블록체인과 분산 암호화 기술에 의한 파일 불법 복제 방지와 소유권 부여 및 배포 시스템

특허출원 : 10-2021-0101701
블록체인 및 분산 암호화를 이용한 파일 유통 시스템

Atomixoft

필수 요소

1. 시스템은 파일이나 정보를 원본 그대로 복사하거나 저장하는 기능이 없다.
   단, 암호화된 파일이나 정보는 저장이 가능하다.
2. 시스템은 파일이나 정보에 대한 어떠한 출력을 할 수 없다.
   단, 시스템 내부에서는 파일이나 정보의 출력과 저장이 방지된 상태에서 복호화 하여 파일의 내용을 조회할 수 있다.
   조회된 내용을 출력을 할 때에는 화면 캡처 및 음원 캡처 등이 방지된 상태에서 출력할 수 있다.
3. 모든 암호값은 해시 처리하여 저장한다.
4. 해시 처리된 암호키는 시스템 내에서 비공개로 사용되며 해시 처리된 암호키의 값은 아무도 볼 수 없다.
5. 해시 처리된 암호키는 시스템 내에서만 사용되기 때문에 실질적인 암호키의 값은 본인을 포함하여 아무도 알 수 없도록 해야 한다.
6. 컨텐츠 출력은 큐빅스토리의 API로 만들어진 어플에서만 사용 가능하며 큐빅스토리 내에서 개인키로 복호화를 했을 때만 출력할 수 있다.
   복호화된 컨텐츠는 다른 이름으로의 저장 등 어떠한 방식으로 파일로 저장할 수 없으며 파일은 오로지 암호화된 상태로 존재해야만 한다.
7. 큐빅스토리 플랫폼의 어플리케이션이나 서비스는 화면 캡처 및 음원 캡처 모두 방지되어야 한다.
   매크로 방지를 비밀번호와 보안과 관련된 모든 정보는 보안 키패드로 입력할 수 있도록 해야 한다.
   VMWare와 같은 가상머신에서 어플이나 서비스를 실행할 수 없도록 해야 한다.
8. 파일은 사용소유권이 있는 사용자만 복호화 하여 조회할 수 있다.

목표

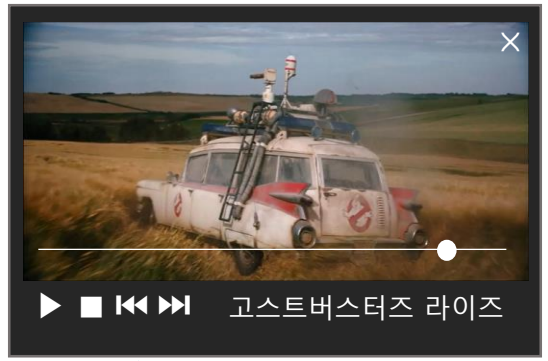
1. 각 노드에 저장되는 파일은 각 노드의 암호키에 의해서 암호화 된다.
2. 각 노드로 파일이 전파될 때 마다 고유의 암호키로 암호화 한다.
3. 큐빅스토리에서는 공개키의 의미는 누구나 공개된 키가 아니라 양방향 키로서 또 하나의 개인화 된 키로서 활용할 수 있도록 한다.
4. 공개키는 암호화할 때 사용 후 바로 파기되어 원본 파일로 복호화 되지 않도록 한다.
5. 개인화 된 공개키라고 하더라도 어떤 노드에 접속하더라도 동일하게 처리할 수 있도록 해야 한다.
6. 개인키는 조회할 때만 사용하도록 한다.
7. 공개키와 개인키는 사용자가 입력한 키값과 임의의 키를 합쳐서 만들어지며 해시 처리하여 실제로 암호화 사용된 키값을 알 수 없도록 한다.
   따라서 공개키 역시 개인화 레벨의 암호화로 처리된다.
8. 암호화를 할 때 파일의 내용을 추출하여 BASE64로 변환하여 암호화를 한다.
9. 블록체인 스마트컨트랙트에 의해서 소유권 번호가 발행되며 파일을 암호화 할 때 파일의 내용과 함께 암호화를 한다.
10. 개인키로 조회할 때 함께 암호화된 소유권 번호를 통해 조회를 하는 사용자가 소유권을 가지고 있는지 확인한다.

파일 암·복호화 과정

공개 암호키는 클라이언트와 노드 마다 다르게 설정되며 따라서 큐빅스토리 플랫폼에서 배포된 모든 파일은 각각 고유의 공개 암호키로 암호화 하게 된다.
공개 암호키와 개인키는 해시 처리하여 암호로 적용되며 적용 후에는 파기하여 노드의 사용자와 클라이언트는 실질적으로 적용된 암호키 값을 알 수 없도록 한다.
복호화된 파일 내용은 파일로 저장하지 않고 BASE64 상태에서 그대로 출력하도록 한다.

공개 암호키
전송하고자 하는 노드의 암호키 + [임의의 값] + 소유권 번호
암호키를 적용할 때 해시 처리 후 시스템 내부에서 비공개로 적용할 것

임의의 값 : 노드의 고유 암호값을 만들기 위해 시스템에 따라 선택적으로 적용할 수 있다.
적용 후 키값은 파기하여 알 수 없도록 한다.
ex) 전송받고자 하는 노드의 개인키, 해당 디바이스의 마더보드 시리얼 번호, 휴대폰 번호 등



파일 내용 출력 화면

파일 원본 → BASE64 변환 → 암호화 처리 → BWE92PQ.. → 복호화 처리 → BASE64 내용

암호화된 파일

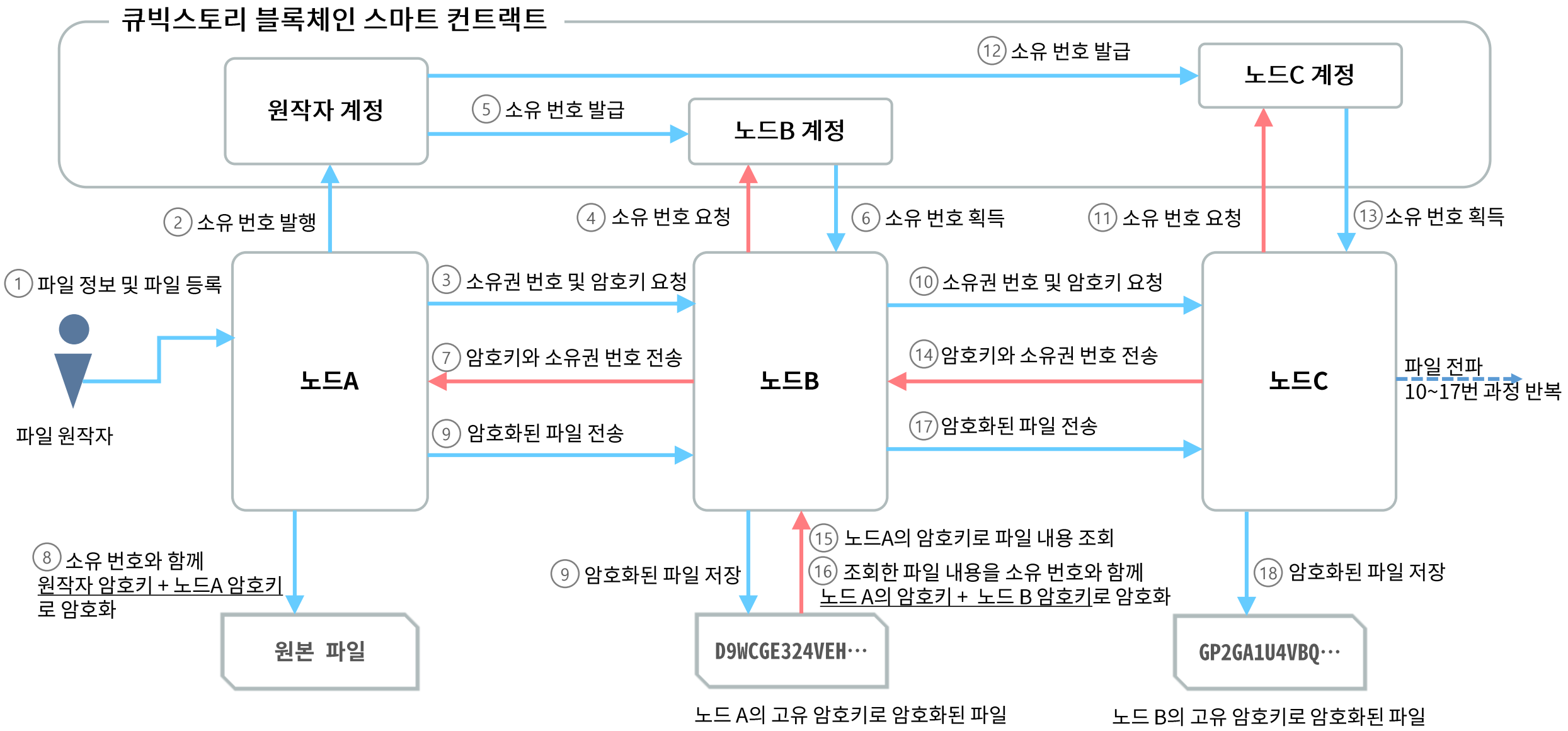
시스템(노드, 클라이언트 앱)

전송 받고자 하는 노드의 개인키
입력한 비밀번호(개인키) + 소유권 번호

암호키를 적용할 때 해시 처리 후 비공개로 적용할 것
적용 후 키값은 파기하여 알 수 없도록 한다.

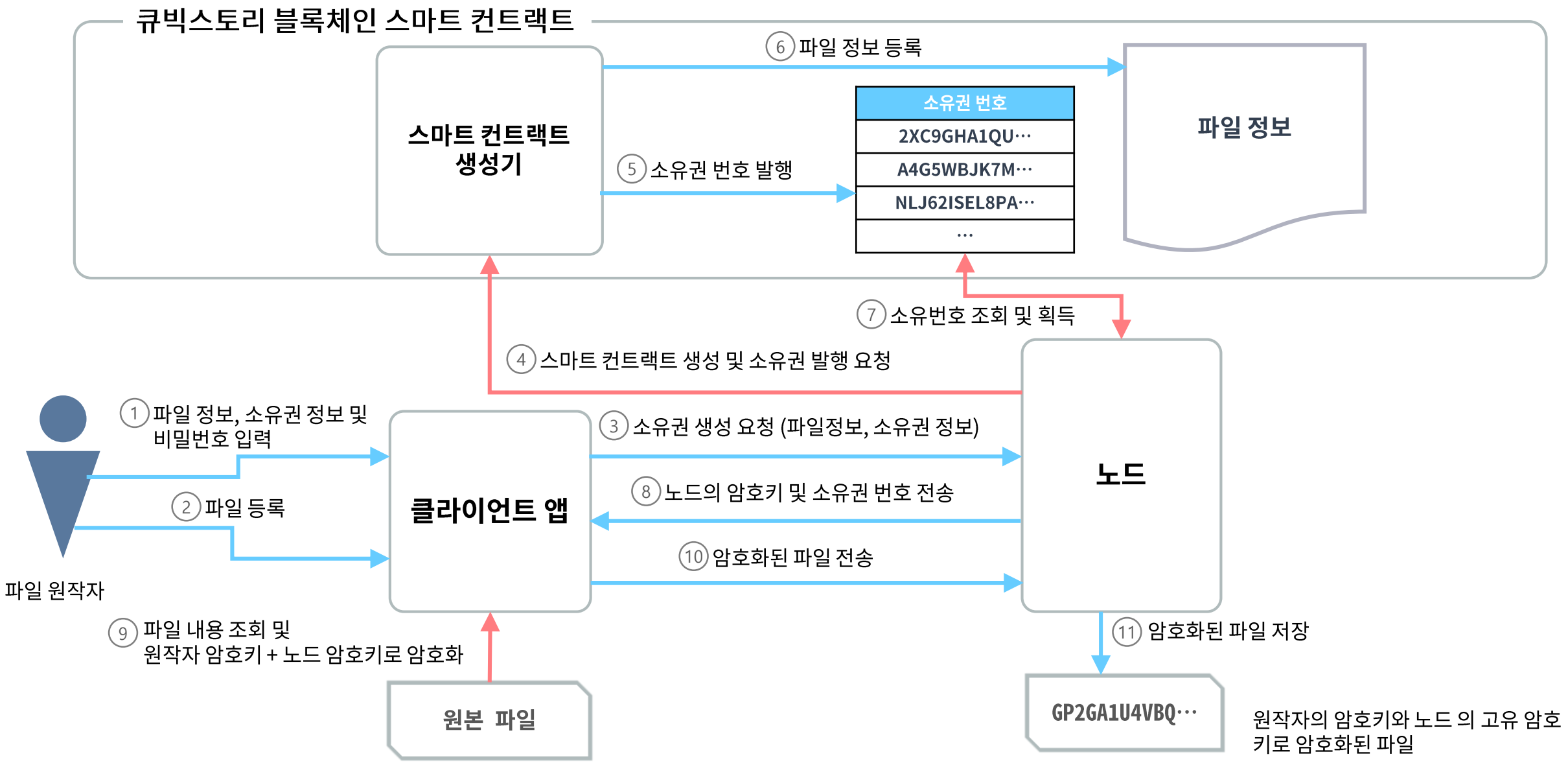
노드 상의 분산 암호화 처리 과정

배포하고자 하는 파일은 플랫폼의 모든 노드와 어플리케이션 내에선 각각의 고유의 암호키로 암호화하여 저장한다.
파일의 내용을 조회 할 때는 소유권 번호와 노드 또는 어플리케이션의 암호키에 의해서만 조회 가능하다.



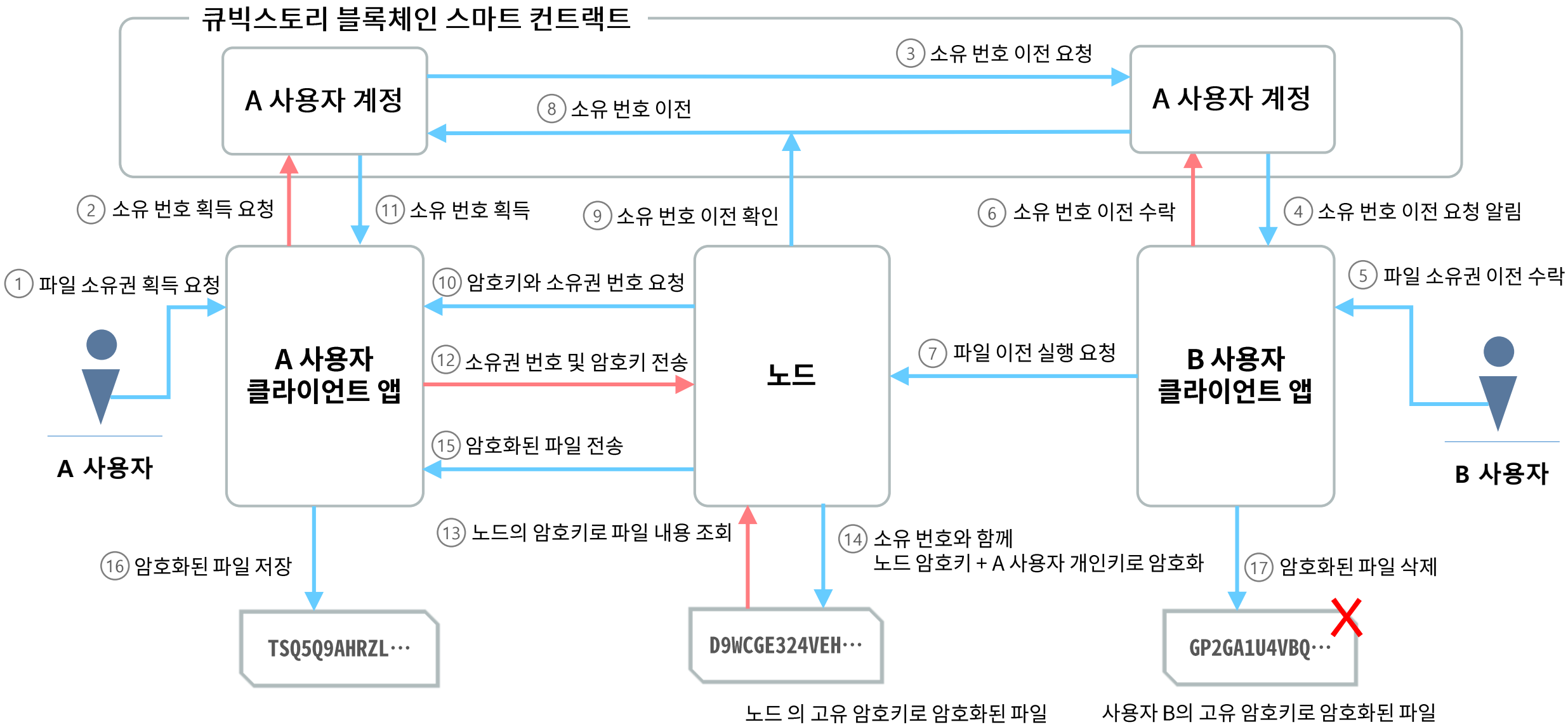
파일 등록 과정

스마트 컨트랙트는 파일 정보를 받고 소유권 번호를 발행하여 파일을 받고자 하는 사용자와 노드에게 발급하도록 한다..
각 노드 또는 사용자에게 배포시에는 소유권 번호를 발급하여 파일의 소유권을 보장하도록 한다.



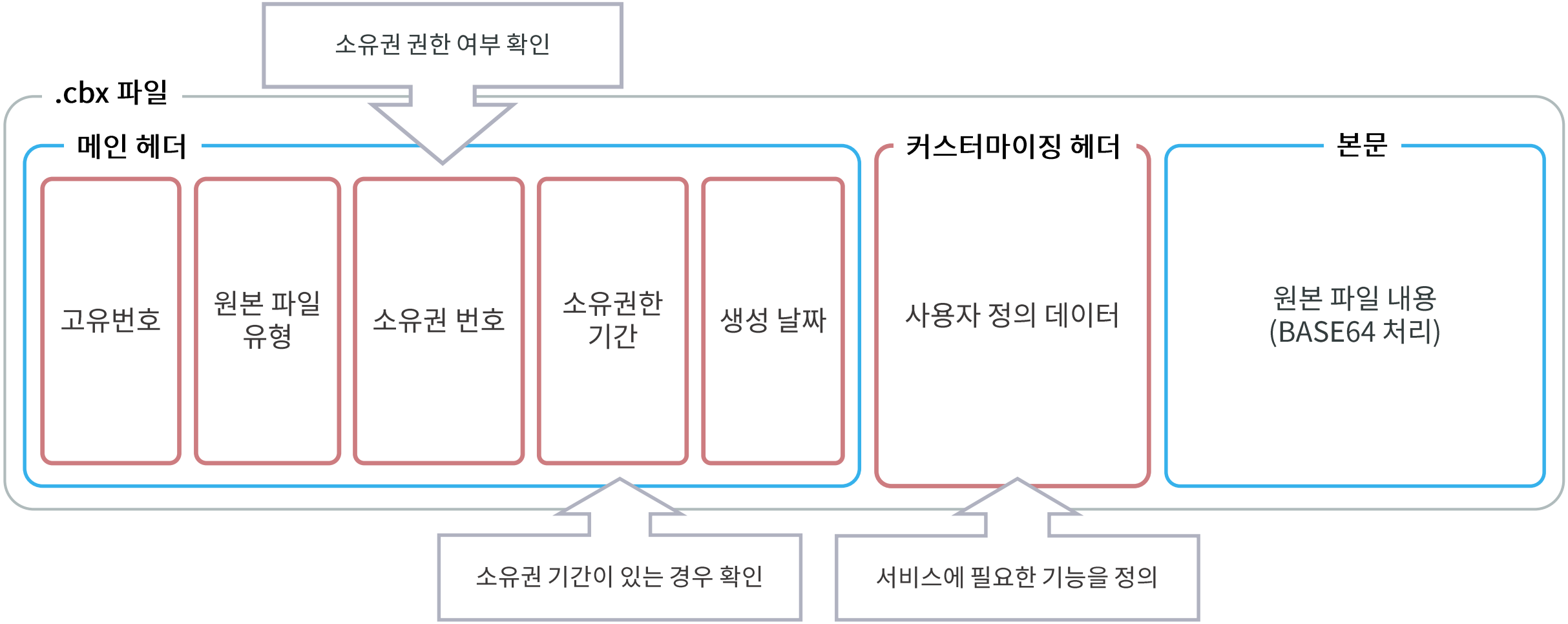
파일 소유권 이전

스마트 컨트랙트는 소유권을 사용자와 사용자 사이에 이전할 수 있으며 소유권이 이전되면 소유권의 원 사용자는 더 이상 파일을 조회할 수 없다.
소유권 이전과 함께 파일을 이전 받을 수 있으며 이때 분산 암호화를 통해 이전 받은 사용자의 암호키로 파일을 암호화 하고 자신의 암호키를 통해 파일 내용을 조회할 수 있다.



.cbx 파일

.cbx 파일은 보호해야 할 파일을 분산 암호화를 하여 처리된 파일로서 파일의 내용을 암호화 처리를 한 후에 시스템에서 사용할 정보들과 함께 하나의 파일로서 처리한다.
메인 헤더에는 분산암호화시 생성된 고유번호, 원본 파일에 대한 파일타입, 소유권 정보 등 서비스 시스템에서 이용할 정보들을 위해 할당된다.
커스터마이징 헤더는 서비스만의 기능을 정의하여 파일의 내용을 이용한 다양한 서비스를 처리할 수 있도록 할당된다.
본문은 파일의 내용을 위한 영역으로 1개 또는 다수의 파일을 저장하기 위해 할당된다.
메인 헤더와 커스터마이징 헤더 그리고 본문을 하나의 데이터로 처리한 후 암호화하여 하나의 .cbx 파일로 생성한다.

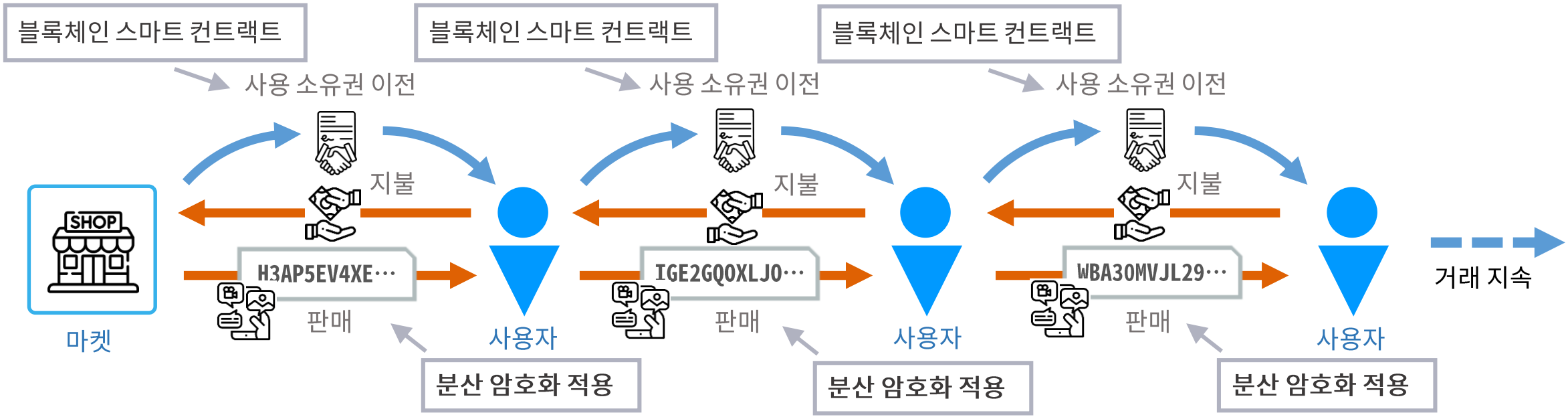


.cbx 파일처리

보호하고자 하는 파일은 큐빅스토리의 노드에 의해 소유권 부여 및 분산 암호화를 통해 .cbx 파일로 변환되며 큐빅스토리를 통해 복호화하여 파일 내용을 저장하고 콘텐츠를 조회한다.
큐빅스토리 플랫폼에서는 .cbx 파일에 의해서만 파일을 조회할 수 있으며 큐빅스토리의 캡처 방지와 녹음 방지 API가 적용된 콘텐츠 뷰어 또는 콘텐츠 플레이어에서만 조회 및 출력된다.

디지털 상품의 n차 거래

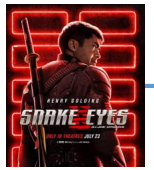
사용자와 사용자 사이에는 소유권 이전을 통해 거래를 할 수 있다.
n차 거래를 할 때 마다 파일은 사용자 각자의 암호키로 암호화 되어 파일을 받는다.



큐빅스토리 플랫폼 클라우드 – 파일 전파 및 배포

사용자는 랜덤하게 임의의로 지정된 노드에 접속하여 파일을 받을 수 있다.
큐빅스토리 플랫폼에서 사용자가 거래를 할 때마다 접속한 노드에서 암호화 되서 파일을 받는다.

특허 출원[출원번호 : 10-2021-0101701]
분산 암호화에 의한 암호화 파일 유통 기술
참고 : 큐빅스토리 기술 문서 참고 [다운로드]
특허출원증명서 참고 [다운로드]



동영상, 음악, 이미지, 텍스트 파일, HTML 게임 등

암호화 후 파일 등록

큐빅스토리 플랫폼 클라우드 네트워크

A 노드
H3AP5EV4XE…
A 노드 암호화 컨텐츠 파일

암호화 후 파일 전송

C 노드
H3AP5EV4XE…
C노드 암호화 컨텐츠 파일

D 노드
WBA30MVJL29…
D노드 암호화 컨텐츠 파일

암호화 후 파일 전송

B 노드
IGE2GQ0XLJ0…
B노드 암호화 컨텐츠 파일

암호화 후 파일 전송

B 사용자의 암호화 후 파일 다운로드

A 사용자의 암호화 후 파일 다운로드

복호화 후 조회

A 사용자

N2UHEW1KM…
A 사용자 암호화 컨텐츠 파일

복호화 후 조회

B 사용자

CA8Y3TLIF…
B 사용자 암호화 컨텐츠 파일

암호화된 파일 배포 결과

파일의 배포 또는 거래가 완료되면 각 노드와 사용자는 동일한 파일에 대해서 각기 다른 암호화로 파일을 소유하게 된다.
따라서 복제를 하더라도 해당 노드 또는 사용자의 소유권과 암호키가 없으면 조회가 불가능하다.

파일

H3AP5EV4XE···

A 노드

A 노드 암호화 파일

IGE2GQOXLJO···

A 사용자
클라이언트

A 사용자 클라이언트
암호화 파일

B노드 암호화 파일

WBA30MVJL29···

B 노드

A 서버 암호화 파일

XU810GOS7NF···

A 서버

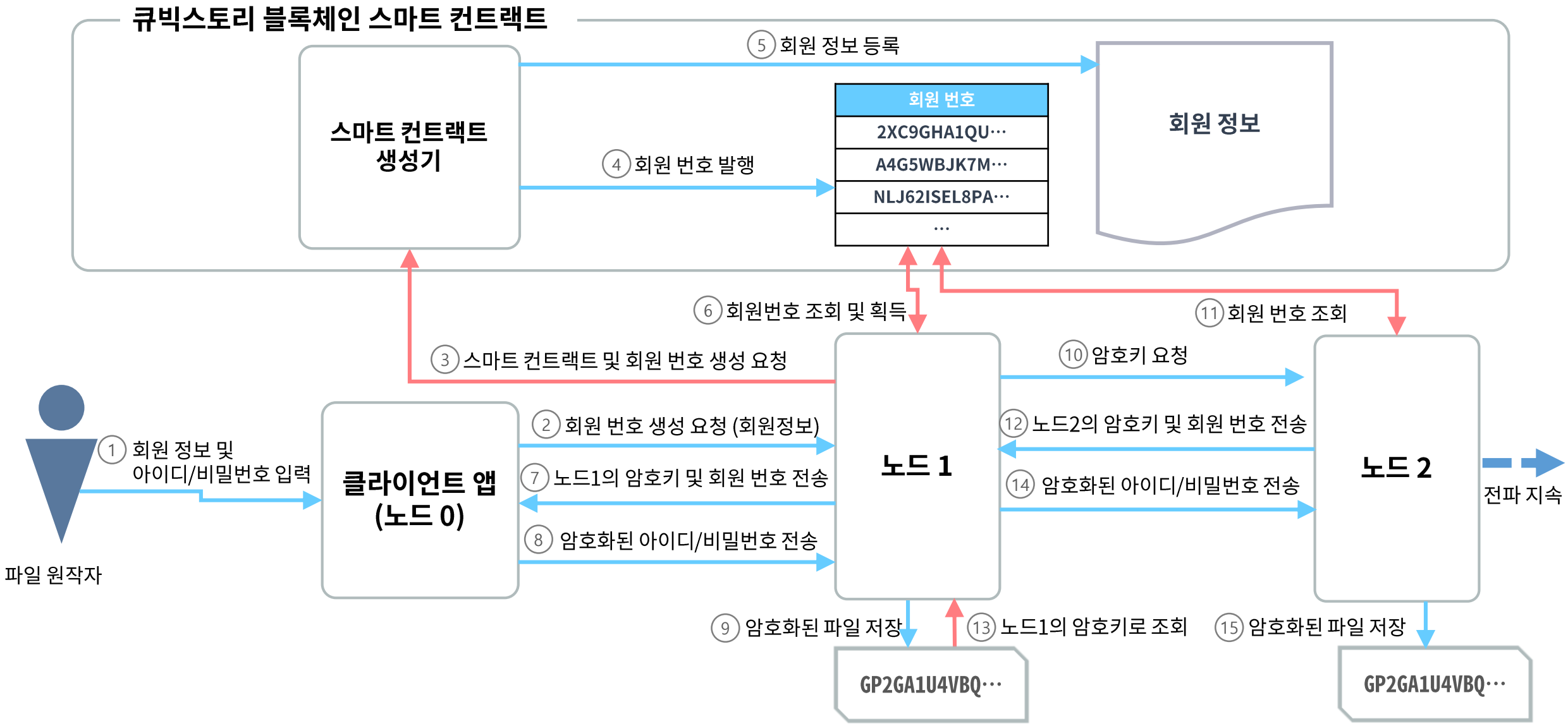
B 사용자 클라이언트
암호화 파일

T9V81ENHYLA···

B 사용자
클라이언트

응용 : 개인정보(노드 보안정보) 관리(1)

개인 정보 또는 노드(블록체인 노드, DB서버, 웹서버 등)의 보안 정보를 분산 암호화를 통해 저장하면 강력한 보안 체계로 개인정보를 관리할 수 있다.
파일의 분산 암호화와 소유권 부여 방법과 동일하게 처리하면 개인 및 노드의 보안 정보를 안전하게 보호하면서 이용할 수 있다.



응용 : 개인정보(노드 보안정보) 관리(2)

사용자나 노드에서 회원 정보 또는 노드(블록체인 노드, DB서버, 웹서버 등)의 보안 정보를 이용하기 위해서 어떤 노드에서도 보안 기능(로그인 등)을 처리할 수 있도록 할 때 분산 암호화를 통해서 보다 안전하고 강력한 보안성으로 처리할 수 있다.
또한 회원 정보 또는 노드의 보안 정보도 분산 암호화 되어 각 노드에 배치되고 있기 때문에 보안이 강화된 상태에서 어느 노드에 접속하든 안전하게 조회 또는 이용할 수 있다.

사용사례 : 강력한 보안이 필요한 개인 정보 서비스 시스템, 암호화폐 비밀번호 관리, 보안이 필요한 SSO 기능, 정품소프트웨어 인증 등