2017/01/23, 2017 동계 세미나-블록체인

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto, 2008

이부형 (boohyung@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

Contents

- Introduction
- Proposed Scheme
- Conclusion

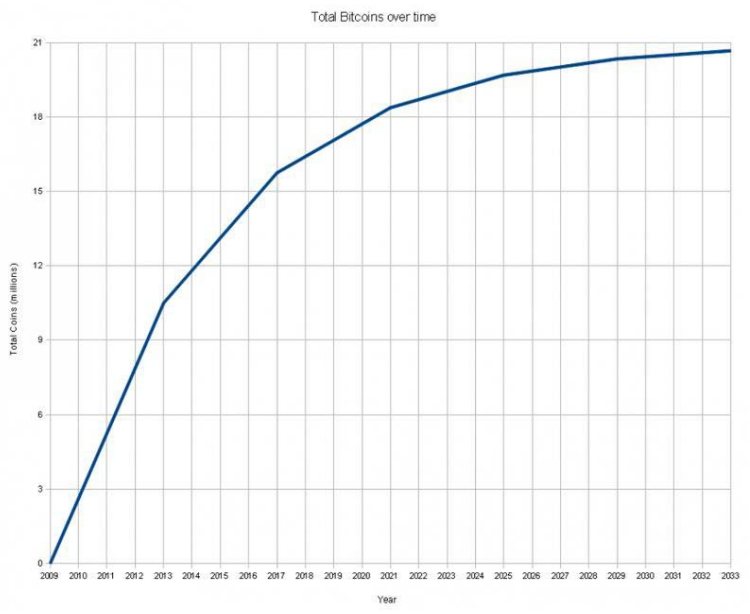
Introduction (1/2)

- 연구배경
  - 과거의 전자상거래 모델: TTP를 사용하여 사용자간의 신뢰를 바탕으로 하는 거래 방식
    - TTP (Trusted Third Party): 제 3의 신뢰기관
    - TTP의 거래 중재를 위해 사용자는 많은 비용을 지불해야 함

- **비트코인: 암호화 기술에 기반한 전자화폐 시스템**
  - 암호화 기술: 해시 함수 & 디지털 서명
  - TTP없이 당사자 간 직접적인 거래를 가능하게 함
  - 이중지불 문제 방지: P2P 분산 네트워크 기반 타임스탬프 서버를 가지는 체인 구조 사용

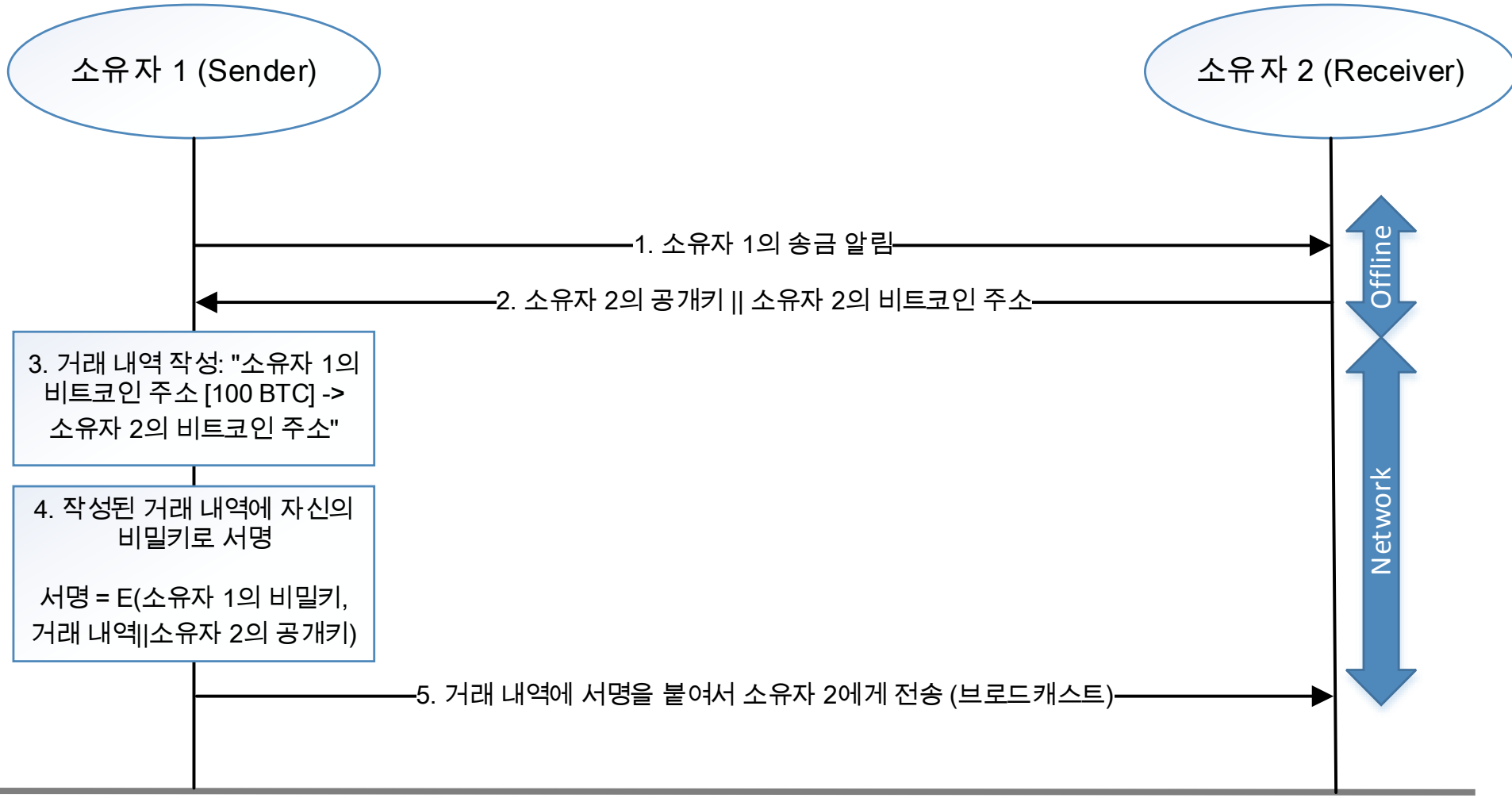
Introduction (2/2)

- 비트코인의 특징
  - 공개키 암호화 기술에 기반
  - 사용자 간의 자발적인 참여를 전제로 하는 P2P 분산 네트워크를 통해 거래 내역을 검증
  - 통화 공급량 고정: 최대 2100만 BTC
  - 오픈 소스: 누구나 수정 및 사용이 가능함



Proposed Scheme (1/22)

- Transactions
  - 예. 소유자 1이 소유자 2에게 100 BTC를 송금할 때

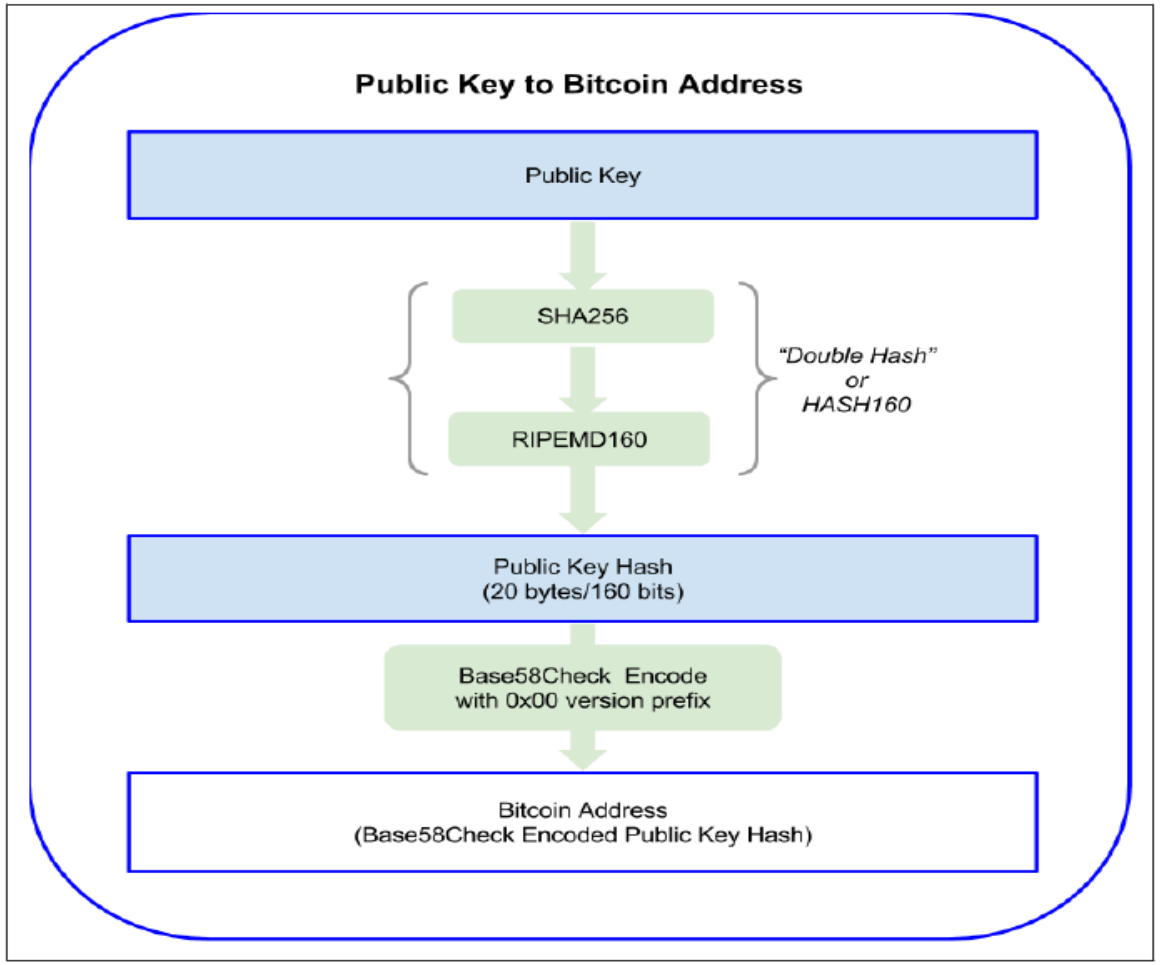


Proposed Scheme (2/22)

- Transactions
  - 비트코인 주소
    - 발신자의 주소: 송신자의 공개키를 기반으로 만듬
    - 수신자의 주소: 수신자의 공개키를 기반으로 만듬
  - 서명 작성과 검증 (ECDSA)
    - 거래를 하기 전에 소유자 1은 거래 내역에 자신의 개인키로 디지털 서명
      - E(소유자 1의 개인키, 거래 || 소유자 2의 공개키)
    - 소유자 2를 포함한 검증자들은 소유자 1의 공개키를 이용하여 서명을 검증
  - 사용자가 원할 경우 비밀번호를 따로 지정하여 비트코인 지갑을 암호화할 수 있음
    - 비트코인 거래 전용 소프트웨어를 사용 (예. Bitcoin Core)

Proposed Scheme (3/22)

- Transactions
  - 공개키 -> 비트코인 주소 변환 방법



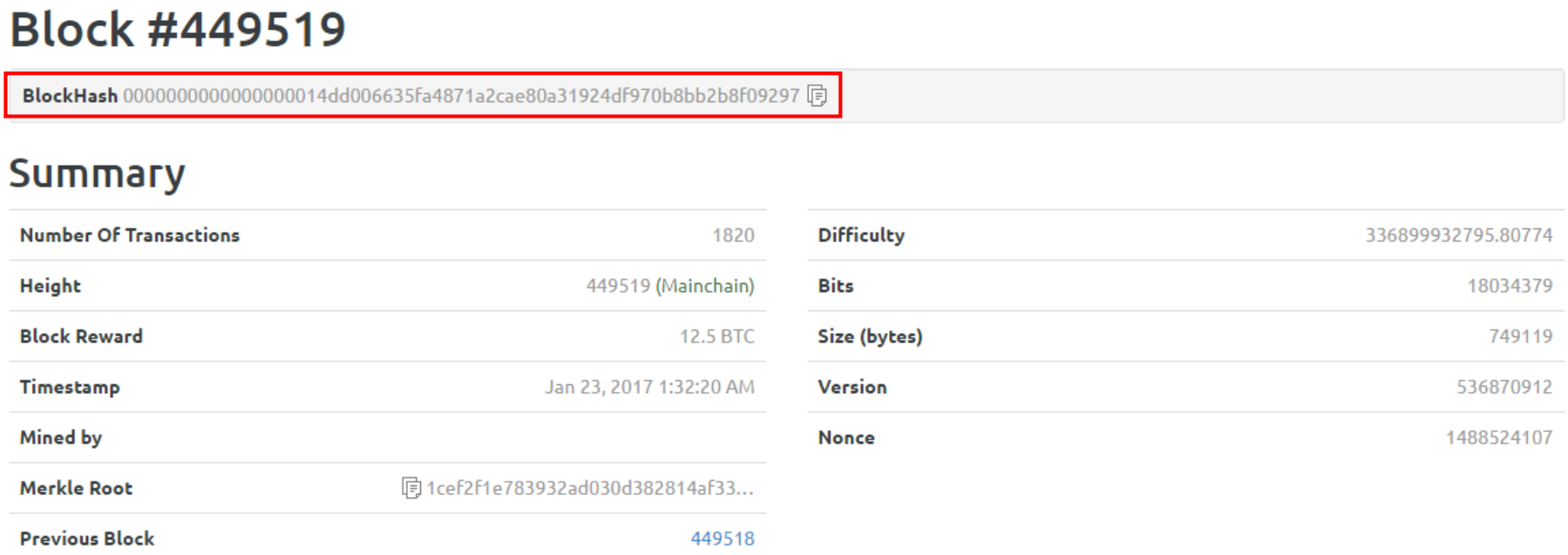
Proposed Scheme (4/22)

- Transactions
  - 비트코인 거래의 특징에 따라 제공되는 기능

특징	기능
모든 노드에게 거래 내역이 공개됨	투명성
디지털 서명 사용	무결성, 부인봉쇄
거래 내역 자체를 암호화하지 않음	기밀성X

Proposed Scheme (5/22)

- Timestamp Server
  - 블록을 발행할 때 블록 헤더에 타임스탬프를 포함하고, 블록 헤더의 해시값을 네트워크에 공개
    - 타임스탬프는 그 시각에 데이터가 존재했음을 입증
    - 과거 거래기록의 누적



Block #449519

BlockHash 00000000000000000014dd006635fa4871a2cae80a31924df970b8bb2b8f09297

Summary

Number Of Transactions	1820	Difficulty	336899932795.80774
Height	449519 (Mainchain)	Bits	18034379
Block Reward	12.5 BTC	Size (bytes)	749119
Timestamp	Jan 23, 2017 1:32:20 AM	Version	536870912
Mined by		Nonce	1488524107
Merkle Root	1cef2f1e783932ad030d382814af33...		
Previous Block	449518		

Proposed Scheme (6/22)

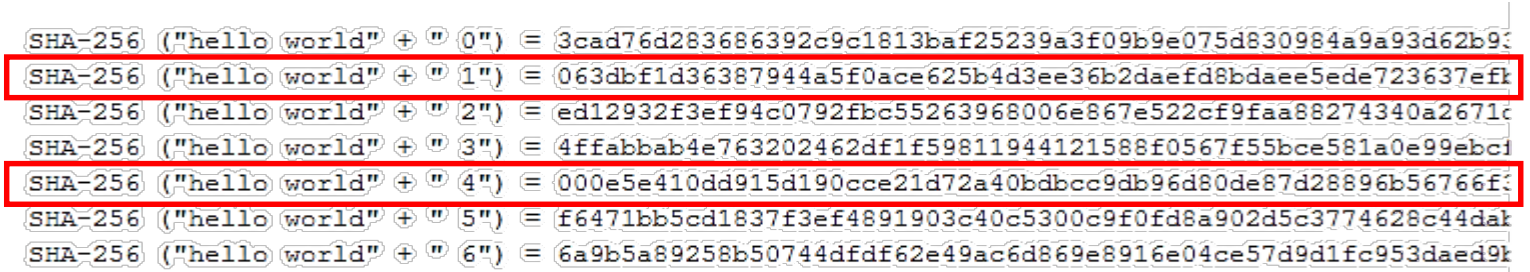
- Proof-of-Work
  - PoW, 작업 증명
  - 일정 시간이 걸리는 연산작업을 통해 거래 사실을 선의의 노드들이 스스로 증명하는 절차를 의미
    - SHA-256 알고리즘으로 다수의 0비트들로 시작되는 암호화 해시값 x를 찾는 과정
    - 작업에 걸리는 시간은 노드가 소유한 컴퓨팅 파워에 따라 달라짐
    - 평균 작업시간: x의 연속되는 0비트의 요구 개수에 따라 지수적으로 증가됨

Proposed Scheme (7/22)

- Proof-of-Work
  - 동작 과정
    - 10분 단위로 발생한 거래를 모두 묶어서 하나의 블록으로 생성한 후 네트워크 내 모든 노드들에게 브로드캐스트
    - 임의의 노드들은 생성된 블록을 검증하기 위해 연산 작업을 수행 (예. $block\ n$에 대한 PoW)
      - $h(h(n-1\ th\ block\ header)||nonce) < x$
        - nonce = 0부터 시작하여 조건을 만족하는 x를 찾을 때까지 1씩 증가하는 값
        - x = 다수의 0비트들로 시작되는 특정한 해시값

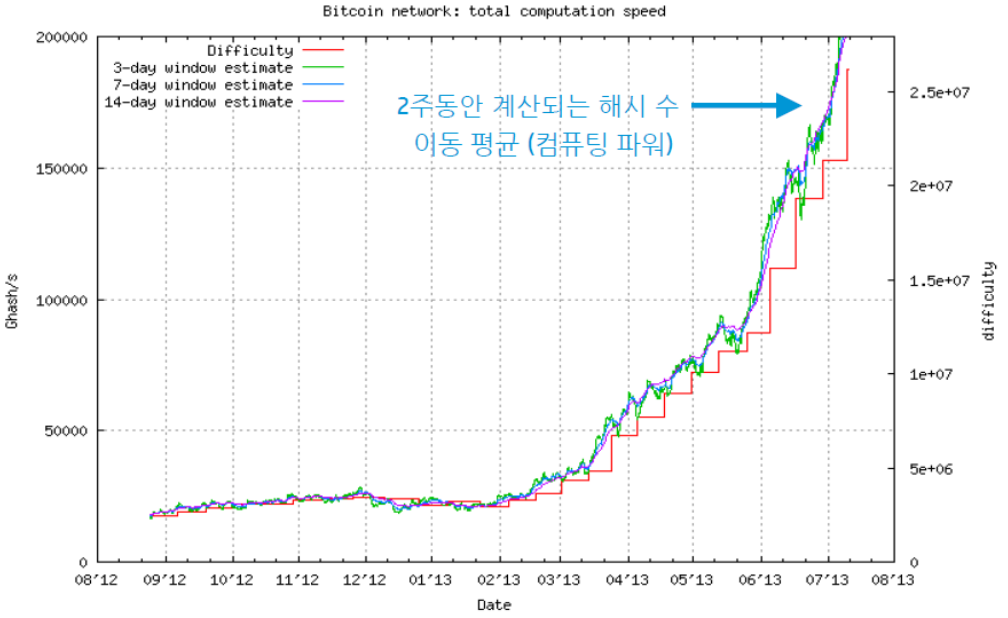
Proposed Scheme (8/22)

- Proof-of-Work
  - 동작 과정
    - 예시: SHA-256(“hello world” || nonce)
      - brute force방식으로 nonce값을 변화시켜가며 해시값을 계산
      - 비트코인에서의 hashcash: 몇 개 이상의 0으로 시작하는 해시값을 만족하는 nonce를 찾는 문제
        - 문자열 대신 블록체인에 추가된 가장 최근 블록의 헤더를 이용
        - 0의 개수를 늘려 난이도를 조정
        - 네트워크 내의 정당한 노드들이 참여하여 hashcash를 해결하는데 총 10분이 소요되도록 유지



Proposed Scheme (9/22)

- Proof-of-Work
  - 0의 자릿수에 따른 난이도
    - 앞자리의 0의 수가 많을 수록 조건을 만족하는 해시값을 찾는 것은 기하급수적으로 어려워짐
    - 난이도 조절은 약 2주마다 업데이트
      - 10분마다 채굴이 성공하면 2주동안 2016개의 블록이 만들어짐
      - 2016개의 블록이 2주일이 되기도 전에 만들어지면 난이도가 올라가고, 2주일보다 늦어지면 난이도가 내려감

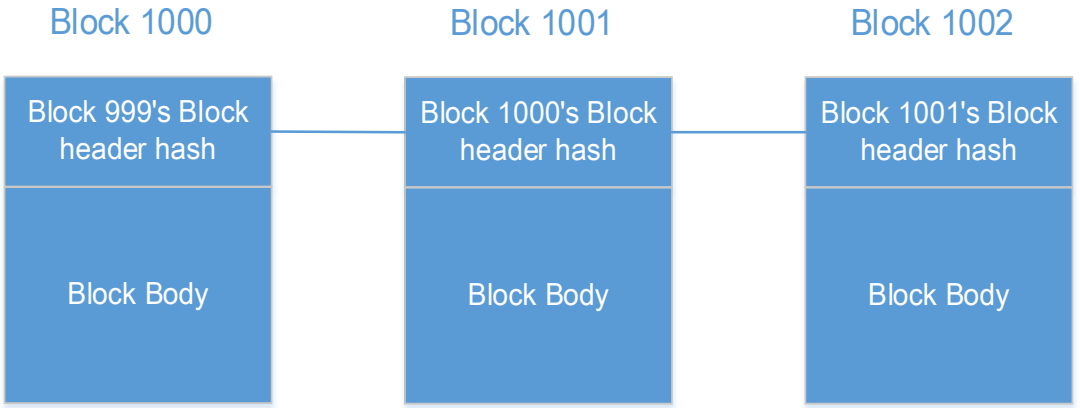


Proposed Scheme (10/22)

- Network
  - 새로운 거래가 발생하면 모든 노드에게 브로드캐스트
  - 각각의 노드들은 새로운 거래 내역들을 블록에 취합
    - 아직 승인되지 않은 블록
    - 10분동안 발생한 모든 거래는 블록에 취합됨
  - 만약, 하나의 노드가 블록에 대한 작업 증명 과정을 성공적으로 수행한다면 모든 노드에게 그 블록을 브로드캐스트
    - 정당한 거래 내역이라면 승인
      - 블록 내의 모든 거래가 이전에 쓰이지 않고 유효한 경우에만 승인
      - 6번의 승인이 발생할 경우 유효한 블록으로 인정됨
  - 다음 블록을 생성할 경우 이전 블록 헤더의 해시값을 다음 블록의 헤더에 넣어 체인 형태로 구성
    - 이전 블록이 정당한 블록임을 승인한다는 의사표현

Proposed Scheme (11/22)

- Network
  - Confirmation의 정의
    - 첫 번째 Confirmation: 임의의 노드의 블록 마이닝 (발행)
    - 두 번째~ : 다음 블록의 발행
      - 다음 블록이 발행될 때 마다 현재 블록의 Confirmation 횟수를 1씩 증가시킴
      - Confirmation = 6 이라면 더 이상 횟수 증가를 하지 않음
        - 모두가 신뢰할 수 있는 거래
        - 예. Block 1000은 Block 1005가 발행된 시점에서 Confirmation counting을 하지 않음



	마이닝에 따른 Confirmation 횟수		
	Block 1000	Block 1001	Block 1002
Block 1000의 발행	1	.	.
Block 1001의 발행	2	1	.
Block 1002의 발행	3	2	1

Proposed Scheme (12/22)

- Network
  - Why 6 Confirmations?
    - 이중지불 공격에 대한 Confirmation 횟수에 따른 공격 성공 확률
      - 가정1: 공격자는 전체 네트워크에서 차지하는 컴퓨팅 파워(hashrate) 중 10% 미만을 보유하고 있음
      - 가정2: 공격 성공 확률이 0.1% 미만이라면 공격에 대한 가능성을 고려하지 않음

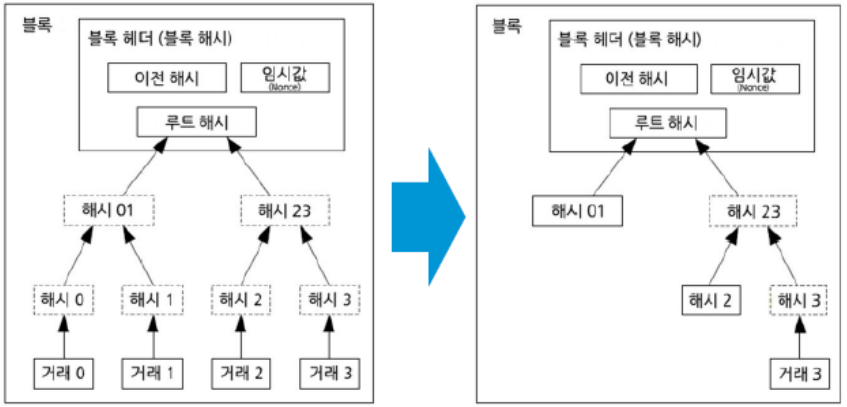
q	1	2	3	4	5	6	7	8	9	10
2%	4%	0.237%	0.016%	0.001%	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
4%	8%	0.934%	0.120%	0.016%	0.002%	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
6%	12%	2.074%	0.394%	0.078%	0.016%	0.003%	0.001%	≈ 0	≈ 0	≈ 0
8%	16%	3.635%	0.905%	0.235%	0.063%	0.017%	0.005%	0.001%	≈ 0	≈ 0
10%	20%	5.600%	1.712%	0.546%	0.178%	0.059%	0.020%	0.007%	0.002%	0.001%
12%	24%	7.949%	2.864%	1.074%	0.412%	0.161%	0.063%	0.025%	0.010%	0.004%
14%	28%	10.662%	4.400%	1.887%	0.828%	0.369%	0.166%	0.075%	0.034%	0.016%
16%	32%	13.722%	6.352%	3.050%	1.497%	0.745%	0.375%	0.190%	0.097%	0.050%
18%	36%	17.107%	8.741%	4.626%	2.499%	1.369%	0.758%	0.423%	0.237%	0.134%
20%	40%	20.800%	11.584%	6.669%	3.916%	2.331%	1.401%	0.848%	0.516%	0.316%
22%	44%	24.781%	14.887%	9.227%	5.828%	3.729%	2.407%	1.565%	1.023%	0.672%
24%	48%	29.030%	18.650%	12.339%	8.310%	5.664%	3.895%	2.696%	1.876%	1.311%
26%	52%	33.530%	22.868%	16.031%	11.427%	8.238%	5.988%	4.380%	3.220%	2.377%
28%	56%	38.259%	27.530%	20.319%	15.232%	11.539%	8.810%	6.766%	5.221%	4.044%
30%	60%	43.200%	32.616%	25.207%	19.762%	15.645%	12.475%	10.003%	8.055%	6.511%
32%	64%	48.333%	38.105%	30.687%	25.037%	20.611%	17.080%	14.226%	11.897%	9.983%
34%	68%	53.638%	43.970%	36.738%	31.058%	26.470%	22.695%	19.548%	16.900%	14.655%
36%	72%	59.098%	50.179%	43.330%	37.807%	33.226%	29.356%	26.044%	23.182%	20.692%
38%	76%	64.691%	56.698%	50.421%	45.245%	40.854%	37.062%	33.743%	30.811%	28.201%
40%	80%	70.400%	63.488%	57.958%	53.314%	49.300%	45.769%	42.621%	39.787%	37.218%
42%	84%	76.205%	70.508%	65.882%	61.938%	58.480%	55.390%	52.595%	50.042%	47.692%
44%	88%	82.086%	77.715%	74.125%	71.028%	68.282%	65.801%	63.530%	61.431%	59.478%
46%	92%	88.026%	85.064%	82.612%	80.480%	78.573%	76.836%	75.234%	73.742%	72.342%
48%	96%	94.003%	92.508%	91.264%	90.177%	89.201%	88.307%	87.478%	86.703%	85.972%
50%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Proposed Scheme (13/22)

- Incentive
  - P2P 네트워크의 신뢰성 확보에 기여하는 참여자들에게 제공되는 보상 체계 마련
    - 정당한 노드들의 생태계 형성에 도움을 줌
  - 인센티브를 얻는 경우
    - 채굴 (마이닝): 블록의 최초 생성자에 대한 보상
      - 컴퓨팅 자원과 전력을 소비한 대가, 2017년 기준 12.5 BTC
      - 채굴이 완료된 하나의 블록에 포함된 모든 거래 수수료의 합 = 채굴 수수료
    - 거래 수수료
      - 빠른 거래 승인과 채굴자들의 수고비로써 거래 당사자들이 거래 시 자발적으로 지급 (권장 사항)
      - 거래 금액에 따라 수수료가 달라짐
      - 거래 수수료 ≠ 거래소 수수료
        - 거래소 수수료: 거래소에서 비트코인 환전 시 사용자가 부담하는 수수료

Proposed Scheme (14/22)

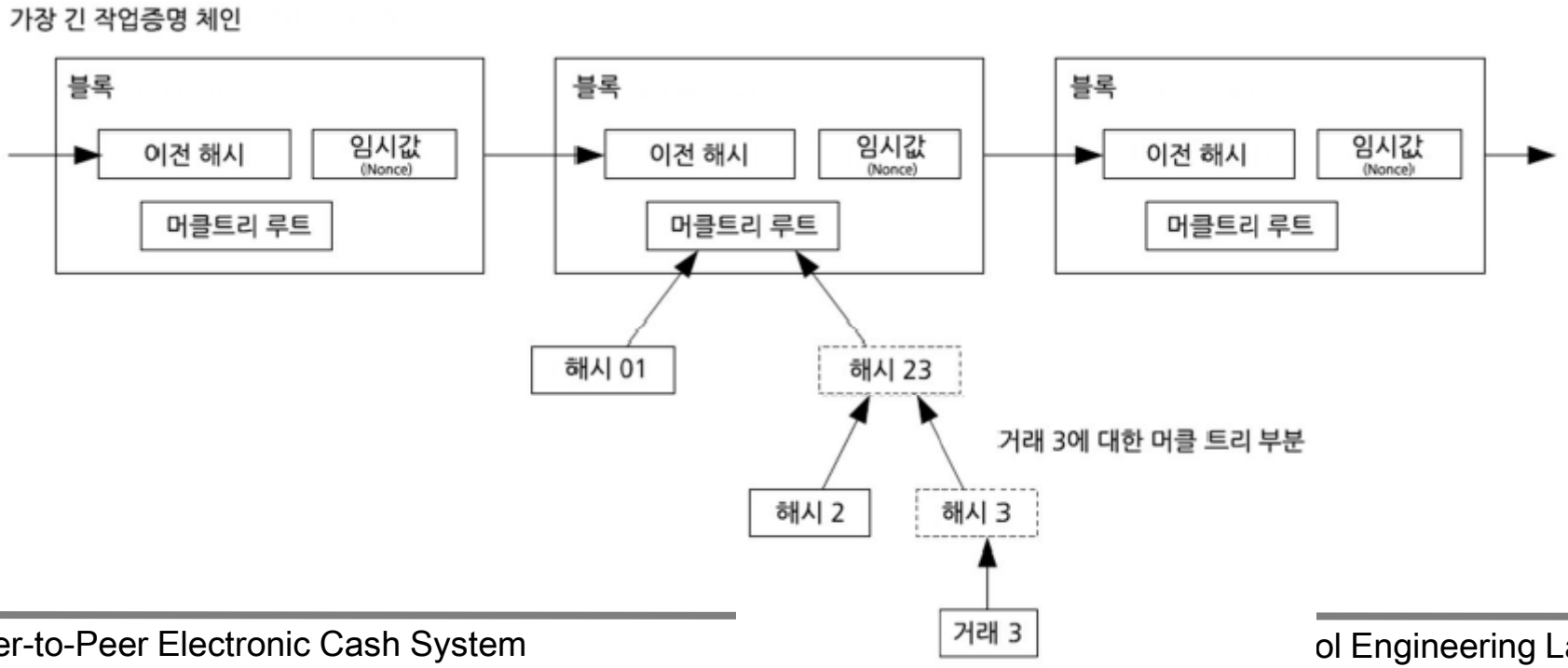
- Reclaiming Disk Space
  - 비트코인 사용자는 모든 블록정보를 로컬에 저장해야 함
    - 10분마다 블록 생성
      - 매년 4.2MB 필요 (80 * 6 * 24 * 365)
    - 저장공간의 한계가 발생
  - 거래 내역은 머클 트리 구조로 해시되고, 머클 루트만 블록 헤더 해시에 포함됨
    - 트리 구조에서 오래된 거래 내역에 대한 가지를 쳐내는 작업을 수행하여 저장 공간을 재확보



머클트리 구조로 해시된 거래 내역들 블록에서 거래 내역 0~2를 제거 후

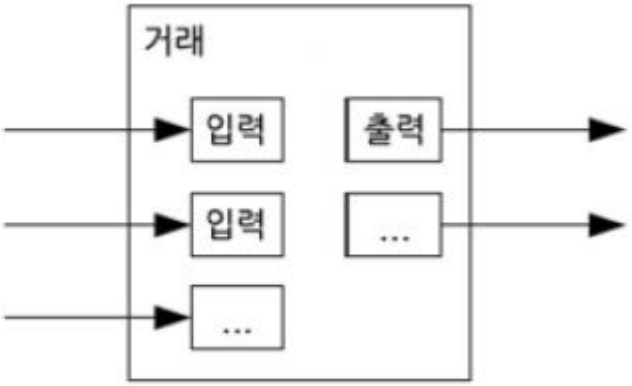
Proposed Scheme (15/22)

- Simplified Payment Verification
  - 지불 입증의 간소화 방안
    - 전체 거래내역을 일일히 확인하지 않고, 네트워크 내 노드들이 해당 거래를 승인했는지 여부를 확인
    - 가장 긴 블록체인의 블록 헤더 정보가 공개됨
      - 입증할 거래 내역이 기록된 블록의 머클트리를 통해 확인 가능



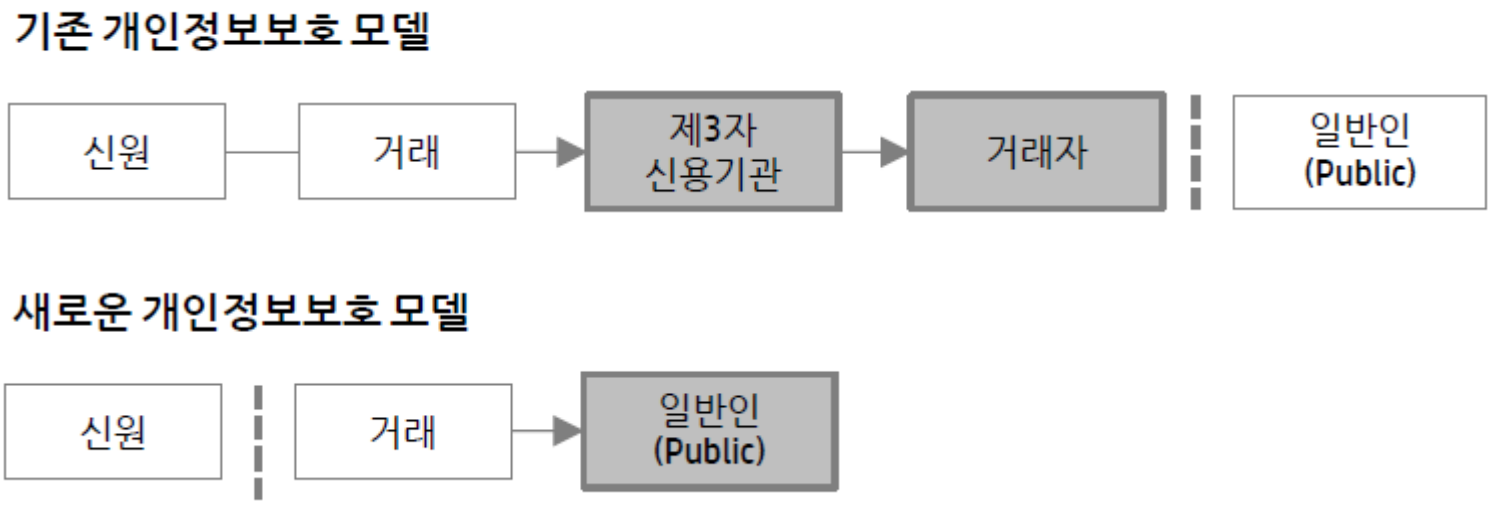
Proposed Scheme (16/22)

- Combining and Splitting Value
  - 거래 내역은 복수의 입력과 출력으로 구성됨 (fan-out)
    - 입력: 사용자가 입력한 금액
    - 출력: 입력한 금액에서 소비한 금액이나 거스름돈
      - 출력을 사용할 수 있는 권한은 거래 내역 내의 수신자 주소에 대응하는 수신자의 비밀키를 통해 획득 가능



Proposed Scheme (17/22)

- Privacy
  - 사용자를 특정할 수 없게 하여 프라이버시를 보호
    - 거래할 때 사용되는 비트코인 주소
      - 사용자의 공개키를 기반으로 생성됨
      - 사용자의 개인정보가 포함되지 않음
    - 개별 거래마다 새로운 공개키-비밀키 쌍을 사용
  - 개인정보보호 모델의 변화



Proposed Scheme (18/22)

- Calculations
  - 비트코인에 대한 공격과 보안
    - 공격: 정당하지 못한 방법을 통한 비트코인의 획득
      - “Double-spending”, 이중 지불
      - 51% Attack: 전체 네트워크의 절반을 넘는 CPU 파워를 공격자가 가진다면 전체 블록체인의 조작이 가능
      - Finney Attack: 51% Attack이 현실적으로 어렵기 때문에 악의적인 miner와 협력하여 정직한 노드보다 체인을 빠르게 생성
  - 공격 시나리오
    - 시나리오 1: 공격자가 다른 체인의 갈래를 빠르게 생성하려고 시도하여 정직한 노드들의 체인을 앞질러 가장 긴 체인을 생성
    - 시나리오 2: 수신자가 일시적으로 돈을 받았다고 믿게 만들고, 일정 시간 뒤에 다시 돈을 자기 자신에게 되돌리려는 발신자의 시도

Proposed Scheme (19/22)

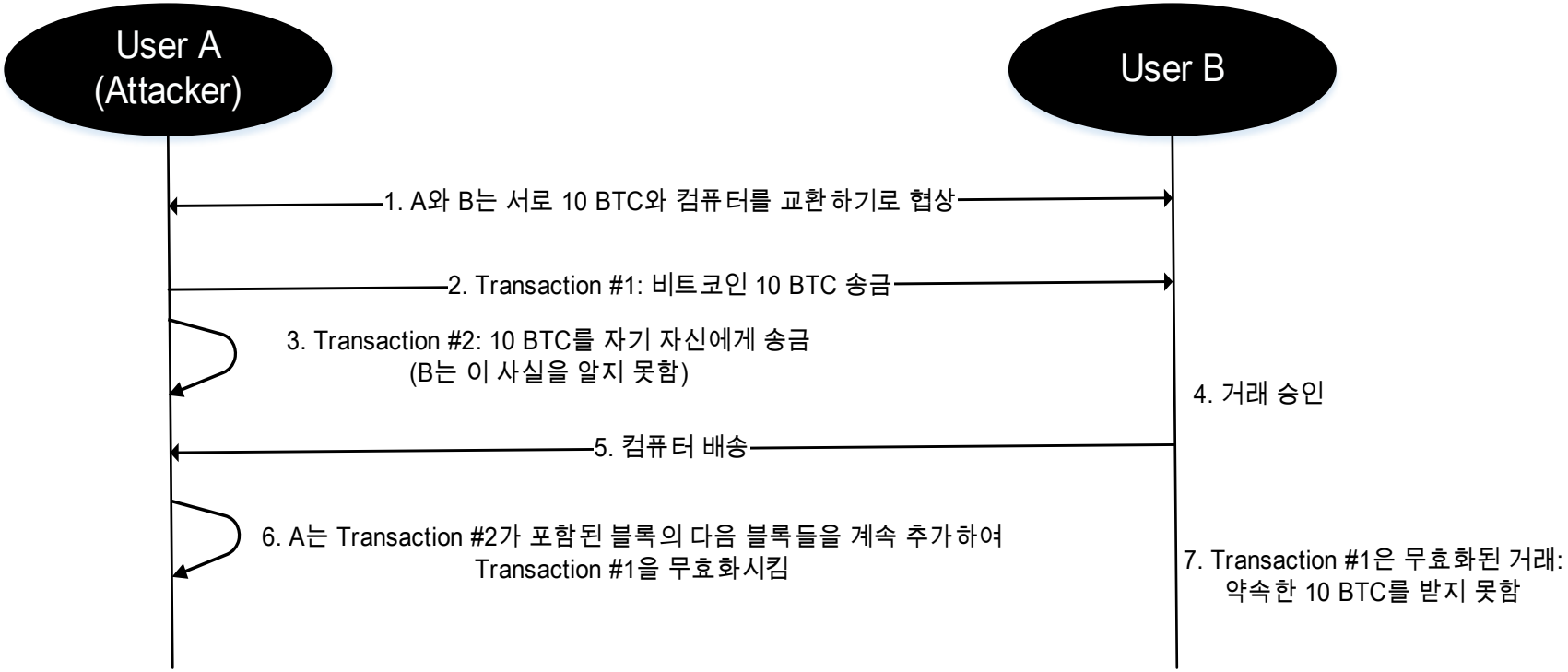
- Calculations
  - 시나리오 1: 정직한 노드가 생성하는 체인과 공격자가 생성하는 체인의 속도 경쟁
    - 두 체인 간의 길이 차이는 이항 분포를 따르게 됨
      - 길이 차이가 +1 or -1
      - $p =$ 정직한 노드가 다음 블록을 만들 확률
      - $q =$ 공격자 노드가 다음 블록을 만들 확률
      - $q_z =$ 공격자가 정당한 체인을 따라잡을 확률(z는 공격자가 따라잡은 정당한 체인의 길이)
    - 공격 횟수 k를 무한대로 늘리면 포아송 근사에 의해 포아송 분포로 계산이 가능
      - $\mu = z\frac{q}{p}$
      - $\sum_{k=0}^{\infty} \frac{\mu^k e^{-\mu}}{k!} \begin{cases} \left(\frac{q}{p}\right)^{z-k} & if\ k \le z \\ 1 & if\ k > z \end{cases}$

Proposed Scheme (20/22)

- Calculations
  - 시나리오 1: 정직한 노드가 생성하는 체인과 공격자가 생성하는 체인의 속도 경쟁
    - Example
      - $P =$ 공격자가 정당한 체인을 따라잡아 해킹에 성공할 확률
        - $q = 0.1$일 때
          - $z = 0 \quad P = 1.0000000$
          - $z = 1 \quad P = 0.2045873$
          - $z = 2 \quad P = 0.0509779$
          - $z = 3 \quad P = 0.0131722$
          - $z = 4 \quad P = 0.0034552$
          - $\vdots$
    - 공격자가 정당한 체인을 따라잡을 확률은 블록 수에 따라 지수적으로 감소
      - q가 증가 할수록 z도 같이 증가

Proposed Scheme (21/22)

- Calculations
  - 시나리오 2의 예
    - 가정 1: 공격자는 선의의 노드(피해자)보다 더 큰 해싱파워를 가지고 있어 빠르게 블록을 생성할 수 있음
    - 가정 2: 피해자는 6-confirmation을 따르지 않음



Proposed Scheme (22/22)

- Calculations
  - 공격 시나리오 2에 대한 방어: 6-confirmation
    - 발생하는 문제: 몇 번의 블록 승인이 있어야 그 블록을 안전하다고 생각할 수 있을까?
    - threshold 기준 이상일 때 실거래가 이루어지도록 함
      - 6번의 블록 승인은 약 1시간 소요 (블록 생성에 약 10분이 걸림)
      - 금액에 따라 승인 횟수를 정함
        - 6번까지 승인할 가치가 없는 정도의 거래라면 3번만 승인
        - 3-confirmation
      - 마이닝에 대한 검증은 약 100번의 블록 승인을 요구

Conclusion

- **P2P 분산 네트워크에서 사용 가능한 전자 화폐: 비트코인**
  - 디지털 서명의 연속 (블록체인): 무결성, 부인봉쇄 기능
  - 거래 장부 공개: 거래 내역의 투명성
  - TTP 없이 거래 당사자 간의 자율적인 거래가능
  - 프라이버시 보호: 비트코인 주소 사용
  - 생성한 블록에 대한 정당성 보장
    - 네트워크 내 노드들 간의 합의 알고리즘: Proof-of-Work
    - 블록 생성을 통한 승인 작업
  - 정직한 노드들 간의 생태계 구축: 인센티브

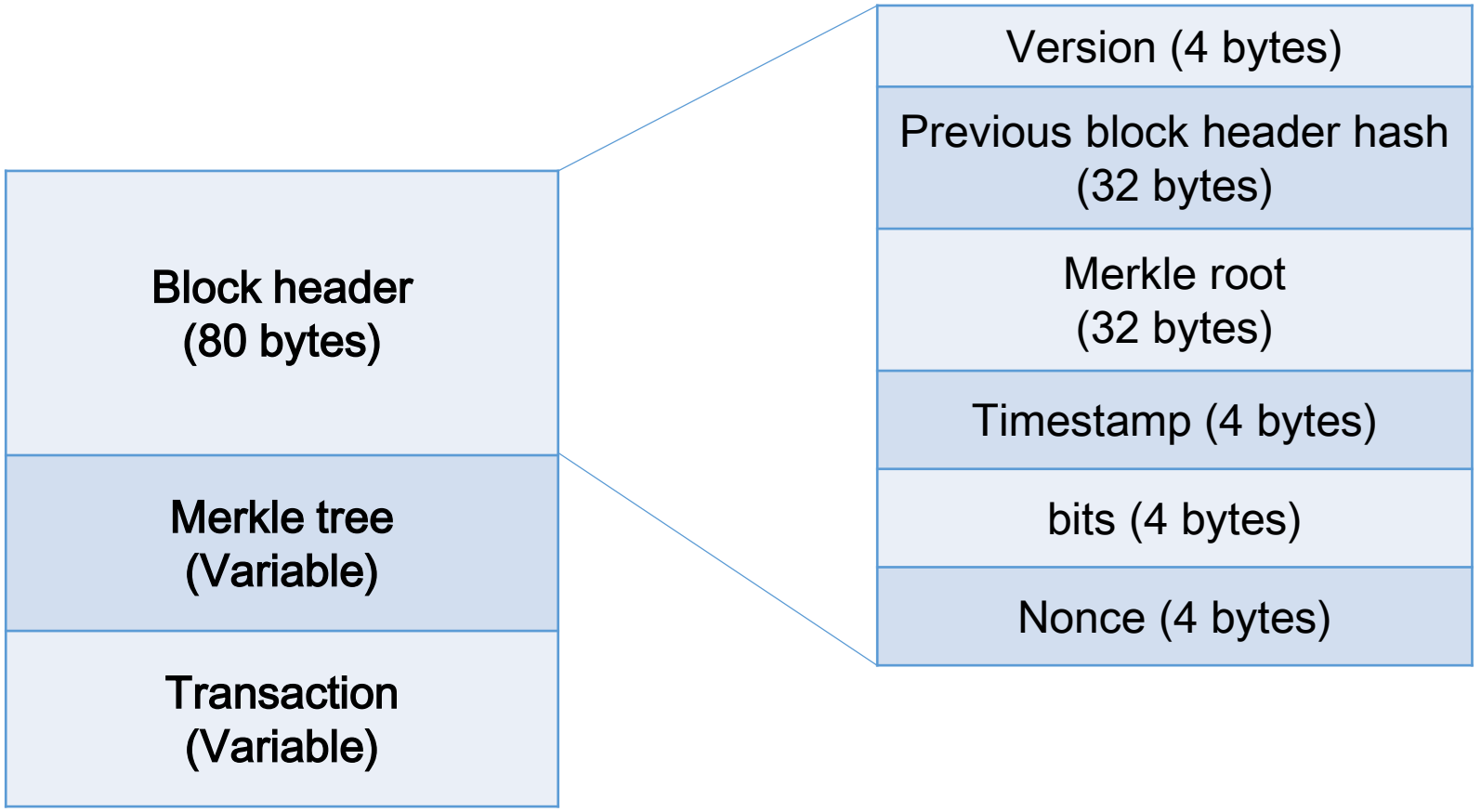
감사합니다!

이부형 (boohyung@pel.smuc.ac.kr)

백업 1: 블록 구조 (1/3)

- 블록 헤더 (80 bytes)
  - Version (4 bytes): 블록체인 버전
  - Previous block header hash (32 bytes): 이전 블록 헤더의 해시 값
  - Merkle root (32 bytes): 머클 트리의 루트 값
  - Timestamp (4 bytes): 블록 발행 시각
  - bits (4 bytes): 난이도 조절 값
  - Nonce (4 bytes): 합의 알고리즘에 사용하는 난수
- 블록 바디 (가변적)
  - Transaction: 10분 동안 발생한 사용자 간의 거래들
  - Merkle tree: 거래 내역의 해시값으로 이루어진 트리 구조
    - 거래 내역의 변조를 막기 위해 사용

백업 1: 블록 구조 (2/3)


Block header
(80 bytes)
Merkle tree
(Variable)
Transaction
(Variable)
Version (4 bytes)
Previous block header hash
(32 bytes)
Merkle root
(32 bytes)
Timestamp (4 bytes)
bits (4 bytes)
Nonce (4 bytes)

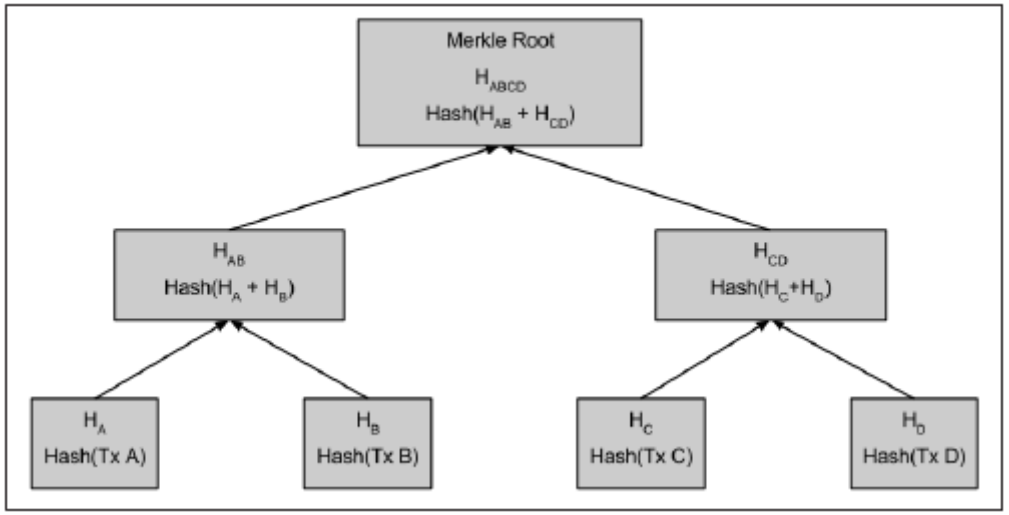
백업 1: 블록 구조 (3/3)

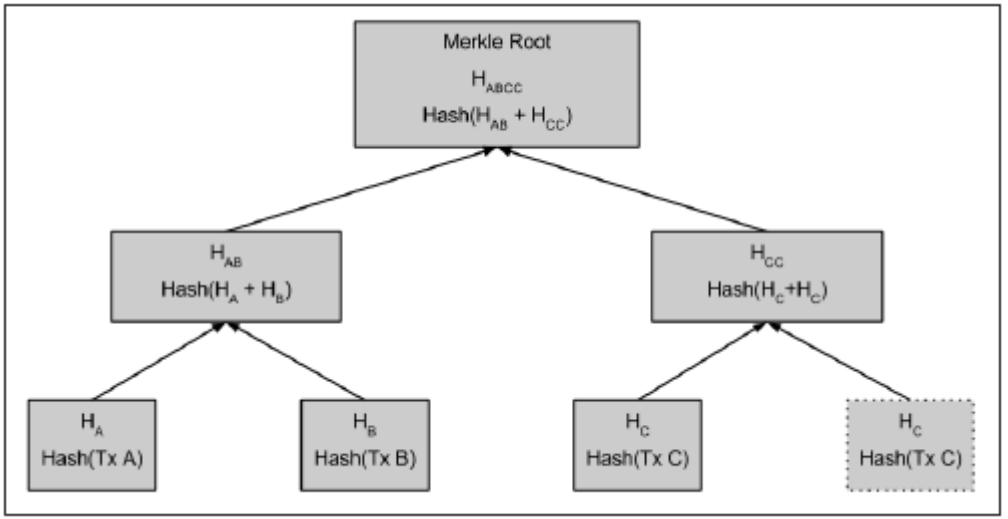
- 블록 구조 예시: Bitcoin (Raw Data)
  - JSON (JavaScript Object Notation) 형태로 기록


```
{
  "hash":"00000000000000041cbd5ba9607416285167b4b6ff65bda651ac0f55e03bbd6",
  "ver":2,
  "prev_block":"000000000000000005fca80796201fe3fbdf2b695b64feb84e60b5d0c4ba2a99",
  "mrkl_root":"59a1914711c0923c2ee4bd43778991ee528b6addce86f8d0f6c18f12f7700823",
  "time":1402209318,
  "bits":408782234,
  "nonce":4027624099,

  "n_tx":27,
  "size":13924,

  "tx":[

      /* 거래 내용이 포함됨 */

   ],
   "mrkl_tree":[

      /* 거래내역의 머클크리 */
   ]
}
```

백업 2: 머클 트리

- 블록 헤더 내의 머클 루트를 이용하여 정당한 거래 내역임을 검증
  - 아래 그림에서
    - $H_A = SHA256(SHA256(Transaction\ A))$
    - $H_{AB} = SHA256(SHA256(H_A + H_B))$
  - 만약 블록의 개수가 홀수라면 가장 최근의 블록의 해시값을 복사하여 머클 루트 계산

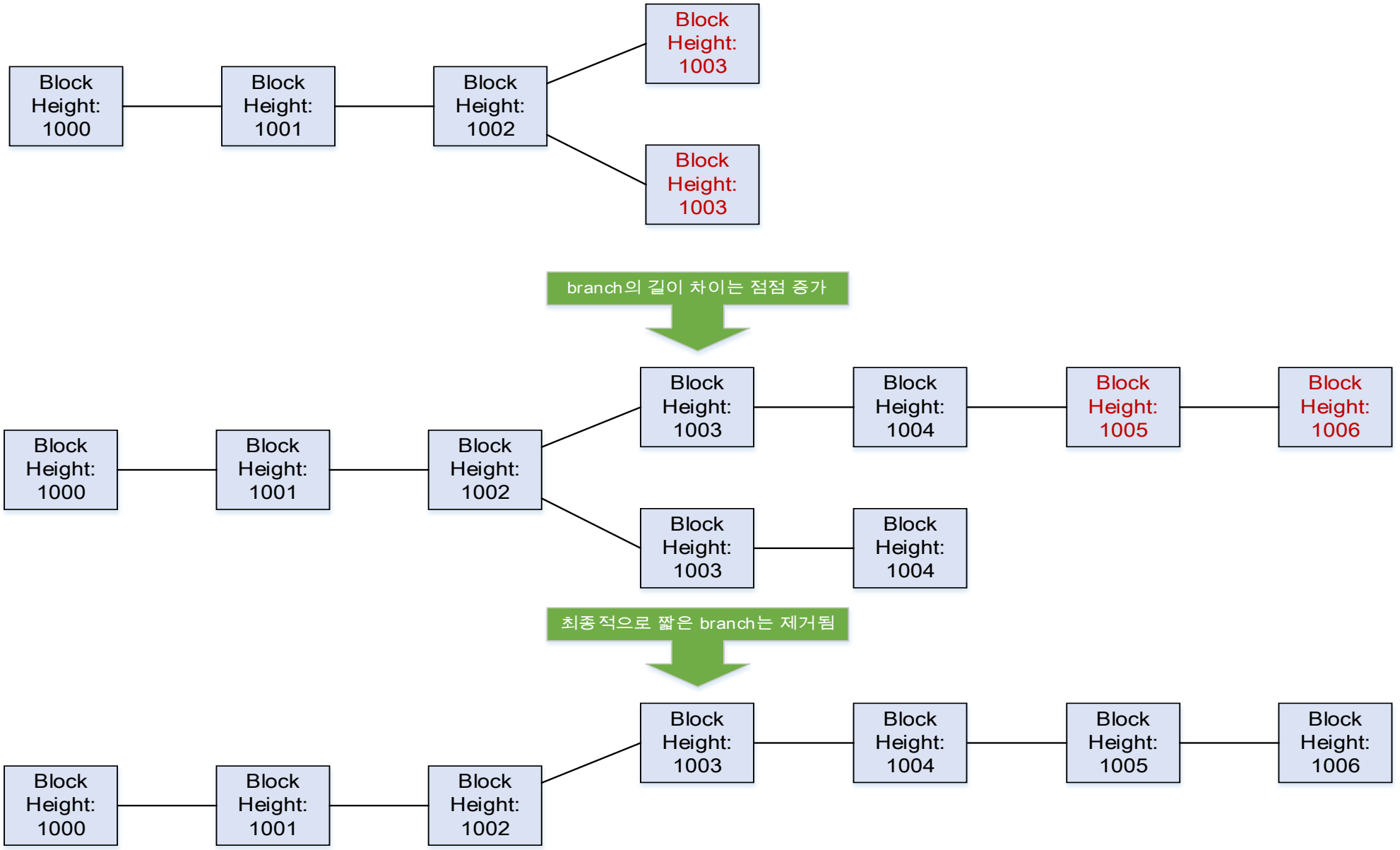




백업 3: Fork (1/2)

- 거의 동시에 두 개의 새로운 블록이 생성되어 체인에 연결된 경우
  - Block Height가 같은 2개의 블록
    - 먼저 도착한 트랜잭션을 처리
    - 나중에 도착한 트랜잭션은 보관(나중에 도착한 트랜잭션이 더 긴 체인을 생성할 수도 있기 때문)
      - 나중에 도착했어도 최종적으로 더 긴 체인이 Main Chain으로 인정되기 때문에 먼저 도착한 트랜잭션이 삭제될 수 있음
    - 노드의 선택에 따른 branch 사이의 경쟁
      - 두 branch의 차이가 크게 벌어질 수록, 많은 수의 노드가 긴 branch로 이동
      - 최종적으로 짧은 branch는 사라짐

백업 3: Fork (2/2)


Block Height: 1000
Block Height: 1001
Block Height: 1002
Block Height: 1003
Block Height: 1003
branch의 길이 차이는 점점 증가
Block Height: 1000
Block Height: 1001
Block Height: 1002
Block Height: 1003
Block Height: 1004
Block Height: 1005
Block Height: 1006
Block Height: 1003
Block Height: 1004
최종적으로 짧은 branch는 제거됨
Block Height: 1000
Block Height: 1001
Block Height: 1002
Block Height: 1003
Block Height: 1004
Block Height: 1005
Block Height: 1006