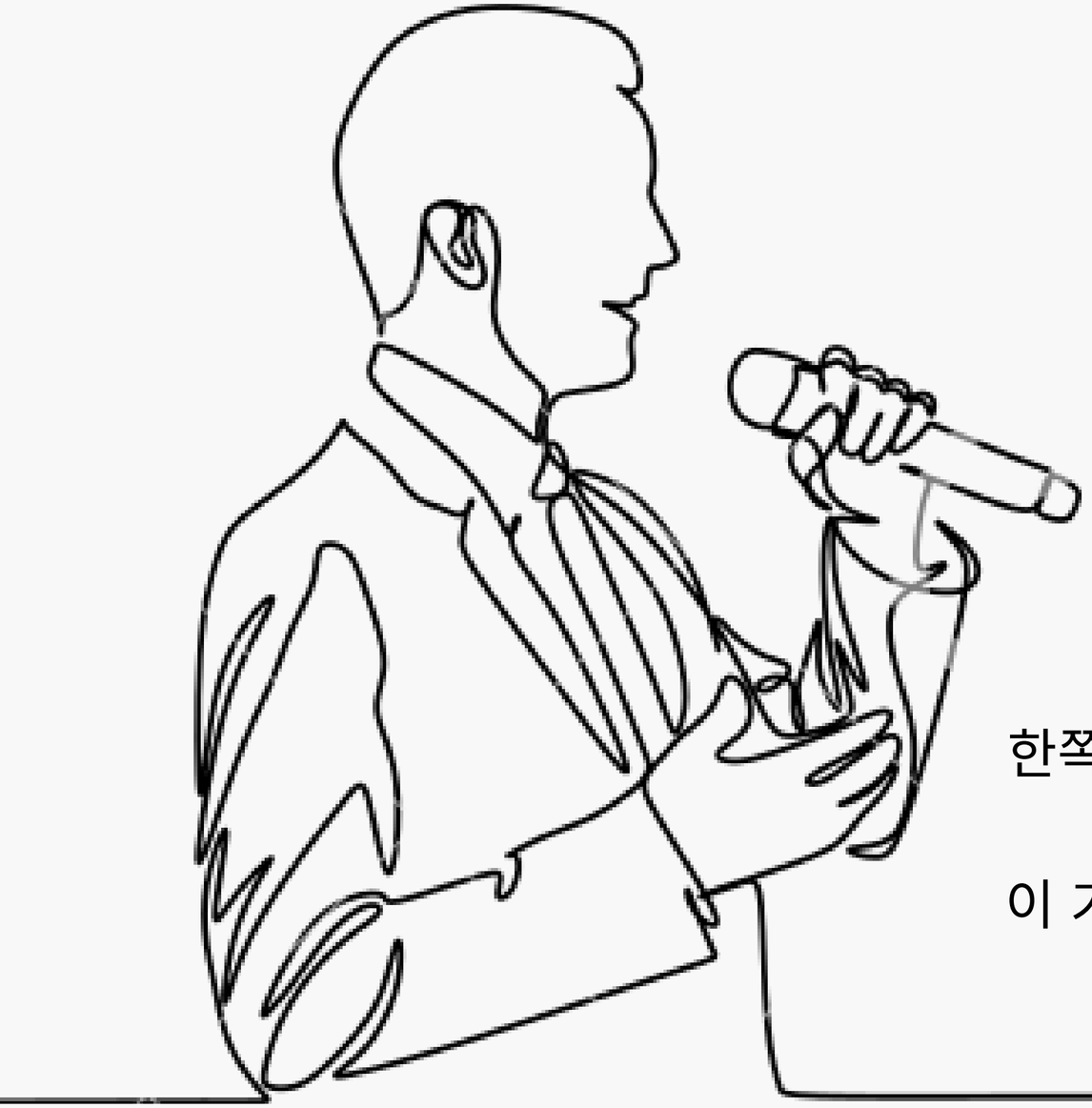
영지식 증명



ZERO-KNOWLEDGE PROOF

영지식 증명이란

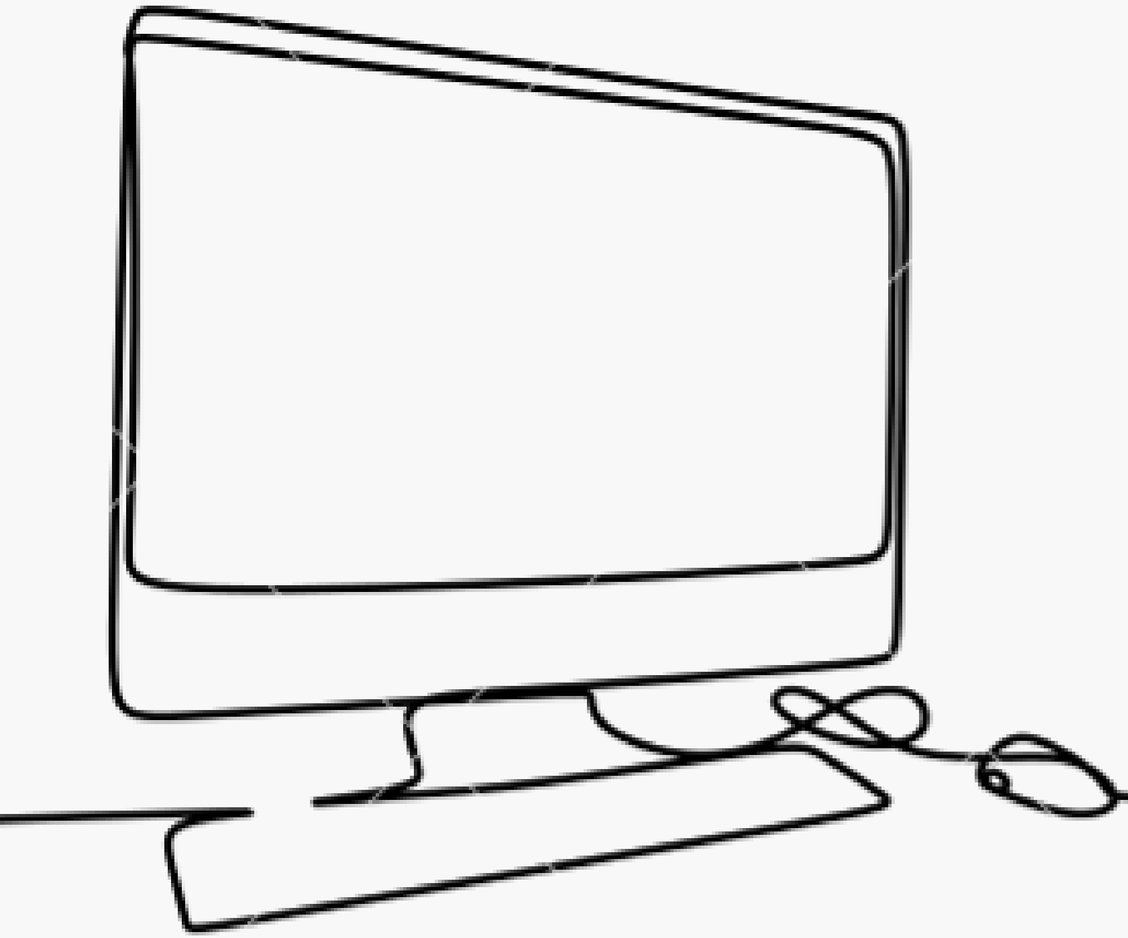
한쪽이 다른 쪽에게 어떠한 정보를 전혀 공유하지 않고, 그 정보에 대한 진실성이나 정확성을 입증하는 암호학적 기술
이 기술은 블록체인에서 개인정보 보호와 거래 비밀, 제3자에 대한 보안 등의 측면에서 중요한 역할을 합니다.

영지식 증명에서는 증명자(Prover)와 검증자(Verifier)라는 두 주체가 있습니다. 증명자는 검증자에게 특정 정보를 공개하지 않으면서 그 정보가 올바르다는 것을 증명하는 것이 목표이고 검증자는 증명자의 특정 정보를 직접 알지 못하는 상태에서 그 정보의 유효성을 확인하는 것이 목표입니다.

블록체인 기술에서 ZKP는 거래의 사생활 보호와 익명성을 보장하는데 사용되며, 대표적인 활용 사례로 Zcash, Monero 등의 암호화폐가 있습니다. 이 같은 활용을 통해 블록체인 기술의 확장성과 투명성을 유지하면서 개인정보와 거래 내용의 보안을 강화할 수 있습니다.

```python
import random

def prover_generate_random_number():
    return random.randint(1, 100)

def prover_calculate_square(random_number):
    return random_number ** 2

def verifier_check_square(random_number, square_result):
    return square_result == random_number ** 2

def zero_knowledge_proof():
    # 증명자가 임의의 숫자를 생성합니다.
    prover_random_number = prover_generate_random_number()

    # 증명자는 임의의 숫자의 제곱을 계산합니다.
    square_result = prover_calculate_square(prover_random_number)

    # 검증자는 증명자가 제공한 결과가 정확한지 확인합니다.
    is_proof_valid = verifier_check_square(prover_random_number, square_result)

    print(f"Prover's random number: {prover_random_number}")
    print(f"Square result: {square_result}")
    return is_proof_valid

if __name__ == "__main__":
    proof_valid = zero_knowledge_proof()
    if proof_valid:
        print("Verifier accepts the proof.")
    else:
        print("Verifier rejects the proof.")

```

PYTHON



ZKAPP

영지식 증명 기술을 사용하는 ZKApps는 개인 정보 및 데이터의 보호와 사생활 보호를 중점으로 한 블록체인 기반의 응용 프로그램을 구축합니다. 이를 통해 다양한 산업 분야에서 실제 활용이 가능하며, 현재 정부, 금융 서비스, 의료 서비스 등 다양한 분야에서 검토 중입니다.

```python
from py_ecc.bn128 import *
from py_ecc import bn128 as bn

# 원시 필드 상수(Scalar field constant) 설정
scalar_field = bn.curve_order

# 두 개의 임의의 점 생성
G = (1, 2)
H = (3, 4)

# 임의의 비밀 키(secret_key) 설정
secret_key = 9

# 행위자가 알고 있는 공개 키(public_key) 계산
public_key = multiply(G, secret_key)

# 증명값(proof) 계산
proof = multiply(H, secret_key)

# 검증자가 올바른 증명값을 받았을 경우, 다음 수식이 참이 된다.
point_x, point_y = proof
verifier_Gx, verifier_Gy = G
verifier_public_key_x, verifier_public_key_y = public_key

# 검증자는 secret_key * H (mod scalar_field) = proof를 확인한다.
assert (point_x - (verifier_public_key_x * point_x) % scalar_field) == 0
assert (point_y - (verifier_public_key_x * point_y) % scalar_field) == 0

print("올바른 증명값이 확인되었습니다.")
```

PYTHON