하는 단계는 네트워크를 통해 상기 데이터 저장소에 액세스하는 단계 및 상기 데이터 저장소에서 상기 응답 데이터 레코드를 생성하기 위해 상기 네트워크를 통해 데이터를 전송하는 단계를 포함하는,

컴퓨터 구현 방법.

청구항 15

제1항 내지 제14항 중 어느 한 항에 있어서,

상기 하나 이상의 검증 당사자는 복수의 검증 당사자인,

컴퓨터 구현 방법.

청구항 16

제15항에 있어서,

상기 저장하는 단계는 하나 이상의 응답 데이터 레코드의 개개의 서브세트만을 상기 검증 당사자 각각에게 이용 가능하게 만들고, 상이한 서브세트는 상기 검증 당사자 중 상이한 검증 당사자에게 이용 가능하게 된 레코드인,

컴퓨터 구현 방법.

청구항 17

제16항에 있어서,

상기 서브세트는 서로 배타적인,

컴퓨터 구현 방법.

청구항 18

제1항 내지 제17항 중 어느 한 항에 있어서,

각각의 응답 데이터 레코드는 상기 개개의 응답의 명시적 값을 포함하는,

컴퓨터 구현 방법.

청구항 19

제18항에 있어서,

각각의 응답 데이터 레코드의 값은 암호화된 형태로만 상기 데이터 저장소에 저장되는,

컴퓨터 구현 방법.

청구항 20

제19항에 있어서,

각각의 응답 데이터 레코드는 암호해독하기 위해 상이한 개개의 암호해독 키를 요구하는,

컴퓨터 구현 방법.

청구항 21

제16항에 종속하는 제20항에 있어서,

상기 암호해독 키 중 상이한 암호해독 키를 상이한 검증 당사자들에게 분배함으로써 상기 상이한 서브세트가 상기 검증 당사자들 중 상이한 검증 당사자들에게 이용가능하게 되는,

컴퓨터 구현 방법.

청구항 22

제1항 내지 제17항 중 어느 한 항에 있어서,

각각의 응답 데이터 레코드는 상기 응답의 명시적 값이 아닌 상기 개개의 응답의 증명만을 포함하고, 상기 증명은 상기 개개의 응답의 변환을 포함하고, 상기 응답의 변환은 응답을 개시하지 않지만, 검증 당사자가 타겟에 의해 리턴된 상기 개개의 응답의 추가 인스턴스에 대해 동일한 변환을 수행하고 상기 증명과 비교함으로써 상기 타겟의 아이덴티티를 검증하는 것을 가능하게 하는,

컴퓨터 구현 방법.

청구항 23

제22항에 있어서,

상기 변환은 해시 또는 이중 해시를 포함하는,

컴퓨터 구현 방법.

청구항 24

제1항 내지 제23항 중 어느 한 항에 있어서,

상기 검증 당사자 중 하나로부터 요청을 수신하는 단계 - 상기 요청은 상기 챌린지 세트 중 하나의 챌린지를 포함함 - ; 및 이에 응답하여, 대응하는 응답의 추가 인스턴스를 생성하기 위해 상기 챌린지의 추가 인스턴스를 상기 PUF 모듈에 입력하고, 그리고 상기 검증 당사자가 상기 데이터 저장소에 저장된 응답과의 비교에 기초하여 상기 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위해 상기 응답의 추가 인스턴스를 상기 검증 당사자에 리턴하는 단계를 포함하는,

컴퓨터 구현 방법.

청구항 25

제1항 내지 제17항 중 어느 한 항에 있어서,

각각의 응답 데이터 레코드는 상기 개개의 응답으로부터 도출된 개개의 공개-개인 키 쌍의 개개의 공개 키를 포함하는,

컴퓨터 구현 방법.

청구항 26

제25항에 있어서,

상기 도출은 상기 개개의 공개-개인 키 쌍을 생성하는 알고리즘에서 시드로서 상기 개개의 응답을 사용하는 것을 포함하는,

컴퓨터 구현 방법.

청구항 27

제25항 또는 제26항에 있어서,

공개 키 중 하나를 사용하여 상기 타겟에 의해 서명된 메시지를 상기 검증 당사자 중 하나에게 전송하거나 이용 가능하게 만드는 단계를 포함하는,

컴퓨터 구현 방법.

청구항 28

제1항 내지 제27항 중 어느 한 항에 있어서,

상기 PUF 모듈은 PUF, 인터페이스 로직 및 결정론적 변형 함수를 포함하고, 상기 인터페이스 로직은:

- 상기 챌린지 세트를 수신하고;
 - 대응하는 1차 응답을 생성하기 위해 기본 챌린지를 상기 PUF에 입력하고; 그리고
 - 상기 응답 세트의 개개의 응답을 생성하기 위해 상기 생성된 기본 응답과 함께 상기 수신된 챌린지 세트 각각을 상기 변형 함수에 입력함으로써 — 상기 변형 함수는 상기 수신된 챌린지 세트로부터의 개개의 챌린지 및 상기 생성된 기본 응답의 함수임 —, 상기 응답 세트의 생성을 유발하는,
- 컴퓨터 구현 방법.

청구항 29

제1항 내지 제28항 중 어느 한 항에 있어서,

각각의 응답 데이터 레코드는 상기 개개의 챌린지의 표시와 관련하여 상기 데이터 저장소에 저장되고, 따라서 상기 검증 당사자가 상기 개개의 챌린지를 결정하거나, 또는 검증에서 상기 타겟에 챌린지하는 데 사용되는 상기 개개의 챌린지에 기초하여 상기 개개의 응답을 복원하는 것을 가능하게 하는,

컴퓨터 구현 방법.

청구항 30

제29항에 있어서,

각각의 개개의 챌린지의 표시는 상기 개개의 응답 데이터와 관련하여 저장된 상기 개개의 챌린지의 명시적 값을 포함하는,

컴퓨터 구현 방법.

청구항 31

제29항에 있어서,

상기 챌린지의 표시는 상기 챌린지 세트가 도출될 수 있는 마스터 챌린지를 포함하는,

컴퓨터 구현 방법.

청구항 32

제1항 내지 제31항 중 어느 한 항에 있어서,

상기 타겟 당사자는, 상기 데이터 저장소가 다수의 타겟 당사자의 대응하는 PUF에 대한 챌린지로부터 기인하는 응답 데이터 세트를 레코딩하는 상기 다수의 타겟 당사자 중 하나인,

컴퓨터 구현 방법.

청구항 33

컴퓨터 장비로서,

하나 이상의 메모리 유닛을 포함하는 메모리; 및

하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 상기 메모리는 상기 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 상기 코드는 상기 처리 장치 상에서 실행될 때 제1항 내지 제32항 중 어느 한 항의 방법을 수행하도록 구성되는,

컴퓨터 장비.

청구항 34

비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 제1항 내지 제32항 중 어느 한 항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

청구항 35

하나 이상의 검증 당사자가 타겟 당사자 또는 상기 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위한 컴퓨터 구현 방법으로서,

상기 방법은, 셋-업 국면에서, 신뢰할 수 있는 제3자에 의해:

개개의 응답을 생성하기 위해, 상기 타겟 당사자가 PUF(physically unclonable function)를 포함하는 PUF 모듈에 개개의 챌린지를 입력한 것으로부터 기인한 응답 데이터를 상기 타겟 당사자로부터 수신하는 단계;

하나 이상의 응답 각각에 대한 개개의 응답 데이터 레코드를 상기 신뢰할 수 있는 제3자의 데이터 저장소에 저장하는 단계 - 각각의 응답 데이터 레코드는 상기 개개의 응답 또는 상기 개개의 응답으로부터 도출된 데이터를 포함함 - ; 및

후속 검증 국면에서 상기 타겟의 아이덴티티를 검증하기 위해 상기 응답 데이터 레코드를 상기 하나 이상의 검증 당사자에게 이용 가능하게 만드는 단계를 포함하고, 각각의 응답에 대한 응답 데이터 레코드는 상기 개개의 응답 또는 상기 개개의 응답으로부터 도출된 데이터를 포함하는,

컴퓨터 구현 방법.

청구항 36

컴퓨터 장비로서,

하나 이상의 메모리 유닛을 포함하는 메모리; 및

하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 상기 메모리는 상기 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 상기 코드는 상기 처리 장치 상에서 실행될 때 제35항의 방법을 수행하도록 구성되는,

컴퓨터 장비.

청구항 37

비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 제35항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

청구항 38

검증 당사자가 타겟 당사자 또는 상기 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하게 하는 컴퓨터 구현 방법으로서,

상기 방법은, 상기 검증 당사자에 의해:

상기 타겟 당사자 또는 상기 검증 당사자의 임의의 장비 외부에 있는 데이터 저장소의 응답 데이터 레코드에 액세스하는 단계 - 상기 데이터 저장소는 신뢰할 수 있는 제3자의 제3자 컴퓨터 장비의 일부이거나 공개 피어-투-피어 공개 매체이고, 상기 응답 데이터 레코드는 a) 상기 타겟과 연관된 PUF(physically unclonable function)를 포함하는 PUF 모듈에 챌린지를 입력한 것으로부터 발생한 응답의 값, 또는 b) 상기 응답의 변환을 포함하는 증명을 포함함 - ;

상기 챌린지를 포함하는 요청을 상기 타겟에 전송하고, 이에 응답하여, 상기 응답의 추가 인스턴스를 다시 수신하는 단계;

비교를 수행하고, 상기 비교 결과가 일치한다는 조건으로 상기 타겟의 아이덴티티를 검증하는 단계를 포함하고, 상기 비교는 a)의 경우에서 상기 응답의 저장된 값이 상기 추가 인스턴스와 일치하는 것, 또는 b)의 경우에서 상기 증명이 상기 추가 인스턴스에 적용된 동일한 변환과 일치하는 것을 포함하는,

컴퓨터 구현 방법.

청구항 39

컴퓨터 장비로서,

하나 이상의 메모리 유닛을 포함하는 메모리; 및

하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 상기 메모리는 상기 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 상기 코드는 상기 처리 장치 상에서 실행될 때 제38항의 방법을 수행하도록 구성되는, 컴퓨터 장비.

청구항 40

비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 제38항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

발명의 설명

기술 분야

[0001] 본 개시는 물리적 복제 방지 함수(physically unclonable function; PUF) 분야에 관한 것이다.

배경 기술

[0002] PUF(physical unclonable function)는 결정론적이지만 예측 불가능한 물리적 현상을 포함하는 함수를 지칭하는 기술 용어이다. PUF는 때로는 또한 물리적 랜덤 함수로서 지칭된다. PUF는 "챌린지"로서 지칭된 입력을 수신하고 PUF에 의해 사용되는 챌린지 및 물리적 현상에 종속하여 대응하는 "응답"으로서 지칭되는 출력을 생성한다. PUF들은 때로는 강한 PUF 및 약한 PUF로 분류된다. 강한 PUF는 통상적으로 어떠한 임의적 챌린지의 값도 취할 수 있는 매우 다수의 상이한 챌린지들에 대한 개개의 응답을 생성할 수 있다. 약한 PUF는 단일 응답 또는 적은 수의 응답들에 대해서만 응답을 생성할 수 있다(통상적으로 챌린지는 어떠한 임의적 값도 취할 수 없음). 즉, 강한 PUF는 매우 다수의 챌린지-응답 쌍들(이는 큰 챌린지-응답 공간을 가짐)을 갖는 반면, 약한 PUF는 단일 챌린지-응답 쌍 또는 제한된 수의 챌린지-응답 쌍들(작은 또는 제한된 챌린지-응답 공간)을 갖는다. 일 정의에 따르면, 약한 PUF는 챌린지 비트들의 수에 선형적인 다수의 응답들을 갖거나 더 일반적으로 다른 파라미터들에서 선형을 초과하여 늘어나지 않는 다수의 응답들을 갖는다.

[0003] 강한 PUF의 알려진 예는 광학 PUF이다. 예컨대, 광학 PUF는 레이저, 광학 센서, 고체 광학 매체에 세팅된 기포들 또는 다른 이러한 인공물들을 갖는 고체 광학 매체를 포함할 수 있다. 레이저는 회절 또는 산란 패턴(이는 매체 내 기포들 또는 인공물의 효과임)을 생성하도록 제어 가능한 각도로 광학 매체를 통해 비친다. 센서는 이 패턴을 감지하도록 배열된다. 챌린지는 레이저의 각도이고, 응답은 감지된 패턴에 기초하여 생성된다.

[0004] 약한 PUF의 예는 SRAM PUF이다. 이 경우에, 챌린지는 SRAM(static random access memory)을 켜는 것이다. 하나의 SRAM으로부터 다른 SRAM으로 약간의 제조 차이들로 인해, SRAM 셀들은 파워 업(power up)시에 0/1 상태들의 고유한 패턴에 빠지게 될 것이며, 이는 이에 따라 개별 SRAM의 특징적인 핑거프린트를 형성한다. PUF는 파워 업 시에 응답으로서 이를 출력하도록 구성된다.

[0005] PUF는 이룰테면, (예컨대, 문서에 서명하거나 암호화하기 위한) 암호화 알고리즘들에 사용하기 위한 키를 생성하기 위한 수단으로 사용될 수 있다. PUF의 또 다른 애플리케이션은 PUF를 통합하는 디바이스 이룰테면, 컴퓨터 디바이스의 식별을 위한 것이다. 주어진 챌린지에 대한 예상된 응답이 이전에 결정된 경우, 검증 당사자는 추후에 챌린지로 타겟 디바이스에 챌린지하여, 예상된 응답을 제공하는지를 체크하고, 그리하여 타겟 디바이스가 예상된 응답과 연관된 디바이스인지를 체크할 수 있다.

[0006] 제한된 챌린지 응답 공간으로 인해, 약한 PUF에 대한 입력-출력(i/o) 인터페이스는 단 하나 또는 제한된 수의 당사자들로 제한되는 경향이 있다(예컨대, 단 하나 또는 제한된 수의 신뢰할 수 있는 당사자들에게만 PUF에 대한 액세스가 물리적으로 또는 합법적으로 승인될 수 있거나 PUF에 대한 인터페이스가 패스워드로 보호될 수 있거나 이런 식임). 즉, 문제의 당사자 또는 당사자들만이, 챌린지를 제출하는 데 필요한 PUF에 대한 입력 및 응답이 다시 수신되는 출력에 대한 액세스를 획득할 수 있다. 반면에 강한 PUF들의 경우, 강한 PUF에 대한 i/o 인터페이스는 다수의 또는 제한되지 않은 수의 당사자들에 — 이들 전부가 반드시 알려지거나 신뢰할 수 있는 당사자들일 필요는 없음 — 광범위하게 이용 가능하게 될 수 있다. 그 이유는, 챌린지 응답 공간이 충분히 커서, 적대자가 모든 챌린지-응답 쌍들을 열거(enumerate)하는 것이 불가능하기 때문이고, 이에 따라 적대자가 PUF에 자유롭게 액세스하는 능력은 약한 PUF들에 대해 해당했을 바와 같은 PUF의 열거(enumeration) 및 스푸핑(spoofing)을 허용함으로써 그의 보안을 손상시키지 않아야 한다.

- [0007] 상이한 기술 영역에서, 블록체인은 블록체인의 복제본이 분산 P2P(Peer-to-Peer) 네트워크(이하 "블록체인 네트워크"로서 지칭됨) 내 복수의 노드들 각각에서 유지되며 널리 공개되는 분산 데이터 구조의 형태를 지칭한다. 블록체인은 데이터의 블록들의 체인을 포함하며, 각각의 블록은 하나 이상의 트랜잭션들을 포함한다. 소위 "코인베이스 트랜잭션(coinbase transaction)들"이 아닌 각각의 트랜잭션은 시퀀스 내 이전 트랜잭션을 다시 가리키며, 이는 하나 이상의 블록들에 걸쳐 있어 하나 이상의 코인베이스 트랜잭션들로 되돌아갈 수 있다. 코인베이스 트랜잭션은 아래에서 추가로 논의된다. 블록체인 네트워크에 제출된 트랜잭션들은 새로운 블록들에 포함된다. 새로운 블록들은 복수의 노드들 각각이 "작업 증명"을 수행하기 위해 경쟁하는 것 즉, 블록체인의 새로운 블록에 포함되기를 기다리는, 순서화되고 유효성 검증된 보류중인 트랜잭션들의 정의된 세트의 표현에 기초하여 암호화 퍼즐을 해결하는 것을 수반하는 "채굴"로서 지칭되는 프로세스에 의해 생성된다. 블록체인은 일부 노드들에서 프루닝(prune)될 수 있으며 블록들의 공개는 단순 블록 헤더들의 공개를 통해 달성될 수 있다는 것이 주의되어야 한다.
- [0008] 블록체인의 트랜잭션들은 다음 목적들: 디지털 자산(예컨대, 다수의 디지털 토큰들)을 전달하는 것, 가상화된 원장 또는 레지스트리 내 엔트리들의 세트를 순서화하는 것, 타임스탬프 엔트리들을 수신 및 프로세싱하는 것, 그리고/또는 인덱스 포인터들을 시간-순서화하는 것 중 하나 이상을 위해 사용될 수 있다. 블록체인 위에 부가적인 기능성을 쌓기 위해 블록체인이 또한 활용될 수 있다. 예컨대, 블록체인 프로토콜들은 트랜잭션의 데이터에의 부가적인 사용자 데이터 또는 인덱스들의 저장을 허용할 수 있다. 단일 트랜잭션 내에 저장될 수 있는 최대 데이터 용량에 대해 미리 지정된 제한이 없고 이에 따라 점점 더 복잡한 데이터가 통합될 수 있다. 예컨대, 이는 블록체인에 전자 문서를 저장하거나, 오디오 또는 비디오 데이터를 저장하는 데 사용될 수 있다.
- [0009] 블록체인 네트워크의 노드들(종종 "채굴자들"로서 지칭됨)은 분산 트랜잭션 등록 및 검증 프로세스를 수행하며, 이는 나중에 보다 자세히 설명될 것이다. 요약하면, 이 프로세스 동안, 노드는 트랜잭션들을 유효성 검증하여 유효한 작업 증명 솔루션을 식별하려고 시도하는 블록 템플릿에 삽입한다. 유효한 솔루션이 발견되면, 새로운 블록이 네트워크의 다른 노드들로 전파되고, 이에 따라 각각의 노드가 블록체인 상에 새로운 블록을 레코딩하는 것을 가능하게 한다. 트랜잭션을 블록체인에 레코딩하기 위해, 사용자(예컨대, 블록체인 클라이언트 애플리케이션)는 트랜잭션을 전파될 네트워크의 노드들 중 하나로 전송한다. 트랜잭션을 수신하는 노드들은 유효성 검증된 트랜잭션을 새로운 블록에 통합하는 작업 증명 솔루션을 찾기 위해 경합할 수 있다. 각각의 노드는 트랜잭션이 유효하기 위한 하나 이상의 조건들을 포함하는 동일한 노드 프로토콜을 시행하도록 구성된다. 유효하지 않은 트랜잭션들은 블록들 내로 통합되거나 전파되지 않을 것이다. 트랜잭션이 유효성 검증되고 그리하여 블록체인 상에 수락된다고 가정하면, 트랜잭션(임의의 사용자 데이터 포함함)은 이에 따라 불변의 공개 레코드로서 블록체인 네트워크 내 노드들 각각에 등록되고 인덱싱된 상태로 유지된다.
- [0010] 최신 블록을 생성하기 위해 작업 증명 퍼즐을 성공적으로 해결한 노드는 통상적으로 디지털 자산의 금액, 즉 다수의 토큰들을 분배하는 "코인베이스 트랜잭션"이라 불리는 새로운 트랜잭션으로 보상을 받는다. 유효하지 않은 트랜잭션들의 검출 및 거절은 네트워크의 에이전트들로서 작용하는 경쟁 노드들의 액션에 의해 시행되며 불법 행위를 보고하고 차단하도록 장려된다. 광범위한 정보 공개는 사용자들이 노드들의 성능을 지속적으로 감사하도록 허용한다. 단순 블록 헤더들의 공개는 참가자들이 블록체인의 지속적인 무결성을 보장하도록 허용한다.
- [0011] "출력 기반" 모델(때로는 UTXO 기반 모델로서 지칭됨)에서, 주어진 트랜잭션의 데이터 구조는 하나 이상의 입력들 및 하나 이상의 출력들을 포함한다. 임의의 지출 가능한 출력은 진행중인 트랜잭션 시퀀스로부터 도출 가능한 디지털 자산의 금액을 지정하는 요소를 포함한다. 지출 가능한 출력은 때로는 UTXO("미지출 트랜잭션 출력")로서 지칭된다. 출력은 출력의 향후 리딤션(redemption)을 위한 조건을 지정하는 잠금 스크립트를 더 포함할 수 있다. 잠금 스크립트는 디지털 토큰들 또는 자산들을 유효성 검증하고 이전하는 데 필요한 조건들을 정의하는 술어이다. (코인베이스 트랜잭션 이외의) 트랜잭션의 각각의 입력은 선행 트랜잭션의 이러한 출력에 대한 포인터(즉, 참조)를 포함하고, 가리키는 출력의 잠금 스크립트를 잠금해제하기 위한 잠금해제 스크립트를 더 포함할 수 있다. 따라서 트랜잭션들의 쌍을 고려하고, 이들을 제1 및 제2 트랜잭션(또는 "타겟" 트랜잭션)이라고 한다. 제1 트랜잭션은 출력을 잠금해제하는 하나 이상의 조건들을 정의하는 잠금 스크립트를 포함하고 디지털 자산의 금액을 지정하는 적어도 하나의 출력을 포함한다. 제2의 타겟 트랜잭션은 제1 트랜잭션의 출력에 대한 포인터를 포함하는 적어도 하나의 입력, 및 제1 트랜잭션의 출력을 잠금해제하기 위한 잠금해제 스크립트를 포함한다.
- [0012] 이러한 모델에서, 제2의 타겟 트랜잭션이 블록체인 네트워크에 전송되어 블록체인에서 전파 및 레코딩될 때, 각각의 노드에 적용되는 유효성에 대한 기준들 중 하나는, 잠금해제 스크립트가 제1 트랜잭션의 잠금 스크립트에

정의된 하나 이상의 조건들 전부 충족하는 것일 것이다. 다른 하나는 제1 트랜잭션의 출력이 다른 더 앞선 유효한 트랜잭션에 의해 이미 리던되지 않았다는 것일 것이다. 이러한 조건들 중 임의의 것에 따라 유효하지 않은 타겟 트랜잭션을 발견한 임의의 노드는 이를 전파하지 않거나(유효한 트랜잭션으로서 전파하지 않으나, 어쩌면, 유효하지 않은 트랜잭션을 등록하기 위해 전파함) 블록체인에 레코딩될 새로운 블록에 이를 포함시키지 않을 것이다.

[0013] 대안적인 유형의 트랜잭션 모델은 계정 기반 모델이다. 이 경우에 각각의 트랜잭션은 과거 트랜잭션들의 시퀀스에서 선행 트랜잭션의 UTXO를 뒤로 참조하기 보다는, 절대 계정 잔액을 참조함으로써 전달될 금액을 정의한다. 모든 계정들의 현재 상태는 블록체인과 별개로 노드들에 의해 저장되며 지속적으로 업데이트된다.

발명의 내용

[0014] 종래에, 타겟 당사자(검증될 당사자, 즉, 증명자)는, 나중에 참조할 수 있도록 로컬에 저장해야 하는 검증 당사자(챌린저)와 직접 하나 이상의 CR 쌍을 셋 업해야 한다. 반면에, 본 개시는 검증 당사자가 셋 업에 수반될 필요가 없고 CR 쌍을 스스로 저장할 필요가 없는 방식을 제공한다.

[0015] 본원에서 개시된 일 양상에 따라, 하나 이상의 검증 당사자가 타겟 당사자 또는 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위한 컴퓨터 구현 방법이 제공된다. 방법은, 셋-업 국면에서, 하나 이상의 검증 당사자 중 임의의 검증 당사자 이외의 셋-업 당사자에 의해: PUF(physically unclonable function)에 기초하여 개개의 하나 이상의 응답 세트를 생성하기 위해 PUF를 포함하는 PUF 모듈에 하나 이상의 챌린지 세트를 입력하는 단계; 및 응답 세트 각각에 대한 개개의 응답 데이터 레코드를 타겟 당사자 또는 하나 이상의 검증 당사자 중 임의의 당사자의 임의의 장비 외부에 있는 데이터 저장소에 저장하는 단계를 포함한다. 데이터 저장소는 제3자 컴퓨터 장비의 일부 또는 공개 피어-투-피어 공개 매체(예컨대, 블록체인) 중 어느 하나이고, 이로써 후속 검증 국면에서 타겟의 아이덴티티를 검증하기 위해 하나 이상의 검증 당사자에게 응답 데이터 레코드를 이용 가능하게 만든다. 각각의 응답에 대한 응답 데이터 레코드는 개개의 응답 자체 또는 그로부터 도출된 데이터를 포함할 수 있다.

[0016] 실시예에서, PUF 모듈은 타겟 당사자 측에서 구현된다. 특히 제3자가 다수의 타겟 당사자에 대한 레코드를 저장하는 서비스인 경우, 이것은 셋-업 시에 타겟 당사자에 대한 PUF 모듈을 제3자가 유지해야 하는 것을 경감시켜주기 때문에 특히 경량의 배열이다. 일부 실시예에서, 제3자는 완전히 제거될 수 있고, 응답 데이터 레코드(예를 들어, 응답 증명)는 블록체인과 같은 공개 매체에 전체적으로 저장될 수 있다.

도면의 간단한 설명

[0017] 본 개시내용의 실시예들의 이해를 보조하기 위해 그리고 그러한 실시예들이 어떻게 실행될 수 있는지를 보여주기 위하여, 단지 예로서 첨부 도면들에 대한 참조가 이루어진다.

도 1은 블록체인을 구현하기 위한 시스템의 개략적인 블록도이다.

도 2는 블록체인에 레코딩될 수 있는 트랜잭션들의 일부 예들을 개략적으로 예시한다.

도 3은 PUF의 챌린지 및 응답을 개략적으로 예시한다.

도 4는 PUF를 포함하는 시스템의 개략적인 블록도이다.

도 5a는 본원에서 개시된 실시예들에 따른 확장된 PUF의 개략적인 블록도이다.

도 5b는 비-확장 동작 모드에서 확장된 PUF의 개략적인 블록도이다.

도 6은 챌린지-응답 쌍들의 분배에서 신뢰할 수 있는 제3자 또는 공개 매체를 수반하는 시스템의 개략도이다.

도 7은 본원에서 개시된 실시예들에 따른 검증 프로세스의 개략적인 흐름도이다.

도 8a 내지 도 8c는 본원에서 개시된 실시예들에 따라, 마스터 챌린지로부터 챌린지들의 세트를 생성하는 방법들을 개략적으로 예시한다.

도 9는 체인 상에 응답 데이터를 레코딩하는 방법을 개략적으로 예시한다.

발명을 실시하기 위한 구체적인 내용

[0018] 사람들 및 기계들 둘 모두를 위한 키-생성 시스템들 및 프라이버시-보호 아이덴티티 시스템들과 같은 시스템들

의 견고성은 PUF(physically unclonable function)들의 참여로 개선될 수 있다. 이들은 서로 또는 블록체인과 같은 공개 시스템과 상호작용하는 당사자들 및/또는 자율 기계들일 수 있다.

[0019] 물리적 시스템들에 기초하고 물리적 디바이스들의 제조 시에 랜덤이고, 결정 불가능하며 반복 불가능한 변동들의 가정에 의해 보호되며 이들 기능들은 사람 아이덴티티와 그의 디바이스 사이에 설정된 링크를 강화하거나, 디바이스 그 자체에 대해 위조 불가능한 고유한 아이덴티티를 또한 설정하는 데 사용될 수 있다.

[0020] 문헌에서, 자신의 뚜렷한 특성들에 의해 카테고리화되는 PUF들은 약한 유형 및 강한 유형으로 카테고리화된다. 아래의 일 양상에 따르면, PUF의 이러한 유형들 둘 모두의 이점들을 갖는 실용적인 PUF 디바이스를 설명하기 위한 일반화된 확장된 PUF(ePUF) 프레임워크가 제공되는데; 즉, ePUF는 구현하기에 실용적이고 비용 효율적인 상태를 유지하면서, 애플리케이션에서 사용될 광범위한 챌린지-응답 쌍들을 생성할 수 있다.

[0021] 보다 일반적으로 PUF들 및 챌린지-응답 쌍들의 관리에 관한 다양한 양상들이 본원에서 개시된다. 이들 상이한 양상들은 개별적으로 또는 임의의 조합으로 사용될 수 있다. 이들은 예컨대, 다음을 포함한다:

[0022] I. PUF의 챌린지-응답 공간을 확장하기 위한 확장된 PUF;

[0023] II. ePU 디바이스들을 사용함으로써 사람 및/또는 디바이스 아이덴티티를 설정하기 위한 일 세트의 블록체인-불가지론적 프로토콜들;

[0024] III. 블록체인을 활용함으로써 이러한 아이덴티티 프로토콜들을 개선하기 위한 프레임워크;

[0025] IV. 챌린지-응답 쌍들의 경량 저장을 위한 기술; 및

[0026] V. 디바이스들에 의한 검증 가능한 컴퓨테이션에 위한 그리고 단순화된 지불 검증(SPV) 프로세스들을 위한 KYC의 구현과 같은 다양한 문제들에 대한 ePUF 디바이스들의 일 세트의 신규 애플리케이션들.

[0027] 1. PUF(PHYSICALLY UNCLONABLE FUNCTION)들 - 서두들

[0028] PUF(physically unclonable function)들이라는 용어는 범용 랜덤 함수로서 작용하는 물리적 시스템들 및 디바이스들의 클래스를 지칭한다. 이러한 PUF들은 종종 마이크로 미만(sub-micron) 스케일에서 그의 물리적 특성들에 의해 고유하게 특성화되며, 이는 물리적 자극으로 이러한 특성들을 프로빙(probing)함으로써 각각의 PUF가 고유하게 식별 및 검증될 수 있음을 의미한다.

[0029] 고레벨에서, PUF는 챌린지들을 응답들에 매핑하는 함수들로서 간주될 수 있는데: 그의 쌍들은 종종 챌린지-응답 쌍(challenge-response pair; CRP)들로서 지칭된다. 이러한 맵 F를 설명하기 위해, 다음과 같이 아래의 표기법이 사용될 수 있다:

$$[0030] \quad F: C \rightarrow R \quad \forall (C, R) \in \Phi_F$$

[0031] 여기서 C,R은 각각 챌린지들 및 응답들을 나타내고 Φ_F 는 PUF에 의해 생성될 수 있는 형태 (C,R)의 모든 챌린지-응답 쌍들의 세트이다.

[0032] PUF의 고유한 물리적 특성들은 통상적으로 실리콘 칩들과 같은 물리적 디바이스들의 제조 시에 내재된 랜덤 프로세스 변동들의 결과이다. 통상적으로, PUF에 대해 다음과 같은 가정이 이루어진다:

[0033] 1. 임의의 형태의 분석에 의해 물리적 시스템의 파라미터들을 완전히 결정하는 것은 다루기 어렵고; 및

[0034] 2. 물리적 시스템의 파라미터들은 PUF로서 사용되는 디바이스의 오리지널 제조자를 포함하여 어떠한 당사자에 의해서도 알려지지 않는다. 이 가정은 종종 제조자-저항(manufacturer-resistance)으로서 지칭된다.

[0035] 이러한 가정들은, 임의적 챌린지들에 대한 예측 불가능하지만 결정론적인 응답을 생성하는데 PUF가 사용되도록 허용한다. 이 챌린지-응답 프로세스는 도 3에 예시된 바와 같이 물리적 흑색 박스처럼 PUF를 취급한다.

[0036] 도 3은 물리적 흑색 박스로서 모델링된 PUF(302)를 도시한다. 제출 당사자(103S)는 PUF(302)에 대한 입력으로서 챌린지 C를 제출하고, 응답으로 PUF(302)는 대응하는 응답 R을 생성한다. 제출 당사자는, PUF(302) 자체가 구현되는 디바이스와 동일하거나 상이한 디바이스일 수 있는 제출 당사자의 컴퓨터 디바이스(미도시)와 같은 디바이스로부터 챌린지를 제출한다.

[0037] 제출 당사자(103S)는 타겟 당사자 또는 디바이스의 아이덴티티에 링크된 예상된 응답들의 세트를 설정하기 위해 셋-업 국면(추후에 논의되는 예들)의 부분으로서 챌린지-응답(CR) 쌍들을 생성하는 당사자일 수 있다. 또는,

제출 당사자(103S)는 생성된 응답이 예상된 응답과 매칭된다는 것을 검증하고 이에 따라 PUF를 소유한 타겟 당사자 또는 PUF(302)를 포함하는 타겟 디바이스의 아이덴티티를 검증하기 위해 추후 검증 국면에서 챌린지를 제출하는 검증 당사자일 수 있다.

[0038] 다른 예시적인 시나리오에서, 제출 당사자(103S)는 (예컨대, 블록체인 트랜잭션에 서명하기 위해) 블록체인 애플리케이션과 같은 암호화 애플리케이션에서 사용하기 위해 생성된 응답을 키로서 사용하거나 키를 생성하기 위한 시드로서 사용하고자 하는 당사자일 수 있다.

[0039] 도 4는 PUF(302)에 대한 인터페이스의 예를 포함하는 시스템을 도시한다. 시스템은 프로세서(402) 및 PUF(302)를 포함한다. 인터페이스는, 메모리에 저장되고 프로세서(402) 상에서 실행되도록 배열된 인터페이스 로직(404)을 포함한다. 인터페이스 로직(404)이 저장되는 메모리는 하나 이상의 저장 매체들(예컨대, 자기 매체 이클테면, 자기 디스크 또는 테이프, 또는 전자 매체 이클테면, ROM, EPROM, EEPROM, 플래시 메모리, SRAM, DRAM 등)을 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다. 프로세서(402)는 하나 이상의 프로세싱 유닛들(예컨대, 범용 프로세서 이클테면, CPU, 또는 애플리케이션 특정 가속기 프로세서 이클테면, GPU, DSP 또는 암호화-프로세서)을 포함할 수 있다. 또한, 인터페이스 로직(404)이 전용 하드웨어 회로, 또는 구성 가능하거나 재구성 가능한 회로 이클테면, PGA 또는 FPGA에서 부분적으로 또는 전체적으로 구현될 수 있다는 것이 배제되지 않는다.

[0040] 제출 당사자(103S)는 인터페이스 로직(404)을 통해 PUF(302)에 챌린지 C를 제출하기 위해 디바이스(미도시)를 사용한다. 제출 당사자(103S)에 의해 사용되는 디바이스는 예컨대, 프로세서(402)가 구현되는 외부 컴퓨터 디바이스 또는 동일한 컴퓨터 디바이스인 컴퓨터 디바이스일 수 있다. 그 후, PUF(302)는 대응하는 응답 R을 인터페이스 로직(404)을 통해 제출 당사자(302)의 디바이스로 다시 리턴한다. 추후에 더 자세히 논의되는 일부 실시예들에서, 인터페이스 로직(404)은 PUF(302)에 대한 액세스를 특정 당사자들, 예컨대, 패스워드, PIN 또는 바이오메트릭 정보와 같은 인식된 크리덴셜들을 제시할 수 있는 당사자들로부터 제한하는 액세스 제어 로직(406)을 포함할 수 있다. 그리고/또는, 프로세서(402)를 포함하는 디바이스에 대한 물리적 인터페이스는 이클테면, 인가된 사람만 액세스할 수 있는 방 또는 콤플렉스(complex)에 위치되게 하거나 잠긴 상자 또는 캐비닛에 유지되게 함으로써 제한될 수 있다. 그러나 대안적인 시스템들에서, 인터페이스 로직(404)은 임의의 당사자가 챌린지들로 질의하는 데 이용 가능하도록 만들어질 수 있다.

[0041] PUF의 챌린지-응답 프로세스는 선택한 응답들로부터 이들 챌린지들을 추출함으로써 의사-랜덤 데이터 값들의 생성을 허용한다. 예컨대, PUF들은 암호화에 사용될 랜덤 반복 가능 데이터를 추출하기 위한 키-생성기들로서 사용될 수 있다. PUF(302)는 결정론적이고 반복 가능한 방식으로 작동하여서, PUF는 다수의 별개의 경우들에 동일한 챌린지가 주어지면 동일한 응답을 산출할 것이라는 것에 주의한다.

[0042] PUF들로서 사용될 수 있는 다수의 상이한 물리적 시스템들이 존재하고 이러한 시스템들을 사용하여 PUF들의 다수의 상이한 구현들이 존재한다. PUF의 예시적인 예는 기포들을 포함하는 광학 매체이며, 이는 레이저에 의해 프로빙될 때, (i) 레이저의 포지션 및 (ii) 광학 매체의 소규모 파라미터들에 의해 결정론적으로 결정되는 응답 회절 또는 '스펙클(speckle)' 패턴을 생성한다.

[0043] 1.1. PUF들의 클래스들

[0044] 1.1.1 약한 PUF: 약한 PUF들은 작은 챌린지-응답 공간을 가짐으로써 특징화되고, 대부분은 CRP 공간의 크기가 $|\Phi_F|=1$ 이 되도록 단일 챌린지만을 갖는다. 일반적으로 약한 PUF에 대한 챌린지-응답 공간은 차수 $O(n)$ 로 간주되며, 여기서 n 은 제어 불가능한 제조 변동들에 영향을 받는 PUF 내 구성요소들의 수이다.

[0045] 약한 PUF들의 경우에, 통상적으로, PUF의 응답들에 대한 액세스가 제한된다고 또한 가정된다. 이는, 약한 PUF에 의해 서비스되는 소수의 CRP들로 인해, 적대자가 합당한 시간에 이러한 모든 쌍들을 열거할 수 있고 이에 따라 PUF의 거동을 에뮬레이트하거나 '스푸핑'할 수 있기 때문이다. 이 제한은 때로는, 약한 PUF들의 거동을 논의할 때 제한된 챌린지-응답 인터페이스로서 지칭된다.

[0046] 이러한 특성들은 약한 PUF들을 키-생성기로서 암호화 애플리케이션에서 사용하기에 가장 자연스럽게 적합하게 만들며, 여기서 PUF에 의해 생성된 하나(또는 소수)의 CRP(들)는 암호화 동작들을 위해 이클테면, 온-디바이스 비-휘발성 메모리(NVM)를 암호화하기 위해 또는 HMAC 대칭 키로서 사용하기 위해 개인 키로서 사용될 수 있다. 이러한 경우들에서, PUF의 응답으로부터 도출된 키는 비밀로 유지되고, 수행된 암호화 프로세스의 보안 및 또한 PUF 그 자체의 보안 둘 모두를 위해 디바이스의 소유자에게만 알려져야 한다.

- [0047] 약한 PUF의 지배적인 그리고 널리 구현된 예는 SRAM PUF이며, 여기서 'SRAM'이라는 용어는 '정적 랜덤-액세스 메모리'를 지칭한다. SRAM PUF의 설계는 SRAM 칩들의 '파워-온(powered-on)' 상태에서의 변동을 활용하며, 이는 각각, 칩이 파워-온될 때 칩의 SRAM 셀이 '0' 또는 '1' 상태에 있는 변동들로 인해 고유한 핑거프린트를 갖는다.
- [0048] 이 경우에, PUF 구성은 (즉, SRAM 칩을 파워-온함으로써) PUF를 프로빙하기 위한 하나의 고정 모드 및 이에 따라 단 하나의 CRP만 존재하기 때문에 약한 것으로 간주된다. 이 경우에, 하나의 그리고 유일한 '챌린지'는 SRAM 칩에 전력을 공급하는 것이며, 응답은 파워-온 상태에서부터 도출된 고유한 핑거프린트이다. 응답의 비밀성을 보장하기 위한 액세스 제어는 또한, SRAM PUF가 사용되는 디바이스 상의 적소의 기존 메모리 액세스 제어 정책 또는 메커니즘, 또는 디바이스 상에서 사용되는 대안적인 메커니즘들을 사용하여 구현될 수 있다.
- [0049] 이를테면, SRAM PUF의 경우에 일부 PUF 구현들의 특징은 PUF들에 의해 생성된 응답들에서 오류-정정을 사용하여, 동일한 챌린지가 조건-불변 및 시간-불변 방식으로 동일한 응답을 산출할 것을 보장한다는 것이다. 이러한 오류-정정 기술들의 세부사항은 당업자들에게 알려져 있다. 일부 경우들에서, 오류-정정 프로세스는 오류-정정을 용이하게 하기 위해 나중에 요청 시 생성되는 응답과 결합되는 헬퍼 데이터(helper data)의 소스를 제공하기 위해 PUF 디바이스들이 초기에 '등록'될 것을 요구할 수 있다.
- [0050] **1.1.2. 강한 PUF들:** 약한 PUF들과 대조적으로, 강한 PUF들은 활용될 수 있는 가능한 챌린지-응답 쌍들(CR-쌍들 또는 CRP들)의 큰 공간을 가짐으로써 특징화된다. CRP들의 이 큰 공간은, 적대자가 다항 시간(polynomial time)에 강한 PUF의 도메인 내의 모든 챌린지-응답 쌍들을 열거하는 것이 실행 불가능한 것으로 간주된다는 것을 의미한다. 이 특성은 강한 PUF들이 일반적으로 보호되지 않는 챌린지-응답 인터페이스를 가질 수 있음을 의미하는데, 그 이유는 적대자가 PUF에 자유롭게 액세스하는 능력은 약한 PUF들에 해당했을 바와 같은 PUF의 열거 및 스푸핑을 허용함으로써 그의 보안을 손상시키지 않을 것이기 때문이다. 이 클래스의 PUF들은 또한 Φ_F 의 큰 서브세트를 알고 있는 적대자의 관점에서, 예측 불가능한 응답들을 생성하는 것으로 여겨지고, 이는 강한 PUF들이 큰 도메인을 통해 암호화 해시 함수와 보다 유사하게 작동한다는 것을 의미한다.
- [0051] 그러나 챌린지 C가 제시될 때 강한 PUF에 의해 응답 R만이 제공되어야 하고 PUF의 내부 작동 또는 동작에 대한 어떠한 다른 정보도 프로세스에서 누출되어서는 안 된다는 제한이 강한 PUF들에 가해진다. 이 제한은 적대자가 PUF의 거동을 뒷받침하는 물리적 시스템을 특성화하려고 시도할 수 있는 다양한 분석 공격들을 완화하기 위한 것이다. 이들은 종종 문헌에서 모델링 공격들로서 지칭된다.
- [0052] 약한 PUF들과 유사하게, 일부 강한 PUF 구성들은 디바이스들에 의해 생성된 응답들의 정확성을 보장하기 위해 오류 정정 기술들에 의존할 수 있다.
- [0053] 강한 PUF들의 기존의 메인 애플리케이션들은 내재적 챌린지-응답 메커니즘을 사용하여 시스템 인증 및 식별을 용이하게 하는 것이다. 이러한 메커니즘들은 직접 두 당사자들 간의 공유 비밀들로서 CRP들의 생성을 수반하는 프로토콜들에 의존하며, 종종 적어도 한 당사자가 다른 당사자에 대한 인증 토큰들로서 사용되도록 CRP들의 테이블을 미리 생성(초기 셋-업)하도록 요구한다.
- [0054] 강한 PUF 구현의 가장 이른 예들 중 하나는 광학 PUF 시스템이었다. 이 구성에서, PUF는 입사광을 산란시키는 제조 변동들의 결과로서 랜덤으로 분포된 물리적 불완전성들을 포함하는 광학 매체를 포함한다.
- [0055] 이 PUF 구조는 광 산란 매체로 지향되는 레이저 빔에 의해 프로빙될 수 있다. 이 경우에, 입사 빔의 방향 및 편광은 챌린지를 형성하고, 관찰된 산란 패턴이 PUF 응답으로서 취해진다.
- [0056] 그러나 이 강한 PUF 구성은, 측정 디바이스가 PUF 디바이스의 나머지 부분과 별개이고 반도체 구성요소들과 직접 통합하기가 또한 어렵다는 사실로 인해, 구현하기 복잡하다. 이는 장치 그 자체와 연관된 비용들 및 일상적인 애플리케이션을 위한 장치의 유용성을 감소시키는 에런지먼트의 휴대성의 결여에 추가된다.
- [0057] 이러한 이슈들 중 일부를 극복하는 APUF(Arbitrator PUF)로서 알려진 전기 통합식 강한 PUF가 이후 제안되었다. 이 구조는 신호 다중화를 활용하고 전기 구성요소들에서의 런타임 지연을 활용한다. 다수의 다른 강한 PUF 구성들이 동시에 제안되었지만, 다수는 널리 사용하기 위한 실질적인 적합성이 부족하고, 다수는 보안 및 잠재적 공격 벡터들과 관련하여 연관된 약점을 갖는다. 예컨대, 매우 문제가 있는 잠재적인 공격은 중간자 공격(man-in-the-middle attack)으로, 이를 통해 공격자는 평문으로 제출된 챌린지들을 가로채고 보증된 컴퓨테이션들을 스푸핑할 수 있다.
- [0058] **1.1.3. 제어된 PUF들:** CPUF(controlled PUF)로서 알려진 PUF의 제3 클래스는 기존의 강한 PUF 구조들을 개선하

지만 이들을 빌딩 블록으로서 사용한다. 이러한 PUF들은 강한 PUF를 사용하고 PUF에 대한 액세스를 제한하는 부가적인 제어 로직을 적용하며, 이는, 보통은 보호되지 않은 챌린지-응답 인터페이스를 가질 수 있는 비-제어(non-controlled) 강한 PUF로부터 제어된 PUF들 구별한다.

- [0059] 도 4에 도시된 바와 같이, 이제 더 큰 PUF 디바이스의 일부인 PUF에 적용된 제어 로직(406)은 PUF(302) 그 자체에 대한 액세스를 중재할 수 있다. 이는 제어 로직 구성요소(406)가 어느 챌린지들이 PUF에 제시될지를 제한할 수 있을 뿐만 아니라, 후속 응답들이 사용자에게 어떻게 드러날지를 제어할 수 있음을 의미한다.
- [0060] CPUF 구성에서, 바람직하게는, 제어 로직 구성요소(406)는 강한 PUF 구성요소 내에 매립되거나 둘러싸여야 한다. CPUF의 일 정의에 따르면, PUF는 분리할 수 없는 방식으로 PUF에 물리적으로 링크된 알고리즘을 통해서만 액세스될 수 있는 경우(즉, 알고리즘을 우회하려는 시도는 PUF의 파괴로 이어질 것임) 제어되는 것으로 여겨진다. 이 매립은 제어 로직의 프로빙을 훨씬 더 어렵게 만들어야 한다.
- [0061] 이는 PUF 구성요소와 제어 로직 구성요소 간에 상호-유익한 관계를 설정할 것이어서, 서로 일 유형의 공격을 완화할 수 있다. 즉, PUF 디바이스 내의 제어 로직의 캡슐화 그 자체는, 물리적 또는 침입 공격들이 PUF 구성요소를 복구할 수 없을 정도로 손상시키고 그의 응답을 변경할 것이기 때문에 이들로부터 제어 로직을 보호하는 반면, 제어 로직은 CRP들 또는 PUF 그 자체의 기본이 되는 내부 물리적 시스템에 대한 다른 정보를 추출하기 위한 프로토콜 레벨 공격들로부터 PUF 구성요소를 자연스럽게 보호한다.
- [0062] CPUF들의 애플리케이션들은 강한 PUF들과 거의 동일하지만, 보다 강력한 방식으로 달성될 수 있다. 특히, 보증된 컴퓨테이션들 및 실행 증명이 위에서 약속된 프로토콜들로 쉽게 달성될 수 있다.
- [0063] 강한 APUF(Arbiter PUF)의 설계를 확장한 CPUF의 초기 예는 제어 로직이 이미 설명한 방식으로 APUF 그 자체와 얽히도록 요구하여서, 제어 로직 및 APUF는 상이한 유형들의 공격으로부터 서로를 상호 보호한다. 제어된 APUF 설계는 시스템의 트랜지언트 응답을 통합함으로써 집적 회로(IC)로부터의 단일 정적 응답으로부터 CRP들의 큰 세트를 생성한다.
- [0064] 제어된 PUF의 다른 알려진 예는 PUF-FSM 구조이다. 이것은 APUF 구성요소 그 자체의 챌린지-응답 인터페이스에 대한 액세스를 제한하는 제어 로직으로서 작용하는 FSM(finite state machine)과 함께 강한 PUF(실제로는 APUF)를 포함한다.
- [0065] **1.2. 논의**
- [0066] **1.2.1. 실용성:** 표준 CMOS(complementary metal-oxide semiconductor) 구성요소와 통합할 수 있으면서도 실용적일 뿐만 아니라 경량인 강한 PUF들을 생성하는 것은 매우 난제라는 것이 본 문헌에서 인정된다. 반대로, SRAM PUF와 같은 약한 PUF들은 생산하기가 저렴하고 집적 회로 아키텍처들과 간단하게 결합될 수 있다.
- [0067] **1.2.2. PUF들에 대한 공격들:** 제안되고 연구된 다수의 상이한 공격들이 존재하며, 여기서 상이한 공격들은 특정 PUF 구성들 또는 클래스들을 타겟팅할 수 있다. 가장 널리 알려진 공격 유형들 중 일부는 다음과 같이 나열된다.
- [0068] · MITM 공격들 — 이러한 공격들은 제어되지 않는 강한 PUF들과 같은 PUF들을 타겟팅하며, 여기서 특히 보증된 컴퓨테이션을 위해 사용되는 경우, 적대자가 PUF의 응답을 가장(impersonate)하거나 스푸핑하기 위해 평균으로 만들어진 챌린지들을 가로챌 수 있다.
- [0069] · 모델링 공격들 — 이러한 공격들은 APUF와 같은 다수의 강한 PUF 구성들에 대한 취약점을 입증하였다.
- [0070] · 선택된 챌린지 공격들 — 이러한 공격들은 또한 강한 PUF들에 영향을 미치고 부분적으로 CPUF 아키텍처들로 이동하는 동기가 된다.
- [0071] 해당 PUF 시스템의 보안을 약화시키는 익스플로잇(exploit)들을 허용하는 다양한 PUF 설계들이 가진 다른 이슈들 이를테면, 일부 경우들에서 고유성의 부족이 또한 존재한다.
- [0072] **1.2.3 보안 모델들:** PUF 구성들의 보안 모델들은 그들의 CRP들이 발생하는 랜덤 프로세스 또는 제조 변동들이 제조자-저항성이고 분석 수단에 의해 PUF의 물리적 시스템을 특성화하는 것은 다루기 어렵다는 가정과 같은 일부 유사점들을 공유하는 경향이 있다. 그러나 3개의 메인 PUF 클래스들에 대한 보안 모델들에서 일부 차이들이 또한 존재한다.
- [0073] · **약한 PUF들** — 약한 PUF의 보안은 그의 CRP들이 비밀로 유지된다는 가정에 의존하고, 그렇지 않으면 디바이스

는 열거되고 가장될 수 있다. 이는, 엔트로피의 소스를 제공하고 암호화 동작들을 위해 해당 엔트로피의 저장을 안전하게 하는데 약한 PUF가 사용될 수 있지만, 실제 CRP 응답 데이터 그 자체는 프로세스에서 공개적으로 드러나지 않는다는 것을 의미한다.

[0074] · **강한 PUF들** — 강한 PUF의 보안은, 그의 CRP 공간이 챌린지 비트들의 수 면에서 지수적이 되는 경향이 있고, 이에 따라 전체 공간의 열거가 합리적인 시간프레임 내에 실행 불가능하다는 사실에 의존한다. 이는 약한 PUF들의 경우와 달리, 강한 PUF의 CRP 응답들이 디바이스에 의해 드러날 수 있음을 의미한다.

[0075] · **제어된 PUF들** — 제어된 PUF의 보안은 프로토콜 레벨 공격들로부터 보호하는 제어 로직 및 물리적 공격들로부터 보호하는 PUF 그 자체의 결합에 의해 결정된다.

[0076] 약한 PUF들과 차별되는 강한 PUF들의 두 가지 특성들은 다음과 같다. 첫째, 강한 PUF는 CRP들의 큰 세트를 갖는다. 이는, 강한 PUF는 큰 챌린지 공간 Φ_F 을 가지며, 여기서 약한 PUF는 통상적으로 그에 대해 사용 가능한 단 하나(또는 소수)의 챌린지(들)를 갖는다는 것을 의미한다. 또한 강한 PUF는 임의의 그리고 모든 알려진 CRP들에 대해 예측 불가능한 것으로 간주된다. 즉, 임의적 수의 CRP들에 대한 지식은 새로운 챌린지의 응답을 예측하는 데 어떠한 이점도 제공하지 않는다.

[0077] 둘째, 강한 PUF는 보호되지 않은 챌린지-응답 인터페이스를 가질 수 있다. 주어진 강한 PUF는 챌린지-응답 인터페이스에 대한 액세스를 제한하기 위해 액세스-제어 로직을 요구하지 않는다고 가정된다. 이는, PUF에 대한 물리적 액세스를 갖는 임의의 당사자는 PUF 또는 그의 물리적 특성들에 대한 어떠한 추가적인 정보도 드러내지 않고 임의적으로 챌린지들을 적용하고 응답을 획득할 수 있음을 의미한다.

[0078] 제어된 PUF는 보호된 챌린지-응답 인터페이스를 갖지만 강한 PUF와 같은 큰 챌린지-응답 공간도 갖는다.

[0079] 2. 확장된 PUF(ePUF)

[0080] 다음은 기본 PUF(302)의 주어진 챌린지-응답(CR) 쌍으로부터 다수의 2차 CR 쌍들을 생성함으로써 PUF의 CR 공간을 확장하기 위한 시스템 및 방법을 개시한다. 이는 본원에서 "확장된 PUF" 또는 "ePUF"로서 지칭될 수 있다. 이 아이디어는 통상적인 강한 PUF 메커니즘(이를테면, 레이저, 광학 매체 및 센서를 요구하는 광학 PUF)의 복잡성 또는 비실용성 없이, 예컨대, 단 하나 또는 제한된 수의 내재적 CR 쌍들을 갖는 약한 PUF의 챌린지-응답 공간을 확장하는 데 사용될 수 있다. 그러나, 원칙적으로, 개시된 기술들은 약한, 강한, 제어된 또는 다른 것이든 간에, 임의의 기본 PUF의 CR 쌍들의 수를 확장하기 위해; 또는 난독화 또는 재사용성과 같은 다른 목적들을 위해 임의의 PUF의 CR 쌍을 변형하는데 보다 일반적으로 사용될 수 있다.

[0081] 도 5a는 본원에서 개시된 실시예들에 따른 확장된 PUF(ePUF)(500)를 도시한다. ePUF(500)는 예컨대, 종래의 약한 PUF일 수 있는 성분 기본(constituent base) PUF(302)를 포함한다. ePUF(500)는 변형 함수(502), 예컨대, 암호화 해시 함수(예컨대, SHA256 등)와 같은 해시 함수를 더 포함한다. ePUF(500)는 또한 인터페이스 로직(404')을 포함하며, 이는 도 4와 관련하여 논의된 인터페이스 로직(404)과 유사할 수 있지만 추가적인 인터페이스 기능을 갖는다. 인터페이스 로직(404') 및 변형 함수(502)는 소프트웨어, 예컨대, 메모리에 저장되고 (변형 함수(502) 및 인터페이스(404'))의 추가적인 기능을 실행하지만, 도 4에 도시된 바와 같은) 프로세서(402) 상에서 실행되도록 배열된 매립된 펌웨어에서 구현될 수 있다. 인터페이스 기능(404') 및 변형 로직(504)이 저장되는 메모리는 하나 이상의 저장 매체들(예컨대, 자기 매체 이를테면, 자기 디스크 또는 테이프, 또는 전자 매체 이를테면, ROM, EPROM, EEPROM, 플래시 메모리, SRAM, DRAM, 퓨즈 래치들 등)을 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다. 인터페이스 기능(404') 및 변형 로직(504)이 실행되는 프로세서는 하나 이상의 프로세싱 유닛들(예컨대, 범용 프로세서 이를테면, CPU, 또는 애플리케이션 특정 가속기 프로세서 이를테면, GPU, DSP 또는 암호화-프로세서)을 포함할 수 있다. 또한, 인터페이스 로직(404') 및/또는 변형 함수(502)가 전용 하드웨어 회로, 또는 구성 가능하거나 재구성 가능한 회로 이를테면, PGA 또는 FPGA에서 부분적으로 또는 전체적으로 구현될 수 있다는 것이 배제되지 않는다.

[0082] 인터페이스 로직(404')은 변형 함수(502)에 그리고 또한, 선택적으로 기본 PUF(302)에 동작 가능하게 커플링된다. 기본 PUF(302)는 변형 함수에 동작 가능하게 커플링된다. 인터페이스 로직(404')은, ePUF(500)가 구현된 동일한 디바이스 또는 외부 디바이스일 수 있는 제출 당사자(103S)의 디바이스(도 5a에 도시되지 않음), 예컨대, 컴퓨터 디바이스로부터 입력을 수신하고 그에 출력을 제공하도록 배열된다. 제출 당사자(103S)는 셋업을 수행하고, 향후 참조를 위해 아이덴티티에 링크될 일 세트의 챌린지들 및 예상된 응답들을 생성하기 위해 ePUF(500)를 사용하는 당사자일 수 있고; 또는 생성된 응답이 이전에 설정된 예상된 응답과 매칭되는지를 검증

하기 위해 나중에 PUF를 사용하는 검증 당사자(또는 검증 당사자에게 제공하기 위해 응답을 생성하는 챌린저)일 수 있다. 다른 예시적인 애플리케이션에서, 제출 당사자(103S)는 ePUF(500)를 사용하여 키를 생성하기 위한 시드로서 또는 키로서 사용하기 위한 응답을 생성할 수 있다. 예컨대, 이것은 메시지를 암호화하거나 서명하기 위한 예컨대, 블록체인 트랜잭션의 일부에 서명하기 위한 암호화 키로서 사용될 수 있다.

[0083] 기본 PUF(302)는 입력으로서 "1차" 챌린지 C_w 를 수신하는 것에 대응하여, 출력으로서 "1차" 응답 R_w 를 생성하도록 동작 가능하다. "1차" 챌린지-응답(CR) 쌍은 본원에서 기본이 되는 성분 PUF(302)의 기본 또는 "천연"(즉, 내재적) CR 쌍을 지칭한다. 일부 실시예들에서, 기본 PUF(302)는 약한 PUF와 유사하게 단일 챌린지 C_w 에 대한 응답으로 단일 기본(즉, 1차) 응답 C_w 만을 생성할 수 있을 수 있다.

[0084] 동작에서, 인터페이스 로직(404')은 제출 당사자(103S)의 디바이스로부터 적어도 "2차" 챌린지 C_i 를 포함하는 챌린지 데이터(챌린지 입력)를 수신한다. 또한, 1차(기본) 챌린지 C_w 는 1차(기본) 응답 R_w 를 생성하기 위해 기본 PUF(302)에 입력된다. 실시예들에서, 제출 당사자(103S)는 ePUF(500)에 대한 챌린지 데이터 입력에 기본 챌린지 C_w 를 포함하도록 요구되고, 인터페이스 로직(404')은 기본 응답 R_w 를 생성하기 위해 이를 기본 PUF(302)로 라우팅한다. 그러나, 다른 실시예들에서, 1차 챌린지 C_w 가 메모리, 퓨즈 래치들 또는 전용 회로와 같은 내부 소스로부터 기본 PUF(302)로 입력되는 것이 배제되지 않는다. 어느 쪽이든, 변형 함수(502)는 입력으로서: a) 제출 당사자로부터 입력 챌린지 데이터에서 수신된 바와 같은 2차 챌린지 C_i , 및 b) 기본 PUF(302)에 의해 생성된 바와 같은 1차 응답 R_w 를 수신하도록 배열된다. 변형 함수(502)는 이들의 조합을 변형 함수(502)에 입력된 C_i 및 R_w 의 특정 조합에 대응하는 고유한 개개의 "2차" 응답 R_i 에 결정론적으로 매핑하도록 구성된 함수이다. 2차 챌린지 응답 쌍들은, 이들이 부분적으로 1차 응답 R_w 에 기초하여 생성되는 1차(기본) CR 쌍의 상단에 계층화(layered)된다는 의미에서 본원에서 "2차"로 지칭될 수 있다. 이들은 또한 "확장된 계층" 또는 "보완적인" 챌린지들 및 응답들이라 칭해질 수 있다.

[0085] 실시예들에서, 변형 함수(502)는 해시 함수, 예컨대, SHA 또는 DSA 해시 함수와 같은 암호화 해시 함수를 포함한다. 해시 함수가 사용될 수 있는 적어도 2개의 상이한 방식들이 존재한다. 첫 번째에서, 변형 함수(502)는 프리이미지(preimage)의 해시를 포함하고, 여기서 프리이미지는 수신된 2차 챌린지 C_i 및 생성된 1차 응답의 조합(예컨대, 컨캐터네이션(concatenation))을 포함한다. 즉 $R_i = H(C_i || R_w)$ 이다. 또는, 보다 일반적으로 프리이미지는 다른 요소들도 및/또는 컨캐터네이션 이외의 다른 형태의 조합도 포함할 수 있다.

[0086] 제2의 대안적인 접근법에서, 변형 함수(502)는 프리이미지의 해시를 포함하고, 여기서 프리이미지는 수신된 2차 챌린지를 포함하고 해시 함수는 생성된 1차 응답으로 초기화된다. 즉 $R_i = H(C_i)$ 이며, 여기서 H 는 R_w 에 의해 초기화된다. 또는, 다시 더 일반적으로, H 의 프리이미지는 적어도 C_i 를 포함하는 한 다른 요소들을 물론 포함할 수 있다. R_w 에 의해 초기화된다는 것은 해시 함수 H 에 의해 정의된 출력들에 대한 프리이미지의 매핑 자체가 R_w 에 의존할 것임을 의미한다. 반면에, 이전의 경우에, H 에 의해 야기된 출력들에 대한 프리이미지의 매핑은 R_w 에 의존하지 않고 오히려 프리이미지가 R_w 에 의존한다. 즉, 이전 단락에서, 프리이미지는 R_w 에 의존하고, 이 단락에서 H 만이 R_w 에 의존한다.

[0087] 여전히 보다 일반적으로, 원칙적으로, ePUF(500)에 의해 수용될 도메인의 각각의 가능한 C_i 에 대해, C_i 및 R_w 의 조합을 R_i 의 개개의 값에 결정론적으로 그리고 고유하게 매핑하는 한, 임의의 함수가 사용될 수 있다.

[0088] 2차 챌린지 C_i 는 다수의 상이한 가능한 값들 중 임의의 것을 취할 수 있고, 변형 함수(502)는 특정 수신된 2차 챌린지 C_i 의 값 및 1차 응답 R_w 의 값에 기초하여 이들을 2차 응답 R_i 의 개개의 값들에 매핑할 것이다. 따라서 ePUF(502)는 주어진 1차(기본) CR 쌍의 CR 공간을 다수의 2차 CR 쌍들로 확장할 수 있다. 실시예들에서, C_i 는 사용된 변수에 의해 지원되는 값들의 범위 내에서 어떠한 임의적 값도 취할 수 있다(예컨대, C_i 가 32비트 정수인 경우, 이는 2^{32} 개의 값들 중 임의의 것을 취할 수 있음).

[0089] 일부 실시예들에서, ePUF(500)는 도 5b에 도시된 바와 같이 대안적인 동작 모드에서 동작할 수 있을 수 있다. 이 경우에, 인터페이스 로직(404')은 입력 챌린지 데이터가 1차 챌린지 C_w 만을 포함함을 검출한다. 이에 응답하여, 이는 C_w 의 수신된 값을 기본 PUF(302)로 라우팅하고 결과적인 1차 응답 R_w 를 제출 당사자(103S)의 디바이스로 역으로 라우팅한다. 즉, 이 실시예에서 ePUF(500)는 또한 "레거시" 또는 "비-확장" 모드에서 동작할 수 있다.

[0090] 선택적으로, 애플리케이션에 의존하여, 인터페이스 로직(404')은 이를테면, 인가된 당사자에 매핑되는 것으로 인식하는 크리덴셜들(예컨대, 패스워드, PIN 또는 바이오메트릭 입력)을 제시할 수 있는 당사자에게만 액세스를 승인함으로써 제한된 수의 가능한 제출 당사자(103S)에게만 액세스를 제한하는 액세스 제어 로직(406)을 포함할

수 있다. 이 경우에, ePUF(500)은 CPUF의 한 형태로서 간주될 수 있다. 대안적으로, ePUF(500)에 대한 물리적 인터페이스는 이를테면, ePUF(500)를 포함하는 디바이스를 제한된 세트의 당사자들만이 액세스가 허용되는 방 또는 대내(premises)에 유지함으로써, 또는 그 디바이스를 잠긴 박스, 캐비닛 또는 방에 유지함으로써 법적으로 또는 물리적으로 보호될 수 있다. 이 경우에, ePUF(500)은 일종의 확장된 약한 PUF와 유사하게 간주될 수 있다.

[0091] PUF에 대한 인터페이스에 관한 이러한 물리적 제한들에 대한 대안으로 또는 추가로, 액세스는 또한 1차 챌린지에 대한 액세스를 제한함으로써 제한될 수 있다. 예컨대, 타겟 당사자(103T)("앨리스", 나중에 논의됨)는 Cw를 아는 유일한 당사자일 수 있다.

[0092] 그러나 다른 대안으로서, 인터페이스 로직(404')에 대한 액세스는 제한되지 않을 수 있으며, 예컨대, 임의의 당사자가 인터넷을 통해 그에 대해 자유롭게 질의할 수 있다. 이 경우에, ePUF(500)는 약한 기본 PUF 메커니즘을 확장함으로써 생성된 일종의 강한 PUF(502)와 유사하게 간주될 수 있다.

[0093] 도 5a에 도시된 어레인지먼트는 본원에서 확장된 PUF(ePUF)로서 지칭되는 PUF 디바이스의 새로운 하이브리드 클래스를 제공하며, 이는 나중에 제시되는 것과 같은 다수의 애플리케이션들을 위한 프레임워크로서 일반적으로 사용될 수 있다.

[0094] ePUF는 다음의 3개 모듈들: 본질적으로 약한 PUF와 같은 기본 PUF(302); 암호화 해시 함수와 같은 변형 함수(502); 및 인터페이스 로직 모듈(404')을 함께 포함하는, 도 5a에 도시된 바와 같은 물리적 디바이스 또는 시스템으로서 정의될 수 있다. 논의된 바와 같이, ePUF(500)는 암호화 해시 함수와 같은 변형 함수(404')를 도입함으로써 정규 PUF(302)에 대해 '확장'될 수 있는데, 그 이유는, 이것이 기본 약한 PUF(302)에 대해 $|\Phi_F| \sim 1$ 로부터, 약한 PUF의 물리적 시스템보다는 해시 함수의 선택에 의해 대신 경계가 정해지는 $|\Phi_F| \gg 1$ 로 고유한 챌린지 공간 Φ_F 의 크기를 증가시키기 때문이다.

[0095] 강한 PUF의 큰 CRP 공간과 약한 PUF의 실용성을 그 자체로 결합하는 시스템을 실현하는 아이디어는 이전에 탐구되었다. 강한 PUF의 캐릭터를 가진 시스템을 생성하기 위해 결합된 동작에서 다수의 FPGA-기반 약한 PUF를 사용하는 것이 알려져 있다. 여기서 의도는 부분적으로 기본 약한 PUF들의 CRP 공간을 '확장'하는 것이다. 그러나 이러한 성질의 기존 구성들은 실제로 제한된다. 위에서 언급된 FPGA 설계의 경우에, 시스템은 FPGA 상에 구축되어야 하며, 여전히 상대적으로 낮은 CRP 공간($\sim 2^{10}$)에 종속된다.

[0096] 본원에서 개시된 ePUF 설계는, 기존의 약한 PUF(302)에 인터페이스 로직 구성요소(404') 및 암호화 해시 함수(또는 다른 그러한 변형 함수)(502)를 추가하기만 하면 된다는 점에서 극도로 경량으로 설계된다. 예컨대, SRAM PUF가 널리 사용되는 약한 PUF(302)로서 선택되는 경우, 2개의 나머지 모듈들(404', 502)의 추가는 상당한 오버헤드를 생성하지 않아야 하는데, 예컨대, 소프트웨어(예컨대, 펌웨어)에서 작은 알고리즘 또는 비교적 간단한 하드웨어 회로 조각으로 구현되어야 한다. 또한, ePUF(500)의 가능한 출력들의 공간은 선택된 해시 또는 변형 함수(502)의 범위로 확장되며, 이는 위의 공간보다 상당히 크다. 예컨대, SHA-256 해시 함수가 선택되는 경우, 가능한 출력들(그리고 이에 따라 CRP들)의 공간이 즉시 2^{256} -1로 증가되고, 해시 함수 모듈 자체를 매핑하는 것 이상으로 하드웨어 오버헤드를 스케일링할 필요가 없다.

[0097] 도 5a는 ePUF(expanded PUF)(500)의 개략적 설계를 도시한다. 암호화 해시 함수가 사용되는 실시예들은 또한 ePUF(500)가 그의 CRP가 예측 불가능하다는 특성을 갖는다는 것을 의미하며, 이는 강한 PUF 시스템들에 대해서도 해당된다.

[0098] ePUF 디바이스의 제어 로직 요소(406)는 또한 이러한 구성에서 일반화될 수 있다. 제어 로직(406)은 예컨대, 애플리케이션에 적절한 경우, SRAM PUF와 유사한 물리적 보안으로서 간단하게 구현될 수 있다.

[0099] 대안적으로, 제어 로직 모듈(406)은 CPUF들과 함께 사용되는 것과 유사한 소프트웨어 제어 모듈로서 구현될 수 있으며, 여기서 이는 실제로, PUF 디바이스 자체 내에 매립되어 이전에 논의된 캡슐화의 상호 보안 이익들을 제공한다. 그러나 여기에서 ePUF 설계가 특히 CPUF들의 설계와 차별되는 포인트는 제어 로직이 이러한 방식으로 구현되기 위해 어떠한 엄격한 요건도 없다는 것이다.

[0100] 제어 모듈(406)에 대한 침입 공격이 반드시 ePUF 설계에서 약한 PUF 구성요소(302)의 거동을 변경한다고 가정할 필요는 없다. 대신, 이 요소의 구현은 사례별로 선택될 수 있다.

[0101] 2.1. ePUF들에 대한 챌린지들 및 응답들

[0102] ePUF에 대응하는 챌린지-응답 쌍들 $(C, R) \in \Phi_F$ 의 세트는 다음의 방식으로 정의될 수 있다:

$$\Phi_F = \{(C_w, R_w), (C_1, R_1), (C_2, R_2), \dots, (C_N, R_N)\},$$

$$F : C_i \rightarrow R_i, \quad \forall i \in (1, N)$$

$$F_w : C_w \rightarrow R_w$$

[0103]

[0104] 여기서 (C_w, R_w) 는 약한 PUF(302)의 기본 챌린지-및-응답에 대응하는 특권 CRP이고 맵 F_w 는 약한 PUF의 고유한 물리적 특성들에 의해 정의된다. 쌍 (C_w, R_w) 은 본원에서 ePUF의 기본 또는 1차 쌍으로서 지칭될 수 있다. 반대로, 맵 F 는 ePUF에 대해 선택된 암호화 해시 함수에 의해 정의된다. 도 5a-b는 ePUF(500)로부터 응답을 추출하는 것을 도시하며, 여기서(도 5b) 챌린지는 C_w 만이고, (도 5a) 챌린지는 또한 C_i 를 포함한다.

[0105] 확장된 PUF의 일부 실시예들에서, 모든 챌린지들 $C_i, i \in \{1, 2, \dots, N\}$ 는 기본 챌린지 C_w 를 동반해야 하며, 기본 응답 R_w 는 도 5a에 도시된 바와 같이 모든 다른 응답들 R_i 를 생성하기 위한 프로세스에 통합된다.

[0106] ePUF를 사용하여 일반 CRP들을 생성하기 위해 도 5a에 묘사된 프로세스는 이 기본 비밀 쌍을 임의의 다른 임의의 챌린지 C_i 에 적용하여 확장함으로써 기본 챌린지-응답 쌍 (C_w, R_w) 을 사용하도록 설계되었다. ePUF로부터 CRP들을 생성하는 데 사용되는 알고리즘이 결정론적 방식으로 기본 쌍 (C_w, R_w) 을 사용하는 경우 특정 용도로 맞춤화될 수 있다. `getResponse()`로 표시된 이러한 알고리즘의 간단한 예는 다음과 같이 작성될 수 있다.

[0107] **getResponse()**:

[0108] 입력들: **Challenge**

[0109] 1. 사용자/클라이언트로부터 **challenge** 를 획득함.

[0110] 2. **challenge == C_w** 인지를 체크?

[0111] i. 만약 그렇다면:

[0112] 1. **R_w** 를 획득하기 위해 **C_w** 로 약한 PUF 모듈을 프로빙함. 2. **Response ← R_w** 로 세팅함.

[0113] ii. 만약 아니라면:

[0114] 1. **Challenge** 를 **C_w** 및 **C_i** 구성요소들로 분리함.

[0115] 2. **R_w** 를 획득하기 위해 **C_w** 로 약한 PUF 모듈을 프로빙함.

[0116] 3. **C_i** 및 **R_w** 를 해시 함수 모듈로 전송함.

[0117] 4. **hash(C_i, R_w, H)** 를 컴퓨팅함

[0118] 5. **Response ← hash(C_i, R_w, H)** 로 세팅함

[0119] 3. **Response** 를 리턴함

[0120] 출력들: **Response**

[0121] 함수 $\text{hash}(C_i, R_w, H)$ 는 암호화 해시 함수 H 를 사용하여 해시 다이제스트를 컴퓨팅하는 데 사용되는 일반 함수이다. 함수 $\text{hash}()$ 는 다양한 방식들로 이룰테면, 간단한 경우 단순히 $H(C_i \parallel R_w)$ 를 컴퓨팅함으로써 구현될 수 있거나, $H(C_i)_{R_w}$ 를 택싱 컴퓨팅(taxing computing)함으로써 구현될 수 있으며, 여기서 값 R_w 는 해시 함수

H의 초기 벡터로 사용되었다. 어느 쪽이든, $\text{hash}()$ 의 출력은 C_i 및 R_w 둘 모두에 의존한다.

[0122] 도 5a 및 도 5b의 다이어그램들은 ePUF(500)에는 선택적으로 제어 로직 모듈(406)을 포함하는 인터페이스 로직(404')이 장비될 수 있음을 도시한다. 실시예들에서, 응답을 생성하는데 취할 수 있는 2개의 가능한 경로들이 있으며, 여기서 챌린지가 단순히 C_w 일 때 도 5b의 경로가 사용되고, 챌린지가 C_w 를 동반하는 새로운 값 C_i 일 때 도 5a의 경로가 사용된다. 이는 결정론적이다.

[0123] 개시된 ePUF 설계는 다음의 이점들 및/또는 다른 것들 중 임의의 것을 제공하기 위해 사용될 수 있다.

[0124] · 선택된 해시 함수의 도메인 및 범위에 의해 정의되는 큰 CRP 공간.

[0125] · PUF 자체로부터 제어 로직을 분리할 수 있는 유연성.

[0126] · 약한 PUF의 보안 프리미티브들.

[0127] 이것은, 사용자는 CPUF 디바이스와 유사하게 ePUF 디바이스를 사용할 수 있지만, PUF에 대한 제어된 액세스는 (I) 약한 PUF의 기본 CRP(C_w, R_w)를 안전하게 저장하는 것, 그리고 (II) 의도된 사용자에게로만 PUF 디바이스에 대한 물리적 액세스를 제한하는 것 둘 모두를 포함한다. 이 모델에서, 기본 쌍(C_w, R_w)은 형태 (C_i, R_i)의 극도로 매우 다수의 다른 CRP들이 도출될 수 있는 마스터 키와 같이 작용하며, 여기서 C_i 가 외부 또는 제3자에 의해 제출될 수 있다.

[0128] 2.2. ePUF의 애플리케이션

[0129] ePUF 디바이스의 가능한 애플리케이션들(사용 사례들)은 크게 적어도 2개의 메인 카테고리로 분류될 수 있다:

[0130] 1. 아이덴티티를 활동들 또는 컴퓨테이션들 동작들에 링크하는 것; 및

[0131] 2. 암호화 동작들을 위한 키-생성기로서 작용하는 것.

[0132] 애플리케이션 1은 기존의 강한 PUF들에 의해 가장 일반적으로 구현되고, 2는 기존의 약한 PUF들에 의해 가장 일반적으로 구현된다. ePUF 구성이 각각의 특성들을 결합한다는 사실은 ePUF가 어느 하나의 애플리케이션에 동등하게 적합한 것으로 취급될 수 있음을 의미한다. 애플리케이션 1에서, 이점은 대부분의 강한 또는 제어된 PUF보다 일반적으로 ePUF를 사용하여 그러한 애플리케이션들을 실제로 훨씬 더 쉽게 구현할 수 있다는 것이다.

[0133] 3. 아이덴티티 링키지 시스템

[0134] 이 섹션에서, 사람 또는 기계 아이덴티티들을 PUF 디바이스들에 링크하기 위한 일반 프레임워크가 개시된다.

[0135] 실시예들은 ePUF(expanded PUF)를 사용할 수 있다. 여기서 의도는, 다수의 상이한 사용 사례에 맞게 용도 변경될 수 있는 강력하면서도 매우 일반화되고 유연한 아이덴티티 시스템을 제공하는 PUF 아키텍처를 공식화하는 것이다. 이 구성에서 캡처하려는 특성들은 다음과 같다:

[0136] · 강한 PUF의 것에 필적하는 큰 CRP 공간;

[0137] · 약한 PUF의 것에 필적하는 실용성; 및

[0138] · CPUF의 것보다 유연한 제어 로직.

[0139] ePUF 설계는 다양한 아이덴티티-설정 프로토콜들에서 사용되는 PUF에 대한 기본 모델로서 사용될 수 있다. 실시예들은 프로세스에서 최종 사용자 또는 기계의 독립성을 허용할 수 있다. ePUF를 사용하도록 또한 용도 변경될 수 있는 기존 체계들이 셋업 동안 PUF 디바이스에 직접 액세스하기 위해 신뢰할 수 있는 제3자에 의존하는 경우, ePUF-기반 제안된 시스템들은 PUF 디바이스의 최종 사용자가 아이덴티티를 대신 설정하고 셋업 동안 제3자가 디바이스에 로컬로 또는 직접 액세스할 필요 없이 앞으로의 인증들에 참여하도록 허용할 수 있다.

[0140] 일부 구현들은 공개 블록체인을 도입함으로써 이러한 아이덴티티-링키지 프로토콜들의 견고성을 개선하고 추가로 확장할 수 있다. 여기서 사용될 수 있는 2개의 개념들은, (A) 변조-방지(tamper-proof) CRP 관리 시스템으로서 블록체인의 사용, 및 (B) 아이덴티티-링키지 프로토콜들에 사용되는 요청-응답 메시지를 중재하고 효율적인 폐지 시스템을 제공하기 위한 타임-스탬핑 서비스로서 블록체인 네트워크의 사용이다.

[0141] 도 6은 본원에서 개시된 실시예들에 따른 아이덴티티 링키지 및 검증을 위한 예시적인 시스템을 도시한다. 도 7은 대응하는 방법을 도시한다.

- [0142] 시스템은 PUF 모듈(603), 타겟 당사자(103T)의 컴퓨터 장비(102T) 및 응답 데이터 저장소(601)를 포함한다. PUF 모듈(603)은 도 5a 및 도 5b와 관련하여 이전에 설명된 바와 같은 ePUF(500)를 포함하거나, 대안적으로 도 3 및 도 4와 관련하여 이전에 설명된 바와 같이 종래의 PUF(302) 또는 PUF에 종래의 인터페이스 로직(404)을 더한 것만을 포함할 수 있다. 응답 데이터 저장소(601)는 제3자 컴퓨터 장비(602)의 일부일 수 있고 신뢰할 수 있는 제3자에 의해 관리되거나 대신, 블록체인과 같은 분산 피어-투-피어 저장 매체일 수 있다. 제3자 장비(602)는 예컨대, 하나 이상의 지리적 사이트들에 위치된 하나 이상의 서버 유닛들을 포함하는 서버 장비를 포함할 수 있다(클라우드 저장 기술은 그 자체로 당업계에 알려져 있음). 시스템은 검증 당사자(103V)의 컴퓨터 장비(102V)를 더 포함할 수 있거나, 일부 대안적인 경우들에서, 검증 당사자는 PUF 모듈(603), 타겟 당사자의 컴퓨터 장비(102T) 또는 제3자 컴퓨터 장비(602)와 직접 상호작용할 수 있다.
- [0143] 검증 당사자(103V), 타겟 당사자(103T) 또는 제3자인지 여부에 관계없이, 사용자 또는 당사자(103) 등의 액션에 대한 본원에서의 임의의 참조는 당사자가 해당 당사자의 컴퓨터 장비(102)를 통해 행동할 가능성을 커버한다. 간결함을 위해, 이는 매번 명시적으로 언급될 필요는 없을 것이지만, 암묵적으로 커버하는 것으로 이해될 것이다. 이는, A) 액션은 컴퓨터 장비에 대한 당사자에 의한 수동 사용자 입력의 제어 하에 수행되거나 이에 의해 트리거되거나, 또는 B) 액션은 당사자를 대신하여 컴퓨터 장비에 의해 자동으로 수행되는 가능성들 둘 모두를 커버한다(당사자가 액션을 수행한다고 언급하는 것은 반드시 해당 당사자의 인간 사용자가 수동으로 해당 액션을 착수한다는 것을 의미하는 것이 아니라, 당사자의 장비가 그/그녀를 대신하여 자율적으로 해당 액션을 수행한다는 의미일 수 있음). 의심의 여지를 없애기 위해, 당사자는 단일 개별 사람 또는 그룹 또는 사람들 또는 조직 예컨대, 회사, 자선 단체, 정부 기관, 시립 또는 학술 기관들을 지칭할 수 있다는 것에 또한 주의한다.
- [0144] 타겟 당사자(103T)의 컴퓨터 장비(102T)는 (예컨대, 제3자 장비(602)에 대한 연결에 의해) 응답 데이터 저장소(601)에 동작 가능하게 연결될 수 있다. 검증 당사자(103V)의 컴퓨터 장비(102V)는 (예컨대, 제3자 장비(602)에 대한 연결에 의해) 응답 데이터 저장소(601)에 동작 가능하게 연결될 수 있다. 타겟 당사자(103T)의 컴퓨터 장비(102T)는 검증 당사자(103V)의 컴퓨터 장비(102V)에 동작 가능하게 연결될 수 있다. 이러한 연결들 중 임의의 것은 하나 이상의 네트워크들, 예컨대, 인터넷 또는 모바일 셀룰러 네트워크와 같은 하나 이상의 광역 네트워크들을 통해 형성될 수 있다. 실시예들에서, 이러한 연결들 중 임의의 것은 개개의 보안 채널을 통해 형성될 수 있는데, 예컨대, 해당 두 당사자들 간에 공유되는 공유 비밀에 기초하여 설정될 수 있다. 본원에서 두 당사자들이 임의의 방식으로 이를테면, 챌린지를 전송하거나 응답 다시 수신하는 등을 함으로써 통신한다고 언급될 때마다, 이것은 이들 통신들이 각자의 컴퓨터 장비(102V, 102T; 102T, 602; 또는 102V, 602) 사이의 임의의 적합한 직접 또는 네트워크 연결을 통해 수행될 수 있는 가능성을 커버한다는 것이 이해될 것이다. 간결함을 위해, 이는 매번 명시적으로 언급될 필요는 없을 것이지만, 암묵적으로 커버되는 것으로 이해될 것이다.
- [0145] 타겟 당사자(103T)는 그의 아이덴티티가 PUF 모듈(603)에 기초하여 검증되는 당사자이거나, PUF 모듈(603)에 기초하여 검증될 디바이스를 소유하거나 달리 그 디바이스를 담당하거나 연관되는 당사자이다. 검증 당사자(103V)는 검증을 수행하는 당사자이다. 다수의 검증 당사자들(103V)(이들 각각은 개개의 컴퓨터 장비(102V)를 통해 행동할 수 있음)이 있을 수 있지만, 예시의 편의를 위해, 단 하나만이 도 6에서 도시된다. PUF 모듈(603)은 타겟 당사자(103T)의 소유일 수 있다. PUF 모듈(603)은 그의/그녀의 컴퓨터 장비(103T)에 통합되거나, 예컨대, 주변 장치로서 또는 로컬 네트워크를 통해 또는 조합으로 이에 연결될 수 있다(예컨대, 인터페이스 로직(404/404')은 컴퓨터 장비(103T) 상에서 구현될 수 있고 PUF(302)는 외부 주변 장치일 수 있음). 대안적으로, PUF 모듈(603)은 신뢰할 수 있는 제3자의 소유일 수 있다. 이는 예컨대, 주변 장치로서 또는 로컬 네트워크 또는 조합으로 제3자 컴퓨터 장비(602)에 통합되거나 이에 연결될 수 있다(예컨대, 인터페이스 로직(404/404')은 제3자 장비(602) 상에서 구현될 수 있고 PUF(302)는 외부 주변 장치일 수 있음).
- [0146] 일반적으로, 타겟 당사자(103T), 검증 당사자(103V) 또는 제3자 중 임의의 당사자는 도 3, 도 4 및 5와 관련하여 이전에 논의된 제출 당사자의 역할을 맡을 수 있다. 타겟 당사자(103T), 검증 당사자(103V) 또는 제3자 중 임의의 것은 제출 당사자의 역할을 맡을 수 있거나, 또는 PUF 모듈(603)을 사용하여, 하나 이상의 CR 쌍들의 세트를 설정하고 이를 나중의 검증 국면에서 사용하기 위해 타겟 당사자(103T)의 아이덴티티에 링크하는 셋-업 당사자의 역할을 맡을 수 있다. 일부 특정 예시적인 시나리오들은 나중에 보다 자세히 논의된다.
- [0147] 응답 데이터 저장소(601)는 셋-업 국면에서 PUF 모듈(603)에 의해 생성된 응답 데이터를 저장한다. 데이터 저장소(601)는 타겟 당사자(103T) 또는 타겟 당사자(103T)의 디바이스일 수 있는 타겟의 아이덴티티의 증거와 관련하여 이 응답 데이터를 저장한다. 검증 당사자(103V)는 응답 데이터 저장소(601)에 대한 액세스를 갖고 이를 사용하여 나중에 검증 국면 동안 타겟의 아이덴티티를 검증할 수 있다. 이를 행하기 위해, 검증 당사자(103V)는 셋-업 국면에서 사용된 챌린지들의 세트에 이전에 포함되었던 챌린지 Ci에 대한 응답 Ri를 생성하도록 타겟

에 챌린지한다. 타겟이 응답 데이터 저장소(601)에 저장된 것에 따라 예상된 응답을 생성할 수 있는 경우, 이는 타겟이 PUF 모듈(603)을 소유하거나 제어하고 있다는 증거가 되고, 따라서 셋-업 국면에서 아이덴티티가 캡처된 동일한 당사자인 것으로 가정될 수 있다.

[0148] 대안적인 변형에서, 응답 데이터 저장소(601)는 예컨대, 시드로서 응답을 사용하여 셋-업 국면에서 생성된 응답(들)에 기초하여 생성된 하나 이상의 개개의 공개-개인 키 쌍들의 하나 이상의 공개 키들을 저장할 수 있다. 타겟이 나중에 개인 키들 중 하나를 사용하여 메시지(예컨대, 문서 또는 블록체인 트랜잭션)에 서명하는 경우, 검증 당사자는 응답 데이터 저장소(601)로부터의 대응하는 공개 키를 사용하여 서명을 검증할 수 있다. 이러한 변형들에서 "응답 데이터"라는 용어는 반드시 응답 R_i 의 명시적 값이나 증명이 아니라, 응답 R_i 으로부터 도출된 데이터를 커버하기 위해 더 넓은 의미로 사용된다는 것에 주의한다.

[0149] 응답 데이터 저장소(601)는 공개적으로 액세스 가능하거나, 액세스가 적어도 하나의 검증 당사자(103V)를 포함하는 하나 이상의 당사자들의 제한된 세트에서만 제한될 수 있다. 이는 제3자 시스템(602) 상에 또는 피어-투-피어 방식으로 호스팅될 수 있거나, 대안적으로 이는 타겟 당사자(103T)의 컴퓨터 장비(102T) 또는 검증 당사자(103V)의 컴퓨터 장비(102V)에서 구현될 수 있다.

[0150] 도 7을 참조하면, 방법은 셋-업 국면(702) 및 검증 국면(704)의 두 국면들을 포함한다. 셋-업 국면에서, 단계(710)에서, 셋업 당사자로서 작용하는 타겟 당사자(103T) 또는 제3자 중 하나는 하나 이상의 챌린지들 $C_i(i=1 \dots n)$, 여기서 $n \geq 1$ 의 세트를 PUF 모듈(603)에 제출한다. 이들은 ePUF(500)가 사용하는 경우에 2차 챌린지들이다. 타겟 당사자(103T)가 PUF 모듈(603)을 소유하고 셋-업을 수행하는 경우에, 챌린지 C_i 는 타겟 당사자(103T)에 의해 생성되거나 제3자 시스템(602) 또는 검증 당사자(103V)로부터 수신될 수 있다. 제3자가 PUF 모듈(603)을 소유하고 셋-업을 수행하는 경우에, 챌린지들은 제3자 시스템(602)에 의해 생성되거나 타겟 당사자(103T) 또는 검증 당사자(103V)로부터 수신될 수 있다. 어느 쪽이든, 이에 응답하여, PUF 모듈(603)은 PUF(302/500)에 기초하여 응답들 R_i 의 대응하는 세트를 생성한다. 이들은 ePUF(500)의 경우에 2차 응답들이다. 따라서 이 방법은 CR 쌍들 $\{C_i, R_i\}$ 의 세트를 생성한다.

[0151] 실시예들에서, PUF 모듈(903)에 대한 액세스는 타겟 당사자(103T)(및 상이한 당사자인 경우, 셋-업 당사자)만이 응답들 R_i 에 대한 액세스를 획득할 수 있도록 제한된다. 이는 패스워드, PIN, 바이오메트릭 데이터 등과 같은 인식된 크리덴셜들을 제시할 수 있는 당사자에게만 액세스를 승인할 수 있는 액세스 제어 로직(404 또는 404')에 의해 달성될 수 있다. 그리고/또는, PUF 모듈(603)에 대한 물리적 인터페이스에 대한 액세스는 이를테면, 잠긴 컨테이너, 캐비닛 또는 방에 PUF 모듈(603)을 유지함으로써 물리적으로 보호될 수 있거나; 또는 이는 이를테면, PUF 모듈(603)을 특정 직원만이 액세스를 허용받을 수 있는 방이나 단지에 저장함으로써 법적으로 보호될 수 있다. 다른 대안 또는 부가적인 제한으로서, ePUF(501)의 경우, 타겟 당사자(103T)(및 실시예들에서, 별개의 셋-업 당사자로서 작용하는 신뢰할 수 있는 제3자)만이 C_w 를 알도록 1차 챌린지 C_w 에 대한 지식이 제한될 수 있다.

[0152] 단계(720)에서, 방법은 응답 데이터 저장소(601)에 응답 데이터를 저장하는 것을 포함한다. 실시예들에서, 저장된 응답 데이터는 생성된 CR 쌍들 $\{C_i, R_i\}$ 의 레코드를 포함한다. 각각의 CR 쌍의 레코드는 쌍의 대응하는 챌린지 C_i 를 표시하는 방식으로 저장된 개개의 응답 R_i 의 레코드를 포함한다. 실시예들에서, 각각의 응답 R_i 의 저장된 레코드는 응답의 명시적 값, 즉 레코드들을 판독할 수 있는 검증 당사자(103V)에게 명시적으로 공개된 R_i 의 실제 값을 포함한다. 값은 평문으로 저장될 수 있거나 검증 당사자가 값을 복호화할 수 있는 복호화 키를 갖는 경우 암호화될 수 있지만 그럼에도 불구하고, 저장된 값은 검증 당사자(103V)에게 명시적으로 공개된다는 의미에서 본원에서의 목적을 위해 여전히 명시적 값인 것으로 언급된다. 대안적으로, 응답의 레코드는 R_i 의 결정론적 변형을 포함하는 응답 R_i 의 "증명"을 포함할 수 있다. 예는 해시 $H(R_i)$ 또는 이중 해시 $H^2(R_i)$ 의 값을 저장하는 것일 것이다. 이는 R_i 에 적용된 동일한 변형(예컨대, $H(R_i)$ 또는 $H^2(R_i)$)이 증명과 매칭되는지를 체크함으로써 응답 R_i 의 값이 저장소에 레코딩된 것과 동일한지를 검증 당사자가 체크하는 것을 가능하게 한다. 이는 응답 R_i 의 실제 값이 공개되지 않는다는 이점이 있다. 따라서 방법의 이 변형은 저장소(601)가 블록체인과 같은 공개 매체인 경우에 특히 유용할 수 있다. 그러나 암호화는 다른 가능성이 있을 것이다.

[0153] 응답 데이터가 암호화된 형태로 저장되는 경우, 응답 데이터의 각각의 조각(예컨대, 각각의 CR 쌍)은 개별적으로 암호화될 수 있어, 각각이 복호화를 위해 상이한 개개의 복호화 키를 요구한다. 대안적으로, 응답 데이터의 서브세트를 또는 전체 세트들(예컨대, 주어진 타겟 당사자(103T)에 대한 모든 CR 쌍들)은 함께 암호화될 수 있어, 모두가 동일한 키를 가진 그룹으로서 함께 복호화 가능하다.

- [0154] 응답 데이터, 예컨대, CR 쌍들은 타겟의 아이덴티티의 증거와 관련하여 응답 데이터 저장소(601)에 저장된다. 예컨대, 타겟 당사자(103T)는 셋-업의 일부로서 여권과 같은 하나 이상의 조각들의 식별 정보들을 생성하도록 요구될 수 있다. 응답 데이터와 관련하여 응답 데이터 저장소(601)에 홀딩된 증거는 응답 데이터와 관련하여 명시적으로 저장되는 이 정보 그 자체의 사본을 (평문으로 또는 검증 당사자(103)에 대해 액세스 가능한 암호화된 형식으로) 포함할 수 있다. 대안적으로 응답 데이터 저장소(601)가 신뢰할 수 있는 제3자 또는 검증 당사자(103V) 자체에 의해 관리되는 경우, 응답 데이터가 특정 아이덴티티와 관련하여 응답 데이터 저장소(601)에 등록된다는 단순한 사실만으로도 충분한 증거로 간주될 수 있다(검증 당사자(103V)가 셋-업 시에 타겟 당사자의 식별 정보를 적절하게 체크한 셋-업 당사자 및 응답 데이터 저장소(601)를 관리하는 당사자 예컨대, 신뢰할 수 있는 제3자를 신뢰한다고 가정함).
- [0155] 검증 국면(704)에서, 단계(730)에서, 검증 당사자(103V)는 검증 동작에서 사용할 응답 데이터를 결정하기 위해 응답 데이터 저장소에 액세스한다. 실시예들에서, 복수의 잠재적 검증 당사자들(103V)이 존재하고, 각각에는 CR 쌍들 중 하나 이상의 상이한 개개의 서브세트를 할당된다. 즉, 응답 데이터 저장소(601)는 주어진 검증 당사자(103V)에게, 해당 당사자에게 할당된 CR 쌍(들)의 예상된 응답(들) Ri만을 공개할 것이다. 예컨대, 이 체계는 신뢰할 수 있는 제3자 시스템(602)에 의해 관리될 수 있다. 이러한 체계는 유리하게는 CR 쌍들을 분리한 채로 유지하여서, 하나의 검증 당사자(103V)는 타겟인 다른 검증 당사자인 척할 수 없게 한다. 그러나 저장소(601)에 대한 액세스가 주어진 모든 검증 당사자들(103V)이 신뢰할 수 있는 경우에, 이것이 필수적인 것은 아니다.
- [0156] 실시예들에서, 검증 당사자(103V)는 자신이 사용할 챌린지를 초기에 알지 못하며, 대응하는 응답 데이터(예컨대, 응답 또는 증명)와 함께 데이터 저장소(601)로부터 챌린지에 액세스함으로써 이를 결정한다. 대안적으로 검증 당사자(103V)는 자신이 사용하고자 하는 챌린지를 미리 알고 있고, 이를 사용하여 데이터 저장소(601)에서 어떤 응답 데이터가 이 챌린지에 매핑되는지를 룩업한다.
- [0157] 검증 당사자(103V)(또는 실제로 임의의 당사자)가 이를테면, 응답 데이터 및/또는 챌린지를 결정하기 위해 블록체인으로 부터의 데이터에 액세스하는 시나리오에서, 블록체인에 액세스하는 것은 직접적으로, 블록체인 네트워크의 노드에 질의하거나, 또는 간접적으로, 블록체인 데이터에 대한 액세스를 추구하는 당사자들 대신에 질의들을 중재하거나 블록체인 데이터를 캐시(cache)하는 중간 서비스에 질의함으로써 수행될 수 있다. 예컨대, 검증 당사자(103V)는 블록체인 네트워크(106)에 직접 연결되지 않은 다른 서비스 제공자로부터의 데이터에 액세스할 수 있지만, 응답 관련 데이터만을 제공하고 어쨌면, 머클(Merkle) 증명을 또한 제공할 수 있다.
- [0158] 단계(740)에서, 검증 당사자(103V)는 PUF 모듈(603)을 소유하거나 제어하는 타겟 당사자(103T)에게 챌린지(Ci)를 제출한다. 이는 단계(730)에서, 검증 당사자(103V)가 응답 데이터 저장소(601)로부터 액세스한 레코드들 중 하나에 대응하는 챌린지이다. 신뢰할 수 있는 제3자가 셋-업 시에 PUF 모듈(603)을 소유하고 있었던 시나리오들에서, PUF 모듈(603)은 셋-업 국면(702)과 검증 국면(704) 사이에서 신뢰할 수 있는 제3자로부터 타겟 당사자(103T)로 물리적으로 전달될 수 있다는 것에 주의한다.
- [0159] 제출된 챌린지 Ci에 대한 응답으로, PUF 모듈(603)은 타겟 당사자(103V)가 검증 당사자에게 리턴하는 대응하는 응답 Ri를 생성한다. 단계(750)에서, 검증 당사자는, 수신된 응답 Ri가 단계(730)에서 응답 데이터 저장소(601)로부터 액세스된 응답 데이터에 따라 예상되는 응답과 일치하는지를 체크한다.
- [0160] 언급된 바와 같이, 셋-업 단계들(702)을 수행하는 당사자는 타겟 당사자(103T) 또는 응답 데이터(예컨대, CR-쌍들)를 저장하는 신뢰할 수 있는 제3자일 수 있다. 추가 변형들에서, 이러한 단계들은 신뢰할 수 있는 오라클(실시예들에서, 데이터 저장소(610)를 포함하는 제3자 컴퓨터 장비(602)를 실행하는 당사자 이외의 다른 제3자)와 같은 다른 조정 당사자에 의해 수행될 수 있다. 그러한 실시예들에서, 데이터 저장소(601)는 (상이한 제3자의) 제3자 시스템(602) 또는 블록체인과 같은 공개 피어-투-피어 매체일 수 있다. 그리고/또는, 또 다른 변형들에서, PUF 모듈(603)에 대한 입력들을 수행하는 당사자와 출력들을 수신하는 당사자 사이에 분리(separation)가 제공될 수 있다.
- [0161] 또한 언급된 바와 같이, 응답 Ri가 응답 데이터 저장소(601)에 레코딩되는 방식에 대해 적어도 두 가지 가능성들이 있다. 첫 번째는 단순히 Ri 자체의 실제 값을 명시적으로 저장하는 것이다. 이 경우에, 단계(750)는 단순히, 저장된 값(이는 셋-업(702)에서 설정됨)을 (검증 국면(704)에서) 제출된 챌린지 Ci에 대한 응답으로 이제 수신된 값 R'i(응답 Ri의 알려진 값)와 비교하는 것을 포함한다. 이들이 매칭되는 경우, 방법은 타겟 당사자(103T)의 아이덴티티가 검증된 것으로 선언되는 단계(760)로 분기한다. 그렇지 않으면, 방법은 타겟 당사자

(103T)의 아이덴티티가 검증되지 않은 것으로 선언되는 단계(770)로 분기한다.

- [0162] 제2 가능성은 Ri의 증명만이 예컨대, 해시 또는 이중 해시가 응답 데이터 저장소(601)에 저장된다는 것이다. 이 경우에, 검증 당사자(103V)는 검증 국면(704)에서 자신이 타겟 당사자(103T)로부터 다시 수신한 응답 R'i에 대한 증명을 생성하는 데 사용된 것과 동일한 변환을 적용한다. 이것이 저장된 증명과 매칭하는 경우, 방법은 타겟 당사자(103T)의 아이덴티티가 검증된 것으로 선언되는 단계(760)로 분기한다. 그렇지 않으면, 방법은 타겟 당사자(103T)의 아이덴티티가 검증되지 않은 것으로 선언되는 단계(770)로 분기한다.
- [0163] 응답 데이터 저장소(601)에서, 대응하는 챌린지 Ci가 각각의 레코딩된 응답 Ri와 연관되는 것으로 표시되는 방식에 대해 적어도 두 가지 가능성들이 있다. 첫 번째는 단순히 각각의 CR 쌍 {Ci, Ri}의 명시적 값을 저장하는 것, 즉 Ri 및 Ci의 실제 값들을 (평문 또는 암호화되어) 저장하는 것이다. 대안적으로, 본원에서 개시된 실시예들에 따른 제2의 더 경량의 방식은 미리 결정된 결정론적 챌린지-도출 함수(f)에 따라 챌린지들(Ci)이 도출될 수 있는 마스터 챌린지(Cm)를 저장하는 것이다.
- [0164] 이는 도 8a에서 예시된다. 각각의 응답 Ri는 개개의 인덱스와 관련하여 저장된다. 함수 f는 응답 데이터 저장소(601)에 저장되거나 검증 당사자(103V)에 미리 알려진다. 어느 쪽이든, 검증 당사자(103V)는 응답들 Ri 중 적어도 하나의 응답의 인덱스 i에 대응하는 챌린지 Ci를 결정하기 위해 마스터 챌린지 Cm을 함수 f에 입력한다. 그 후, 검증 당사자(103V)는 타겟을 검증하기 위해 이 챌린지 Ci를 사용한다.
- [0165] 그러한 일부 실시예들에서, 함수 f는 또한 식별 정보(806)의 함수일 수 있으며, 이는 단일 조각의 식별 정보 또는 복수의 조각들의 식별 정보(802)(예컨대, 여권 정보, 어머니의 결혼 전 이름 및 지문 정보)의 조합(804)(예컨대, 컨캐터네이션)일 수 있다. 이는 타겟 당사자(103T)의 식별 정보를 포함할 수 있다. 이는 특정 타겟 당사자(103T) 특유의 챌린지들의 세트 Ci를 가능하게 하며, 이는 예컨대, 동일한 제3자 시스템(602)이 상이한 타겟 당사자들에 대한 챌린지 세트들을 생성하는 데 사용되는 경우 고유성이 중요할 수 있으므로, 보안상의 이유들로 유리하다. 타겟 당사자(103T)의 여권 정보 또는 어머니의 결혼 전 이름과 같은 개인 식별 정보를 사용하는 것은, 개인 식별 정보가 그/그녀가 이미 알고 있거나 갖고 있는 것이고 비공개로 유지되는 경향이 있기 때문에 좋은 선택이다.
- [0166] 대안적으로 또는 부가적으로, 식별 정보(806)는 검증 당사자(103V)의 식별 정보를 포함할 수 있어서, f는 특정 검증 당사자(103V)의 아이덴티티의 함수이다. 이는 특정 검증 당사자(103V)에 하나 이상의 특정 챌린지들의 특정 서브세트를 할당하는 데 사용될 수 있어서, 상이한 검증 당사자들(103V)에는 검증(704)에서 사용할 상이한 챌린지들 Ci이 주어진다.
- [0167] 일부 실시예들에서, 마스터 챌린지 Cm이 어떻게 형성되는지에 관계없이, 챌린지들 Ci는 도 8b에 도시된 바와 같이 $C1 = f(Cm)$, $C2 = f(C1)$, 이러한 식이 되도록 체인 방식으로 마스터 챌린지 Cm에 매핑될 수 있다. 즉, 제1 챌린지 C1는 마스터 챌린지 Cm에 함수 f를 적용함으로써 결정되고, 그 후 제2 챌린지 C2는 동일한 함수 f를 제1 챌린지에 적용함으로써 결정되고 이러한 식이다. 예로서, f는 해시 함수를 포함할 수 있다.
- [0168] 다른 변형에서, 챌린지 Ci는 도 8c에 도시된 바와 같이 계층적 방식으로 마스터 챌린지 Cm에 매핑될 수 있다. 이는 나중에 더 자세히 논의될 것이다.
- [0169] 체인 접근법은 더 경량이고 f()가 루트 키 이외의 임의의 데이터를 요구하지 않는 경우 루트 정보로부터 복구하기가 또한 더 쉽다. 계층적 도출의 경우에, 트리의 인덱스들이 추가될 것이며, 이는 다음과 같은 간단한 체인에 대해 필요로 되지 않을 것이다: C_m , $H(C_m)$, $H(H(C_m))$..., 예컨대, 여기서 f()는 해시 함수일 뿐이다.
- [0170] f()의 형태 또는 마스터 챌린지가 식별 정보 및/또는 다른 정보를 포함하는지에 관계없이, 실시예들에서, 마스터 챌린지 Cm는 셋-업(702) 동안 타겟 당사자(103T)로부터 제3자 시스템(602)에 의해 수신될 수 있다. 그 후, 제3자는 검증(704)에서 향후 사용을 위해 데이터 저장소(601)에 수신된 마스터 챌린지를 (예컨대, 로컬로 또는 체인 상에) 저장한다. 대안적으로, 제3자 시스템(602)은 타겟 당사자(103T)로부터 챌린지들의 세트 Ci를 수신하고, 예컨대, 함수 f()의 역을 적용함으로써 그로부터 마스터 챌린지 Cm을 도출한다. 이러한 접근법들의 변형들에서, 제3자 시스템(602)은 타겟 당사자(103T) 이외의 다른 곳으로부터, 예컨대, 오라클 또는 조정 당사자(미도시)로부터 식별 정보, 마스터 챌린지 또는 챌린지들의 세트를 수신할 수 있다. 이러한 접근법들의 조합이 또한 사용될 수 있다(예컨대, 하나의 조각의 식별 정보는 타겟 당사자로부터 수신되고 하나는 다른 곳으로부터 획득됨). 또는, 추가 대안들에서, 제3자는 관여되지 않고 타겟 당사자(103T)는 마스터 챌린지를 체인 상에(또는 일부 다른 피어-투-피어 공개 매체에) 스스로 저장한다.
- [0171] 도 7의 방법의 추가 변형들에서, 응답 데이터 저장소(601)에 저장된 응답 데이터는 셋-업 시에 생성된 CR

쌍(들)의 레코드를 포함하지 않을 수 있다. 대신에, 응답 데이터는 공개-개인 키 쌍의 공개 키 또는 이러한 공개 키들의 세트를 포함할 수 있으며, 여기서 하나 이상의 키 쌍들 각각은 셋업 국면(702)으로부터의 개개의 PUF 응답 Ri에 기초하여 생성되었다. 예컨대, 응답 Ri는 공개-개인 키-쌍 생성 알고리즘에서 시드로 사용될 수 있다. 이러한 실시예들에서, 방법은, 단계(730)에서 검증 당사자가 저장된 공개 키들 중 하나에 액세스하고 단계(740)에서 검증 당사자(103V)가 타겟의 PUF 모듈(603)에 입력될 챌린지 Ci를 제출하지 않는 것을 제외하고는, 도 7에 제시된 바와 같이 진행한다. 대신, 검증 당사자(103V)는 타겟에 의해 (알려지게) 서명된 메시지(예컨대, 문서, 파일 또는 블록체인 트랜잭션의 일부)를 획득한다. 이 메시지는 타겟 당사자(103T)에 의해 그/그녀에게 전송될 수 있거나 검증 당사자(103V)가 블록체인 또는 웹사이트와 같은 공개 매체로부터 자율적으로 메시지에 액세스할 수 있다. 어느 쪽이든, 단계(750)에서, 체크는 (당업계에 그 자체로 잘 알려진 공개-개인 키 서명 검증 기술들에 기초하여) 메시지에 적용된 서명을 검증하기 위해 저장소(601)로부터 액세스된 공개 키를 사용하는 것을 포함한다.

[0172] 다음은 이제 본원에서 개시된 실시예들에 따라 ePUF들 또는 PUF들에 대한 일부 예시적인 아이덴티티-설정 및 검증 프로토콜들을 보다 일반적으로 설명한다. 증명자 앨리스(타겟 당사자 103T) 및 검증자 밥(검증 당사자 103V)을 고려한다. PUF 아이덴티티 시스템에 적어도 3개의 상이한 챌린지 유형들이 존재한다. 예로서, 아래는 ePUF의 관점에서 설명될 것이지만, 보다 일반적으로 임의의 PUF 디바이스가 사용될 수 있다(PUF 모듈(603)을 포함하는 임의의 디바이스).

[0173] **1. 원격 PUF 챌린지** — 검증자는 밥에 의해 제출된 챌린지에 대한 앨리스로부터의 응답을 요청함으로써 원격으로 증명자에 챌린지한다. 이 모드는 검증자가 증명자의 PUF로부터 예상되는 응답(들)을 알고 있고, 또한 PUF가 합법적인 소유자에 의해 소유된다고 가정한다.

[0174] **2. 로컬 PUF 챌린지** — 검증자는 앨리스에 의해 제어되는 PUF 디바이스와 상호작용함으로써 로컬로 증명자에 챌린지한다. 이 모드는, 검증자가 증명자의 아이덴티티에 관한 어떤 것을 알고 있지만 그의 PUF의 거동에 대해서는 아무것도 모른다고 가정한다.

[0175] **3. 암호화 챌린지** — 검증자는 이를테면, 증명된 공개 키에 증명 가능하게 링크된 키로 메시지에 서명함으로써, 자신의 아이덴티티와 관련된 일부 암호화 요건들을 충족시키도록 증명자에 챌린지한다.

[0176] 유형들 1 및 2의 경우, 챌린지는 증명자와 검증자 둘 모두의 관점들로부터 명시적으로 PUF 모듈(603)에 의존한다. 이러한 경우에 챌린지 및 이에 따른 대응하는 검증 프로세스는 본질적으로 PUF 디바이스(PUF 모듈(603))을 포함하는 디바이스, 예컨대, 앨리스의 컴퓨터 장비(102T)의 동작에 링크된다. 이 경우들에, PUF 디바이스들의 물리적 상태가 아이덴티티에 고유하게 바인딩될 수 있는 PUF 디바이스들의 특성이 사용되고, 이에 따라 PUF는 활용중인 아이덴티티 시스템에서 중심 역할을 한다.

[0177] '원격' 및 '로컬'이라는 용어들은 구체적으로 챌린지를 할 때 검증자와 증명자의 PUF 간의 상호작용을 지칭하는 것에 주의한다. 이는 원격 챌린지 프로토콜이 미리 증명자와 검증자 간의 로컬 상호작용을 수반하는 셋업 국면을 갖는 것을 배제하지 않는다.

[0178] 그러나 경우 3에서, 챌린지 및 검증 프로세스는 증명자의 관점으로부터 PUF 디바이스와 관련되기만 하면 된다. 검증은 자신의 챌린지에 대한 응답을 생성하는 데 증명자에 의해 PUF가 사용되었는지 여부를 검증자가 아는 것에 의존하지 않는다. 이 경우에, 방법은 아이덴티티를 디바이스 자체에 링크하는데 있어 그의 유용성 보단, 단순히 앨리스에 대한 키-생성기로서 PUF의 유용성을 사용하는 것이다.

[0179] 다음에서, 위에서 언급된 3개의 동작 모드들 각각에서 아이덴티티 시스템들에 대한 셋업 및 검증, 및 선택적 업데이트 및 폐지 프로세스들에 대한 예시적인 구현들이 제공된다. 실시예들에서, 일반적으로 신뢰할 수 있는 제3자는 PUF 기반 아이덴티티 시스템과 관련된 프로세스들에 수반된다. 이는 그러한 아이덴티티 시스템들이 아이덴티티 및 관련 크리덴셜들에서 무결성 및 신뢰를 유의미하게 보장하기 위해 그러한 제3자를 요구하는 경향이 있기 때문이다. 이러한 시스템에서 개인의 아이덴티티가 설정되고 사용되는 경우, 해당 신뢰할 수 있는 제3자는 인증 기관, 정부 기관 또는 은행과 같은 금융 서비스 제공자일 수 있다.

[0180] 아이덴티티가 기계 또는 비-인간 엔티티에 대해 설정되어야 하는 경우, 제3자는 디바이스 제조업체, 발행자, 규제 기관 또는 일부 다른 관련 액터일 수 있다. 이 경우는 특히 사물 인터넷(IoT) 또는 추가로, 사물 블록체인(Bot) 패러다임에 적합하며, 여기서 아이덴티티는 일부 목표를 달성하기 위해 공동으로 작업들 또는 계산들을 수행할 수 있는 디바이스들의 네트워크의 상이한 멤버들에 할당될 것이다.

[0181] **3.1. 원격 PUF 시스템**

[0182] **3.1.1. 셋업:** 원격 PUF 챌린지들의 경우, 증명자에게 챌린지 C를 제출하는 검증자는 예상되는 응답 R을 미리 알고 있다고 가정한다. 이것은, 이 경우 셋업 프로세스가 앨리스와 다른 당사자 간에 CRP들의 세트(즉, 적어도 하나)를 설정해야 하며, CRP들의 세트는 나중에 앨리스의 아이덴티티를 인증하는 데 사용될 수 있는, 이들 간의 공유 비밀을 도출하는 데 사용될 수 있다는 것을 의미한다.

[0183] 이전에 언급된 바와 같이, 앨리스가 아이덴티티를 설정하도록 장비된 일반 제3자와 이 공유 비밀을 설정하고, 이 제3자가 나중에 앨리스와의 검증 프로세스에 참여하는 검증 당사자일 수도 있고 아닐 수도 있다고 가정한다. 검증 당사자가 아이덴티티-설정 제3자와 별개인 경우, 검증 당사자는 제3자로부터 공유 비밀(들)에 사용되는 관련 CRP 정보를 획득할 수 있다고 가정한다.

[0184] 여기에서 셋업 국면에 대해, 앨리스가 항상 PUF 디바이스에 대한 액세스를 갖는 유일한 당사자인지 또는 신뢰할 수 있는 제3자가 또한 셋업 국면 동안에만 PUF 디바이스에 대한 액세스를 가질 수 있는지에 따라 카테고리화되는 2개의 별개의 옵션들이 존재한다.

[0185] **경우 1: 앨리스는 PUF에 대한 단독 액세스를 가짐**

[0186] 1. ePUF 디바이스가 제조되어 앨리스에게 배포된다.

[0187] 2. 앨리스는 신뢰할 수 있는 제3자에 접촉함으로써 자신의 아이덴티티를 자신의 ePUF 디바이스에 링크하도록 신청한다.

[0188] i. 제3자는 앨리스에 대한 식별 계정을 설정하고 그녀의 아이덴티티의 증명을 요청한다.

[0189] ii. 앨리스는 제3자에게 관련 식별 문서들 또는 크리덴셜들을 제공한다.

[0190] iii. 제3자는 앨리스의 아이덴티티를 검증한다.

[0191] 3. 앨리스 및 제3자는 나머지 셋업 프로세스를 위해 (예컨대, 표준 Diffie-Hellman 키 교환을 통해) 보안 통신 채널을 설정한다.

[0192] i. 앨리스 및 제3자는 공개 키들 P_A , P_T 를 각각 교환한다.

[0193] ii. 앨리스 및 제3자는 $S = S_A \cdot P_T = P_A \cdot S_T$ 와 같이 나머지 셋업 통신들에 대해 임시 비밀을 독립적으로 설정한다.

[0194] iii. 앨리스 및 제3자는 S에 의해 보호되는 채널 예컨대, AES 암호화 채널을 통해 통신하기 시작한다.

[0195] 4. 제3자는 앨리스에게 보안 채널을 통해 챌린지들의 세트 $C_1, C_2, \dots, C_n$ 를 전송한다.

[0196] 5. 앨리스는 ePUF 디바이스로부터 응답들 $R_1, R_2, \dots, R_n$ 을 획득한다.

[0197] 6. 앨리스는 보안 채널을 통해 제3자에게 응답들 $R_1, R_2, \dots, R_n$ 을 전송한다.

[0198] 7. 제3자는 앨리스의 아이덴티티 계정에 대해 응답 CRP 세트 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 를 저장한다.

[0199] **경우 2: 제3자가 셋업 동안 PUF에 액세스함**

[0200] 1. 제3자는 기본 쌍 및 해시 함수에 대한 지식을 갖는다. 예컨대, ePUF 디바이스가 제조되고 신뢰할 수 있는 제3자*에게 배포된다.

[0201] 2. 제3자는 디바이스로부터 기본 CRP(C_w, R_w)를 획득한다.

[0202] 3. 앨리스는 제3자에 접촉함으로써 아이덴티티-링크 ePUF 디바이스를 신청한다. 이는 보안되지 않은 통신 채널을 통해 행해질 수 있다.

[0203] i. 제3자는 앨리스에 대한 식별 계정을 설정하고 그녀의 아이덴티티의 증명을 요청한다.

[0204] ii. 앨리스는 제3자에게 관련 식별 문서들 또는 크리덴셜들을 제공한다.

- [0205] iii. 제3자는 앨리스의 아이덴티티를 검증하고 ePUF 디바이스 및 그의 기본 쌍(C_w, R_w)을 앨리스의 계정에 할당한다. 공유 비밀은 이 CRP이거나 이 CRP의 도출물이다.
- [0206] 4. 제3자는 ePUF 디바이스를 앨리스에게 전송한다.
- [0207] (*디바이스는 앨리스에게 먼저 배포되고 그 후 앨리스에 전송될 수 있다. 그러나 대부분의 경우, 디바이스가 제3자에게 직접 배포하는 것이 더 합리적일 것이다. 예컨대, 디바이스가 스마트 직불 카드인 경우, PUF 셋업 후에, 카드는 제조업체로부터 발행 은행으로 그리고 그 후 발행 은행으로부터 고객인 앨리스에게 전달될 수 있다.)
- [0208] 셋업 프로토콜은 검증 프로세스 동안 나중에 앨리스의 아이덴티티(또는 PUF-포함 디바이스)를 인증하는 데 사용되도록 앨리스와 신뢰할 수 있는 제3자 간의 공유 비밀(들)을 설정한다. 또한, 경우들은 이들 둘 모두가 앨리스와 신뢰할 수 있는 제3자 간의 보안 통신을 수신하는 것이 바람직하다는 점에서 유사하다.
- [0209] 다만, 두 경우들 간의 차이는 경우 1은 보안 통신 채널을 설정함으로써 보안 통신을 달성하는 반면, 경우 2는 물리적 보안을 통해 이를 달성한다는 점이다.
- [0210] 경우 1 및 2의 두 프로토콜들 사이에 주목해야 할 다른 차이는 경우 2에서, 신뢰할 수 있는 제3자가 PUF 없이 앨리스만큼 많은 CRP들을 도출할 수 있는 반면, 경우 1에서, 이 당사자는 고정된 수의 쌍들을 저장해야 한다는 것이다.
- [0211] 이는 PUF 디바이스로 사용자를 셋업하기 위한 기존 프로토콜에 비해 경우 2의 이점인데, 그 이유는 경우 2는 신뢰할 수 있는 제3자가 임의적 수의 CRP들을 원격으로 생성할 수 있는 반면, 기존 프로토콜에서, 신뢰할 수 있는 제3자가 이를 행하기 위해서는 최종 사용자 또는 장치 제조업체와 협력할 필요가 있을 수 있기 때문이다. 앨리스가 보안 채널을 통해 밥에게 기본 쌍(C_w, R_w)을 전송하는 단계가 추가되는 경우(제3자가 악의적인 방식으로 기본 쌍을 사용하지 않는다고 신뢰함), 동일한 기술적 이점이 경우 1에서 달성될 수 있다.
- [0212] 셋-업 국면에서 보안 통신의 사용은 검증 프로세스와 같은 향후 통신들이 보안되지 않은 채널들을 통해 송신되도록 허용한다는 것에 주의한다. 이는 검증들이, 검증 시에 양 당사자들이 온라인 상태일 필요가 있는 것과 같은 기술적 제한들이 더 적게 발생하는 것을 허용한다는 이점을 갖고, 이 일회 셋업 프로세스에서 추가적인 보안 통신 오버헤드만을 요구한다.
- [0213] **3.1.2. 검증:** 원격 PUF 검증 모드에서, 셋업 국면에 2개의 별개의 경우들이 있었다는 점을 기억하자 — 이는 아래 상세된 바와 같이 약간 상이한 원격 검증 프로토콜들에 반영된다.
- [0214] **경우 1: 앨리스는 PUF에 대한 단독 액세스를 가짐**
- [0215] 1. 밥은 셋업 동안 앨리스 및 제3자에 의해 설정된 세트 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 로부터 (C_1, R_1) 과 같은 미사용 CRP를 획득한다.
- [0216] i. 밥이 또한 신뢰할 수 있는 제3자인 경우, 밥은 세트로부터 요소를 단순히 리트리브(retrieve)한다.
- [0217] ii. 밥이 신뢰할 수 있는 제3자가 아닌 경우, 밥은 앨리스에 대해 미사용 CRP를 요청함으로써 제3자와 통신한다.
- [0218] 2. 밥들은 챌린지 C_1 를 앨리스에게 전송한다.
- [0219] 3. 앨리스는 ePUF 디바이스로부터 후보 응답 R'_1 을 획득하고 이를 밥에게 전송한다.
- [0220] 4. 밥은 $R'_1 == R_1$ 인지를 검증한다.
- [0221] i. 만약 그렇다면, 검증이 통과된다.
- [0222] ii. 만약 아니라면, 검증은 실패한다.
- [0223] 5. 쌍 (C_1, R_1) 은 후속적으로 신뢰할 수 있는 제3자에 의해 제거되어, 나머지 챌린지-응답 쌍들의 세트 $\{(C_2, R_2), (C_3, R_3), \dots, (C_n, R_n)\}$ 를 남긴다.

- [0224] 단계 1.ii에서, CRP들의 단일 사용 성질은 임의적 밥이 특정 CRP를 사용하여 앨리스를 '가장'하는 것이 가능하지 않다는 것을 보장하는데, 그 이유는 신뢰할 수 있는 제3자는 각각의 주어진 상황에서 각각의 쌍의 사용을 간단히 모니터링할 수 있고 모든 각각의 인증 시도에 대해 새로운 CRP를 사용해야 하기 때문이라는 것에 주의한다.
- [0225] **경우 2: 제3자가 셋업 동안 PUF에 액세스함**
- [0226] 1. 밥은 검증을 위해 새로운 챌린지 C를 생성한다. 이는 랜덤으로, 또는 일부 다른 데이터(예컨대, 알려진 KYC 데이터, 바이오메트릭들, 이미지들)로부터 결정론적으로 수행될 수 있다.
- [0227] 2. 밥들은 챌린지 C를 앨리스에게 전송한다.
- [0228] 3. 앨리스는 ePUF 디바이스로부터 후보 응답 R'을 획득하고 이를 밥에게 전송한다.
- [0229] 4. 밥은 예상된 응답 R을 획득한다.
- [0230] i. 밥이 신뢰할 수 있는 제3자인 경우, 밥은 $R = \text{hash}(C, R_w, H)$ *를 컴퓨팅함으로써 직접 응답을 계산할 수 있다.
- [0231] ii. 밥이 신뢰할 수 있는 제3자가 아닌 경우, 밥은 C를 제3자에게 전송하고 응답 R을 요청한다.
- [0232] 5. 밥은 $R' = R$ 인지를 검증한다.
- [0233] i. 만약 그렇다면, 검증이 통과된다.
- [0234] ii. 만약 아니라면, 검증은 실패한다.
- [0235] (*이는 제3자가 셋업 프로토콜(경우 2) 동안 기본 쌍(C_w, R_w)을 획득 — 이는 R_w 가 그들에게 알려져 있음을 암시함 — 했기 때문이다. 또한, 해시 함수 H는 모든 사람이 아니더라도 적어도 제3자에게 알려져 있다고 즉, SHA-256과 같은 공개 표준이라고 가정함.)
- [0236] **3.1.3. 업데이트:** 검증(및 로그인과 같은 다른 유용한 프로토콜)에서 새로운 CRP들의 단일 사용 성질을 고려하여 앨리스 및 제3자가 새로운 CRP들을 설정하기 위한 프로세스를 지정하는 것이 또한 바람직할 수 있다.
- [0237] **경우 1: 앨리스는 PUF에 대한 단독 액세스를 가짐.** 이 경우, 셋업에서와 같이, 앨리스와 제3자 간에 챌린지들 및 응답들을 송신하기 위해 다른 보안 채널이 설정된다. 우리는 앨리스가 형태 $S = H(R_i)$ 또는 이와 유사한 형태의 공유 비밀을 설정하기 위해 형태 (C_i, R_i) 의 적어도 하나의 남은 CRP를 갖거나 DH 키 교환으로부터 이전 공유 비밀 $S = S_A \cdot P_T = P_A \cdot S_T$ 에 대한 액세스를 갖는다고 가정한다.
- [0238] 1. 앨리스 및 제3자는 공유 비밀 S를 사용하여 보안 통신 채널을 설정한다. 이것은 다수의 방식으로 도출될 수 있으며, 프로토콜은 이에 대해 불가지론적이다.
- [0239] 2. 제3자는 앨리스에게 보안 채널을 통해 챌린지들의 세트 $C_1, C_2, \dots, C_n$ 를 전송한다.
- [0240] 3. 앨리스는 ePUF 디바이스로부터 응답들 $R_1, R_2, \dots, R_n$ 을 획득한다.
- [0241] 4. 앨리스는 보안 채널을 통해 제3자에게 응답들 $R_1, R_2, \dots, R_n$ 을 전송한다.
- [0242] 5. 제3자는 앨리스의 아이덴티티 계정에 대해 응답 CRP 세트 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 를 저장한다.
- [0243] 적어도 단계들 2-5는 셋-업 단계들 4-7과 동일하다는 것에 주의한다.
- [0244] 채널을 통해 제3자(C_w, R_w)에게 알리는 앨리스에 대한 이전 코멘트를 또한 참조한다.
- [0245] **경우 2: 제3자가 셋업 동안 PUF에 액세스함.** 이 경우, 제3자는 간접적으로 임의적 수의 CRP들을 생성할 수 있는데, 그 이유는 이들이 기본 쌍(C_w, R_w) 및 해시 함수 H() 둘 모두에 대한 지식을 갖기 때문이다. 이는 이 경우 상호작용식 업데이트에 대한 어떠한 요건도 없음을 의미한다.
- [0246] **3.1.4. 페지:** 아이덴티티 시스템의 추가 부분은 특정 ePUF 디바이스가 페지되어서 더 이상 아이덴티티 목적들로

사용되지 않는 것일 수 있다. 페지 프로세스는 간단하고 (i) 사용자인 앨리스와 독립적으로 제3자에 의한 페지, 또는 (ii) 페지 요청으로서 전달된 앨리스에 의한 페지로서 수행될 수 있다.

[0247] 제1 경우는 ePUF들 또는 다른 것을 수반하는 어떠한 기술적 수단도 요구하지 않는다. 제2 경우는, ePUF 특유의 프로토콜 또는 솔루션을 요구하지 않는데, 그 이유는, 제1 경우에서 페지에 대한 양호한 예는 앨리스가 ePUF를 포함하는 물리적 디바이스를 분실한 경우 또는 ePUF가 어떤 식으로든 손상된 경우이기 때문이다.

[0248] 그러나 페지 프로세스에서 ePUF를 선택적으로 활용하려는 경우 — 여기서 앨리스는 여전히 디바이스의 물리적 제어를 가짐 —, 앨리스의 요청은 이를테면, 각각의 경우 키로서 CRP 응답 또는 비밀을 사용하는 HMAC 또는 암호화된 메시지에 의해 그녀 및 제3자가 설정한 CRP들 중 하나(또는 그의 도출된 공유 비밀)를 사용하여 인증되도록 규정될 수 있다. 그러나 위에서 언급된 이유로, 이것은 어떻게든지 시스템의 엄격한 요건으로 간주되지 않는다.

[0249] 3.2. 로컬 PUF 시스템

[0250] 3.2.1. 셋업: 로컬 PUF에 대해 사용될 수 있는 셋업은 원격 PUF에 대한 셋업과 정확히 동일하지만, 로컬 및 원격 경우들 간의 차이는 아래의 검증 단계가 수행되는 방법이다.

[0251] 3.2.2. 검증: 이 시나리오에서, 검증이 로컬로 수행된다. 이는 검증 프로세스가 증명자(앨리스) 및 검증자(밥) 둘 모두가 동일한 물리적 위치에 있을 것을 요구한다는 것을 의미한다.

[0252] 이 시나리오는 예컨대, 앨리스가 자신의 ePUF 디바이스를 사용하여 로컬로 조사와 상호작용하도록 법적으로 요구되는 (인간 아이덴티티에 대한) 법원 절차, 또는 시스템의 관리자가 로컬로 특정 디바이스의 응답을 명시적으로 체크하고자 할 수 있는 IoT 시스템의 분석이 수행되는 (디바이스 아이덴티티에 대한) 법원 절차와 관련이 있을 수 있다. 이것은 또한 지불 시나리오들과 관련이 있을 수 있다.

[0253] 이러한 프로세스가 적용 가능한 다른 시나리오들은 사고 후 차량에 대한 진단들을 포함할 수 있으며, 여기서 관계자들은 정확히 어떤 디지털 구성요소가 명령을 발행했는지 결정하고자 한다. 이 경우에, 입력 C는 일부 환경 또는 동적 조건들일 수 있으며, 응답 R은 디바이스에 의해 주어진 명령의 일부일 것이다.

[0254] 아래에 약속된 로컬 PUF 검증 프로토콜과 이전 원격 PUF 검증 프로토콜의 차이는 검증자가 ePUF의 응답을 미리 알고 있음을 이 로컬 프로토콜이 가정하지 않는다는 것이다. 즉, 로컬 검증 프로세스 동안 생성된 응답은 검증자가 미리 사용 가능하지 않다.

[0255] 그러나 이 시나리오에서, 검증 프로세스에 사용된 챌린지가 어떤 방식으로든 유의미할 가능성이 높다. 예컨대, 아이덴티티가 그의 매립된 ePUF 구성요소의 기본 쌍(C_w, R_w)으로 간주될 수 있는 기계를 고려한다. 검증 프로세스는, 주어진 입력 C로부터 출력 R을 이전에 산출한 것이 이 특정 디바이스였다는 것을 검증하기 위해 수행될 수 있다.

[0256] 1. 밥은 해당 CRP(C,R)에 기초하여 ePUF 디바이스에 제출할 적절한 챌린지 C를 획득한다.

[0257] 2. 밥은 ePUF 디바이스에 대한 액세스를 획득한다.

[0258] 3. 밥은 ePUF 디바이스를 사용하여 후보 응답 $R' = \text{hash}(C, R_w, H)$ 을 생성한다.

[0259] 4. 밥은 $R' == R$ 인지를 검증한다.

[0260] i. 만약 그렇다면, 검증이 통과된다.

[0261] ii. 만약 아니라면, 검증은 실패한다.

[0262] 이러한 시나리오들에서, 밥은 후보 응답 R'을 미리 아는 것이 아니라 오히려, PUF 디바이스로부터 그가 지금 수신한 응답이 이전에 생성된 응답과 매칭된다고 검증하고 있다. 예컨대, 이것은 사람(앨리스) 또는 응답을 생성한 디바이스가 (예컨대, 법원에) 지금 존재하는 동일한 사람 또는 디바이스임을 검증하는 데 사용될 수 있다. 예컨대, 디지털 구성요소의 예에서, 이것은 일부 입력 챌린지 C에 기초한 R의 생성 시 명령을 발행하도록 구성되었을 것이다. 예컨대, 디바이스가 자율 주행 자동차이고 구성요소가 "앞차가 너무 가깝습니다"라는 데이터로부터 도출되거나 이를 포함하는 챌린지를 수신하는 경우, 응답 R이 생성되고, R은 구성요소를 트리거하여 브레이크들을 밟으라는 명령들을 발행한다. 따라서 후향적 진단 검증(retrospective diagnostic verification)에서, 검증자는 자동차가 느려졌다고 믿고, 해당 응답을 트리거하기 위해 조건들이 실제로 "앞차

가 너무 가까웠다"는 것을 검증하고자 한다.

3.2.3. 업데이트: 업데이트된 CRP들을 생성하기 위한 프로세스는, 해당 시나리오의 핵심적인 차이가 검증에만 적용되므로, 원격 경우에 대해 제시된 것과 동일한 로직을 따를 수 있다.

3.2.4 폐지: 원격 폐지에 대해 설명된 동일 기술들이 또한 여기에서 유효하다.

3.3. 암호화 PUF 시스템

3.3.1 셋업: 이 경우에, 앨리스는 표준 암호화 수단을 사용하지만, 프로세스에서 ePUF 디바이스를 사용하여 제3자와 아이덴티티를 설정한다.

이 시나리오에서, 제3자는 선택적으로 ePUF가 프로세스에서 사용되었다는 지식을 가질 수 있다. 마찬가지로, 이러한 방식으로 설정된 아이덴티티의 경우, 아이덴티티의 검증자는 ePUF 디바이스가 아이덴티티 검증 프로세스에 관여됨을 알 수도 있고 모를 수도 있다. 요컨대, 다음 프로토콜은 디바이스의 소유자인 앨리스가, ePUF 디바이스가 아이덴티티 시스템에 관여된다는 지식을 가지고 있다는 것만 규정한다.

1. ePUF 디바이스가 제조되어 앨리스에게 배포된다.

2. 앨리스는 신뢰할 수 있는 제3자에게 접촉함으로써 암호화 아이덴티티를 설정하도록 신청한다.

i. 제3자는 앨리스에 대한 식별 계정을 설정하고 그녀의 아이덴티티의 증명을 요청한다.

ii. 앨리스는 제3자에게 관련 식별 문서들 또는 크리덴셜들을 제공한다.

iii. 제3자는 앨리스의 아이덴티티를 검증한다.

3. 앨리스는 자신의 아이덴티티에 대한 암호화 링크를 설정하기 위한 예컨대, 자신의 CRP를 사용하여 보증된 비대칭 키 쌍을 설정하기 위한 암호화 방법을 선택한다.

i. 제3자는 앨리스로부터 공개 키 P_A 를 획득하며, 여기서 $P_A = s_A \cdot G$ 는 EC 키 쌍이다.

ii. 제3자는 앨리스가 개인 키 s_A 를 사용하여 메시지 m 에 (예컨대, ECDSA를 통해) 서명하도록 요청한다.

iii. 앨리스는 ECDSA 서명 $Sig(P_A, m)$ 을 생성하여 제3자에게 전송한다.

iv. 제3자가 서명을 검증한다.

4. 서명이 유효한 경우, 제3자는 앨리스의 아이덴티티에 대해 키 P_A 를 보증한다.

단계 3은 사용자가 선택한 암호화 체계를 사용하는 것을 수반하지만, 프로세스에 수반된 관련 키는 앨리스에게만 알려진 CRP 응답의 도출물일 것이라고 가정한다. 위에서 선택된 예에서, 이는 개인 키 s_A 가 $s_A = H(R_2)$ 와 같이 특정 ePUF 응답 R 로부터 도출될 것임을 의미한다.

3.3.2 검증: 암호화 경우에서, 이전에 상세된 암호화 셋업 국면 동안 설정된 암호화 정보를 사용하여 아이덴티티 검증이 수행된다. 이 경우에, 보증된 EC 비대칭 키 쌍이 셋업 동안 앨리스의 아이덴티티에 대해 설정되었고 우리는 검증을 위해 해당 키를 이제 사용하는 예를 든다.

그러나 아래의 프로토콜은, 적절한 경우 임의의 다른 체계들에 대한 기존 셋업 및 검증 프로토콜들을 간단히 대체함으로써 그러한 암호화 체계에 대해 간단히 적응될 수 있다. 여기서 차이는 홀더 앨리스에 대한 악의적인 손상 위험을 감소시키는 셋업 및 검증 프로세스를 위한 보안 키 생성기로서 ePUF 디바이스를 사용하는 것이다.

1. 밥은 아이덴티티-링크 정보 P_A 예컨대, 보증된 키를 획득한다.

i. 밥이 신뢰할 수 있는 제3자인 경우, 그는 단순히 앨리스의 계정으로 부터 P_A 를 리트리브한다.

ii. 밥이 신뢰할 수 있는 제3자가 아닌 경우, 그는 제3자와 통신하고 앨리스에 대한 보증된 공개 키를 요청한다.

2. 밥은 앨리스가 서명할 메시지를 선택하고, 앨리스에게 전송한다.

- [0286] 3. 앨리스는 메시지 m 에 대해 서명을 생성한다.
- [0287] i. 앨리스가 자신의 보증된 키로 서명하고자 하는 경우, 그녀는 서명 $Sig(P_A, m)$ 을 생성한다.
- [0288] ii. 앨리스가 일회-사용 도출 키로 서명하고자 하는 경우, 그녀는 서명 $Sig(P_\alpha, m)$ 를 생성하며, 여기서 $P_\alpha = P_A + H(d) \cdot G$ 및 d 는 일부 일회-사용 데이터*이다.
- [0289] 4. 앨리스는 서명을 밥에게 전송한다. 이 지점에서, 앨리스는 또한, 밥이 아직 데이터 d 를 알지 못하는 경우 이를 전송할 수 있다.
- [0290] 5. 밥은 P_A (및 해당되는 경우, d)를 사용하여 공개 키에 대해 서명을 검증한다.
- [0291] i. 서명 검증이 통과되는 경우, 아이덴티티 검증이 통과된다.
- [0292] ii. 서명 검증이 실패하는 경우, 아이덴티티 검증이 실패한다.
- [0293] (* 이 데이터는 인보이스 메시지, 바이오메트릭 퍼지-매칭 데이터와 같이 검증과 관련될 수 있다. 데이터 d 는 밥 또는 앨리스에 의해 선택될 수 있다. 대안적으로, d 는 예컨대, Diffie-Hellman 키 교환 및/또는 HMAC를 사용하여 도출된, 앨리스 및 밥에게 알려진 공유 비밀일 수 있다.)
- [0294] 위의 암호화 검증 프로세스는 아이덴티티가 EC 또는 PGP 키와 같은 유사한 암호화 프리미티브(cryptographic primitive)로 설정된 경우, 이전 섹션에서 설명된 바와 같이 독립적으로 설정된 아이덴티티에 또한 적용될 수 있다.
- [0295] **3.3.3. 업데이트:** 여기에서 앨리스의 아이덴티티를 업데이트하는 프로세스는 키 생성에서 ePUF 디바이스의 사용에 의존하지 않고, 이에 따라 여기에서 임의의 특정 방법을 규정할 필요가 없다. 대신, P_A 와 같은 보증된 키를 업데이트하는 표준 방법들이 사용될 수 있다.
- [0296] ePUF가 기존 프로세스(들)에 의해 요구되는 임의의 요구되는 서명 또는 다른 암호화 프로세스를 위한 키 생성에 수반될 것임이 간단히 가정될 수 있다.
- [0297] **3.3.4. 폐지:** 유사하게, 여기에서 특정 폐지 프로토콜을 규정할 필요는 없지만, 표준 메커니즘에 따른다. 다시 한 번, ePUF가 적절한 암호화 동작들을 위한 키 생성기로서 백그라운드에서 수반될 것임이 가정될 수 있다.
- [0298] **3.4. 독립적 PUF 메커니즘**
- [0299] **3.4.1 셋업:** ePUF 디바이스를 사용하여 아이덴티티를 설정하기 위한 독립적인 경우에서, 엔티티가 임의의 제3자와 독립적으로 인간 아이덴티티를 설정하거나 폐쇄된 시스템 내에서 디바이스 아이덴티티를 설정하고자 하는 시나리오를 고려한다. 이 프로세스에 수반된 유일한 당사자는 ePUF 디바이스의 '소유자'이자 이후 검증들 동안에서의 최종 증명자인 앨리스이다.
- [0300] **경우 1: 앨리스는 인간 아이덴티티를 설정함**
- [0301] 1. 앨리스는 ePUF 디바이스를 획득한다.
- [0302] 2. 앨리스는 챌린지 C 로 ePUF를 프로빙한다.
- [0303] 3. 앨리스는 ePUF로부터 응답 R 을 획득한다.
- [0304] 4. 앨리스는 쌍 (C, R) 을 사용하여 자신에 대한 아이덴티티를 설정한다.
- [0305] i. 앨리스는 보증되지 않은 아이덴티티 키 P_A 를 설정하기 위해 암호화 셋업을 사용할 수 있다.
- [0306] ii. 앨리스는 자신의 아이덴티티에 대해 자신의 아이덴티티 키를 공개한다.
- [0307] 5. 앨리스는 응답의 이중 해시 $H^2(R)$ 와 같은 증명을 자신의 CRP에 공개하고자 할 수 있다.
- [0308] 앨리스가 그녀 자신에 대한 '자주적(self-sovereign)' 아이덴티티를 설정하는 이 경우는 그녀만이 제어하는 디바이스에 대해 고유하고 재현 가능한 디바이스 식별자를 제공하는 데 일정 정도 유용하다. 그러나 이러한 아이덴티티 시스템에서 신뢰할 수 있는 제3자의 기여는 검증자가 나중에 증명자의 아이덴티티와 증명자의 디바이스

사이의 링크를 신뢰해야 함을 의미한다. 이는 실세계에서 매우 제한된 애플리케이션들을 가질 수 있다.

[0309] **경우 2: 엘리스는 디바이스에 대한 아이덴티티를 설정함**

- [0310] 1. 엘리스는 ePUF 디바이스를 획득한다.
- [0311] 2. 엘리스는 챌린지 C로 ePUF를 프로빙한다.
- [0312] 3. 엘리스는 ePUF로부터 응답 R을 획득한다.
- [0313] 4. 엘리스는 (C,R) 쌍을 사용하여 그녀의 시스템 내 디바이스에 대한 아이덴티티를 설정한다.
- [0314] i. 엘리스는 쌍 (C,R)을 자신의 디바이스에 매핑한다.
- [0315] ii. 엘리스는 자신의 모든 디바이스들 및 CRP 매핑들의 데이터베이스를 유지한다.
- [0316] 5. 엘리스는 응답의 이중 해시 $H^2(R)$ 와 같은 증명을 자신의 CRP에 공개하고자 할 수 있다.

[0317] 디바이스에 대한 '자주적' 아이덴티티를 생성하는 위의 경우에서, 관리자가 단순히 폐쇄된 시스템 내에서 상이한 디바이스들을 간단히 식별하려고 하는 폐쇄된 시스템 내에서 설계가 매우 유용할 수 있음을 알 수 있다. 이는 또한 나중에 다른 사람들에게 증명하는 데 유용할 수 있다. 그러나 셋업 동안 신뢰할 수 있는 제3자의 결여는 시나리오에 의존하여, 디바이스들이 변경되지 않았음을 외부 검증자에게 확신시키는 데 있어 증명자를 여전히 제한한다.

[0318] 경우 1 및 경우 2는 동일한 프로세스로 간주될 수 있지만 의도된 목적이 상이할 수 있다는 것에 주의한다. 따라서 경우 1 및 경우 2는, 인간 또는 기계에 대한 '자주적' 아이덴티티를 생성하기 위한 방법으로 함께 간주될 수 있으며, 여기서 후자의 경우, 시스템 관리자(이를테면, IoT 시스템에서, 엘리스)는 그녀 자신이 신뢰할 수 있는 엔티티이다. 둘 모두의 경우들에서, 엘리스가 신뢰할 수 있는 엔티티이다.

[0319] **3.4.2 검증:** 이 경우에 대한 검증 프로세스는 주어진 챌린지로 ePUF 디바이스를 프로빙하고 그의 응답을 검사하는 것만큼 간단하다. 외부 당사자들에 대한 더 복잡한 증명들 또는 증거가 이들에 대해 아이덴티티를 증명하기 위해 그 위에 구축될 필요가 있을 수 있다.

[0320] **3.4.3 업데이트:** 이 경우에 대한 업데이트 프로세스는 관리자(이 경우 엘리스)가 향후 사용을 위해 추가적인 CRP들을 열거하는 셋업 프로세스의 단순한 반복이다.

[0321] **3.4.4. 폐지:** 이 시나리오에서, 아이덴티티 폐지의 유일한 유형은, 이 프로세스에 수반되는 제3자가 없기 때문에 관리자(엘리스)가 독립적으로 아이덴티티를 폐지하고자 하는 경우이다. 이것은, 폐지가 엘리스가 ePUF 디바이스의 사용의 중단 및 그녀의 CRP들의 데이터베이스의 퍼징(purging) 만큼 간단할 수 있다.

[0322] 이후 섹션에서, 블록체인 증명 및 증거에 의해 이 자주적 폐지가 보다 강력하게 만들어져서, 나중에 외부 당사자를 확신시킬 수 있는 방식들이 개시된다.

[0323] **3.5. 아이덴티티-기반 CRP 관리**

[0324] 위의, 특히 원격 PUF 기반 아이덴티티 시스템에서, 셋업 및 검증 프로토콜들에서 아이덴티티를 인증하는 데 사용되는 CRP들의 일회-사용 성질은 수반된 당사자들에게 CRP 관리 난제를 제시한다.

[0325] 예컨대, 신뢰할 수 있는 제3자가 셋업 동안 PUF 디바이스에 액세스하지 않는 경우, 향후 검증들을 위해 제3자가 저장하도록 다수의 CRP들 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 이 열거되는 것이 바람직할 수 있다. 또한, ePUF 자체가 챌린지들 대 응답들의 결정론적 의사-랜덤 매핑으로서 작용하기 때문에, 응답들은 서로 관련이 없는 것처럼 보일 것이다. 따라서 신뢰할 수 있는 제3자가 그의 사용자들 또는 클라이언트들을 위해 CRP들의 세트들을 표로 작성하고 저장해야 하는 부담은 매우 다수의 사용자들에게 서비스를 제공해야 하는 경우 스케일링 이슈를 빠르게 제시할 것이다.

[0326] 도 8a는 본원에서 개시된 실시예들에 따라 식별 데이터로부터 챌린지들의 결정론적 도출을 예시한다.

[0327] 그러한 실시예들에 따르면, 신뢰할 수 있는 제3자에 대한 부담 이슈들을 해결하기 위해, CRP 관리는 주로 챌린지들 $C_1, C_2, \dots, C_n$ 의 생성에서 다뤄진다. 여기에서 아이디어는 단일 마스터 챌린지, 또는 마스터 챌린지가 도출되는 마스터 데이터로부터 챌린지들이 결정론적으로(및 어쩌면, 또한 계층적으로) 도출되어야 한다는 것이다.

이 개념은 신뢰할 수 있는 제3자(또는 다른 관련 당사자)가, 비트코인 시나리오에서 '지갑 시드'로서 지칭되는 마스터 데이터만 사용하여 모든 관련 챌린지들을 복구하도록 허용하게 설계되었다는 점에서 일회-사용 비트코인 키들을 관리하기 위해 계층적 결정론적(HD) 지갑들을 사용하는 것과 유사하다.

[0328] 이러한 일부 실시예들에서, 엘리스(타겟 당사자(103T))의 식별 데이터(806)는 이전 섹션들에서 제안된 것들과 같은 아이덴티티 시스템들에서 어떤 CRP들이 사용되는지를 결정하기 위해 매우 다양한 챌린지들을 생성하기 위한 마스터 데이터로서 사용된다. 식별 데이터 자체는 상이한 데이터 요소들(802)의 조합(804)을 포함할 수 있지만, 조합에서 그들은 바람직하게는 다음의 특성들을 갖는다:

[0329] · **고유성** — 식별 데이터는 그것이 관련된 엔티티에 대해 고유함; 및

[0330] · **비밀성** — 식별 데이터는 그것이 관련된 엔티티(또는 그의 소유자)에게만 알려짐.

[0331] 식별 데이터의 성분들의 간단한 예들은 여권 번호, 국민 보험 번호, 이름, 생년월일 또는 보안 질문에 대한 답변들(예컨대, 어머니의 결혼 전 성) 또는 디바이스 식별의 경우 일련 번호들 및 제조 정보를 포함할 수 있다. 그러나 고유성을 보존하기 위해 퍼지-매직 기술(fuzzy-magic technique)들을 사용하여 추출될 수 있는 지문 또는 얼굴 인식 데이터와 같이 보다 진보된 기술 수단에 의해 획득된 데이터가 또한 사용될 수 있다는 것이 인식된다.

[0332] 실시예들에서, 챌린지들의 세트가 도출되는 마스터 입력으로서 사용되는 '식별 데이터'는 위의 다중성을 포함할 수 있다. 이에 대한 하나의 이유는, 이전 섹션들에서 프로토콜들 중 일부가 제3자 및/또는 외부 검증 당사자와의 챌린지들의 공유에 의존한다는 것을 감안하면, 가능한 한 많은 신뢰할 수 있는 제3자들과 관련하여 정보가 비밀성을 유지하도록 보장하기 위함이다. 다수의 구성요소들을 포함하는 식별 데이터는 증명 당사자인 엘리스의 동의 없이, 임의의 제3자가 완전히 복제하는 것이 더 어려울 것이다.

[0333] 식별 데이터를 사용하여 결정론적으로 CRP들을 생성하기 위한 메커니즘이 도 8a에서 도시된다. 식별 데이터의 성분 부분들은 컨캐터네이션, 비트별 연산들(예컨대, XOR) 또는 임의의 다른 관련 조합 동작일 수 있는 프로세스 'A'(804)에 의해 먼저 결합되며, 이 동작은 원시 데이터를 난독화된 형식으로 변환함으로써 프라이버시를 보존하고자 할 수 있다는 것에 주의한다.

[0334] 그 후, 식별 데이터는 해시 함수 또는 유사한 프로세스를 통해 마스터 챌린지 C_m 으로 전환된다. 마지막으로, 마스터 챌린지는 도출 함수 $f()$ 를 사용하여 일회-사용 챌린지들의 시퀀스 $C_1, C_2, \dots, C_n$ 결정론적으로 도출하는 데 사용된다. 실시예들에서, 도 8b에 도시된 바와 같이, 도출 함수 $f()$ 는 해시 함수 및 논스의 주입을 포함할 수 있어서, 각각의 연속적인 챌린지가 $C_i = SHA256(C_{i-1}, i)$ 로서 생성되며, 여기서 i 는 논스로서 역할을 한다.

[0335] 프로세스 A, 식별 데이터로부터의 챌린지 C_m 의 생성 및 도출 함수 $f()$ 는 모두 특정 구현의 필요에 의존하여 구성될 수 있다.

[0336] 도 8c는 다른 특정 예, 즉 챌린지들의 계층적이고 결정론적인 도출을 도시한다(응답들은 도시되지 않음). 도 8b에 도시된 바와 같이 계층적 방식으로 마스터 C_m 으로부터 단일-사용 챌린지 C_i 를 도출하는 것이 바람직할 수 있다. 이 경우에, 이전 경우에서와 같이 특정 챌린지의 생성이 이전 챌린지들 전부에 의존할 필요가 없다는 사실에 의해 CRP 관리가 추가로 개선된다.

[0337] 아이덴티티 데이터에 기초하여 챌린지들의 결정론적 도출의 사용은 아이덴티티 프로토콜들에서 증명자 엘리스 및 신뢰할 수 있는 제3자 둘 모두에 대한 저장 오버헤드를 감소시킨다. 어느 당사자든 식별 데이터(또는 그의 서브세트)만을 저장하고, 요구되는 경우 및 요구될 때, 필요한 챌린지를 재컴퓨팅하는 것이 가능하다.

[0338] 또한, 엘리스는 트레이드 오프와 함께, 원하는 만큼 각각의 식별 서비스와 많은 정보를 공유하거나 보유하도록 선택함으로써 자신의 프라이버시를 맞춤화하는 옵션을 또한 갖고, 이는 그녀가 더 많은 데이터를 직접 저장할 수 있다는 것과 절충된다.

[0339] 4. 예시적인 블록체인 시스템

[0340] 다음은 본 개시내용의 특정 실시예들에서 사용될 수 있는 예시적인 블록체인 시스템을 설명한다. "엘리스" 및 "밥"은 두 당사자들에 대한 임의적 이름들일 뿐이며, 엘리스 및 밥은 선행 섹션이나 다음 섹션들에서와 같이 이

섹션에서 반드시 동일한 역할을 가질 필요는 없다.

[0341] 일부 실시예들에서, PUF의 출력에 기초한 응답 데이터는 예컨대, 선행 섹션에서 논의된 바와 같이 체인 상에 저장될 수 있다. 체인 상에 저장된 응답 데이터는 실제 응답 자체, 또는 실제 응답의 변환 이블테면, 해시 또는 이중 해시(소위 증명 또는 해시-커밋) 또는 PUF 응답으로부터 도출된 공개-개인 키 쌍의 공개 키의 형태를 취할 수 있다. 온-체인 응답 데이터가 어떤 형태를 취하든지, 다른 검증 당사자가 아이덴티티의 증거로서 제시된 타겟 응답 또는 서명이 예상대로인지 체크하는 것을 가능하게 하는 것이다. 추가 실시예들에서, 블록체인은 챌린지-응답 쌍들을 업데이트 또는 폐지하는 것과 같이 이들을 관리하는 수단으로서 사용될 수 있다.

[0342] 다음은 이러한 특징들을 구현하는 데 사용할 수 있는 블록체인 시스템의 예를 설명한다.

[0343] 4.1. 예시적인 시스템 개요

[0344] 도 1은 블록체인(150)을 구현하기 위한 예시적인 시스템(100)을 도시한다. 시스템(100)은 패킷-교환 네트워크(101), 통상적으로 인터넷과 같은 광역 인터넷네트워크를 포함할 수 있다. 패킷-교환 네트워크(101)는 패킷-교환 네트워크(101) 내에서 P2P(peer-to-peer) 네트워크(106)를 형성하도록 배열될 수 있는 복수의 블록체인 노드들(104)을 포함한다. 예시되지는 않았지만, 블록체인 노드들(104)은 거의 완전한 그래프로서 배열될 수 있다. 따라서, 각각의 블록체인 노드(104)는 다른 블록체인 노드들(104)에 고도로 연결된다.

[0345] 각각의 블록체인 노드(104)는 피어들의 컴퓨터 장비를 포함하며, 노드들(104) 중 상이한 노드들은 상이한 피어에 속한다. 각각의 블록체인 노드(104)는 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU(central processing unit)들, 가속기 프로세서들, 애플리케이션 특정 프로세서 및/또는 FPGA(field programmable gate array)들, 및 다른 장비 이블테면, ASIC(application specific integrated circuit)들을 포함하는 프로세싱 장치를 포함한다. 각각의 노드는 또한 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 포함한다. 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 드라이브(SSD), 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다.

[0346] 블록체인(150)은 데이터의 블록들의 체인(151)을 포함하며, 여기서 블록체인(150)의 개개의 사본은 분산 또는 블록체인 네트워크(106) 내 복수의 블록체인 노드들(104) 각각에서 유지된다. 위에서 언급된 바와 같이, 블록체인(150)의 사본을 유지하는 것은 반드시 블록체인(150)을 전부 저장하는 것을 의미하지는 않는다. 대신, 블록체인(150)은 각각의 블록체인 노드(150)가 각각의 블록(151)의 블록 헤더(아래에서 논의됨)를 저장하는 한, 정리된 상태의 데이터일 수 있다. 체인의 각각의 블록(151)은 하나 이상의 트랜잭션들(152)을 포함하며, 여기서 이 맥락에서 트랜잭션은 일종의 데이터 구조를 지칭한다. 데이터 구조의 성질은 트랜잭션 모델 또는 체계(scheme)의 일부로서 사용되는 트랜잭션 프로토콜의 유형에 의존할 것이다. 주어진 블록체인은 전반에 걸쳐 하나의 특정 트랜잭션 프로토콜을 사용할 것이다. 하나의 공통 유형의 트랜잭션 프로토콜에서, 각각의 트랜잭션(152)의 데이터 구조는 적어도 하나의 입력 및 적어도 하나의 출력을 포함한다. 각각의 출력은 재산으로서 디지털 자산의 양을 표현하는 금액을 지정하며, 그의 예는 출력이 암호학적으로 잠겨 있는 사용자(103)이다(이는 잠금해제되고 그리하여 리딤(redeem) 또는 지출되기 위해 그 사용자의 서명 또는 다른 솔루션을 요구함). 각각의 입력은 선행 트랜잭션(152)의 출력을 뒤로 가리키고, 그리하여 트랜잭션들을 링크한다.

[0347] 각각의 블록(151)은 또한 블록들(151)에 대한 순차적인 순서를 정의하기 위해 체인에서 이전에 생성된 블록(151)을 뒤로 가리키는 블록 포인터(155)를 포함한다. (코인베이스 트랜잭션 외의) 각각의 트랜잭션(152)은 트랜잭션들의 시퀀스들에 대한 순서를 정의하기 위해 이전 트랜잭션에 대한 역 포인터를 포함한다(트랜잭션들(152)의 시퀀스들은 분기가 허용됨을 주의함). 블록들의 체인(151)은 체인의 최초 블록이었던 제네시스(genesis) 블록(Gb)(153)까지 완전히 거슬러 올라간다. 체인(150) 초반의 하나 이상의 오리지널 트랜잭션들(152)은 선행 트랜잭션이 아닌 제네시스 블록(153)을 가리켰다.

[0348] 블록체인 노드들(104) 각각은 트랜잭션들(152)을 다른 블록체인 노드들(104)로 포워딩하고 그리하여 트랜잭션들(152)이 네트워크(106) 전체에 전파되게 하도록 구성된다. 각각의 블록체인 노드(104)는 블록들(151)을 생성하고 동일한 블록체인(150)의 개개의 사본을 그들 개개의 메모리에 저장하도록 구성된다. 각각의 블록체인 노드(104)는 또한 블록(151)에 통합되기를 기다리는 트랜잭션들(152)의 순서화된 세트(또는 "풀")(154)를 유지한다. 순서화된 풀(154)은 종종 "멤풀(mempool)"로서 지칭된다. 본원에서 이 용어는 임의의 특정 블록체인, 프로토콜 또는 모델에 제한되는 것으로 의도되지 않는다. 이는 노드(104)가 유효한 것으로 수락하고 노드(104)가 동일한 출력을 지출하려고 시도하는 임의의 다른 트랜잭션들을 수락하지 않을 의무가 있는 트랜잭션들의 순서화된 세트

를 지칭한다.

- [0349] 주어진 현재 트랜잭션(152j)에서, 그(또는 각각의) 입력은 트랜잭션들의 시퀀스에서 선행 트랜잭션(152i)의 출력을 참조하는 포인터를 포함하여, 그러한 출력이 현재 트랜잭션(152j)에서 "지출"되거나 리딤됨을 지정한다. 일반적으로, 선행 트랜잭션은 순서화된 세트(154) 또는 임의의 블록(151)의 임의의 트랜잭션일 수 있다. 선행 트랜잭션(152i)은 현재 트랜잭션(152j)이 생성되거나 심지어 네트워크(106)로 전송될 때 반드시 존재할 필요는 없지만, 선행 트랜잭션(152i)은 현재 트랜잭션이 유효하기 위해 존재하고 유효성 검증될 필요가 있을 것이다. 따라서 본원에서 "선행(preceding)"이라 함은 포인터들에 의해 링크된 논리적 시퀀스의 선행자를 지칭하며, 반드시 시간적 시퀀스의 전송 또는 생성 시간은 아니고, 따라서 트랜잭션들(152i, 152j)은 순서와 다르게(out-of-order)(고아 트랜잭션들에 대한 아래 논의 참조) 전송되거나 생성되는 것을 반드시 배제하지 않는다. 선행 트랜잭션(152i)은 앞선(antecedent) 트랜잭션 또는 선행자(predecessor) 트랜잭션으로 동등하게 칭해질 수 있다.
- [0350] 현재 트랜잭션(152j)의 입력은 또한 입력 인가, 예컨대, 선행 트랜잭션(152i)의 출력이 잠겨 있는 사용자(103a)의 서명을 포함한다. 차례로, 현재 트랜잭션(152j)의 출력은 새로운 사용자 또는 엔티티(103b)에 암호학적으로 잠길 수 있다. 따라서 현재 트랜잭션(152j)은 선행 트랜잭션(152i)의 입력에서 정의된 금액을 현재 트랜잭션(152j)의 출력에서 정의된 바와 같은 새로운 사용자 또는 엔티티(103b)에 전달할 수 있다. 일부 경우들에서 트랜잭션(152)은 다수의 사용자들 또는 엔티티들(이들 중 하나는 잔돈(change)을 주기 위해 오리지널 사용자 또는 엔티티들(103a)일 수 있음) 사이에서 입력 금액을 분할하기 위해 다수의 출력들을 가질 수 있다. 일부 경우에서 트랜잭션은 또한 하나 이상의 선행 트랜잭션들의 다수의 출력들로부터 금액들을 수집하고 현재 트랜잭션의 하나 이상의 출력들에 재분배하기 위해 다수의 입력들을 가질 수 있다.
- [0351] 비트코인과 같은 출력-기반 트랜잭션 프로토콜에 따르면, 개별 사용자 또는 조직과 같은 당사자(103)가 새로운 트랜잭션(152j)을 제정(enact)하기를 원할 때(수동으로 또는 당사자에 의해 사용되는 자동화된 프로세스에 의해) 제정 당사자는 자신의 컴퓨터 단말(102)로부터 수령인에게 새로운 트랜잭션을 전송한다. 제정 당사자 또는 수령인은 결국, 이 트랜잭션을 네트워크(106)의 블록체인 노드들(104) 중 하나 이상(이는 요즘에는, 통상적으로 서버들 또는 데이터 센터들이지만, 원칙적으로는 다른 사용자 단말들일 수 있음)에 전송할 것이다. 그것은 또한 새로운 트랜잭션(152j)을 제정하는 당사자(103)가 일부 예들에서는 수령인이 아니라, 블록체인 노드들(104) 중 하나 이상에 직접 트랜잭션을 전송할 수 있는 것이 배제되지 않는다. 트랜잭션을 수신한 블록체인 노드(104)는 블록체인 노드들(104) 각각에 적용되는 블록체인 노드 프로토콜에 따라 트랜잭션이 유효한지를 체크한다. 블록체인 노드 프로토콜은 통상적으로 블록체인 노드(104)가 새로운 트랜잭션(152j)의 암호화 서명이 예상되는 서명과 매칭되는지를 체크하도록 요구하며, 이는 트랜잭션들(152)의 순서화된 시퀀스에서 이전 트랜잭션(152i)에 의존한다. 이러한 출력-기반 블록체인 프로토콜에서, 이는 새로운 트랜잭션(152j)의 입력에 포함된 당사자(103)의 암호화 서명 또는 다른 인가가 새로운 트랜잭션이 할당하는 선행 트랜잭션(152i)의 출력에 정의된 조건과 매칭되는지를 체크하는 것을 포함하며, 여기에서 이 조건은 통상적으로 적어도 새로운 트랜잭션(152j)의 입력의 암호화 서명 또는 다른 인가가 새로운 트랜잭션의 입력이 링크되는 이전 트랜잭션(152i)의 출력을 잠금해제한다는 것을 체크하는 것을 포함한다. 조건은 선행 트랜잭션(152i)의 출력에 포함된 스크립트에 의해 적어도 부분적으로 정의될 수 있다. 대안적으로 이는 단순히 블록체인 노드 프로토콜만으로 고정되거나, 이들의 조합으로 인한 것일 수 있다. 어느 쪽이든, 새로운 트랜잭션(152j)이 유효한 경우, 블록체인 노드(104)는 이를 블록체인 네트워크(106) 내 하나 이상의 다른 블록체인 노드들(104)에 포워딩한다. 이러한 다른 블록체인 노드들(104)은 동일한 블록체인 노드 프로토콜에 따라 동일한 테스트를 적용하고, 이에 따라 새로운 트랜잭션(152j)을 하나 이상의 추가 노드들(104)로 포워딩하는 식이다. 이러한 방식으로, 새로운 트랜잭션은 블록체인 노드들(104)의 네트워크 전반에 걸쳐 전파된다.
- [0352] 출력-기반 모델에서, 주어진 출력(예컨대, UTXO)이 할당(예컨대, 지출)되는지 여부에 대한 정의는 그것이 블록체인 노드 프로토콜에 따라 다른 전방 트랜잭션(152j)의 입력에 의해 유효하게 리딤되었는지의 여부이다. 트랜잭션이 유효하기 위한 다른 조건은 리딤을 시도하는 선행 트랜잭션(152i)의 출력이 다른 트랜잭션에 의해 이미 리딤되지 않은 것이다. 재차, 유효하지 않은 경우, (무효로서 플래깅되고 경고를 위해 전파되지 않는 한) 트랜잭션(152j)은 블록체인(150)에 레코딩되거나 전파되지 않을 것이다. 이는 트랜잭터가 동일한 트랜잭션의 출력을 한번 초과로 할당하고자 시도하는 이중-지출을 경계한다. 반면, 계정-기반 모델은 계정 잔액을 유지함으로써 이중-지출을 경계한다. 재차, 트랜잭션들의 정의된 순서가 존재하기 때문에, 계정 잔액은 임의의 한 시간에 단일의 정의된 상태를 갖는다.
- [0353] 트랜잭션들을 유효성 검증하는 것 외에도, 블록체인 노드들(104)은 또한 일반적으로 "작업 증명"에 의해 지원되는 채굴로서 지칭되는 프로세스에서 트랜잭션들의 블록들을 생성하는 첫 번째가 되기 위해 경쟁한다. 블록체인

노드(104)에서, 새로운 트랜잭션들은 블록체인(150) 상에 레코딩된 블록(151)에 아직 나타나지 않은 유효한 트랜잭션들의 순서화된 풀(154)에 추가된다. 그 후, 블록체인 노드들은 암호화 퍼즐을 해결하도록 시도함으로써 트랜잭션들의 순서화된 세트(154)로부터 트랜잭션들(152)의 새로운 유효한 블록(151)을 조립하기 위해 경쟁한다. 통상적으로 이는 "논스(nonce)"가 계류중인 트랜잭션들(154)의 순서화된 풀의 표현과 컨캐터네이팅되고(concatenated) 해시될 때, 해시의 출력이 미리 결정된 조건을 충족시키도록 논스 값을 검색하는 것을 포함한다. 예컨대, 미리 결정된 조건은 해시의 출력이 미리 정의된 특정 수의 선행 0들을 갖는 것일 수 있다. 이는 작업 증명 퍼즐의 단 하나의 특정 유형일 뿐이며 다른 유형들도 배제되지 않는다는 것에 주의한다. 해시 함수의 속성은 해시 함수가 그의 입력에 대해 예측 불가능한 출력을 갖는다는 것이다. 따라서 이 검색은 무차별 대입(brute force)에 의해서만 수행될 수 있고, 이에 따라 퍼즐을 해결하고자 하는 각각의 블록체인 노드(104)에서 상당한 양의 프로세싱 자원을 소비한다.

[0354] 퍼즐을 해결하고자 하는 제1 블록체인 노드(104)는 이를 네트워크(106)에 발표하고, 그 솔루션을 증명으로서 제공한다, 이는 그 후 네트워크의 다른 블록체인 노드(104)들에 의해 쉽게 체크될 수 있다(해시에 대한 해가 주어지면, 그 해가 해시의 출력으로 하여금 조건을 충족시키게 한다는 것을 체크하는 것은 간단함). 제1 블록체인 노드(104)는 블록을 수락하고 이에 따라 프로토콜 규칙들을 시행하는 다른 노드들의 임계 컨센서스에 블록을 전파한다. 트랜잭션들의 순서화된 세트(154)는 그 후 블록체인 노드들(104) 각각에 의해 블록체인(150)에 새로운 블록(151)으로서 레코딩된다. 블록 포인터(155)가 또한 체인에서 이전에 생성된 블록(151n-1)을 뒤로 가리키는 새로운 블록(151n)에 할당된다. 예컨대, 작업 증명 솔루션을 생성하는 데 요구되는 해시 형태의 상당량의 노력은 블록체인 프로토콜의 규칙들에 따르려는 제1 노드(104)의 의도를 시그널링한다. 이러한 규칙들은 트랜잭션이 이전에 유효성 검증된 트랜잭션과 동일한 출력을 할당 - 이는 이중 지출로서 달리 알려짐 - 하는 경우 트랜잭션을 유효한 것으로 수락하지 않는 것을 포함한다. 일단 생성되면, 블록(151)은 블록체인 네트워크(106) 내 블록체인 노드들(104) 각각에서 인식 및 유지되기 때문에 수정될 수 없다. 블록 포인터(155)는 또한 블록들(151)에 순차적인 순서를 부과한다. 트랜잭션들(152)은 네트워크(106)의 각각의 블록체인 노드(104)에서 순서화된 블록들에 레코딩되기 때문에, 이는 이에 따라, 트랜잭션들의 변경 불가능한 공개 원장을 제공한다.

[0355] 임의의 주어진 시간에 퍼즐을 해결하기 위해 경쟁하는 상이한 블록체인 노드들(104)은 솔루션을 검색하기 시작한 시기 또는 트랜잭션이 수신된 순서에 의존하여, 임의의 주어진 시간에 아직 공개되지 않은 트랜잭션들 풀(154)의 상이한 스냅샷들에 기초하여 퍼즐을 해결할 수 있다는 것에 주의한다. 누구든 각자의 퍼즐을 먼저 해결하는 사람은 어느 트랜잭션들(152)이 어떤 순서로 다음의 새로운 블록(151n)에 포함되는지를 정의하고, 공개되지 않은 트랜잭션들의 현재 풀(154)은 업데이트된다. 그 후 블록체인 노드들(104)은 공개되지 않은 트랜잭션들(154)의 새롭게 정의된 순서화된 풀로부터 블록을 생성하기 위해 계속 경쟁한다. 발생할 수 있는 임의의 "포크(fork)" - 이는 2개의 블록체인 노드들(104)이 서로 매우 짧은 시간 내에 그의 퍼즐을 해결하여서, 블록체인에 대한 상충되는 뷰(view)가 노드들 사이에 전파되는 경우임 - 를 해결하기 위한 프로토콜이 또한 존재한다. 요컨대, 가장 길게 성장하는 포크의 갈래가 확정적인 블록체인(150)이 된다. 동일한 트랜잭션들이 포크들 둘 모두에 나타날 것이므로, 이는 네트워크의 사용자들 또는 에이전트들에게 영향을 미치지 않아야 한다.

[0356] 비트코인 블록체인(및 대부분의 다른 블록체인들)에 따르면, 새로운 블록(104)을 성공적으로 구성하는 노드에는 (하나의 에이전트 또는 사용자로부터 다른 에이전트 또는 사용자로 디지털 자산의 금액을 이전하는 에이전트-간 또는 사용자-간 트랜잭션과 대조적으로) 디지털 자산의 부가적인 정의된 양을 분배하는 새로운 특별한 종류의 트랜잭션들에서 디지털 자산의 부가적인 수락된 금액을 새롭게 할당하는 능력이 승인된다. 이 특별한 유형의 트랜잭션은 일반적으로 "코인베이스 트랜잭션"으로서 지칭되지만, "초기 트랜잭션" 또는 "생성 트랜잭션"이라고도 칭해질 수 있다. 그것은 통상적으로 새로운 블록(151n)의 제1 트랜잭션을 형성한다. 작업 증명은 나중에 이 특별한 트랜잭션이 리딩되도록 허용하는 프로토콜 규칙들을 따르도록 새로운 블록을 구성하는 노드의 의도를 시그널링한다. 블록체인 프로토콜 규칙은 이 특별한 트랜잭션이 리딩되기 전에 만기 기간 예컨대, 100개의 블록들을 요구할 수 있다. 종종 일반(비-생성) 트랜잭션(152)이 또한 그 트랜잭션이 공개된 블록(151n)을 생성한 블록체인 노드(104M)를 추가로 보상하기 위해, 그의 출력들 중 하나에 부가적인 트랜잭션 수수료를 지정할 것이다. 이 수수료는 일반적으로 "트랜잭션 수수료"로서 지칭되고 아래에서 논의된다.

[0357] 트랜잭션 유효성 검증 및 공개와 관련된 자원들로 인해, 통상적으로 적어도 블록체인 노드들(104) 각각은 하나 이상의 물리적 서버 유닛들, 또는 심지어 전체 데이터 센터를 포함하는 서버의 형태를 취한다. 그러나 원칙적으로 임의의 주어진 블록체인 노드(104)는 사용자 단말 또는 함께 네트워킹된 사용자 단말들의 그룹의 형태를 취할 수 있다.

[0358] 각각의 블록체인 노드(104)의 메모리는 블록체인 노드 프로토콜에 따라 각자의 역할 또는 역할들을 수행하고 트

랜잭션들(152)을 처리하기 위해 블록체인 노드(104)의 프로세싱 장치 상에서 실행되도록 구성된 소프트웨어를 저장한다. 본원에서 블록체인과 노드(104)에 기인한 임의의 액션은 각자의 컴퓨터 장비의 프로세싱 장치 상에서 실행되는 소프트웨어에 의해 수행될 수 있다는 것이 이해될 것이다. 노드 소프트웨어는 애플리케이션 계층, 또는 운영 체제 계층이나 프로토콜 계층과 같은 하위 계층 또는 이들의 임의의 조합에서 하나 이상의 애플리케이션들로 구현될 수 있다.

[0359] 또한 네트워크(101)에는 소비 사용자들의 역할을 하는 복수의 당사자들(103) 각각의 컴퓨터 장비(102)가 연결되어 있다. 이러한 사용자들은 블록체인 네트워크(106)와 상호작용할 수 있지만 트랜잭션들을 유효성 검증하거나 블록들을 구성하는 데 참여하지 않는다. 이러한 사용자들 또는 에이전트들(103) 중 일부는 트랜잭션들에서 전송자들 및 수령인들로서 작용할 수 있다. 다른 사용자들은 반드시 전송자들 또는 수령인들로서 작용할 필요 없이 블록체인(150)과 상호작용할 수 있다. 예컨대, 일부 당사자들은 블록체인(150)의 사본을 저장하는 저장 엔티티들로서 작용할 수 있다(예컨대, 블록체인 노드(104)로부터 블록체인의 사본을 획득함).

[0360] 당사자들(103) 중 일부 또는 전부는 상이한 네트워크, 예컨대, 블록체인 네트워크(106) 위에 오버레이된 네트워크의 부분으로서 연결될 수 있다. 블록체인 네트워크의 사용자들(종종 "클라이언트"로서 지칭됨)은 블록체인 네트워크(106)를 포함하는 시스템의 일부로서 언급될 수 있지만; 이러한 사용자들은 블록체인 노드들에서 요구되는 역할을 수행하지 않기 때문에 블록체인 노드들(104)이 아니다. 대신에, 각각의 당사자(103)는 블록체인 네트워크(106)와 상호작용할 수 있고 그리하여 블록체인 노드(106)에 연결(즉, 통신)함으로써 블록체인(150)을 활용할 수 있다. 제1 당사자(103a) 및 그/그녀의 개개의 컴퓨터 장비(102a) 및 제2 당사자(103b) 및 그/그녀의 개개의 컴퓨터 장비(102b)인 두 당사자들(103) 및 이들의 개개의 장비(102)가 예시 목적으로 도시된다. 훨씬 더 많은 이러한 당사자들(103) 및 이들의 개개의 컴퓨터 장비(102)가 존재하고 시스템(100)에 참여할 수 있지만, 편의상 그것들은 예시되지 않는다는 것이 이해될 것이다. 각각의 당사자(103)는 개인 또는 조직일 수 있다. 순전히 예시로서, 제1 당사자(103a)는 본원에서 앨리스(Alice)로서 지칭되고 제2 당사자(103b)는 밥(Bob)으로서 지칭되지만, 이것이 제한적이지 않고 본원에서 앨리스 또는 밥에 대한 임의의 참조는 각각 "제1 당사자" 및 "제2 당사자"로 대체될 수 있다는 것이 인지될 것이다.

[0361] 각각의 당사자(103)의 컴퓨터 장비(102)는 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU들, GPU들, 다른 가속기 프로세서들, 애플리케이션 특정 프로세서들 및/또는 FPGA들을 포함하는 개개의 프로세싱 장치를 포함한다. 각각의 당사자(103)의 컴퓨터 장비(102)는 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 더 포함한다. 이 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 SSD, 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다. 각각의 당사자(103)의 컴퓨터 장비(102) 상의 메모리는 프로세싱 장치 상에서 실행되도록 배열된 적어도 하나의 클라이언트 애플리케이션(105)의 개개의 인스턴스를 포함하는 소프트웨어를 저장한다. 본원에서 주어진 당사자(103)에 기인한 임의의 액션은 개개의 컴퓨터 장비(102)의 프로세싱 장치 상에서 실행되는 소프트웨어를 사용하여 수행될 수 있다는 것이 이해될 것이다. 각각의 당사자(103)의 컴퓨터 장비(102)는 적어도 하나 사용자 단말, 예컨대, 데스크 톱 또는 랩톱 컴퓨터, 태블릿, 스마트폰, 또는 스마트워치와 같은 웨어러블 디바이스를 포함한다. 주어진 당사자(103)의 컴퓨터 장비(102)는 또한 사용자 단말을 통해 액세스되는 클라우드 컴퓨팅 자원들과 같은 하나 이상의 다른 네트워킹된 자원들을 포함할 수 있다.

[0362] 예컨대, 서버로부터 다운로드되거나, 또는 이동식 저장 디바이스 이를테면, 이동식 SSD, 플래시 메모리 키, 이동식 EEPROM, 이동식 자기 디스크 드라이브, 자기 플로피 디스크 또는 테이프, 광학 디스크 이를테면, CD 또는 DVD ROM 또는 이동식 광학 드라이브 등 상에서 제공되는 클라이언트 애플리케이션(105)은 적절한 컴퓨터-판독 가능 저장 매체 또는 매체들 상에서 임의의 주어진 당사자(103)의 컴퓨터 장비(102)에 초기에 제공될 수 있다.

[0363] 클라이언트 애플리케이션(105)은 적어도 "지갑" 기능을 포함한다. 이는 2개의 메인 기능성들을 갖는다. 이들 중 하나는 개개의 당사자(103)가 트랜잭션들(152)을 생성하고 인가(예컨대, 서명)하여 하나 이상의 비트코인 노드들(104)에 전송하여, 이어서 블록체인 노드들(104)의 네트워크 전반에 걸쳐 전파되고 그리하여 블록체인(150)에 포함되는 것을 가능하게 하는 것이다. 남은 하나는 개개의 당사자에게 자신이 현재 소유하고 있는 디지털 자산의 금액을 다시 보고하는 것이다. 출력-기반 시스템에서, 이 제2 기능성은 블록체인(150) 전반에 걸쳐 흩어져 있는 해당 당사자에 속하는 다양한 트랜잭션들(152)의 출력들에서 정의된 금액들을 대조하는 것을 포함한다.

[0364] 참고: 다양한 클라이언트 기능성이 주어진 클라이언트 애플리케이션(105)에 통합되는 것으로서 설명될 수 있다

만, 이는 반드시 제한적인 것은 아니며, 대신 본원에서 설명된 클라이언트 기능성은 API를 통해 인터페이싱하거나 하나가 남은 하나에 플러그인하는 두 개 이상의 개별 애플리케이션들의 세트에서 구현될 수 있다. 보다 일반적으로, 클라이언트 기능성은 애플리케이션 계층 또는 운영 체제와 같은 하위 계층 또는 이들의 임의의 조합에서 구현될 수 있다. 다음은 클라이언트 애플리케이션(105)과 관련하여 설명될 것이지만, 이것이 제한적이지 않다는 것이 인지될 것이다.

[0365] 각각의 컴퓨터 장비(102) 상의 클라이언트 애플리케이션 또는 소프트웨어(105)의 인스턴스는 네트워크(106)의 블록체인 노드들(104) 중 적어도 하나에 동작 가능하게 커플링된다. 이는 클라이언트(105)의 지갑 기능이 트랜잭션들(152)을 네트워크(106)로 전송하는 것을 가능하게 한다. 클라이언트(105)는 또한 개개의 당사자(103)가 수령인인 임의의 트랜잭션들에 대해 블록체인(150)에 질의하기 위해(또는 실시예들에서, 블록체인(150)은 그의 공개 가시성을 통해 부분적으로 트랜잭션들의 신뢰를 제공하는 공공 시설(public facility)이므로, 실제로 블록체인(150)에서 다른 당사자들의 트랜잭션을 검사하기 위해) 블록체인 노드들(104)에 접촉할 수 있다. 각각의 컴퓨터 장비(102) 상의 지갑 기능은 트랜잭션 프로토콜에 따라 트랜잭션들(152)을 공식화(formulate) 하고 전송하도록 구성된다. 위에서 제시된 바와 같이, 각각의 블록체인 노드(104)는 블록체인 노드 프로토콜에 따라 트랜잭션들(152)을 유효성 검증하고 트랜잭션들(152)을 포워딩하여 이들을 블록체인 네트워크(106) 전체에 전파하도록 구성된 소프트웨어를 실행한다. 트랜잭션 프로토콜 및 노드 프로토콜은 서로 대응하며, 주어진 트랜잭션 프로토콜은 주어진 트랜잭션 모델을 함께 구현하도록 주어진 노드 프로토콜을 따른다. 동일한 트랜잭션 프로토콜이 블록체인(150) 내 모든 트랜잭션들(152)에 사용된다. 동일한 노드 프로토콜이 네트워크(106) 내 모든 노드들(104)에 의해 사용된다.

[0366] 주어진 당사자(103), 이를테면 앨리스가 블록체인(150)에 포함될 새로운 트랜잭션(152j)을 전송하기를 원할 때, 그녀는 (자신의 클라이언트 애플리케이션(105)의 지갑 기능을 사용하여) 관련 트랜잭션 프로토콜에 따라 새로운 트랜잭션을 공식화한다. 그 후, 그녀는 클라이언트 애플리케이션(105)으로부터 그녀가 연결되는 하나 이상의 블록체인 노드들(104)에 트랜잭션(152)을 전송한다. 예컨대, 이는 앨리스의 컴퓨터(102)에 가장 잘 연결된 블록체인과 노드(104)일 수 있다 임의의 주어진 블록체인 노드(104)가 새로운 트랜잭션(152j)을 수신할 때, 주어진 노드는 블록체인 노드 프로토콜 및 각자의 역할에 따라 이를 처리한다. 이는 새롭게 수신된 트랜잭션(152j)이 "유효"하기 위한 특정 조건을 충족시키는지를 먼저 체크하는 것을 포함하며, 그의 예들은 곧 보다 자세히 논의될 것이다. 일부 트랜잭션 프로토콜들에서, 유효성 검증을 위한 조건은 트랜잭션들(152)에 포함된 스크립트들에 의해 트랜잭션 단위로 구성 가능할 수 있다. 대안적으로, 조건은 단순히 노드 프로토콜의 내장 피처이거나, 스크립트 및 노드 프로토콜의 조합으로 정의될 수 있다.

[0367] 새롭게 수신된 트랜잭션(152j)이 유효한 것으로 간주되기 때문에 테스트를 통과한다는 것을 조건으로(즉, 그것이 "유효성 검증"된다는 조건으로), 트랜잭션(152j)을 수신하는 임의의 블록체인 노드(104)는 새로운 유효성 검증된 트랜잭션(152)을 그 블록체인 노드(104)에서 유지되는 블록체인들(154)의 순서화된 세트(154)에 추가할 것이다. 또한, 트랜잭션(152j)을 수신하는 임의의 블록체인 노드(104)는 유효성 검증된 트랜잭션(152)을 네트워크(106)의 하나 이상의 다른 블록체인 노드들(104)로 전방으로 전파시킬 것이다. 각각의 블록체인 노드(104)가 동일한 프로토콜을 적용하기 때문에, 트랜잭션(152j)이 유효하다고 가정하면, 이는 그것이 곧 전체 네트워크(106)에 걸쳐 전파될 것임을 의미한다.

[0368] 일단 주어진 블록체인 노드(104)에서 유지되는 계류중인 트랜잭션(154)의 순서화된 풀에 허용되면, 블록체인 노드(104)는 새로운 트랜잭션(152)을 포함하여 154의 각자의 풀의 최신 버전 상에서 작업 증명 퍼즐을 해결하기 위해 경쟁하기 시작할 것이다(다른 블록체인 노드들(104)은 트랜잭션들의 상이한 풀(154)에 기초하여 퍼즐을 해결하려고 시도할 수 있지만 누구든 먼저 해결하는 사람은 최신 블록(151)에 포함된 트랜잭션들의 세트를 정의할 것임을 상기한다). 결국 블록체인 노드(104)는 앨리스의 트랜잭션(152j)을 포함하는 순서화된 풀(154)의 일부에 대한 퍼즐을 해결할 것이다. 새로운 트랜잭션(152j)을 포함하는 풀(154)에 대한 작업 증명이 완료되면, 이는 변경 불가능하게 블록체인(150)의 블록들(151) 중 하나의 부분이 된다. 각각의 트랜잭션(152)은 이전 트랜잭션에 대한 역 포인터를 포함하여서, 트랜잭션들의 순서가 또한 변경 불가능하게 레코딩된다.

[0369] 상이한 블록체인 노드들(104)은 주어진 트랜잭션의 상이한 인스턴스들을 먼저 수신하고 이에 따라 하나의 인스턴스가 새로운 블록(151)에 공개되기 전에 어떤 인스턴스가 '유효'한지에 관한 상충되는 뷰들을 가질 수 있으며, 이 때 모든 블록체인 노드들(104)은 공개된 인스턴스가 유일한 유효 인스턴스라는 것에 동의한다. 블록체인 노드(104)가 하나의 인스턴스를 유효한 것으로 수락하고 그 후 제2 인스턴스가 블록체인(150)에 레코딩되었음을 발견하는 경우, 해당 블록체인 노드(104)는 이를 수락해야 하며 초기에 수락된 인스턴스(즉, 블록

(151)에서 공개되지 않은 인스턴스)를 폐기(즉, 유효하지 않은 것으로 취급)할 것이다.

[0370]

일부 블록체인 네트워크들에 의해 동작되는 트랜잭션 프로토콜의 대안적인 유형은 계정-기반 트랜잭션 모델의 일부로서 "계정-기반" 프로토콜로서 지칭될 수 있다. 계정-기반의 경우에, 각각의 트랜잭션은 과거 트랜잭션들의 시퀀스에서 선행 트랜잭션의 UTXO를 뒤로 참조하기 보다는, 절대 계정 잔액을 참조함으로써 전달될 금액을 정의한다. 모든 계정들의 현재 상태는 블록체인과 별개로 해당 네트워크의 노드들에 의해 저장되며 지속적으로 업데이트된다. 이러한 시스템에서, 트랜잭션들은 계정의 실행 중인 트랜잭션 집계(또한 "포지션"이라 불림)를 사용하여 순서화된다. 이 값은 그의 암호화 서명의 일부로 발신인에 의해 서명되고 트랜잭션 참조 계산의 부분으로서 해시된다. 게다가, 선택적 데이터 필드가 또한 트랜잭션에 서명할 수 있다. 이 데이터 필드는 예컨대, 이전 트랜잭션 ID가 데이터 필드에 포함된 경우 이전 트랜잭션을 뒤로 가리킬 수 있다.

[0371]

4.2. UTXO-기반 모델

[0372]

도 2는 예시적인 트랜잭션 프로토콜을 예시한다. 이는 UTXO-기반 프로토콜의 예이다. 트랜잭션(152)(약칭 "Tx")은 블록체인(150)의 기본 데이터 구조이다(각각의 블록(151)은 하나 이상의 트랜잭션들(152)을 포함함). 다음은 출력-기반 또는 "UTXO" 기반 프로토콜을 참조하여 설명될 것이다. 그러나 이것은 모든 가능한 실시예들로 제한되지 않는다. 예시적인 UTXO-기반 프로토콜이 비트코인을 참조하여 설명되지만, 다른 예시적인 블록체인 네트워크들 상에서 동일하게 구현될 수 있다는 것에 주의한다.

[0373]

UTXO-기반 모델에서, 각각의 트랜잭션("Tx")(152)은 하나 이상의 입력들(202) 및 하나 이상의 출력들(203)을 포함하는 데이터 구조를 포함한다. 각각의 출력(203)은 (UTXO가 아직 리딤되지 않은 경우) 다른 새로운 트랜잭션의 입력(202)에 대한 소스로서 사용될 수 있는 미지출 트랜잭션 출력(unspent transaction output; UTXO)을 포함할 수 있다. UTXO는 디지털 자산의 금액을 지정하는 값을 포함한다. 이는 분산 원장 상의 세팅된 수의 토큰들을 표현한다. UTXO는 또한 다른 정보 중에서도, 그것이 발생한 트랜잭션의 트랜잭션 ID를 포함할 수 있다. 트랜잭션 데이터 구조는 또한 입력 필드(들)(202) 및 출력 필드(들)(203)의 크기의 표시자를 포함할 수 있는 헤더(201)를 포함할 수 있다. 헤더(201)는 또한 트랜잭션의 ID를 포함할 수 있다. 실시예들에서, 트랜잭션 ID는 (트랜잭션 ID 자체는 제외한) 트랜잭션 데이터의 해시이고 노드들(104)에게 제출된 원시 트랜잭션(152)의 헤더(201)에 저장된다.

[0374]

앨리스(103a)가 해당 디지털 자산의 금액을 밥(103b)에게 전달하는 트랜잭션(152j)을 생성하기를 원한다고 하자. 도 2에서 앨리스의 새로운 트랜잭션(152j)은 "Tx_j"로서 라벨이 지정된다. 이는 시퀀스의 선행 트랜잭션(152i)의 출력(203)에서 앨리스에게 잠긴 디지털 자산의 금액을 취하고, 이 중 적어도 일부를 밥에게 전달한다. 선행 트랜잭션(152i)은 도 2에서 "Tx₀"로 라벨이 지정된다. Tx₀ 및 Tx_i은 임의의 라벨일 뿐이다. 이들은, Tx₀이 블록체인(151)의 최초 트랜잭션이거나, Tx_i이 풀(154)에서 바로 다음 트랜잭션이라는 것을 반드시 의미하지는 않는다. Tx_i은 앨리스에게 잠긴 미지출 출력(203)을 여전히 갖는 임의의 선행(즉, 앞선) 트랜잭션을 뒤로 가리킬 수 있다.

[0375]

선행 트랜잭션(Tx₀)은 앨리스가 자신의 새로운 트랜잭션(Tx_i)을 생성할 때, 또는 적어도 그녀가 그것을 네트워크(106)에 전송할 때까지 이미 유효성 검증되고 블록체인(150)의 블록(151)에 포함되었을 수 있다. 이는 그 시간에 이미 블록들(151) 중 하나에 포함되었거나, 순서화된 세트(154)에서 여전히 대기 중일 수 있으며, 이 경우에 곧 새로운 블록(151)에 포함될 것이다. 대안적으로 Tx₀ 및 Tx_i이 생성되고 네트워크(106)에 함께 전송될 수 있거나 또는 노드 프로토콜이 "고아" 트랜잭션들을 버퍼링하도록 허용하는 경우 Tx₀는 Tx_i 이후에도 전송될 수 있다. 트랜잭션들의 시퀀스의 맥락에서 본원에서 사용된 바와 같은 "선행" 및 "후속"이라는 용어들은 (트랜잭션이 다른 트랜잭션을 뒤로 가리키고, 이와 같이 계속되는) 트랜잭션들에서 지정된 트랜잭션 포인터들에 의해 정의된 바와 같은 시퀀스에서의 트랜잭션들의 순서를 지칭한다. 이들은 "선행자(predecessor)" 및 "후행자(successor)", 또는 "앞선(antecedent)"과 "후위의(descendant)", "부모" 및 "자식" 등으로 동등하게 대체될 수 있다. 이는 그것들이 생성되고, 네트워크(106)로 전송되거나, 임의의 주어진 블록체인 노드(104)에 도달하는 순서를 반드시 의미하지는 않는다. 그럼에도 불구하고, 선행 트랜잭션(앞선 트랜잭션 또는 "부모")을 가리키는 후속 트랜잭션(후위의 트랜잭션 또는 "자식")은 부모 트랜잭션이 유효성 검증될 때까지 그리고 유효성 검증되지 않는 한 유효성 검증되지 않을 것이다. 그의 부모 이전에 블록체인과 노드(104)에 도달하는 자식은 고아로 간주된다. 이는 노드 프로토콜 및/또는 노드 거동에 의존하여 부모를 기다리기 위해 특정 시간 동안 버퍼링되거나 폐기될 수 있다.

- [0376] 선행 트랜잭션(T_{x_0})의 하나 이상의 출력들(203) 중 하나는, 본원에서 $UTXO_0$ 으로서 라벨이 지정되는 특정 $UTXO$ 를 포함한다. 각각의 $UTXO$ 는 $UTXO$ 에 의해 표현되는 디지털 자산의 금액을 지정하는 값 및 후속 트랜잭션이 유효성 검증되고 따라서 $UTXO$ 가 성공적으로 리딤되기 위하여 후속 트랜잭션의 입력(202)에서 잠금해제 스크립트에 의해 만족되어야 하는 조건을 정의하는 잠금 스크립트를 포함한다. 통상적으로, 잠금 스크립트는 특정 당사자(그것이 포함된 트랜잭션의 수혜자)에게로 금액을 잠근다. 즉, 잠금 스크립트는, 통상적으로 후속 트랜잭션의 입력의 잠금해제 스크립트가 선행 트랜잭션이 잠겨 있는 당사자의 암호화 서명을 포함하는 조건을 포함하는 잠금해제 조건을 정의한다.
- [0377] 잠금 스크립트(일명 `scriptPubKey`)는 노드 프로토콜에 의해 인식되는 도메인 특정 언어로 작성된 코드 조각이다. 이러한 언어의 특정 예는 블록체인 네트워크에 의해 사용되는 "스크립트(Script)"(대문자 S)라 불린다. 잠금 스크립트는 트랜잭션 출력(203)을 지출하는 데 어떤 정보가 필요한지, 예컨대, 앨리스의 서명 요건을 지정한다. 잠금해제 스크립트들은 트랜잭션들의 출력에서 나타난다. 잠금해제 스크립트(일명 `scriptSig`)는 잠금 스크립트 기준들을 충족시키는 데 필요한 정보를 제공하는 도메인 특정 언어로 작성된 코드 조각이다. 예컨대, 이는 밥의 서명을 포함할 수 있다. 잠금해제 스크립트들은 트랜잭션들의 입력(202)에 나타난다.
- [0378] 따라서 예시된 예에서, T_{x_0} 의 출력(203)의 $UTXO_0$ 은 $UTXO_0$ 가 리딤되기 위해(엄밀히, $UTXO_0$ 을 리딤하고자 시도하는 후속 트랜잭션이 유효하기 위해) 앨리스의 서명 $Sig P_A$ 를 요구하는 잠금 스크립트 $[Checksig P_A]$ 를 포함한다. $[Checksig P_A]$ 는 앨리스의 공개-개인 키 쌍으로부터의 공개 키 P_A 의 표현(예컨대, 해시)을 포함한다. T_{x_1} 의 입력(202)은 (예컨대, 실시예에서, 전체 트랜잭션 T_{x_0} 의 해시인 그의 트랜잭션 Id인 $TxID_0$ 에 의해) T_{x_1} 을 뒤로 가리키는 포인터를 포함한다. T_{x_1} 의 입력(202)은 T_{x_0} 내에서 $UTXO_0$ 을 식별하는 인덱스를 포함하여, T_{x_0} 의 임의의 다른 가능한 출력들 사이에서 그것을 식별한다. T_{x_1} 의 입력(202)은 앨리스의 암호화 서명을 포함하는 잠금해제 스크립트 $\langle Sig P_A \rangle$ 를 더 포함하며, 이는 앨리스가 키 쌍으로부터 자신의 개인 키를 데이터의 미리 정의된 부분(때로는 암호화에서 "메시지"라 불림)에 적용함으로써 생성된다. 유효한 서명을 제공하기 위해 앨리스에 의해 서명될 필요가 있는 데이터(또는 "메시지")는 잠금 스크립트, 노드 프로토콜 또는 이들의 조합에 의해 정의될 수 있다.
- [0379] 새로운 트랜잭션 T_{x_1} 이 블록체인 노드(104)에 도달할 때, 노드는 노드 프로토콜을 적용한다. 이는 잠금해제 스크립트가 잠금 스크립트에 정의된 조건(이 조건은 하나 이상의 기준들을 포함할 수 있음)을 충족시키는지를 체크하기 위해 잠금 스크립트 및 잠금해제 스크립트를 함께 실행하는 것을 포함한다. 실시예들에서, 이는 2개의 스크립트들을 컨케터네이팅(concatenating)하는 것을 수반한다.
- [0380] $\langle Sig P_A \rangle \langle P_A \rangle || [Checksig P_A]$
- [0381] 여기에서 "||"는 컨케터네이션을 표현하고 "<...>"는 스택 상에 데이터를 배치하는 것을 의미하고, "[...]"는 잠금 스크립트(이 예에서, 스택-기반 언어)에 의해 구성된 함수이다. 동등하게, 스크립트들을 컨케터네이팅하는 대신, 스크립트들은 공통 스택을 사용하여 번갈아 실행될 수 있다. 어느 쪽이든, 함께 실행될 때, 스크립트들은 T_{x_0} 의 출력의 잠금 스크립트에 포함된 바와 같은 앨리스의 공개 키 P_A 를 사용하여, T_{x_1} 의 입력의 잠금해제 스크립트가 데이터의 예상되는 부분에 서명하는 앨리스의 서명을 포함한다는 것을 입증한다. 이 인증을 수행하기 위하여 데이터의 예상되는 부분 자체("메시지")가 또한 포함될 필요가 있다. 실시예들에서, 서명된 데이터는 T_{x_1} 전체를 포함한다(이에 따라, 평문으로 데이터의 서명된 부분을 지정하는 별개의 요소가 포함될 필요가 없는 데, 그 이유는 그것이 이미 본질적으로 존재하기 때문임).
- [0382] 공개-개인 암호화에 의한 인증의 세부사항들은 당업자에게 친숙할 것이다. 기본적으로, 앨리스가 자신의 개인 키를 사용하여 메시지에 서명한 경우, 앨리스의 공개 키 및 평문의 메시지를 감안하여, 노드(104)와 같은 다른 엔티티는 메시지가 앨리스에 의해 서명된 것임이 틀림없다는 것을 입증할 수 있다. 서명은 통상적으로 메시지를 해시하는 것, 해시에 서명하는 것, 그리고 이를 서명으로서 메시지에 태깅하고, 이에 따라 공개 키의 임의의 보유자(holder)가 서명을 인증하는 것을 가능하게 하는 것을 포함한다. 따라서 여기에서 특정 데이터 조각 또는 트랜잭션의 일부 등에 서명하는 것에 대한 임의의 참조는 실시예들에서 해당 데이터 조각 또는 트랜잭션 일부의 해시에 서명하는 것을 의미할 수 있다는 것에 주의한다.
- [0383] T_{x_1} 의 잠금해제 스크립트가 T_{x_0} 의 잠금 스크립트에 지정된 하나 이상의 조건들을 충족시키는 경우(이에 따라, 보

여진 예에서, 앨리스의 서명이 Tx_1 에서 제공되고 인증된 경우), 블록체인과 노드(104)는 Tx_1 이 유효한 것으로 간주한다. 이는 블록체인 노드(104)가 계류중인 트랜잭션들(154)의 순서화된 풀에 Tx_1 을 추가할 것임을 의미한다. 블록체인 노드(104F)는 또한 트랜잭션 Tx_1 을 네트워크(106) 내 하나 이상의 다른 블록체인 노드들(104)로 포워딩할 것이어서, 그 트랜잭션이 네트워크(106) 전반에 걸쳐 전파될 것이다. Tx_1 이 유효성 검증되고 블록체인(150)에 포함되면, 이는 지출된 것으로 Tx_0 으로부터의 UTXO₀를 정의한다. Tx_1 은 그것이 미지출 트랜잭션 출력(203)을 지출하는 경우에만 유효할 수 있다는 것에 주의한다. 다른 트랜잭션(152)에 의해 이미 지출된 출력을 지출하려고 시도하는 경우, 다른 모든 조건들이 충족되는 경우조차도 Tx_1 은 유효하지 않을 것이다. 따라서 블록체인과 노드(104)는 또한 선행 트랜잭션 Tx_0 에서 참조된 UTXO가 이미 지출되었는지(즉, 다른 유효한 트랜잭션에 대한 유효한 입력을 이미 형성했는지)를 체크할 필요가 있다. 이는 트랜잭션들(152) 상에 정의된 순서를 부과하는 것이 블록체인(150)에 대해 중요한 하나의 이유이다. 실제로, 주어진 블록체인 노드(104)는 트랜잭션들(152)이 지출된 UTXO들(203)을 마킹하는 별개의 데이터베이스를 유지할 수 있지만, 궁극적으로 UTXO가 지출되었는지를 정의하는 것은 블록체인(150)의 다른 유효한 트랜잭션에 대한 유효한 입력이 이미 형성되었는지의 여부이다.

[0384] 주어진 트랜잭션(152)의 모든 출력들(203)에서 지정된 총 금액이 모든 그의 입력들(202)에 의해 가리켜지는 총 금액보다 큰 경우, 이는 대부분의 트랜잭션 모델들에서 무효에 대한 다른 근거이다. 따라서 이러한 트랜잭션들은 전파되지도 않고 블록(151)에 포함되지 않을 것이다.

[0385] UTXO-기반 트랜잭션 모델에서, 주어진 UTXO는 전체로서 지출될 필요가 있다는 것에 주의한다. 다른 프랙션(fraction)이 지출되면서, 지출된 것으로 UTXO에서 정의된 금액의 프랙션을 "남겨둘" 수는 없다. 그러나 UTXO로부터의 금액은 다음 트랜잭션의 다수의 출력들 사이에서 분할될 수 있다. 예컨대, Tx_0 의 UTXO₀에 정의된 금액은 Tx_1 의 다수의 UTXO들 사이에서 분할될 수 있다. 따라서 앨리스가 UTXO₀에 정의된 모든 금액을 밥에게 주기를 원하지 않는 경우, 앨리스는 Tx_1 의 제2 출력에서 자신에게 잔돈을 주거나, 다른 당사자에게 지불하는데 나머지를 사용할 수 있다.

[0386] 실제로, 앨리스는 또한 일반적으로 블록(151)에 그녀의 트랜잭션(104)을 성공적으로 포함시키는 비트코인 노드(104)에 대한 수수료를 포함할 필요가 있을 것이다. 앨리스가 그러한 수수료를 포함시키지 않는 경우, Tx_0 은 블록체인 노드들(104M)에 의해 거부될 수 있고, 이에 따라 기술적으로 유효하더라도, 전파되어 블록체인(150)에 포함되지 않을 수 있다(노드프로토콜은 블록체인 노드들(104)이 원하지 않는 경우 이들에게 트랜잭션들(152)을 수락하도록 강요하지 않음). 일부 프로토콜들에서, 트랜잭션 수수료는 자체의 별개의 출력(203)을 요구하지 않는다(즉, 별개의 UTXO가 필요하지 않음). 대신, 주어진 트랜잭션(152)의 입력(들)(202)에 의해 가리켜지는 총 금액과 출력(들)(203)에 지정된 총 금액 사이의 임의의 차이는 트랜잭션을 공개한 블록체인 노드(104)에게 자동으로 주어진다. 예컨대, UTXO₀에 대한 포인터가 Tx_1 에 대한 유일한 입력이고 Tx_1 은 단 하나의 출력 UTXO₁만을 갖는다고 하자. UTXO₀에 지정된 디지털 자산의 금액이 UTXO₁에 지정된 금액보다 큰 경우, 차이는 UTXO₁을 포함하는 블록을 생성하기 위한 작업 증명 경쟁에서 승리한 노드(104)에 의해 할당될 수 있다. 그러나 대안적으로 또는 부가적으로, 트랜잭션 수수료가 트랜잭션(152)의 UTXO들(203) 중 자체 UTXO에서 명시적으로 지정될 수 있다는 것이 반드시 배제되는 것은 아니다.

[0387] 앨리스 및 밥의 디지털 자산들은 블록체인(150)의 임의의 위치의 임의의 트랜잭션들(152)에서 그들에게 잠겨 있는 UTXO로 구성된다. 따라서 통상적으로, 주어진 당사자(103)의 자산들은 블록체인(150) 전반에 걸친 다양한 트랜잭션들(152)의 UTXO들에 걸쳐 흩어져 있다. 블록체인(150)의 어떤 위치에도 주어진 당사자(103)의 총 잔액을 정의하는 숫자는 전혀 없다. 클라이언트 애플리케이션(105)에서 지갑 기능의 역할은, 개개의 당사자에게 잠겨 있으며 다른 전방 트랜잭션에서 아직 지출되지 않은 모든 다양한 UTXO들의 값들을 함께 대조하는 것이다. 비트코인 노드들(104) 중 임의의 것에 저장된 블록체인(150)의 사본을 질의함으로써 이것이 수행될 수 있다.

[0388] 스크립트 코드는 종종 도식적으로(즉, 정확한 언어를 사용하지 않음) 표현된다는 것에 주의한다. 예컨대, 특정 기능을 표현하기 위해 작업 코드(opcode)들이 사용될 수 있다. "OP..."는 스크립트 언어의 특정 작업코드(opcode)를 지칭한다. 예로서, OP_RETURN은 잠금 스크립트의 선두에서 OP_FALSE가 앞에 있을 때 트랜잭션 내에 데이터를 저장하고 그리하여 데이터를 블록체인(150)에 변경 불가능하게 레코딩할 수 있는 트랜잭션의 지출 불가능한 출력을 생성하는 스크립트 언어의 작업코드이다. 예컨대, 데이터는 블록체인에 저장하고자 하는 문서를

포함할 수 있다.

[0389] 통상적으로, 트랜잭션의 입력은 공개 키 P_A 에 대응하는 디지털 서명을 포함한다. 실시예들에서, 이는 타원 곡선 secp256k1 을 사용하는 ECDSA에 기초한다. 디지털 서명은 특정 데이터 조각에 서명한다. 일부 실시예들에서, 주어진 트랜잭션에 대해, 서명은 트랜잭션 입력의 일부, 및 트랜잭션 출력들의 전부 또는 일부에 서명할 것이다. 서명되는 출력들의 특정 부분들은 SIGHASH 플래그에 의존한다. SIGHASH 플래그는 일반적으로 어느 출력들이 서명되는지를 선택하기 위해 서명의 끝에 포함된 4-바이트 코드이다(이에 따라, 서명 시에 고정됨).

[0390] 잠금 스크립트는 때로는, 그것이 통상적으로 개개의 트랜잭션이 잠겨 있는 당사자의 공개 키를 포함한다는 사실을 지칭하는 "scriptPubKey"라 칭해진다. 잠금해제 스크립트는 때로는 그것이 통상적으로 대응하는 서명을 제공한다는 사실을 지칭하는 "scriptSig"라 칭해진다. 그러나, 보다 일반적으로, UTXO가 리딤되기 위한 조건이 서명을 인증하는 것을 포함하는 것이 블록체인(150)의 모든 애플리케이션들에서 필수적인 것은 아니다. 보다 일반적으로 스크립팅 언어는 임의의 하나 이상의 조건들을 정의하는 데 사용될 수 있다. 따라서 보다 일반적인 용어들 "잠금 스크립트" 및 "잠금해제 스크립트"가 선호될 수 있다.

[0391] 4.3. 사이드 채널

[0392] 도 1에 도시된 바와 같이, 앨리스 및 밥의 컴퓨터 장비(102a, 102b) 각각 상의 클라이언트 애플리케이션은 각각 추가적인 통신 기능성을 포함할 수 있다. 즉, 추가적인 기능성은 (어느 한 당사자 또는 제3자의 주도로) 앨리스(103a)가 밥(103b)과 별개의 사이드 채널(107)을 설정하는 것을 가능하게 한다. 사이드 채널(107)은 블록체인 네트워크와 별개로 데이터 교환을 가능하게 한다. 이러한 통신을 때로는 "오프-체인(off-chain)" 통신으로서 지칭된다. 예컨대, 이는 당사자들 중 하나가 네트워크(106)로 브로드캐스팅하기로 선택할 때까지, (아직) 트랜잭션이 블록체인 네트워크(106) 상에 등록되거나 체인(150)으로 진행됨 없이 앨리스와 밥 사이에서 트랜잭션(152)을 교환하는 데 사용될 수 있다. 이러한 방식으로 트랜잭션을 공유하는 것은 때로는 "트랜잭션 템플릿" 공유하는 것으로서 지칭된다. 트랜잭션 템플릿은 완전한 트랜잭션을 형성하기 위해 요구되는 하나 이상의 입력들 및/또는 출력들이 없을 수 있다. 대안적으로 또는 부가적으로, 사이드 채널(107)은 키들, 협상된 금액들 또는 조건들, 데이터 콘텐츠 등과 같은 임의의 다른 트랜잭션 관련 데이터를 교환하는 데 사용될 수 있다.

[0393] 사이드 채널(107)은 블록체인 네트워크(106)와 동일한 패킷 교환 네트워크(101)를 통해 설정될 수 있다. 대안적으로 또는 부가적으로, 사이드 채널(301)은 상이한 네트워크 이블테면, 모바일 셀룰러 네트워크, 또는 로컬 영역 네트워크 이블테면, 로컬 무선 네트워크, 또는 심지어, 앨리스 및 밥의 디바이스들(102a, 102b) 사이의 직접 유선 또는 무선 링크를 통해 설정될 수 있다. 일반적으로, 본원의 임의의 위치에서 지칭되는 바와 같은 사이드 채널(107)은 "오프-체인", 즉 블록체인 네트워크(106)와 별개로 데이터를 교환하기 위한 하나 이상의 네트워크 기술들 또는 통신 매체들을 통한 임의의 하나 이상의 링크들을 포함할 수 있다. 하나 초과개의 링크가 사용되는 경우, 오프-체인 링크들의 번들(bundle) 또는 모음은 전체적으로 사이드 채널(107)로서 지칭될 수 있다. 따라서 앨리스 및 밥이 사이드 채널(107)을 통해 특정 정보 조각들 또는 데이터 등을 교환한다고 하면, 이는 이러한 모든 데이터 조각들이 정확히 동일한 링크 또는 심지어 동일한 유형의 네트워크를 통해 전송되어야 한다는 것을 반드시 의미하는 것은 아니란 것에 주의한다.

[0394] 사이드 채널(107)은 앨리스 및 밥과 같은 당사자들 사이의 안전한 개인 오프 체인 통신을 가능하게 하기 위해 알려진 보안 통신 기술들을 채용하는 보안 채널을 포함할 수 있다. 예컨대, 보안 채널은 보안 채널을 통해 통신하는 당사자들 간에 공유되는 공유 비밀에 기초할 수 있다. 이러한 채널은 예컨대, 검증 당사자(103V)와 타겟 당사자(103T) 사이에서 통신하기 위해 이블테면, 검증 당사자(103V)가 타겟 당사자에 의해 홀딩되는 PUF(302/500)에 챌린지를 제출하고 대응하는 응답을 다시 수신하는 것을 가능하게 하기 위해 사용될 수 있다.

[0395] 5.블록체인 기반 PUF 아이덴티티 증명

[0396] 이전 섹션들에서 언급된 바와 같이, 응답의 레코드로서 역할을 하는 응답 데이터는 신뢰할 수 있는 제3자 시스템(602)을 사용하기 보다는, 공개 블록체인 상에 저장될 수 있다. 응답 데이터는 셋업 시에 결정된 데이터이고, 타겟 당사자(103T)("앨리스")에 의한 타겟의 아이덴티티의 어서션(assertion)을 테스트하기 위해 검증 당사자(103V)("밥")에 의해 나중에 사용될 수 있다. 재차, 앨리스 및 밥은 임의적 레이블들일 뿐이며 여기서 앨리스 및 밥은 섹션 4(여기서 밥은 앨리스의 트랜잭션 출력을 지출함)에 주어진 블록체인의 일반적인 개요에서와 동일한 역할을 맡을 필요는 없다는 것에 주의한다.

[0397] 이전에 논의된 바와 같이, 주어진 CR 쌍 $\{C_i, R_i\}$ 에 대한 응답 데이터(체인 상에 저장되든 아니면 다른 곳에 저

장되든지 간에)는, 셋-업 국면(702)에서 결정되고 검증 당사자(103V)에 의한 향후 참조를 위해 저장되는 바와 같은 다음 중 임의의 것을 포함할 수 있다:

- [0398] i) 챌린지 C_i 및/또는 응답 R_i 의 명시적 값(평문 또는 암호화됨), 또는
- [0399] ii) 개개의 응답 R_i 에 대한 특정 챌린지 C_i 가 도출될 수 있는 마스터 챌린지 C_m 에 링크된 응답 R_i 의 명시적 값, 또는
- [0400] iii) 챌린지 C_i 의 명시적 값과 함께 응답 R_i 의 증명(예컨대, 해시 또는 이중 해시), 또는
- [0401] iv) 개개의 응답 R_i 에 대한 특정 챌린지가 도출될 수 있는 마스터 챌린지 C_m 에 링크된 응답 R_i 의 증명(예컨대, 해시 또는 이중 해시), 또는
- [0402] v) 응답 R_i 로부터 도출된 공개-개인 키 쌍의 공개 키.
- [0403] 도 9에 도시된 바와 같이, 어떤 형태를 취하든, 셋-업 국면(702)에서, 이러한 응답 데이터(901)는 블록체인(150) 상에 레코딩된 트랜잭션(152S)의 출력(203)에 저장될 수 있다. 이는 이하, 저장 트랜잭션으로서 지칭될 수 있다. 이는 예컨대, 위의 섹션 4에서 논의된 기술들을 사용하여 체인 상에 레코딩될 수 있고, 해당 섹션에서 앨리스가 반드시 타겟 당사자(103T)일 필요는 없고 해당 섹션에서 밥이 반드시 검증 당사자(103V)일 필요는 없음에 다시 한 번 주의하고 — 실제로 현재 앨리스로서 지칭되는 타겟 당사자(103T)는 체인 상에 레코딩될 저장 트랜잭션(152S)을 공식화하고 전송하는 당사자일 수 있다. 다른 예로서, 신뢰할 수 있는 제3자는 셋-업 시에 생성된 응답 데이터(901)를 포함함으로써 타겟 당사자(103T)가 완료하고 그 후 체인 상에 레코딩되도록 포워딩하기 위한 저장 트랜잭션의 템플릿을 공식화할 수 있다. 타겟 부분(103T)은 블록체인 네트워크(106)를 통해 전파되도록 블록체인 노드들(104) 중 하나에 저장 트랜잭션(152S)을 직접 전송할 수 있거나, 신뢰할 수 있는 제3자와 같은 다른 당사자를 통해 이를 간접적으로 전송할 수 있다. 또 다른 예로서, 타겟 당사자(103T)는 신뢰할 수 있는 제3자가 저장 트랜잭션(152)으로 공식화하고 체인 상에 레코딩되도록 전송하기 위해, 신뢰할 수 있는 제3자에게 자신의 응답 데이터(901)를 전송할 수 있다.
- [0404] 응답 데이터(901)는 저장 트랜잭션(152S)의 지출 불가능한 출력에 저장될 수 있다. 예컨대, 이는, 스크립트 프로토콜을 사용하는 경우, OP_RETURN, 또는 OP_FALSE 및 그에 뒤따르는 OP_RETURN을 사용하여 지출 불가능하게 만들어질 수 있다(BTC 또는 BCH와 같은 일부 블록체인 프로토콜들에서, OP_RETURN의 임의의 포함은 출력을 지출 불가능하게 만드는 반면, BSV와 같은 다른 프로토콜들에서, OP_FALSE와 OP_RETURN 둘 모두가 출력을 지출 불가능하게 만드는 데 요구됨). BTC(Bitcoin), BTC(Bitcoin Cash) 및 BSV(Bitcoin Satoshi Vision)는 앞서 설명된 블록체인 시스템의 상이한 예시적인 구현들이다.
- [0405] 대안적으로, 응답 데이터(901)는 저장 트랜잭션(152S)의 지출 가능한 출력에 매립될 수 있다. 예컨대, 이는 OP_FALSE 없이 OP_RETURN을 포함함으로써 지출 가능하게 유지될 수 있다. 다른 예로서, OP_DROP 코드 바로 앞에 데이터를 포함함으로써 지출 가능한 잠금 스크립트에 그 데이터를 매립할 수 있다. 이는 BTC, BCH 및 BSV에 동일하게 적용될 것이다.
- [0406] 실시예들에서 주어진 타겟 당사자(103T)의 다수의 상이한 CR 쌍들 $\{C_i, R_i\}$ 의 세트의 응답 데이터(901)가 저장될 수 있다. 이들은 저장 트랜잭션(152)의 동일한 출력(203) 또는 상이한 출력들(203)에 저장될 수 있거나, 동일한 출력에 일부 및 상이한 출력에 일부의 조합으로 저장될 수 있다. 그들은 동일한 저장 트랜잭션(152S)에 저장될 수 있거나, 상이한 CR 쌍들의 응답 데이터(901)가 상이한 저장 트랜잭션들(152S)에 저장될 수 있거나, 동일한 트랜잭션에 일부 및 상이한 트랜잭션에 일부의 조합으로 저장될 수 있다.
- [0407] 온-체인 저장은 반드시 계정-기반 모델로 제한되는 것은 아니란 것에 주의한다. 대안적인 전개들에서, 응답 데이터(901)는 계정-기반 모델의 하나 이상의 트랜잭션들의 하나 이상의 스마트 계약들에 저장될 수 있다.
- [0408] 검증 국면(704)에서, 검증 당사자(103V)가 타겟의 아이덴티티를 검증하고자 할 때, 그/그녀는 저장 트랜잭션(152S)으로부터 특정 CR 쌍 하나에 대응하는 응답 데이터(901)를 획득하기 위해 블록체인(150)에 액세스한다. 실시예들에서, 이것은 검증 당사자(103V)에게 특정 챌린지 C_i 에 대응하는 응답 R_i 또는 해당 응답 R_i 의 증명(예컨대, 해시 또는 이중 해시)을 제공한다. 검증 당사자(103V)는 또한 타겟 당사자(103V)에 챌린지 C_i 를 제출하고, 이에 응답하여, 수신된 챌린지 C_i 를 PUF 모듈(603)에 입력함으로써 타겟 당사자(103T)(또는 그들의 디바이스)가 생성하는(알려진) 응답 R'_i 를 다시 수신한다. 그 후 검증 당사자(103V)는 리턴된 응답 R'_i 를 체인 상의 저장 트랜잭션(152S)으로부터 리트리브된 버전과 비교하거나, 증명에 사용된 수신된 응답에 동일한 변환(예컨대, $H(R'_i)$ 또는 $H^2(R'_i)$)을 적용하고 이를 체인 상의 저장 트랜잭션(152S)으로부터 리트리브된 증명

과 비교한다. 어느 쪽이든, 비교가 매칭을 제공하는 경우, 타겟은 검증된다.

- [0409] 검증 당사자(103V)는 블록체인 네트워크(106)의 노드들(104) 중 하나를 통해, 또는 대안적으로 블록체인에 응답 데이터(즉, 트랜잭션)가 포함되었다는 머클 증명을 또한 제공할 수 있는 임의의 외부 당사자로부터 응답 데이터를 획득함으로써 블록체인(150)에 액세스할 수 있다.
- [0410] 응답 데이터(901)가 블록체인(150)과 같은 공개 매체에 저장되는 실시예들에서, 실제 응답 값 R_i 자체가 공개적으로 또는 제한 없이 공개되지 않는 것이 바람직할 수 있다. 그렇지 않으면, 임의의 악의적인 당사자는 체인 상에서 R_i 를 본 다음 C_i 로 쉼진한 경우 타겟 당사자(103T)인 것처럼 가장할 수 있다. 이에 따라, 대신에 체인 상에 홀딩되는 응답 데이터(901)로서 R_i 의 증명(예컨대, $H(R_i)$ 또는 $H^2(R_i)$)만을 저장하거나, 아니면 R_i 의 명시적 값을 암호화된 형태로 저장하는 것이 바람직할 수 있다. 또는, 일부 경우들에서, 증명이 암호화된 형태로 체인 상에 저장될 수 있다.
- [0411] 잠재적으로 다수의 검증 당사자들이 있는 경우에, R_i 또는 그의 증명을 암호화된 형태로 저장하는 것은 타겟 당사자(103T) 또는 신뢰할 수 있는 제3자가, 어떤 검증 당사자들(103V)이 CR 쌍들 중 어느 것에 대응하는 저장 데이터(901)를 획득할 수 있는지를 제어하는 것을 가능하게 한다. 이것은 특정 응답 데이터(901) 조각에 대한 복호화 키를 주어진 검증 당사자에게만 제공하고 다른 응답 데이터(901) 조각에 대한 복호화 키를 다른 검증 당사자에게만 제공함으로써 달성될 수 있다. 복호화 키들의 분배는 타겟 당사자(103T) 또는 신뢰할 수 있는 제3자에 의해 관리될 수 있다. 각각의 검증 당사자 또는 검증 당사자들의 서브세트에는 응답 데이터(901)(예컨대, CR 쌍) 조각들의 개개의 서브세트에 액세스하기 위한 하나 이상의 복호화 키들의 자체 서브세트가 제공된다. 바람직하게는 서브세트들은 서로 배타적이다. 그러나 다른 구현들에서, 서브세트들은 오버랩할 수 있다(예컨대, 동일한 조직 내의 상이한 그룹들은 CR 쌍의 오버랩하는 서브세트들에 대한 액세스를 가질 수 있음).
- [0412] 이에 대한 변형으로서, 응답 데이터(901)(예컨대, CR 쌍들)가 온-체인 대신 제3자 시스템(602)에 저장되는 경우, 복호화 키들을 분배하는 대신(또는 그에 더하여), 각각의 검증 당사자가 CR 쌍들(또는 더 일반적으로 응답 데이터)의 자체 서브세트에만 액세스할 수 있도록 보장하기 위한 다른 수단이 사용될 수 있다. 예컨대, 신뢰할 수 있는 제3자 시스템(602)은 각각의 검증 당사자에 대해 패스워드로 보호된 계정을 유지할 수 있으며, 그들은 자신의 쉼진(들)에 대한 액세스를 획득하도록 로그인하기 위해 패스워드로 보호되는 계정이 요구되고 이것은 자체 CR 쌍(들)에 대한 액세스만을 그들에게 제공한다.
- [0413] 이러한 방식들은 보안에 유리할 수 있다. 주어진 CR 쌍의 응답 R_i 가 하나의 검증 당사자(103V)에게 공개되어야 하는 경우, 동일한 CR 쌍이 다른 검증 당사자(103V)에 대해 사용되지 않는 것이 바람직할 수 있다. 그렇지 않으면, 제1 검증 당사자(103V)는 그/그녀가 타겟 당사자(103T)인 다른 검증 당사자인 것처럼 가장하기 위해 지금-알려진 응답 R_i 를 사용할 수 있다. 그러나 응답 데이터(901)에 대한 액세스를 갖는 모든 잠재적 검증 당사자들(103V)이 신뢰되는 경우, 이를 방지하기 위한 조치를 취하는 것이 필수적이진 않다.
- [0414] 추가 변형들에서, 체인 상에 저장된 응답 데이터(901)는 대응하는 응답 R_i 에 기초하여(예컨대, 이를 시드로서 사용함) 셋-업 시 생성된 공개-개인 키 쌍의 공개 키인 타겟 당사자(103T)의 공개 키의 형태를 취할 수 있다. 이 경우에, 검증 당사자(103V)는 저장 트랜잭션(152S)으로부터 공개 키에 액세스하고 이를 사용하여 타겟 당사자(103T)에 의해 서명된 메시지를 대응하는 개인 키로 검증한다. 일부 경우들에서, 공개 키들은 암호화된 형태로 체인 상에 저장될 수 있어서, 상이한 검증 당사자들(103V)에 의한 사용을 위해 상이한 공개 키들이 할당될 수 있다.
- [0415] 도 9에 또한 도시된 바와 같이, 출력(예컨대, UTXO-기반) 모델을 사용하는 실시예들에서, 이것은 CR 쌍들(또는 그로부터 도출된 키들)을 관리하기 위한 효율적인 메커니즘을 제공하기 위해 이용될 수 있다. 여기에서 관리되는 예컨대, 일단 CR 쌍 또는 키가 이미 소비된(검증 시에 사용된) 경우, CR 쌍 또는 키를 업데이트하거나 폐지하는 것을 포함할 수 있다.
- [0416] 이를 위해, 새로운 수정자 트랜잭션(152M)이 블록체인(150) 상에 레코딩된다. 이는 폐지되거나 업데이트될 응답 데이터(901)의 조각이 저장되는 저장 트랜잭션(152S)의 출력들(203) 중 하나를 가리키는 입력(202)을 갖는다. 이것은 "지출", "리덤" 또는 "할당"으로서 지칭될 수 있다(그러나 이것이 반드시 금전적 가치의 이전을 암시하는 것은 아니란 것에 주의함). 이것은 검증 당사자(103V)에 의해 인식되는 계층-2 프로토콜의 레벨에서 가리키는 저장 트랜잭션(152S) 또는 출력(203)의 응답 데이터(901)가 더 이상 사용되지 않는다는 것을 의미하는 것으로 이해된다. 수정자 트랜잭션(152M) 자체가 자체 출력들 중 하나에 응답 데이터(901')를 포함하는

경우, 이것은 새로운 응답 데이터(901')가 이전 응답 데이터(901)(예컨대, 새로운 CR 쌍)의 대체를 표현한다는 것을 표현하는 것으로 간주된다. 검증 당사자가 검증 동작에 사용할 응답 데이터를 찾기 위해 블록체인(150)에 액세스하는 경우, 검증 당사자는 교체된 버전이 아닌 업데이트된 버전(901')을 사용할 것이다. 반면에, 수정자 트랜잭션(152U)이 교체 응답 데이터(901)를 포함하지 않는 경우, 이는 자신이 가리키는 출력(203) 또는 저장 트랜잭션(152S) 내 응답 데이터(901)를 단순히 폐지하는 것으로 간주된다.

[0417] 일부 실시예들에서, 응답 데이터(901)는 저장 트랜잭션(152S)의 지출 가능한 출력에 매립되고, 응답 데이터(901)(예컨대, CR 쌍)가 저장된 특정 출력(203)을 지출(즉, 할당 또는 리덤)함으로써 폐지되거나 업데이트될 수 있다. 이러한 일부 실시예들에서, 상이한 CR 쌍들에 대응하는 응답 데이터(901)의 상이한 조각들은 동일한 저장 트랜잭션(203)의 개별 출력들(203)에 저장되고 개별적으로 폐지되거나 업데이트될 수 있다.

[0418] 다른 실시예들에서, 응답 데이터(901)는 저장 트랜잭션(152S)의 지출 불가능한 출력에 저장되고, 저장 트랜잭션(152S)의 상이한 지출 가능한 출력을 지출(즉, 할당 또는 리덤)함으로써 폐지되거나 업데이트될 수 있다. 이러한 일부 실시예들에서, 다수의 상이한 CR 쌍들(동일하거나 상이한 지출 불가능한 출력들에 저장됨)에 대응하는 저장 데이터(901)의 다수의 조각들은 동일한 트랜잭션(152S)의 동일한 지출 가능한 출력을 지출함으로써 폐지되거나 업데이트될 수 있다.

[0419] 예시적인 사용 경우로서, CR 쌍에 대응하는 응답 데이터(901)의 조각의 레코드는 일단 소비되면, 즉 검증 시에 사용되면 폐지되거나 업데이트될 수 있다. 이는 응답 데이터(901)가 Ri의 명시적 레코드이든, 증명이든, 아니면 Ri로부터 도출된 공개 키든지 간에 적용될 수 있다. 어느 쪽이든, 이는 보안 이유들로 유리할 수 있어서, 이제 세상에 릴리스된 응답 데이터는 더 이상 재사용 가능하지 않다.

[0420] 수정자 트랜잭션(152M)은 타겟 당사자(103T)에 의해 체인 상에 레코딩되도록 공식화 및 전송될 수 있다. 그것은 전파를 위해 블록체인 노드(104)에 직접 전송되거나 중간 당사자를 통해 노드(104)에 간접적으로 전송될 수 있다. 대안적으로, 신뢰할 수 있는 제3자는 타겟 당사자가 (예컨대, 대체 응답 데이터(901'))에 서명 및/또는 이를 추가함으로써 완료하기 위한 템플릿 트랜잭션을 전송하고 그 후 체인 상에 레코딩되도록 직접 또는 간접적으로 노드(104)로 포워딩할 수 있다. 다른 가능성으로서, 신뢰할 수 있는 제3자는 수정자 트랜잭션(152M)을 공식화할 수 있고(아마도, 타겟 부분(103T)으로부터 전송된 일부 데이터 또는 템플릿에 기초하여, 이는 예컨대, 대체 응답 데이터(901'))를 포함함), 그 후 신뢰할 수 있는 제3자는 체인 상에 레코딩되도록 노드(104)로 수정자 트랜잭션(152M)을 포워딩할 수 있다. 이러한 모든 옵션들은 또한 저장 트랜잭션(152S)이 블록체인(150) 상에 레코딩되는 방식에도 적용될 수 있다는 것에 주의한다.

[0421] 따라서, 위에서 논의된 다양한 개념들에 따르면, i) 아이덴티티(또는 공개 키와 같은 다른 관련된 정보)를 UTXO에 링크하고 이 UTXO의 지출 상태를 아이덴티티 크리덴셜들의 유효성에 대한 프로ksi로서 사용하고; 및 ii) 셋업, 폐지, 업데이트 및 검증과 같은 효율적인 아이덴티티-관리 동작들을 수행하기 위해 트랜잭션들의 세트를 설정하기 위한 시스템이 제공된다. 모든 당사자들이 항상 서로 통신할 필요가 있기보단, 모든 당사자들이 블록체인을 참고하여 CRP들이 소비되거나 아이덴티티들이 폐지되는 때를 확인할 수 있으므로, 통신들의 수가 감소된다는 점에서 프로세스가 보다 효율적이다.

[0422] 이러한 기술들은 예컨대, 검증에 사용되는 CRP 데이터를 처리하기 위해 제3자 KYC(know your customer) 제공자에 대한 의존도를 최소화함으로써 이전에 제시된 바와 같이 아이덴티티를 PUF 디바이스에 링크하기 위한 프레임워크를 확장하는 데 사용될 수 있다. 이 목표는 KYC 제공자의 역할 또는 기능들 중 상당한 일부를 공개 블록체인으로 부분적으로 대체함으로써 달성될 수 있고, 이를 통해 사용자는 임의의 제3자와 독립적으로, ePUF 디바이스와 관련된 그 자신의 아이덴티티 크리덴셜들을 인스턴스화할 수 있다.

[0423] 아이덴티티 시스템에서 신뢰할 수 있는 제3자의 역할이 반드시 완전히 회피되는 것은 아니라, 어느 쪽이든, 아이덴티티 관리의 프로세스가 개선되어서, 프로세스에 대한 신뢰할 수 있는 제3자의 참여 및 이들에게 부과되는 관련 부담이 적어도 감소될 수 있다.

[0424] 5.1. UTXO 세트에 대한 PUF 아이덴티티의 링크

[0425] 이전 섹션들에서 논의된 것들과 같은 아이덴티티 시스템들 상에서 블록체인 사용이 개선될 수 있게 하는 제1 양상은 공개 블록체인의 미지출 트랜잭션 출력 세트(UTXO 세트)를 사용하여 PUF 아이덴티티와 관련된 CRP들을 관리하는 것이다.

[0426] 이 섹션에서, CRP들을 UTXO 세트의 멤버들에 매핑하고 각각의 특정 CRP가 아이덴티티 검증 프로세스에서 소비되었는지에 관한 표시로서 '지출' 또는 '미지출' 상태로서 그의 상태를 사용하기 위한 2개의 별개의 예시적인 메

커니즘들이 개시된다. 제1 메커니즘은 지출 가능한 UTXO들에 CRP 데이터를 매핑하는 것을 수반하며, 제2 메커니즘은 CRP 데이터를 지출 가능한 UTXO들에 페어링하는 것을 수반한다. 어느 경우든, CRP들과 관련된 추가적인 데이터 또는 해당 아이덴티티가 또한 선택적으로 시스템에 포함될 수 있다.

- [0427] **5.1.1. 지출 가능한 UTXO들에의 매핑:** 제1 메커니즘은 CRP들을 지출 가능한 UTXO들에 바인딩하는 것이며, 이 지출 가능한 UTXO들은 조건들은 향후 입력들에 의해 충족될 수 있고 이에 따라 향후 지출 트랜잭션들에 의해 소비될 수 있는 스크립트들을 포함하는 트랜잭션 출력이다.
- [0428] 이러한 매핑을 구현하기 위한 다수의 상이한 옵션들이 있지만, 우리의 목적을 위해, 이는 일반적으로 적어도 다음 잠금 스크립트로 구성될 것이다:
- [0429] $[Checksig\ P]\ OP_RETURN\ \langle Rep(C,R) \rangle$
- [0430] 여기서 $[Checksig\ P]$ 는 표준 P2PKH(pay-to-public-key-hash) 잠금 스크립트이고, $Rep(C,R)$ 은 특정 챌린지-응답 쌍(C,R)의 표현이다.
- [0431] 이 잠금 스크립트는 지출 트랜잭션에 대해 유효한 서명 Sig P를 제공함으로써 간단하게 잠금해제될 수 있으며, 여기서 서명은 공개 키 P에 대해 유효한 것으로 간주된다. 작업 코드 OP_RETURN을 뒤따르는 임의의 데이터는 지출 트랜잭션을 유효성 검증할 때 고려되지 않을 것이고 이에 따라, 이 데이터는 블록체인 유효성 검증기(blockchain validator)들과 관련하여 임의적이고 포맷이 지정되지 않은 것으로 취급될 수 있다는 것이 주의되어야 한다.
- [0432] 위 스크립트에서 OP_RETURN 코드를 뒤따르는 데이터는 챌린지-응답 쌍(C,R)의 표현 $Rep(C,R)$ 이다. 이 표현은 해당 사용 경우에 의존하여, 다양한 방식으로 이루어질 수 있다. 그러나 합리적인 예는 PUF를 소유한 증명자 앨리스에게만 알려진 키 k를 사용하여 CRP를 암호화하는 것일 것이다. 이 경우에, 우리는 다음 표현들 중 임의의 것을 가질 수 있다:
- [0433] $Rep(C,R) = Encrypt(C,k)$,
- [0434] $Rep(C,R) = Encrypt(R,k)$,
- [0435] $Rep(C,R) = Encrypt(C \parallel R,k)$.
- [0436] 이러한 표현들은 앨리스가 나중에 그녀의 UTXO에 포함된 챌린지, 응답 또는 CRP를 각각 리트리브하거나 증명하는 것을 허용할 것이다.
- [0437] **부가적인 스크립트 부담들:** 향후 출력을 지출하는 입력 스크립트 상에 부가적인 조건들을 포함하도록 이전에 보여진 기본 잠금 스크립트를 확장하는 것이 가능하다. 이러한 추가 조건의 합리적인 예는 다음 스크립트일 것이다:
- [0438] $[Checksig\ P][Hash\ Puzzle\ H^2(R)]\ OP_RETURN\ \langle Rep(C,R) \rangle$
- [0439] 여기서 $[Hash\ Puzzle\ H^2(R)] = OP_HASH160\ \langle H(R) \rangle\ OP_EQUAL$ 이다. 다른 해시 함수 작업 코드들이 사용할 수 있다는 것에 주의한다. 이 수정된 스크립트는 이제 지출자가 공개 키 P에 대한 유효한 서명을 제공하는 것 외에도, 챌린지 R의 해시를 드러내도록 요구한다. 여기서의 아이디어는, 이것이 일부 시나리오들에서, 지출자가 문제 R의 챌린지와 결국 관련되는 정보 $H(R)$ 를 알고 있다는 지식 증명으로 사용될 수 있다는 것이다.
- [0440] **트랜잭션 모델들:** 사용될 트랜잭션 잠금 스크립트들의 정확한 구조가 결정되면, CRP들을 저장, 증명 및 관리하는 방식으로 이러한 스크립트들을 포함하는 트랜잭션들을 구조화하는 방법에 관한 선택이 이루어질 수 있다.
- [0441] CRP들 및 연관된 잠금 스크립트들을 일대일 방식으로 UTXO들에 매핑하는 것이 본원에서 개시된다. 즉, 이러한 스크립트를 포함하는 모든 각각의 UTXO는 특정 PUF 디바이스와 관련된 정확히 하나의 CRP에 대응할 것이다.
- [0442] 그 후, 이러한 UTXO들을 트랜잭션들로 구성하는 방법에 대한 몇 가지 옵션들이 있다. 가장 가능성이 높은 가능한 옵션은 다음과 같다:

- [0443] 1. 트랜잭션당 하나의 CRP.
- [0444] 2. 트랜잭션당 하나의 CRP 세트.
- [0445] 3. 트랜잭션당 하나의 PUF.

[0446] 제1 옵션은 이를테면, 유언(a will)을 업데이트하기 위해 매우 드물게 사용되는 PUF들의 경우와 같은 일부 경우들에서 적용 가능할 수 있으며, 다수의 CRP들이 서로 명확하게 링크되지 않는다는 이점을 갖는다. 이것은 하나의 CRP의 소비 및 드러냄(reveal)이 임의의 다른 것들과 독립적으로 드러날 수 있으므로 극도의 프라이버시가 요구되는 상황들에서 또한 유용할 수 있다.

[0447] 아래 표 1의 트랜잭션은 제1 옵션의 예시적인 구현이다. 트랜잭션은 단일 입력 및 출력만을 포함하고, 이에 따라 각각의 CRP는 상이한 트랜잭션에 포함될 것임을 알 수 있다. 그의 출력이 지출될 때, 감사 목적들을 위해서가 아니라, 아이덴티티 시스템에 대한 이 트랜잭션의 관련성이 사실상 종료된다.

$TxID_{CRP-Single}$			
입력		출력	
아웃포인트	스크립트	값	스크립트
$TxID_{prev}$	$< SIG_A > < P_A >$	x BSV	$[Checksig P_A^1] OP_RETURN <Rep(C_1, R_1)>$

[0448]

[0449] 표 1: 단일 트랜잭션에서 UTXO에 매핑된 단일 CRP.

[0450] 다수의 CRP들이 단일 트랜잭션에서 개개의 UTXO에 각각 매핑되게 하는 제2 옵션은, CRP 소비의 예상된 빈도가 상당히 더 높은 은행 카드들과 같은 사용 경우에 대해 더 바람직할 수 있다. 아래 표 2의 트랜잭션은 이것이 달성될 수 있는 방법을 보여준다.

[0451] 앨리스에 의해 생성되었을 가능성이 높은 입력 서명은 출력들의 전체 세트에 대해 서명하도록 만들어질 수 있다는 것에 주의한다. 이는 하나의 공개 키 P_A 로부터 다수의 UTXO들 및 이에 따라, 다수의 CRP들로의 일대다 링크를 제공하는 동시에, 별개의 CRP들 자체에 대한 UTXO들의 일대일 매핑을 유지한다. 또한, 각각의 출력/CRP는 재사용을 회피하기 위해 그 자신의 연관된 공개 키(모두가 앨리스에 의해 소유됨)를 갖는다고 가정된다.

$TxID_{CRP-Set}$			
입력		출력	
아웃포인트	스크립트	값	스크립트
$TxID_{prev}$	$< SIG_A > < P_A >$	x BSV	$[Checksig P_A^1] OP_RETURN <Rep(C_1, R_1)>$
		x BSV	$[Checksig P_A^2] OP_RETURN <Rep(C_2, R_2)>$
		x BSV	$[Checksig P_A^3] OP_RETURN <Rep(C_3, R_3)>$
		...	
		x BSV	$[Checksig P_A^n] OP_RETURN <Rep(C_n, R_n)>$

[0452]

[0453] 표 2: 단일 트랜잭션의 개개의 UTXO들에 매핑된 CRP들의 세트

[0454] 위에 표시된 옵션은 또한 시간이 지남에 따라 CRP 세트들을 업데이트하는 실시예들과 잘 통합될 수 있으며, 여기서 업데이트된 세트가 생성될 때마다, 해당 세트에 대해 새로운 트랜잭션이 발행될 수 있다. 게다가, 병렬 독립적(즉, 온-체인으로 관련되지 않은) 트랜잭션들을 통해 동일한 PUF에 대해 동시에 다수의 상이한 CRP 세트들이 생성 및 발행될 수도 있다. 이것은 다수의 상이한 신뢰할 수 있는 제3자들(예컨대, 상이한 은행들)과 아이덴티티를 설정하는 데 유용할 수 있어서, 아이덴티티들 둘 모두는 독립적으로 설정되지만, 여전히 동일한 PUF에 의해 앵커링된다(anchored).

[0455] 단일 트랜잭션이 단일 PUF를 표현하는 데 사용되는 제3 옵션은 단순히 업데이트들이 가능하지 않은 옵션 2의 보다 제한적인 버전이다. 이것은 PUF-포함 디바이스에 특정 '수명'이 부여된 경우에 대해 적용 가능할 수 있으며, 여기서 이는 사용자에게 새로운 디바이스로 발급받기 전에 미리 결정된 수의 인증들에 대해서만 사용될 수 있다.

[0456] 5.1.2. 지출 가능한 UTXO들과의 페어링

[0457] 지출 가능한 UTXO들 내에 CRP들을 매핑하는 것에 대한 대안은 단순히 이러한 출력들과 CRP들을 페어링하는 것이다. 이 경우에, 디지털 인증서들에 대한 기존 작업과의 차이는 앨리스가 임의의 제3자와 독립적으로 아이덴티티를 증명하기를 원할 수 있을 때, 트랜잭션이 앨리스에 의해 구성되고 서명될 수 있다는 것이다.

$TxID_{CRP-Set}$			
입력		출력	
아웃포인트	스크립트	값	스크립트
$TxID_{Prev}$	$< SIG_A > < P_A >$	x BSV	OP_DUP OP_HASH160 $< P_A^1 >$ OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_0 OP_RETURN $< Rep(C_1, R_1) >$
		x BSV	OP_DUP OP_HASH160 $< P_A^2 >$ OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_0 OP_RETURN $< Rep(C_2, R_2) >$
		...	
		x BSV	OP_DUP OP_HASH160 $< P_A^n >$ OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_0 OP_RETURN $< Rep(C_n, R_n) >$

[0458] 표 3: CRP들에 매핑된 지출 가능한 UTXO들을 포함하는 트랜잭션.
[0459]

[0460] 위의 다이어그램에서 n 개의 CRP들과 관련된 $2n$ 개의 출력들을 포함하는 예시적인 트랜잭션을 볼 수 있으며, 여기서 각각의 지출 가능한 출력은 CRP들 중 하나에 매핑될 수 있으며 CRP 표현 자체는 대응하는 지출 불가능한 출력(예컨대, OP_FALSE OP_RETURN)에 포함되었다. 또한, CRP들을 트랜잭션들 및 UTXO들로 구성하는 3개의 가능한 변형들이 여기에 유사하게 적용된다는 것이 주의되어야 한다.

[0461] 5.1.3. 논의

[0462] **CRP-관리의 이익들:** CRP들을 UTXO들에 매핑하는 개념은 이전 섹션들로부터의 아이덴티티 프로토콜들의 사용자들에 대한 CRP 관리 및 처리를 상당히 개선할 수 있다. 이점은 블록체인 네트워크(106) 및 그로부터 신뢰할 수 있는 리트리벌을 용이하게 할 수 있는 서비스 제공자에 대한 CRP의 저장 및 록업이 부분적으로 오프로딩될 수 있다는 것이다.

[0463] 특정 PUF의 모든 '라이브' CRP들을 UTXO들에 매핑함으로써, 아이덴티티 시스템에서 주어진 PUF에 대해 현재 사용 가능한 CRP들에 관한 정확한 정보를 위해 UTXO 세트의 상태를 질의함으로써 CRP 업데이트 프로세스를 개선하는 것이 가능하다.

[0464] 우리가 설명한 블록체인 및 UTXO-CRP 매핑 규약들을 활용하는 간단한 프로세스의 예는 다음과 같다:

- [0465] 1. 앨리스는 PUF 디바이스를 획득하고 $(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)$ 로서 CRP들의 세트를 열거한다.
- [0466] 2. 앨리스는 표 2에 도시된 바와 같이 트랜잭션 $TxID_{CRP-Set}$ 를 생성하고 블록체인 네트워크에 브로드캐스팅한다.
- [0467] 3. 앨리스는 제3자에게 자신의 아이덴티티를 인증하도록 시간이 지남에 따라 다수의 CRP들을 소비한다.

- [0468] 4. 엘리스는 이제, 다음 주 동안 그녀의 예상되는 활동들을 커버하기에 충분한 CRP들을 갖는지를 체크하기를 원한다.
- [0469] 1. 엘리스는 $TxID_{CRP-Set}$ 의 어떤 UTXO들이 현재 미지출인지를 묻도록 블록체인 노드(104) 또는 SPV-유사 서비스 제공자에게 질의한다.
- [0470] 2. 블록체인 노드 또는 서비스 제공자는 아직 미지출인 트랜잭션 $TxID_{CRP-Set}$ 의 출력들의 수로 응답한다.
- [0471] 5. 리턴된 수가 충분하지 않은 경우, 엘리스는 신뢰할 수 있는 제3자와 아이덴티티 업데이트 프로세스를 생성하거나, 또는 독립적으로 설정된 아이덴티티에 대해 더 많은 CRP들을 단순히 열거할 수 있다. 그렇지 않으면, 엘리스는 어떠한 액션도 취하지 않는다.
- [0472] **매립 대 페어링:** 지출 가능한 출력들 내에 CRP들을 매립할지 아니면 단순히 그들을 출력들과 페어링할지에 관한 선택은 엘리스에게 이러한 경우들을 구별하는 2개의 상이한 이익들 간의 선택을 제공한다.
- [0473] CRP가 지출 가능한 출력들 내에 매립된 경우, 이것은 블록체인 네트워크(106)를 유지하는 블록체인 노드들(104)이 이러한 출력의 데이터를 쉽게 사용 가능하게 유지하도록 장려한다. 이는, 엘리스의 질의들에 대한 응답들이 더 빨라질 수 있다는 것을 의미하며, 더 중요한 것은 블록체인 노드들이 이러한 트랜잭션 출력들의 원시 데이터를 엘리스에게 다시 서빙할 수 있을 가능성이 매우 높다는 것이다.
- [0474] 이전에 논의된 바와 같이, CRP의 표현 $Rep(C,R)$ 이 포함되어서, 쉼민지, 응답 또는 둘 모두의 원시(또는 난독화된) 데이터를 포함하게 되는 경우, 이는 엘리스가 블록체인 네트워크(106)로부터 관련 정보를 리트리브할 수 있을 것임을 의미한다. 이는 엘리스가, 로컬 저장소를 교체하고 블록체인(150)으로 더 경량 시스템을 동작시키도록 허용하는데, 그 이유는 지출 가능한 출력들에의 데이터의 매립은 그녀의 데이터가 아무튼, 높은 가용성을 가질 가능성을 증가시키기 때문이다.
- [0475] 대조적으로, CRP들이 지출 가능한 출력들과만 페어링되는 경우, 엘리스는 자신에 대해 얼마나 많은 CRP들이 이용 가능한지만을 결정할 수 있을 수 있지만, 반드시 비트코인 노드들로부터 표현 데이터 자체를 리트리브할 필요는 없다. 이는 엘리스가 자신의 CRP 세트를 로컬로 유지하지 않는 경우, 블록체인 노드 네트워크(106) 외부의 에이전트를 참고해야 함을 의미할 수 있다.
- [0476] **이중-해시들의 사용:** 위의 예시적인 구현들에서, 이중 해시 $H^2(Data)$ 는 일부 데이터의 온-체인 표현으로서 사용될 수 있음을 보여준다. 이러한 방식으로 이중-해시를 사용하는 이유는, 이것이 단일-해시가 온-체인에서도 드러내는 것을 허용하여, 원칙적으로 당사자가 결국 Data에 연결되는 $H(Data)$ 를 알고 있다는 지식 증명처럼 작용하기 때문이다.
- [0477] 이는 예컨대, 엘리스가 $H(R)$ 을 제공하는 제3자와 R의 실제 값을 공유한다는 점은 감안하면, 제3자에 의해 충족될 수 있는 지출 부담(spending encumbrance)으로서 엘리스에 의해 $H^2(R)$ 가 온-체인으로 레코딩되는 PUF-아이덴티티 상황에서 유용할 수 있다.
- [0478] **다중-당사자 서명:** 이 섹션에 자세히 설명된 트랜잭션들은 PUF 아이덴티티에 대한 엘리스의 증명을 보조하기 위해 다수의 상이한 당사자들로부터 더 많은 서명을 포함할 수 있다는 것이 또한 타당하다. 예컨대, 엘리스 아이덴티티에서 검증자의 신뢰를 향상시키기 위한 방법으로서 엘리스 및 제3자 아이덴티티 제공자 둘 모두가 CRP 트랜잭션들의 입력(들)에 서명하는 것이 바람직할 수 있다. 이는 상대-서명자(counter-signer)가 블록체인 트랜잭션들에 서명하는 데 사용되는 엘리스의 공개 키(들)를 증명할 수 있는 인증 기관인 경우 특히 적절하다. 다수의 서명들(예컨대, '다중-서명')만에 대한 대안으로서, 예컨대, 임계 서명들 또는 키-분할 기술들(예컨대, 샤미르 비밀 공유(Shamir Secret Sharing))을 통해 다수의 당사자들이 서명 프로세스에 포함될 수 있다.
- [0479] **5.2. 트랜잭션들을 이용한 효율적인 아이덴티티-관리**
- [0480] 블록체인이 이전에 제시된 것들과 같은 PUF-기반 아이덴티티 시스템들과 함께 사용될 수 있는 부가적인 방식은 효율적인 수단으로서 PUF 디바이스에 의해 보호되는 아이덴티티 키 또는 토큰을 폐지하는 것이다.
- [0481] 디지털 인증서 관리에 대해 행해진 이전 작업에서, 인증서들을 온-체인으로 발행 및 폐지하는 것이 알려져 있으며, 대응하는 인증서 검증 프로세스가 이에 동반된다. 엘리스가 자신의 PUF-기반 아이덴티티를 온-체인으로 증명할 때 기꺼이 인증 기관과 협력하는 시나리오를 고려한다. 엘리스가 자신의 아이덴티티에 대한 인증서를 온-

체인으로 등록하는 프로세스는 다음과 같다:

- [0482] 1. CA(인증 기관)는 앨리스의 아이덴티티를 검증한다.
- [0483] 2. CA는 인증서 트랜잭션을 생성한다. 이 트랜잭션에는 다음의 입력들 및 출력들을 갖는다:
- [0484] a. 입력: 공개 키 및 CA의 서명을 포함하는 잠금해제 스크립트를 갖는 CA의 UTXO.
- [0485] b. 출력 1: P2PKH 잠금 스크립트.
- [0486] c. 출력 2: 앨리스의 공개 키를 포함하는 OP_RETURN 출력.
- [0487] 3. 트랜잭션이 브로드캐스팅되고, 일단 채굴되면, CA는 앨리스에게 트랜잭션 ID $TxID_{ctx-pk}$ 를 제공한다.
- [0488] 이 프로세스는 앨리스 및 인증 기관이 CA에 의해 서명된 트랜잭션을 생성하기 위해 협력하는 것으로 끝나며, 이는 앨리스의 공개 키에 대한 인증서를 포함하는 하나의 지출 불가능한 출력, 및 CA가 인증서를 폐지하는 데 사용할 수 있는 인증서와 페어링된 지출 가능한 출력을 포함한다.
- [0489] 본원에서 개시된 실시예들은, 앞서 설명된 방법들 중 하나와 같은, PUF-기반 아이덴티티를 설정하는 방법 및 위의 디지털 인증서들에 대해 약속된 방법의 하이브리드를 사용한다. 여기에서 PUF 아이덴티티 시스템에 추가된 요소는, UTXO를 지출함으로써, 일반적인 신뢰할 수 있는 제3자(CA와 유사)가 CRP 또는 관련된 공개 키를 '폐지'할 수 있는 것이다.
- [0490] 신뢰할 수 있는 제3자가 앨리스의 공개 키에 대한 인증서를 폐지하는 경우는 앞서 논의된 암호화 아이덴티티 설정과 관련된다.
- [0491] 체인 상에 저장되거나 증명된 CR-쌍들(CRP들)의 경우, 본원에서 개시된 실시예들은 인증 프로세스에서 일단 CRP들이 사용되면 신뢰할 수 있는 제3자가 CRP들을 폐지하도록 허용하는 체계를 제공한다. 예시적인 방법은 다음과 같다:
- [0492] 1. 앨리스 및 신뢰할 수 있는 제3자는 (예컨대, 앞서 설명된 바와 같은) 아이덴티티 셋업 프로토콜을 실행한다.
- [0493] 2. 앨리스 및 신뢰할 수 있는 제3자는 이제 블록체인을 사용하여 1에서 생성된 CRP들 또는 이제 1단계 후에 획득 가능한 CRP들을 관리하고자 한다:
- [0494] a. 앨리스는 CRP들을 트랜잭션 출력들에 매핑하는 CRP 매핑 트랜잭션 $TxID_{(CRP-Set)}$ 을 생성한다. 이는 아래 표 4에서 보여진다.
- [0495] b. 앨리스 및 신뢰할 수 있는 제3자 둘 모두는 $TxID_{(CRP-Set)}$ 에 서명한다.
- [0496] 3. CRP 매핑 트랜잭션 $TxID_{(CRP-Set)}$ 은 브로드캐스팅되고 블록체인 블록에 공개된다.

$TxID_{CRP-Set}$			
입력		출력	
아웃포인트	스크립트	값	스크립트
$TxID_{prev}$	$< SIG_A > < P_A >$ $< SIG_{TTP} > < P_{TTP} >$	x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_1, R_1)>$
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_2, R_2)>$
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_3, R_3)>$
		...	
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_n, R_n)>$

- [0498] 표 4: 신뢰할 수 있는 제3자에 의한 CRP 폐지/소비를 허용하는 CRP 매핑 트랜잭션.
- [0499] 이 프로세스에서 생성된 매핑 트랜잭션은 위의 표 4에서 보여진다. 이는 앞서 표 2에서 보여진 CRP 매핑 트랜잭션과 매우 유사하지만, 신뢰할 수 있는 제3자 및 앨리스 둘 모두가 입력에 서명하고 CRP들에 매핑된 UTXO들 각각은 향후 트랜잭션에서 지출함으로써 신뢰할 수 있는 제3자에 의해 폐지될 수 있다는 차이가 있다.
- [0500] 이는 직접적인 통신 없이 CRP들의 폐지가 처리되도록 허용하고, TTP가 사용자를 대신하여 폐지를 수행할 수 있으며, 이는 시스템에서 앨리스의 부담을 추가로 감소시키고 그녀의 아이덴티티 관리가 훨씬 더 경량이 될 수 있기 때문에 유리하다.
- [0501] 6. 결론
- [0502] 개시된 기술들의 다른 변형들 또는 사용 사례들은 본원에서의 개시가 주어지면 당업자에게 명백해질 수 있다. 본 개시내용의 범위는 설명된 실시예에 의해 제한되는 것이 아니라 첨부된 청구들에 의해서만 제한된다.
- [0503] 예컨대, 위의 일부 실시예들은 비트코인 네트워크(106), 비트코인 블록체인(150) 및 비트코인 노드(104) 관점에서 설명되었다. 그러나 비트코인 블록체인은 블록체인(150)의 하나의 특정 예이며 위의 설명은 모든 블록체인에 일반적으로 적용될 수 있다는 것이 인지될 것이다. 즉, 본 발명은 결코 비트코인 블록체인으로 제한되지 않는다. 보다 일반적으로, 비트코인 네트워크(106), 비트코인 블록체인(150) 및 비트코인 노드(104)에 대한 위의 임의의 참조는 각각 블록체인 네트워크(106), 블록체인(150) 및 블록체인 노드(104)에 대한 참조로 대체될 수 있다. 블록체인, 블록체인 네트워크 및/또는 블록체인 노드는 위에서 설명된 바와 같이 비트코인 블록체인(150), 비트코인 네트워크(106) 및 비트코인 노드(104)의 설명된 성질들 중 일부 또는 전부를 공유할 수 있다.
- [0504] 본 발명의 바람직한 실시예들에서, 블록체인 네트워크(106)는 비트코인 네트워크이고 비트코인 노드들(104)은 블록체인(150)의 블록들(151)을 생성, 공개, 전파 및 저장하는 설명된 기능들 모두를 적어도 수행한다. 이러한 기능들의 전부는 아니지만 하나 또는 일부만을 수행하는 다른 네트워크 엔티티들(또는 네트워크 요소들)이 있을 수 있음을 배제되지 않는다. 즉, 네트워크 엔티티는 블록들을 생성 및 공개하지 않고 블록들을 전파 및/또는 저장하는 기능을 수행할 수 있다(이러한 엔티티들은 바람직한 비트코인 네트워크(106)의 노드들로 간주되지 않음을 상기한다).
- [0505] 본 발명의 다른 실시예들에서, 블록체인 네트워크(106)는 비트코인 네트워크가 아닐 수 있다. 이러한 실시예들에서, 노드가 블록체인(150)의 블록들(151)을 생성, 발행, 전파 및 저장하는 기능들 전부는 아니지만 적어도 하나 또는 일부를 수행할 수 있다는 것이 배제되지 않는다. 예컨대, 이러한 다른 블록체인 네트워크들에서, "노드"는 블록들(151)을 생성 및 공개하지만 해당 블록들(151)을 저장하고 그리고/또는 다른 노드에 전파하지 않도록 구성된 네트워크 엔티티를 지칭하는 데 사용될 수 있다.
- [0506] 보다 더 일반적으로, 위의 "비트코인 노드"(104)라는 용어에 대한 임의의 참조는 "네트워크 엔티티" 또는 "네트워크 요소"라는 용어로 대체될 수 있으며, 이러한 엔티티/요소는 블록들을 생성, 공개, 전파 및 저장하는 역할들의 일부 또는 전부를 수행하도록 구성된다. 이러한 네트워크 엔티티/요소의 기능들은 블록체인 노드(104)를 참조하여 위에서 설명된 동일 방식으로 하드웨어로 구현될 수 있다.
- [0507] 위의 실시예들은 단지 예로서만 설명되었다는 것이 인지될 것이다. 보다 일반적으로, 다음 스테이트먼트들 중 임의의 하나 이상에 따른 방법, 장치 또는 프로그램이 제공될 수 있다.
- [0508] 스테이트먼트 1. 하나 이상의 검증 당사자가 타겟 당사자 또는 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위한 컴퓨터 구현 방법으로서, 방법은, 셋-업 국면에서, 하나 이상의 검증 당사자 이외의 셋-업 당사자에 의해: PUF(physically unclonable function)에 기초하여 각각의 하나 이상의 응답 세트를 생성하기 위해 PUF를 포함하는 PUF 모듈에 하나 이상의 챌린지 세트를 입력하는 단계; 및 하나 이상의 검증 당사자 중 임의의 검증 당사자 또는 타겟 당사자의 임의의 장비 외부에 있는 데이터 저장소에 응답 세트의 각각의 응답에 대한 개개의 응답 데이터 레코드를 저장하는 단계를 포함하고, 데이터 저장소는 제3자 컴퓨터 장비의 일부이거나 공개 피어-투-피어 공개 매체이고, 이로써 후속 검증 국면에서 타겟의 아이덴티티를 검증하도록 하나 이상의 검증 당사자에게 응답 데이터 레코드를 이용 가능하게 만들고, 각각의 응답에 대한 응답 데이터 레코드는 개개의 응답 또는 개개의 응답으로부터 도출된 데이터를 포함한다.
- [0509] 스테이트먼트 2. 스테이트먼트 1의 방법에 있어서, 셋-업 당사자는 타겟 당사자이고, 방법은 타겟 당사자의 컴퓨터 장비를 이용하여 타겟 당사자에 의해 수행된다.

- [0510] 스테이트먼트 3. 스테이트먼트 2의 방법에 있어서, PUF 모듈은 타겟 당사자의 컴퓨터 장비에 통합된다.
- [0511] 스테이트먼트 4. 스테이트먼트 2의 방법에 있어서, PUF 모듈은 타겟 당사자의 컴퓨터 장비에 대한 외부 주변기기이다.
- [0512] 스테이트먼트 5. 스테이트먼트 2 내지 스테이트먼트 4 중 어느 하나의 방법에 있어서, 데이터 저장소는 제3자 컴퓨터 장비의 일부이고, 타겟 당사자는 제3자로부터 챌린지 세트를 수신하고, PUF 모듈에 입력된 챌린지 세트는 제3자로부터 수신된 세트이다.
- [0513] 스테이트먼트 6. 스테이트먼트 1의 방법에 있어서, 셋-업 당사자는 신뢰할 수 있는 제3자이다.
- [0514] 스테이트먼트 7. 스테이트먼트 6의 방법에 있어서, 데이터 저장소는 제3자 컴퓨터 장비의 일부이고, 방법은 제3자 컴퓨터 장비를 사용하여 신뢰할 수 있는 제3자에 의해 수행된다.
- [0515] 스테이트먼트 8. 스테이트먼트 7의 방법에 있어서, 제3자 컴퓨터 장비는 하나 이상의 지리적 사이트에서 하나 이상의 서버 유닛을 포함하는 서버 장비를 포함한다.
- [0516] 스테이트먼트 9. 스테이트먼트 7 또는 스테이트먼트 8의 방법에 있어서, PUF 모듈은 제3자 컴퓨터 장비의 외부 주변기기 또는 착탈식 구성요소이고, 방법은, 타겟 당사자가 검증 국면에서 검증 당사자로부터의 챌린지에 응답하는 것을 가능하게 하도록, 셋-업 국면 후에, PUF 모듈을 타겟 당사자에게 물리적으로 보내는 단계를 포함한다.
- [0517] 스테이트먼트 10. 스테이트먼트 1 내지 스테이트먼트 6 중 어느 하나의 방법에 있어서, 데이터 저장소는 피어-투-피어 공개 매체이다.
- [0518] 스테이트먼트 11. 스테이트먼트 10의 방법에 있어서, 피어-투-피어 공개 매체는 블록체인이다.
- [0519] 스테이트먼트 12. 스테이트먼트 11의 방법에 있어서, 블록체인은 출력 기반 트랜잭션 모델을 사용하고, 블록체인은 각각 적어도 하나의 입력 및 적어도 하나의 출력을 갖는 복수의 트랜잭션을 포함하고, 각각의 트랜잭션의 입력은 다른 트랜잭션의 출력을 지시하고, 응답 데이터 레코드는 블록체인에 레코딩된 하나 이상의 트랜잭션의 하나 이상의 출력에 저장된다.
- [0520] 스테이트먼트 13. 스테이트먼트 12의 방법에 있어서, 방법은, 응답 데이터 레코드들 중 만료된 응답 데이터 레코드를 저장하는 트랜잭션의 출력을 지시하는 입력을 갖는 새로운 트랜잭션을 상기블록체인 상에 레코딩함으로써 만료된 응답 데이터 레코드를 업데이트 또는 폐지하는 단계를 더 포함한다.
- [0521] 스테이트먼트 14. 스테이트먼트 1 내지 스테이트먼트 13 중 어느 하나의 방법에 있어서, 저장하는 단계는 네트워크를 통해 데이터 저장소에 액세스하는 단계 및 데이터 저장소에서 응답 데이터 레코드를 생성하기 위해 네트워크를 통해 데이터를 전송하는 단계를 포함한다.
- [0522] 스테이트먼트 15. 스테이트먼트 1 내지 스테이트먼트 14 중 어느 하나의 방법에 있어서, 하나 이상의 검증 당사자는 복수의 검증 당사자이다.
- [0523] 스테이트먼트 16. 스테이트먼트 15의 방법에 있어서, 하나 이상의 검증 당사자는 복수의 검증 당사자이고, 저장하는 단계는 하나 이상의 응답 데이터 레코드의 개개의 서브세트만이 검증 당사자 각각에게 이용 가능하게 만들고, 상이한 서브세트는 검증 당사자 중 상이한 검증 당사자에게 이용 가능하게 된 레코드이다.
- [0524] 스테이트먼트 17. 스테이트먼트 16의 방법에 있어서, 서브세트는 서로 배타적이다.
- [0525] 스테이트먼트 18. 스테이트먼트 1 내지 스테이트먼트 17 중 어느 하나의 방법에 있어서, 응답 데이터 레코드 각각은 개개의 응답의 명시적 값을 포함한다.
- [0526] 스테이트먼트 19. 스테이트먼트 18의 방법에 있어서, 각각의 응답 데이터 레코드의 값은 암호화된 형태로만 데이터 저장소에 저장된다.
- [0527] 스테이트먼트 20. 스테이트먼트 19의 방법에 있어서, 각각의 응답 데이터 레코드는 암호해독하기 위해 상이한 개개의 암호해독 키를 요구한다.
- [0528] 스테이트먼트 21. 스테이트먼트 16에 종속하는 스테이트먼트 20의 방법에 있어서, 암호해독 키 중 상이한 암호해독 키를 상이한 검증 당사자들에게 분배함으로써 상이한 서브세트가 검증 당사자들 중 상이한 검증 당사자들에게 이용가능하게 된다.

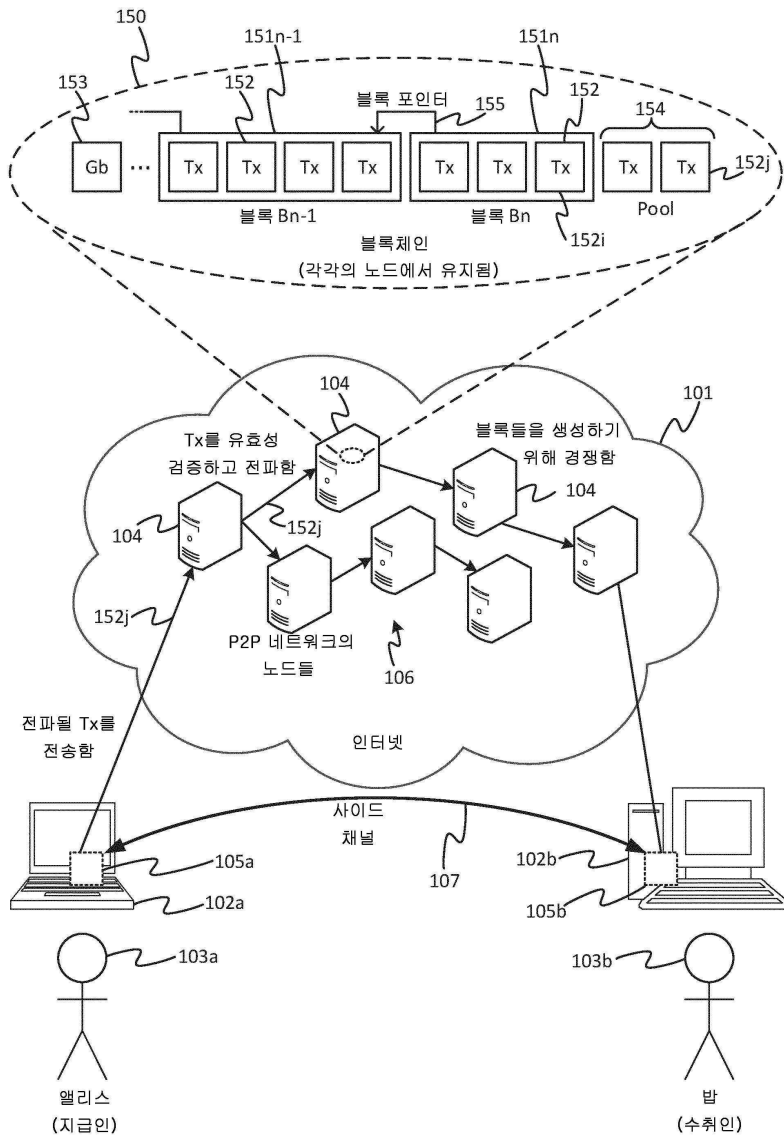
- [0529] 스테이트먼트 22. 스테이트먼트 1 내지 스테이트먼트 17 중 어느 하나의 방법에 있어서, 각각의 응답 데이터 레코드는 응답의 명시적 값이 아닌 개개의 응답의 증명만을 포함하고, 증명은 개개의 응답의 변환을 포함하고, 응답의 변환은 응답을 개시하지 않지만, 검증 당사자가 타겟에 의해 리턴된 개개의 응답의 추가 인스턴스에 대해 동일한 변환을 수행하고 증명과 비교함으로써 타겟의 아이덴티티를 검증하는 것을 가능하게 한다.
- [0530] 스테이트먼트 23. 스테이트먼트 22의 방법에 있어서, 변환은 해시 또는 이중 해시를 포함한다.
- [0531] 스테이트먼트 24. 스테이트먼트 1 내지 스테이트먼트 23 중 어느 하나의 방법에 있어서, 검증 당사자 중 하나로부터 요청을 수신하는 단계 - 요청은 챌린지의 세트 중 하나의 챌린지를 포함함 - ; 및 이에 응답하여, 대응하는 응답의 추가 인스턴스를 생성하기 위해 챌린지의 추가 인스턴스를 PUF 모듈에 입력하고, 그리고 검증 당사자가 데이터 저장소에 저장된 응답과의 비교에 기초하여 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위해 응답의 추가 인스턴스를 검증 당사자에 리턴하는 단계를 포함한다.
- [0532] 스테이트먼트 25. 스테이트먼트 1 내지 스테이트먼트 17 중 어느 하나의 방법에 있어서, 각각의 응답 데이터 레코드는 개개의 응답으로부터 도출된 개개의 공개-개인 키 쌍의 개개의 공개 키를 포함한다.
- [0533] 스테이트먼트 26. 스테이트먼트 25의 방법에 있어서, 상기 도출은 개개의 공개-개인 키 쌍을 생성하는 알고리즘에서 시드로서 개개의 응답을 사용하는 것을 포함한다.
- [0534] 스테이트먼트 27. 스테이트먼트 25 또는 26의 방법에 있어서, 공개 키 중 하나를 사용하여 타겟에 의해 서명된 메시지를 검증 당사자 중 하나에게 전송하거나 이용 가능하게 만드는 단계를 포함한다.
- [0535] 스테이트먼트 28. 스테이트먼트 1 내지 스테이트먼트 27 중 어느 하나의 방법에 있어서, PUF 모듈은 PUF, 인터페이스 로직 및 결정론적 변형 함수를 포함하고, 인터페이스 로직은: 챌린지 세트를 수신하고; 대응하는 1차 응답을 생성하기 위해 기본 챌린지를 PUF에 입력하고; 그리고 상기 응답 세트의 개개의 응답을 생성하기 위해 생성된 기본 응답과 함께 수신된 챌린지 세트 각각을 변형 함수에 입력함으로써 - 변형 함수는 수신된 챌린지 세트로부터의 개개의 챌린지 및 생성된 기본 응답의 함수임 - 응답 세트의 생성을 유발한다.
- [0536] 스테이트먼트 29. 스테이트먼트 1 내지 스테이트먼트 28 중 어느 하나의 방법에 있어서, 각각의 응답 데이터 레코드는 개개의 챌린지의 표시와 관련하여 데이터 저장소에 저장되고, 따라서 검증 당사자가 개개의 챌린지를 결정하거나, 또는 검증에서 타겟에 챌린지하는 데 사용되는 개개의 챌린지에 기초하여 개개의 응답을 록업하는 것을 가능하게 한다.
- [0537] 스테이트먼트 30. 스테이트먼트 29의 방법에 있어서, 각각의 개개의 챌린지의 표시는 개개의 응답 데이터와 관련하여 저장된 개개의 챌린지의 명시적 값을 포함한다.
- [0538] 스테이트먼트 31. 스테이트먼트 29의 방법에 있어서, 챌린지의 표시는 챌린지 세트가 도출될 수 있는 마스터 챌린지를 포함한다.
- [0539] 스테이트먼트 32. 스테이트먼트 1 내지 스테이트먼트 31 중 어느 하나의 방법에 있어서, 타겟 당사자는, 데이터 저장소가 다수의 타겟 당사자의 대응하는 PUF에 대한 챌린지로부터 기인하는 응답 데이터 세트를 레코딩하는 다수의 타겟 당사자 중 하나이다.
- [0540] 스테이트먼트 33. 컴퓨터 장치는 하나 이상의 메모리 유닛을 포함하는 메모리; 및 하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 메모리는 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 코드는 처리 장치 상에서 실행될 때 스테이트먼트 1 내지 스테이트먼트 32 중 어느 하나의 방법을 수행하도록 구성된다.
- [0541] 스테이트먼트 34. 컴퓨터 프로그램은 비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 스테이트먼트 1 내지 스테이트먼트 32 중 어느 하나의 방법을 수행하도록 구성된다.
- [0542] 스테이트먼트 35. 하나 이상의 검증 당사자가 타겟 당사자 또는 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하는 것을 가능하게 하기 위한 컴퓨터 구현 방법으로서, 방법은, 셋-업 국면에서, 신뢰할 수 있는 제3자에 의해: 개개의 응답을 생성하기 위해, 타겟 당사자가 PUF(physically unclonable function)를 포함하는 PUF 모듈에 개개의 챌린지를 입력한 것으로부터 기인한 응답 데이터를 타겟 당사자로부터 수신하는 단계; 하나 이상의 응답 각각에 대한 개개의 응답 데이터 레코드를 신뢰할 수 있는 제3자의 데이터 저장소에 저장하는 단계 - 각각의 응답 데이터 레코드는 개개의 응답 또는 개개의 응답으로부터 도출된 데이터를 포함함 - ; 및 후속 검증 국면에서 타겟의 아이덴티티를 검증하기 위해 응답 데이터 레코드를 하나 이상의 검증 당사자에게 이용 가능하게 만드는 단계를 포함하고, 각각의 응답에 대한 응답 데이터 레코드는 개개의 응답 또는 개개의 응답

으로부터 도출된 데이터를 포함한다.

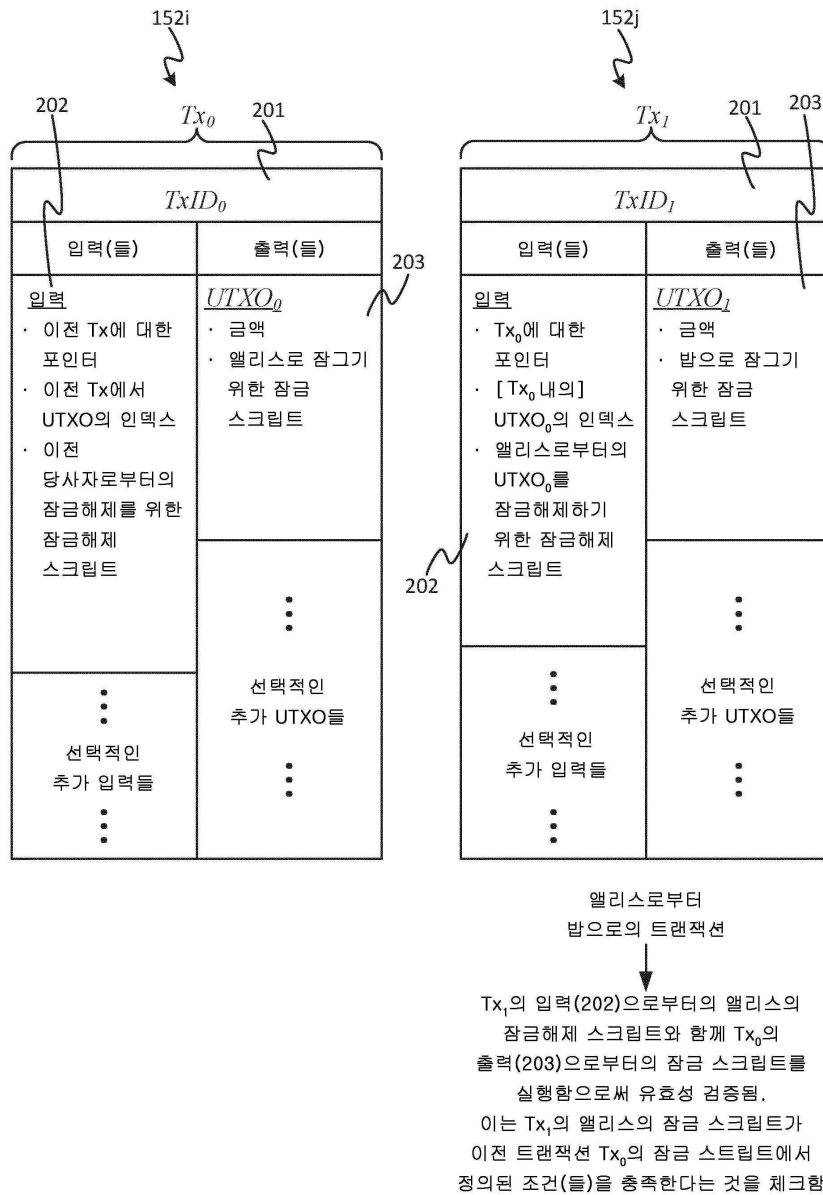
- [0543] 스테이트먼트 36. 컴퓨터 장치는 하나 이상의 메모리 유닛을 포함하는 메모리; 및 하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 메모리는 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 코드는 처리 장치 상에서 실행될 때 스테이트먼트 35의 방법을 수행하도록 구성된다.
- [0544] 스테이트먼트 37. 컴퓨터 프로그램은 비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 내지 스테이트먼트 35의 방법을 수행하도록 구성된다.
- [0545] 스테이트먼트 38. 검증 당사자가 타겟 당사자 또는 타겟 당사자의 디바이스를 포함하는 타겟의 아이덴티티를 검증하게 하는 컴퓨터 구현 방법으로서, 방법은, 검증 당사자에 의해: 타겟 당사자 또는 검증 당사자의 임의의 장비 외부에 있는 데이터 저장소의 응답 데이터 레코드에 액세스하는 단계 - 데이터 저장소는 신뢰할 수 있는 제3자의 제3자 컴퓨터 장비의 일부이거나 공개 피어-투-피어 공개 매체이고, 응답 데이터 레코드는 a) 타겟과 연관된 PUF(physically unclonable function)를 포함하는 PUF 모듈에 챌린지를 입력한 것으로부터 발생한 응답의 값, 또는 b) 응답의 변환을 포함하는 증명을 포함함 - ; 챌린지를 포함하는 요청을 타겟에 전송하고, 이에 응답하여, 응답의 추가 인스턴스를 다시 수신하는 단계; 비교를 수행하고, 비교 결과가 일치한다는 조건으로 타겟의 아이덴티티를 검증하는 단계를 포함하고, 비교는 a)의 경우에서 응답의 저장된 값이 추가 인스턴스와 일치하는 것, 또는 b)의 경우에서 증명이 추가 인스턴스에 적용된 동일한 변환과 일치하는 것을 포함한다.
- [0546] 스테이트먼트 39. 컴퓨터 장치는 하나 이상의 메모리 유닛을 포함하는 메모리; 및 하나 이상의 처리 유닛을 포함하는 처리 장치를 포함하고, 메모리는 처리 장치 상에서 실행되도록 배열된 코드를 저장하고, 코드는 처리 장치 상에서 실행될 때 스테이트먼트 38의 방법을 수행하도록 구성된다.
- [0547] 스테이트먼트 40. 컴퓨터 프로그램은 비일시적 컴퓨터 판독 가능 매체 상에서 구현되고, 하나 이상의 프로세서 상에서 실행될 때 내지 스테이트먼트 38의 방법을 수행하도록 구성된다.

도면

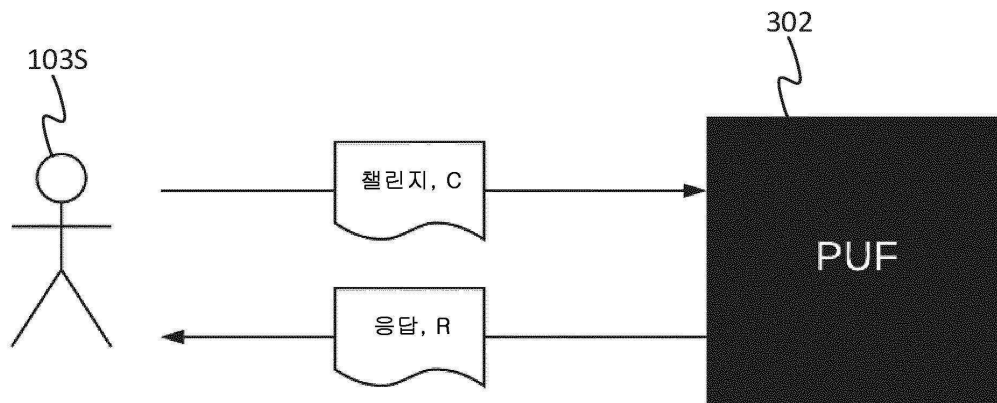
도면1



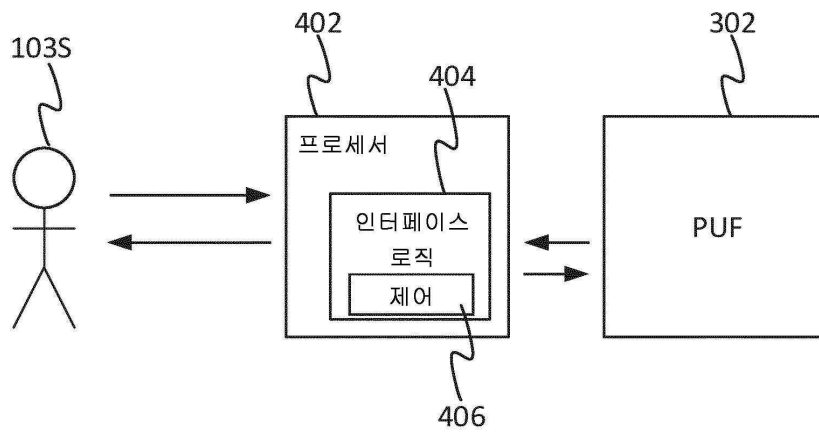
도면2



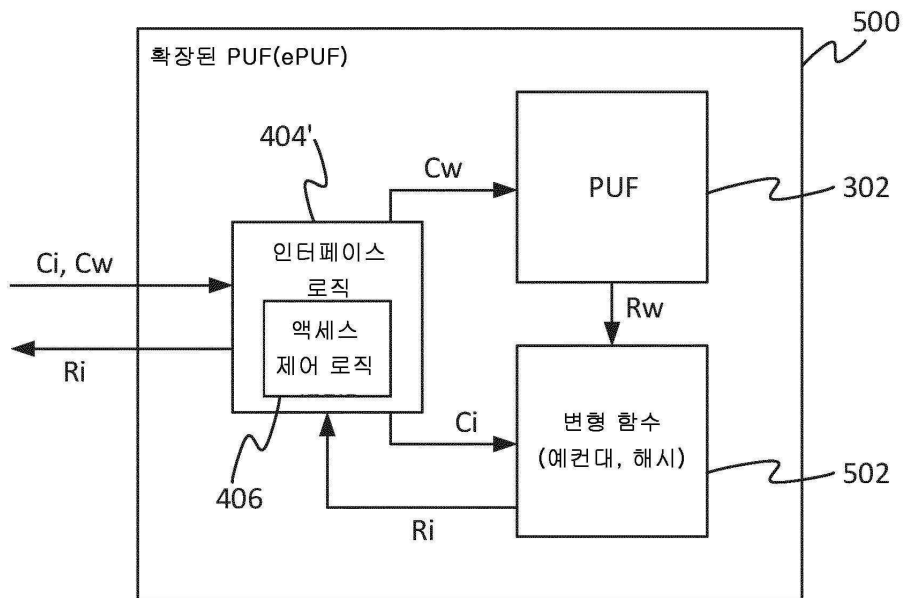
도면3



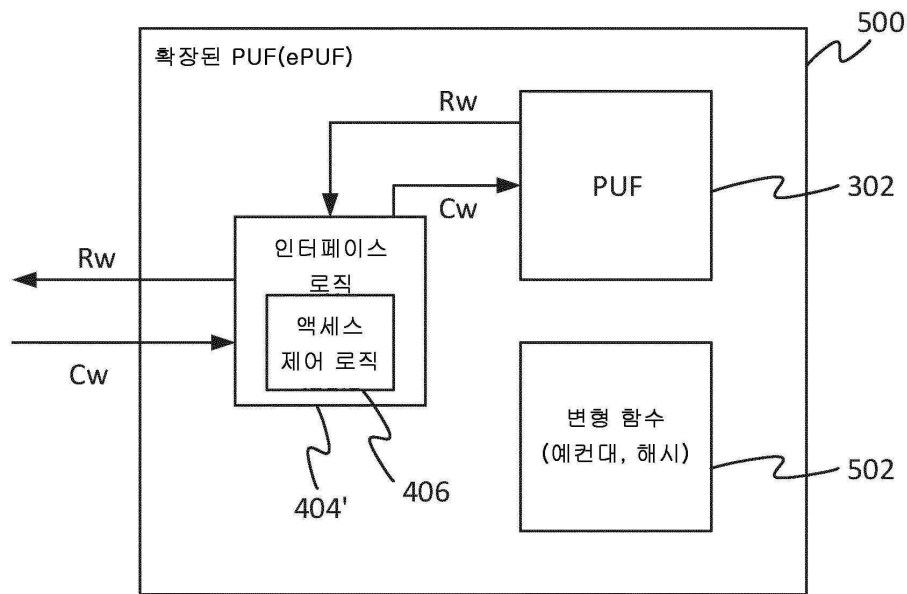
도면4



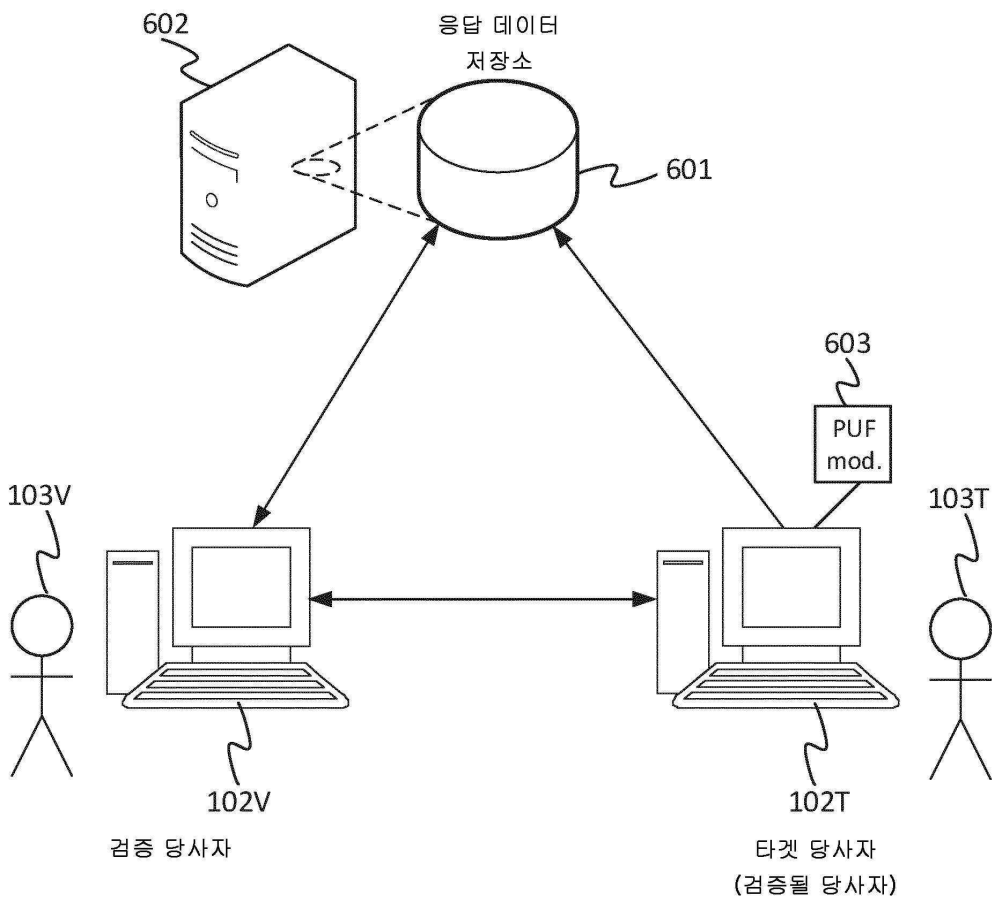
도면5a



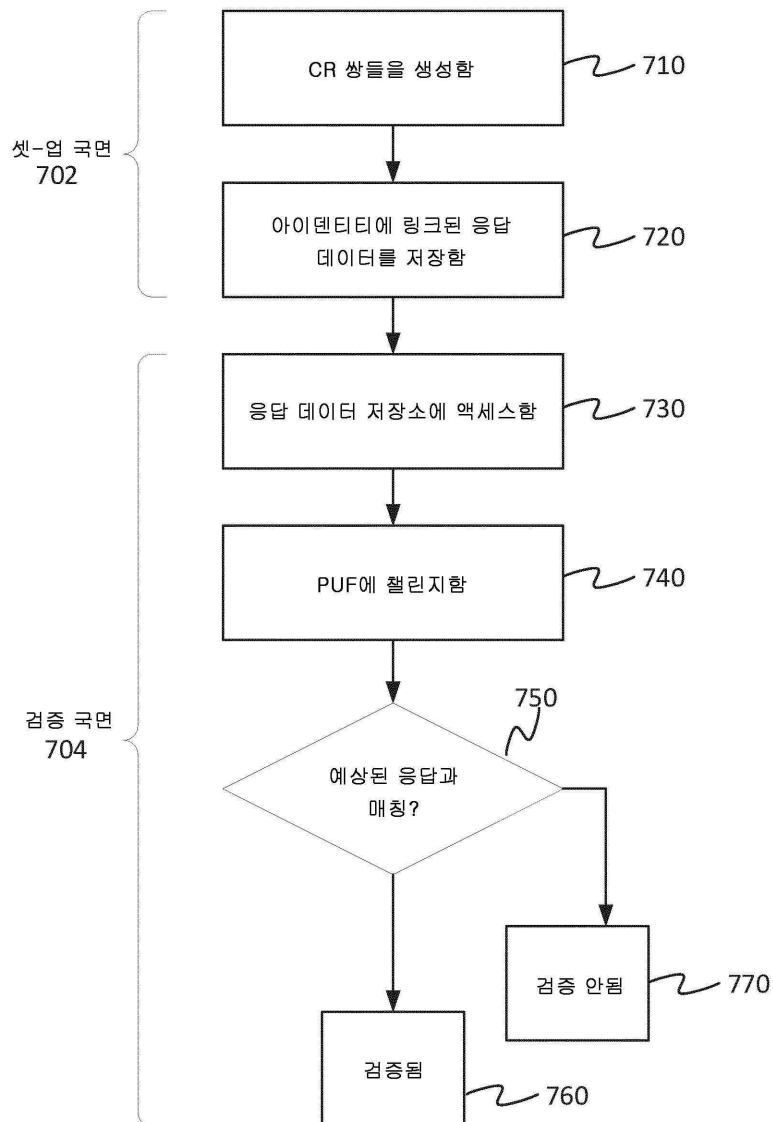
도면5b



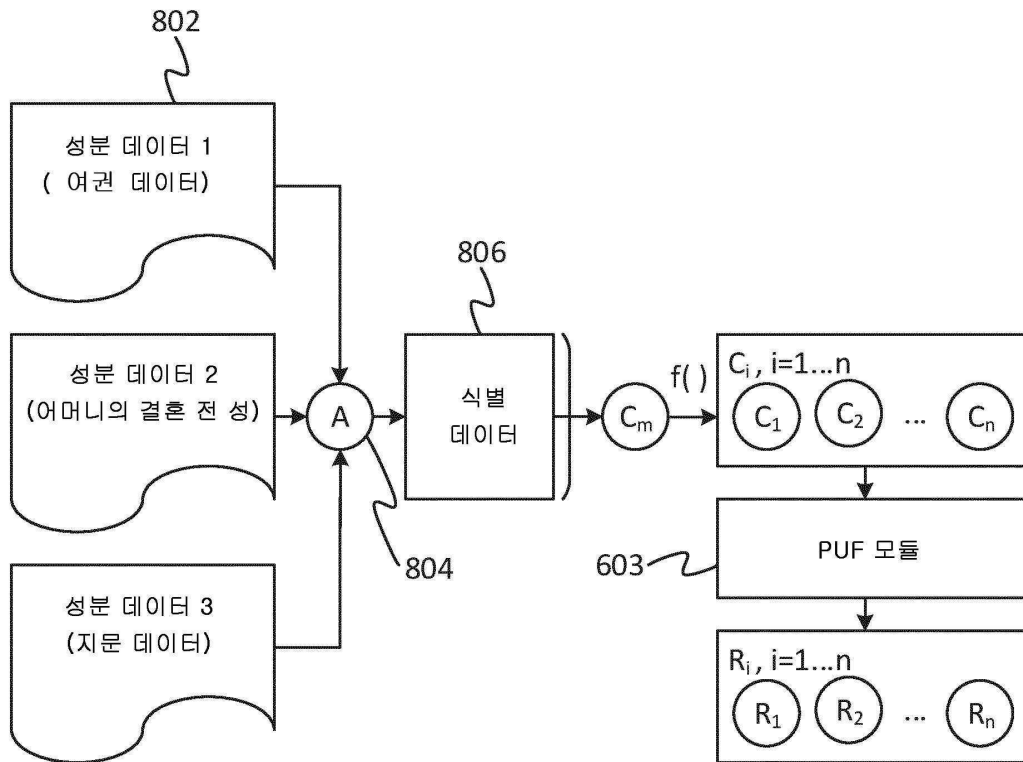
도면6



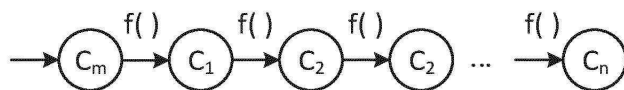
도면7



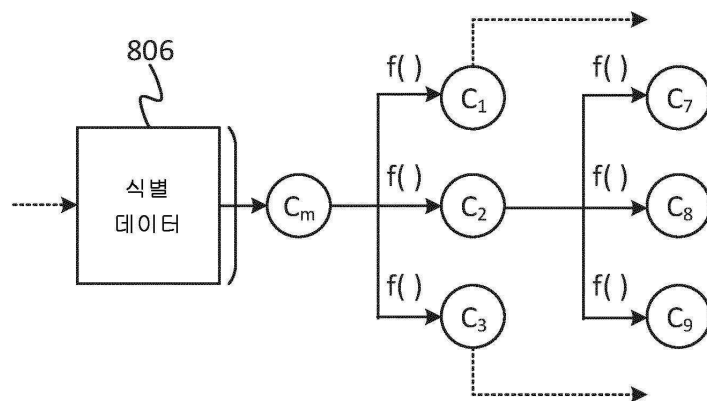
도면8a



도면8b



도면8c



도면9

