(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0071094
(43) 공개일자 2023년05월23일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01)
(52) CPC특허분류
H04L 9/3218 (2013.01)
H04L 9/3247 (2013.01)
(21) 출원번호 10-2022-0152762
(22) 출원일자 2022년11월15일
심사청구일자 2022년11월15일
(30) 우선권주장
1020210156822 2021년11월15일 대한민국(KR)

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
류재철
대전광역시 유성구 어은로 57 한빛아파트 132-801
김근영
대전광역시 유성구 장대로 87 퍼스트캐슬 202호
김준석
대전광역시 유성구 대학로 53
(74) 대리인
이은철, 김재문

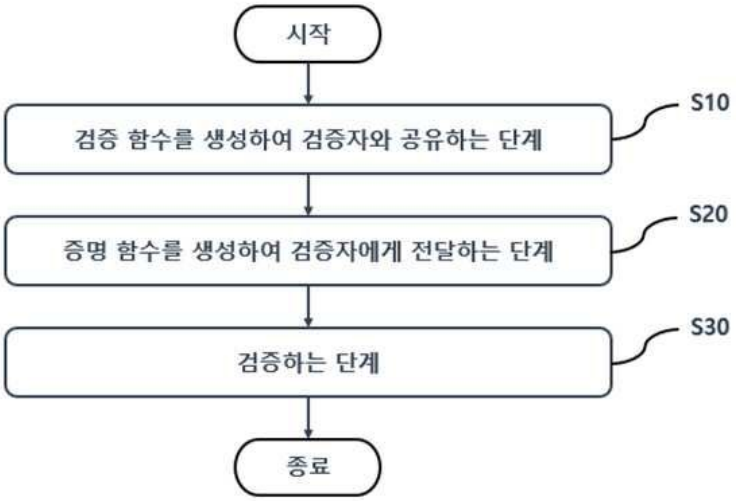
전체 청구항 수 : 총 4 항

(54) 발명의 명칭 **Verifiable Computation을 활용한 영지식증명 컨트랙트 생성 방법**

(57) 요 약

본 발명은, 사용자(Holder)가 전체 크리덴셜(Credential)를 입력하였을 때 상기 전체 크리덴셜 중 검증하고자 하는 특정 크리덴셜이 검증 조건을 만족하는지 계산하는 검증 함수를 생성하여 검증자(Verifier)와 공유하는 단계(S10); 상기 사용자가 상기 검증 함수에 입력하였을 때 특정 크리덴셜이 검증 조건을 만족하는지 출력하게 하는 증명 함수를 생성하여 상기 검증자에게 전달하는 단계(S20); 및 상기 검증자가 상기 검증 함수에 상기 증명 함수를 입력하여 특정 크리덴셜이 조건을 만족하는지에 대해 검증하는 단계(S30);을 포함하는 것을 특징으로 한다.

대 표 도 - 도4



이 발명을 지원한 국가연구개발사업

과제고유번호	1711126223
과제번호	2020-0-00321-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D,정보화)
연구과제명	5G 서비스 환경에서 프라이버시가 보장되는 자기통제형 분산 디지털 신원 관리 및 보안 기술 개발
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

사용자(Holder)가 전체 크리덴셜(Credential)를 입력하였을 때 상기 전체 크리덴셜 중 검증하고자 하는 특정 크리덴셜이 검증 조건을 만족하는지 계산하는 검증 함수를 생성하여 검증자(Verifier)와 공유하는 단계(S10);

상기 사용자가 상기 검증 함수에 입력하였을 때 특정 크리덴셜이 검증 조건을 만족하는지 출력하게 하는 증명 함수를 생성하여 상기 검증자에게 전달하는 단계(S20); 및

상기 검증자가 상기 검증 함수에 상기 증명 함수를 입력하여 특정 크리덴셜이 조건을 만족하는지에 대해 검증하는 단계(S30);을 포함하는 영지식 증명 컨트랙트 생성 방법.

청구항 2

제 1 항에 있어서,

검증자(Verifier)와 공유하는 단계(S20)는,

특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 출력하게 하는 증명 함수를 생성하는 것을 특징으로 하는 영지식 증명 컨트랙트 생성 방법.

청구항 3

제 2 항에 있어서,

상기 검증자에게 전달하는 단계(S20)는,

상기 검증 함수에 전체 크리덴셜을 입력하였을 때의 함숫값을 함께 전달하는 것을 특징으로 하는 영지식 증명 컨트랙트 생성 방법.

청구항 4

제 3 항에 있어서,

검증하는 단계(S30)는,

상기 증명 함수에 상기 함숫값을 대입한 결과를 상기 검증 함수에 입력하여 특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 검증하는 것을 특징으로 하는 영지식 증명 컨트랙트 생성 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 Verifiable Computation을 활용한 영지식증명 컨트랙트 생성 방법에 관한 것으로서, 특히 사용자가 검증 과정에서 어떠한 값도 드러내지 않은 채 연산의 결과가 옳음을 증명할 수 있는 Verifiable Computation을 활용한 영지식증명 컨트랙트 생성 방법에 관한 것이다.

배 경 기 술

[0003] 도 1은 오프체인에서 사용자가 졸업증(Credential)을 발급받아 검증자에게 전송하는 DID(Decentralized

Identity) 과정을 나타낸다. 도 1을 참조하면, 졸업증을 검증하는 과정에서 검증자에게 필요 이상의 정보(학위를 증명하고자 가정했을 때, 대학교와 같은 정보)가 드러나는 문제가 발생한다.

[0004] 도 2는 선택적 노출이 적용된 영지식증명이 적용된 DID(Decentralized Identity) 과정을 나타낸다. 도 2를 참조하면, 사용자는 발급자로부터 영지식증명 중에서도 Selective Disclosure(선택적 노출, 드러내고 싶은 정보만 드러내는 것)가 가능한 Credential을 발급받게 되고 이후 사용자는 드러내고 싶은 정보만을 드러내어 검증자에게 학위 증명서(Presentation)를 전송할 수 있게 된다. 이로써 필요 이상의 정보는 드러나지 않는다는 장점이 있다(추가적으로 드러내고자 하는 학위 정보의 정확한 내용도 드러나지만, range proof와 같이 특정 값 이상이라는 것을 이용해 숨길 수도 있음).

[0005] 도 3은 TTA PG502에서 제정한 표준을 나타낸다. 도 3을 참조하면, 제정된 표준은 Credential을 3개의 type으로 분류하여 정리하였다. 기존의 type 1, type 2의 경우는 도 1과 같이 RSA Signature를 사용하여 영지식증명을 사용할 수 없는 케이스이고, type 3의 경우는 도 2와 같이 CL Signature, BBS+ Signature 등을 사용하여 영지식증명 및 Selective Disclosure를 사용할 수 있는 케이스이다. 이처럼, Credential을 생성할 때, 어떤 서명 알고리즘을 사용하냐에 따라서 영지식증명 및 Selective Disclosure 지원 여부를 정할 수 있다.

[0006] 기존 주로 사용하고 있는 RSA Signature의 경우는 PKI 시스템을 활용해 발급자의 공개키를 블록체인에 업로드함으로써 Credential의 진위를 파악할 수 있다. 하지만, Selective Disclosure를 지원하지 않기 때문에 검증자에게 필요 이상의 claim이 드러난다는 단점이 있다. 이에 비해 CL Signature의 경우에는 Credential을 생성할 때부터 평문의 claim이 아닌 암호화된 claim을 발급해 줌으로써 프라이버시를 보호할 수 있고, 사용자 또한 검증자에게 presentation을 제출할 때 드러내고 싶은 정보만 복호화를 한 후에 claim 그대로 전송하면 되기 때문에 Selective Disclosure를 지원할 수 있다. 하지만, claim의 타입에 따라 range proof가 가능한 경우에는 claim을 숨길 수 있지만, range proof가 불가능한 타입일 경우에는 claim도 노출될 수 있다는 문제점이 있다(string 타입의 경우 range proof가 대부분 불가능).

발명의 내용

해결하려는 과제

[0008] 본 발명은 기존 서명 알고리즘(Library)을 활용하여 CL Signature를 제공했던 방법과는 달리 Verifiable Computation을 이용해 프라이버시 및 개발자(사용자)의 편리성을 제공해주는 도구를 만들고자 한다.

과제의 해결 수단

[0010] 상기 목적을 달성하기 위하여 본 발명은, 사용자(Holder)가 전체 크리덴셜(Credential)를 입력하였을 때 상기 전체 크리덴셜 중 검증하고자 하는 특정 크리덴셜이 검증 조건을 만족하는지 계산하는 검증 함수를 생성하여 검증자(Verifier)와 공유하는 단계(S10); 상기 사용자가 상기 검증 함수에 입력하였을 때 특정 크리덴셜이 검증 조건을 만족하는지 출력하게 하는 증명 함수를 생성하여 상기 검증자에게 전달하는 단계(S20); 및 상기 검증자가 상기 검증 함수에 상기 증명 함수를 입력하여 특정 크리덴셜이 조건을 만족하는지에 대해 검증하는 단계(S30);을 포함하는 것을 특징으로 한다.

[0011] 바람직하게는, 상기 검증자에게 전달하는 단계(S20)는, 특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 출력하게 하는 증명 함수를 생성할 수 있다.

[0012] 바람직하게는, 상기 검증자에게 전달하는 단계(S20)는, 상기 검증 함수에 전체 크리덴셜을 입력하였을 때의 함숫값을 함께 전달할 수 있다.

[0013] 바람직하게는, 검증하는 단계(S30)는, 상기 증명 함수에 상기 함숫값을 대입한 결과를 상기 검증 함수에 입력하여 특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 검증할 수 있다.

발명의 효과

[0015] 본 발명은 range proof가 적용되지 않는 타입에도 Verifiable Computation을 활용하여 어떠한 값도 드러내지 않

은 연산의 결과가 옳음을 증명할 수 있다는 이점이 있다.

[0016] 또한, 본 발명은 모든 타입에서 Selective Disclosure가 지원되는 영지식증명을 제공할 수 있다는 이점이 있다.

도면의 간단한 설명

[0018] 도 1은 오프체인에서 사용자가 졸업증을 발급받아 검증자에게 전송하는 과정을 나타낸다.

도 2는 선택적 노출이 적용된 영지식증명이 적용된 DID(Decentralized Identity) 과정을 나타낸다.

도 3은 TTA PG502에서 제정한 표준을 나타낸다.

도 4는 본 발명의 실시예에 따른 영지식증명 컨트랙트 생성 방법의 흐름도를 나타낸다.

도 5는 본 발명의 실시예에 따른 영지식증명 컨트랙트 생성 방법과 기존의 도구를 비교한 그림을 나타낸다.

도 6은 본 발명의 실시예에 따른 Verifiable Computation 과정을 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0019] 이하, 첨부된 도면들에 기재된 내용들을 참조하여 본 발명을 상세히 설명한다. 다만, 본 발명이 예시적 실시 예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일 참조부호는 실질적으로 동일한 기능을 수행하는 부재를 나타낸다.

[0020] 본 발명의 목적 및 효과는 하기의 설명에 의해서 자연스럽게 이해되거나 보다 분명해 질 수 있으며, 하기의 기재만으로 본 발명의 목적 및 효과가 제한되는 것은 아니다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이, 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다.

[0021] 도 4는 본 발명의 실시예에 따른 영지식증명 컨트랙트 생성 방법의 흐름도를 나타낸다. 도 4를 참조하면, 영지식증명 컨트랙트 생성 방법은 검증 함수를 생성하여 검증자와 공유하는 단계(S10), 검증자에게 전달하는 단계(S20), 및 검증하는 단계(S30)를 포함할 수 있다.

[0022] 이하에서 설명하는 영지식증명이란 상대방에게 어떠한 정보도 제공하지 않은 채, 자신이 해당 정보를 가지고 있다는 사실을 증명하는 것을 의미한다.

[0023] 영지식증명 컨트랙트 생성 방법은 영지식증명이 가능한 type 3가 이용될 수 있다. 영지식증명 컨트랙트 생성 방법은 기존 서명 알고리즘(Library)를 활용하여 CL(Camenisch-Lysyanskaya) Signature를 제공했던 방법과는 달리 Verufuable Computation을 이용해 프라이버시를 보호할 수 있고, 개발자(사용자)에게 편리성을 제공할 수 있다.

[0024] 영지식증명 컨트랙트 생성 방법은 claim의 타입에 따라 range proof의 가능 여부와 관계없이 Selective Disclosure를 지원할 수 있다. 즉, 영지식증명 컨트랙트 생성 방법은 range proof가 적용되지 않는 타입에도 Verifiable Computation을 활용하여 어떠한 값도 드러내지 않은 연산의 결과가 옳음을 증명할 수 있다. 영지식증명 컨트랙트 생성 방법은 모든 타입에서 Selective Disclosure가 지원되는 영지식증명을 제공할 수 있다.

[0025] 도 5는 본 발명의 실시예에 따른 영지식증명 컨트랙트 생성 방법과 기존의 도구를 비교한 그림을 나타낸다. 도 5를 참조하면, 기존의 도구에서 사용자는 기존의 서명 알고리즘을 통해 드러내고 싶은 정보만을 드러내는 프레젠테이션을 생성하게 된다. 사용자가 프레젠테이션을 생성할 때, 크리덴셜 내부의 모든 값(발급자의 서명값 및 랜덤값, 그리고 공개된 모든 값들)을 활용하여 검증자에게 제공할 proof를 생성하게 된다. 이후 검증자는 사용자에게 받은 값을 바탕으로 하여 발급자의 서명 검증 과정 및 사용자가 숨기고자 하는 값이 변경되지 않았음을 검증한다. 하지만 이 과정에서 range proof(특정 값 이상)를 사용하지 못하면 검증 과정에서 claim 값이 드러나게 된다. 또한, 알고리즘을 활용하기 위해 각 플랫폼 환경에 맞게 Library를 변경 또는 확장하여 사용해야 하기에 플랫폼이 달라질 때마다 추가적인 작업이 필요하다. 즉, CL signature를 사용할 것인지, BBS+ Signature를 사용할 것인지 선택해야 하고, 선택 후에 특정 알고리즘에 대한 구현물을 붙여야 한다.

[0026] 기존의 도구와 비교하여 영지식증명 컨트랙트 생성 방법은 range proof가 적용되지 않는 타입에도 Verifiable Computation을 활용하여 어떠한 값도 드러내지 않은 채 연산의 결과가 옳음을 증명할 수 있는 도구를 생성할 수

있다.

[0027] 도 6은 본 발명의 실시예에 따른 Verifiable Computation 과정을 나타낸다. 도 6을 참조하면, Verifiable Computation은 검증 함수(F( ))가 공유되었을 때, 사용자는 로컬에서 특정 입력값 x에 대한 F(x)값인 y와 $proof_F$ (y)를 검증자에게 전달한다. 이후 검증자는 x를 알지 못하더라고 verification 과정 및 $proof_F$(y)에 대한 검증을 통해 연산 결과에 대한 참·거짓 여부를 판단할 수 있다.

[0028] 이하에서는 Verifiable Computation을 활용한 영지식 증명 컨트랙트 생성 방법의 각 단계에 대해 구체적으로 설명한다.

[0029] 검증 함수를 생성하여 검증자와 공유하는 단계(S10)에서는 사용자(Holder)가 전체 크리덴셜(Credential)를 입력하였을 때 전체 크리덴셜 중 검증하고자 하는 특정 크리덴셜이 검증 조건을 만족하는지 계산하는 검증 함수를 생성하여 검증자(Verifier)와 공유할 수 있다.

[0030] 일반적인 DID에서는 검증자가 검증에 필요한 검증 함수를 생성하고, 사용자가 증명에 필요한 증명 함수를 생성한다. 즉, 일반적인 DID에서는 증명자가 증명에 필요한 매개 변수와 증인 변수(정보를 증명하는 데 필요한 비밀 입력값)를 증명 함수에 넣고 증명 데이터를 생성하면, 검증자는 검증에 필요한 매개 변수와 증명 데이터를 검증 함수에 넣고 증명이 유효한지 아닌지를 판별한다. 일반적인 DID에서는 영지식증명의 대상이 크리덴셜 내부 모든 값이므로 내부 모든 로직을 알아야 검증할 수 있다. 따라서, 일반적인 DID에서는 검증 과정에서 공개값이 드러나게 될 수 있다.

[0031] 검증 함수를 생성하여 검증자와 공유하는 단계(S10)에서는 사용자가 검증 함수를 생성하여 검증자에게 공유하므로 검증 과정에서 값이 드러나지 않을 수 있다. 또한, 검증 함수를 생성하여 검증자와 공유하는 단계(S10)에서는 전체 크리덴셜(Credential)을 입력하였을 때 전체 크리덴셜 중 검증하고자 하는 특정 크리덴셜이 검증 조건을 만족하는지 계산하는 검증 함수를 생성하므로 영지식증명의 대상이 크리덴셜 내부의 모든값이 아니라 크리덴셜을 이용한 연산이므로 크리덴셜 내부의 로직을 알 필요가 없다. 따라서, 본 발명은 플랫폼별 SDK(Software Development Kit)에 쉽게 적용할 수 있다는 이점이 있다.

[0032] 검증자에게 전달하는 단계(S20)에서는 사용자가 검증 함수에 입력하였을 때 특정 크리덴셜이 검증 조건(ex, 석사 이상)을 만족하는지 출력하게 하는 증명 함수를 생성하여 검증자에게 전달할 수 있다. 검증자에게 전달하는 단계(S20)에서 검증자에게 전달된 증명 함수는 검증 함수에 입력되어 특정 크리덴셜이 조건에 만족하는지에 대한 진위 판단에 사용될 수 있다.

[0033] 검증자에게 전달하는 단계(S20)에서는 특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 출력하게 하는 증명 함수를 생성할 수 있다. 검증자에게 전달하는 단계(S20)에서는 특정 크리덴셜이 검증 조건을 만족하는지 뿐만 아니라 발급자의 서명을 검증할 수 있는 증명 함수를 생성함으로써, range proof를 사용하지 못하는 경우에도 사용자는 아무런 정보도 제공하지 않은 채 연산에 대한 검증을 통한 증명을 할 수 있기 때문에 사용자의 프라이버시를 보호할 수 있다.

[0034] 검증자에게 전달하는 단계(S20)에서는 검증 함수에 전체 크리덴셜을 입력하였을 때의 함숫값을 함께 전달할 수 있다. 검증자에게 전달하는 단계(S20)에서 전달한 함숫값과 증명 함수를 검증 함수에 입력하여 특정 크리덴셜에 대한 진위를 판단할 수 있다.

[0035] 검증하는 단계(S30)에서는 검증자가 검증 함수에 증명 함수를 입력하여 특정 크리덴셜이 조건을 만족하는지에 대해 검증할 수 있다. 검증하는 단계(S30)에서는 전체 크리덴셜(x)의 내부 값을 알지 못하더라도 검증 과정 및 증명 함수에 대한 검증을 통해 연산 결과에 대한 참·거짓 여부에 따라 특정 크리덴셜의 진위를 판단할 수 있다. 따라서, 검증하는 단계(S30)에서는 전체 크리덴셜(x)의 내부 값이 전혀 드러나지 않아 검증자가 크리덴셜 값을 알 수 없으므로 사용자의 프라이버시가 보호될 수 있다.

[0036] 검증하는 단계(S30)에서는 증명 함수에 함숫값을 대입한 결과를 검증 함수에 입력하여 특정 크리덴셜이 검증 조건을 만족함과 동시에 검증된 발급자의 서명인지 검증할 수 있다. 검증하는 단계(S30)에서는 발급자의 서명이 포함된 증명 함수를 생성함으로써 모든 검증 과정이 연산에 의한 검증이 되어 검증에 있어 어떠한 정보도 드러나지 않는다.

[0037] 본 발명의 실시예에 따른 Verifiable Computation을 활용한 영지식증명 컨트랙트 생성 방법은 기존의 CL Signature로 이루어진 크리덴셜뿐만 아니라 BBS+ Signature에 대한 크리덴셜도 커버할 수 있다. 영지식증명 컨트랙트 생성 방법은 BBS+ Signature를 활용하여 최적화 및 사용자 친화적으로 도구를 개발함으로써 기존의 도구

에 비해 월등히 높은 성능을 제공할 수 있다.

[0038] 추가적으로, 영지식증명 컨트랙트 생성 방법은 크리덴셜을 On-chain에서도 검증할 수 있는 도구에 사용될 수 있다. 영지식증명 컨트랙트 생성 방법은 Verifiable Computation을 지원하는 Zokrates과 같은 도구를 활용하여 Off-chain에서 생성된 Proof를 On-chain에서 검증해줄 수 있는 컨트랙트를 생성할 수 있다. 영지식 증명 컨트랙트 생성 방법은 컨트랙트로 검증을 진행할 경우, 기존 프라이빗 블록체인 환경에 비해 사용자의 검증 결과가 공개되어 있으므로 어디서나 활용할 수 있다는 장점이 있다. 이 경우 사용자는 자신의 신원에 대해 검증받은 내용(석사 이상)이 컨트랙트 상에 저장되어 공개되어 있으므로 투명성을 얻을 수 있고, 타 검증기관에 같은 내용을 증명하고자 할 때 매번 Proof를 만들어 전달하지 않고도 기존에 인증받은 자신의 신원에 대한 검증 결과(석사 이상)를 활용할 수 있다(컨트랙트를 통해 석사 이상임을 증명받았을 경우, 해당 트랜잭션을 학위 증명이 필요한 다른 기관에 보내줌으로써 매번 검증자의 환경에 맞는 증명을 생성할 필요가 없다).

[0039] 이상에서 대표적인 실시예를 통하여 본 발명을 상세하게 설명하였으나, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 상술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리 범위는 설명한 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 특허청구범위와 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태에 의하여 정해져야 한다.

부호의 설명

도면

도면1



도면2



도면3

		TYPE 1	TYPE 2	TYPE 3
형태		Credential	Credential ... Credential	Credential
발급	속성 (Claim)	N	N	N
	인증서 (Credential)	1	N	1
Presentation 생성 방법		Credential 변경없이 Presentation에 사용 (자기통제 불가능)	Credential 변경없이 Presentation에 사용 (자기통제 불가능)	영지식증명을 사용하여 Presentation에 사용 (자기통제 가능)
영지식증명 제공 여부		X	X	O
선택적 노출		불가능	Credential Level로 가능 (발급기관 의존형)	Claim Level로 가능 (발급기관 독립적)
사용자 연결성		사용자의 서명이 포함된 Presentation	사용자의 서명이 포함된 Presentation	사용자의 서명이 아닌 일회용 Presentation (Anonymous Credential)
암호 알고리즘		RSA Signature, ECDSA, ECDDA	RSA Signature, ECDSA, ECDDA	CL Signature (Camenisch-Lysyanskaya)

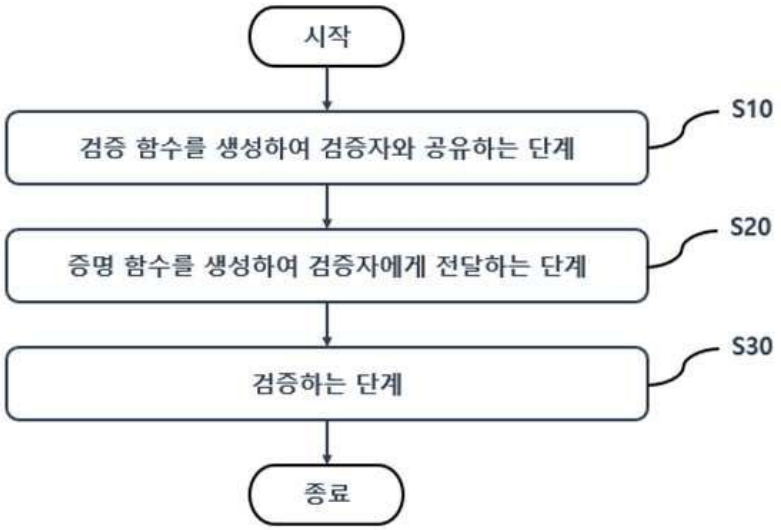


TYPE 1,2 Credential

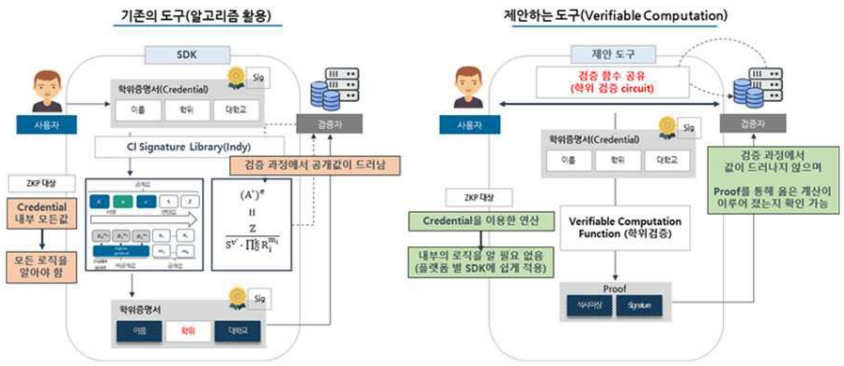


TYPE 3 Credential

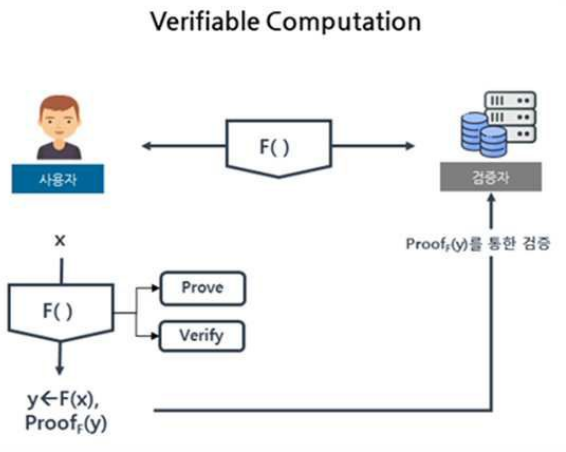
도면4



도면5



도면6


Verifiable Computation
사용자
F( )
검증자
x
F( )
Prove
Verify
y←F(x),
Proof$_F$(y)
Proof$_F$(y)를 통한 검증