# Hash Based Signature(HBS)

https://youtu.be/atxRhDo5b6M

# Lamport Signature Scheme

- Lamport signature?
  - **일회성 서명으로 하나의 메시지에 안전하게 서명**하는데 사용
  - 큰 해시 함수를 갖고 있어 RSA보다 안전
  - 메시지 크기에 비해 비밀키, 공개키, 서명 크기가 크다는 단점
  - **사용되지 않은 공개키는 공개되면 안 됨**
- **키 생성**
  - 개인키 : 256-bit 길이의 **256개의 랜덤값 쌍** 생성 (2 x 256 = 512개)
  - 공개키 : SHA-256 등을 사용하여 256개의 개인키 쌍의 해시 값을 각각 구함
    - **256개의 개인키 쌍에 대한 해시 값 512개의 공개키 생성**

https://ko.wikipedia.org/wiki/%EB%9E%A8%ED%8F%AC%ED%8A%B8_%EC%84%9C%EB%AA%85

# Lamport Signature Scheme

- **서명 생성**


https://www.geeksforgeeks.org/lamport-one-time-signature-scheme/

# Lamport Signature Scheme

- **서명 검증**

  - 검증자가 올바른 서명(A의 서명)을 확인하기 위해서는 메시지의 해시값(256-bit) 구하기
  - 메시지의 해시값을 이용해 A의 공개키 중 256개 선택

  - A가 서명 생성한 것과 동일하게 생성
    = A의 공개된 서명 값 256개에 대한 해시 값 (일치하면 서명은 올바른 것)

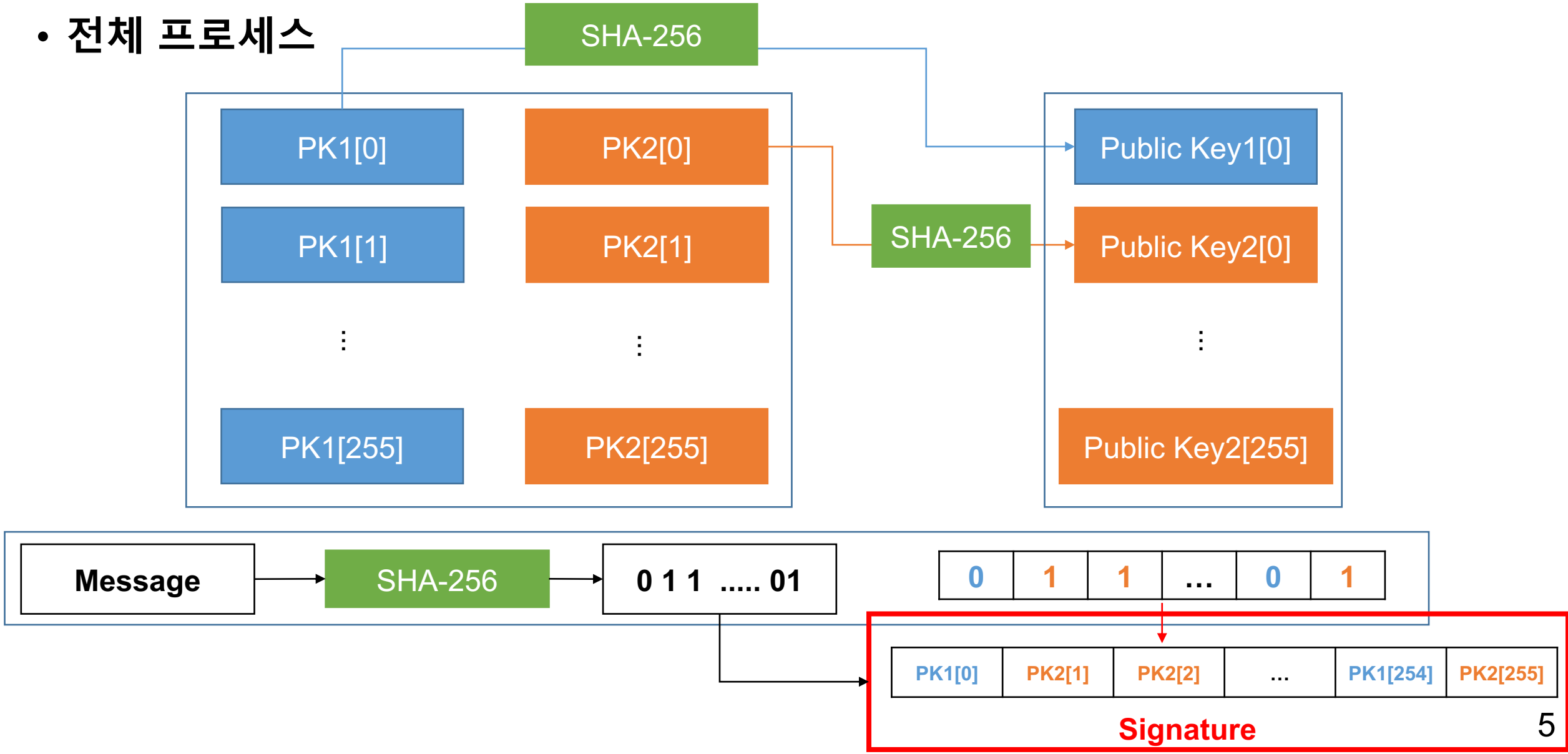
Message (256-bit)
ex) 011…01

| 0 | 1 | 1 | … | 0 | 1 |
|---|---|---|---|---|---|

| PubKey1 [0] | PubKey2 [1] | PubKey2 [2] | … | PubKey1 [254] | PubKey2 [255] |
|---|---|---|---|---|---|

# Lamport Signature Scheme

- 전체 프로세스

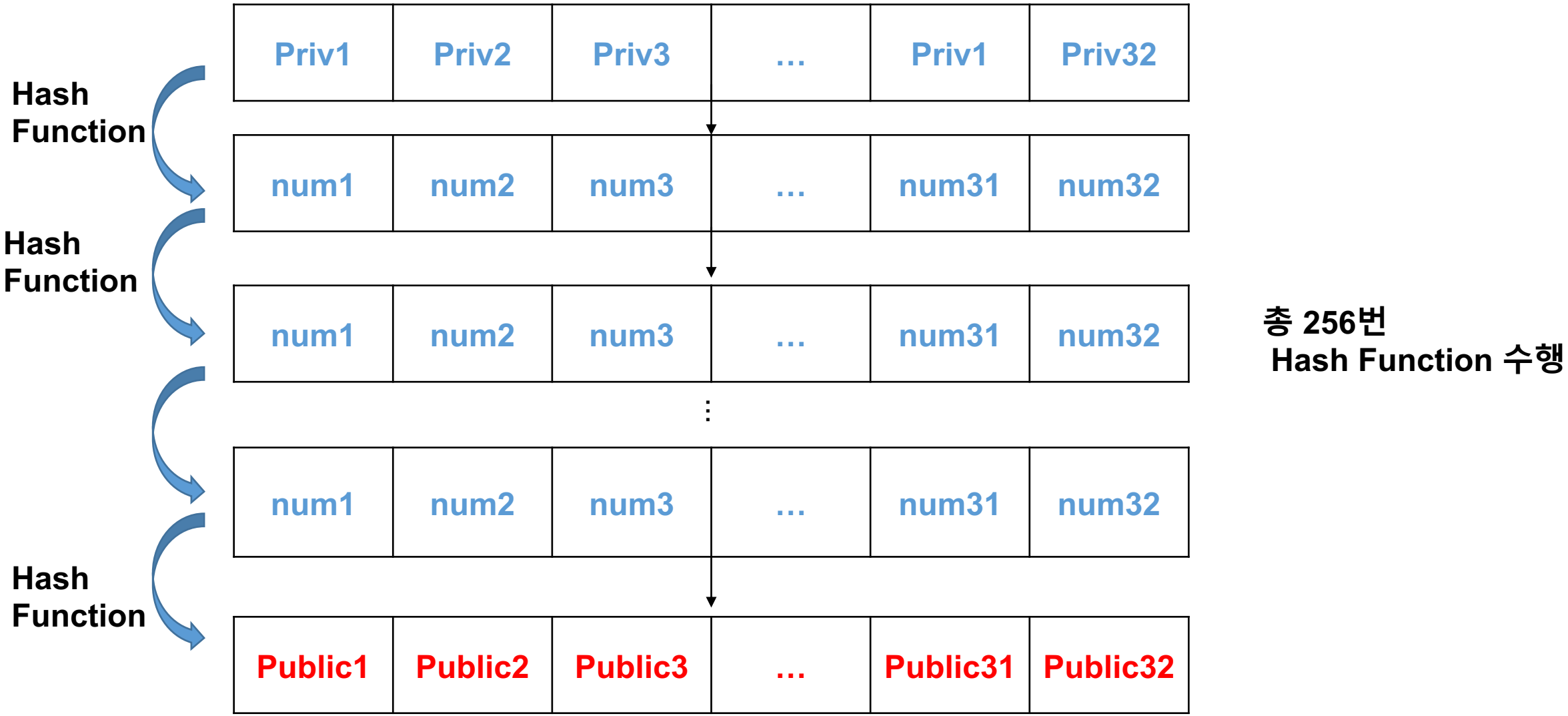


# Winternitz Signature Scheme

- Lamport의 경우, 서명의 크기가 매우 크다는 단점
- 이를 보완하여, **메시지 다이제스트의 일부 비트를 동시에 서명하는 기법**
- 개인키 : 32개의 256-bit 생성
- 공개키 : 32개의 개인키 각각을 256번 해시
  - 32개의 256-bit 획득

# Winternitz Signature Scheme

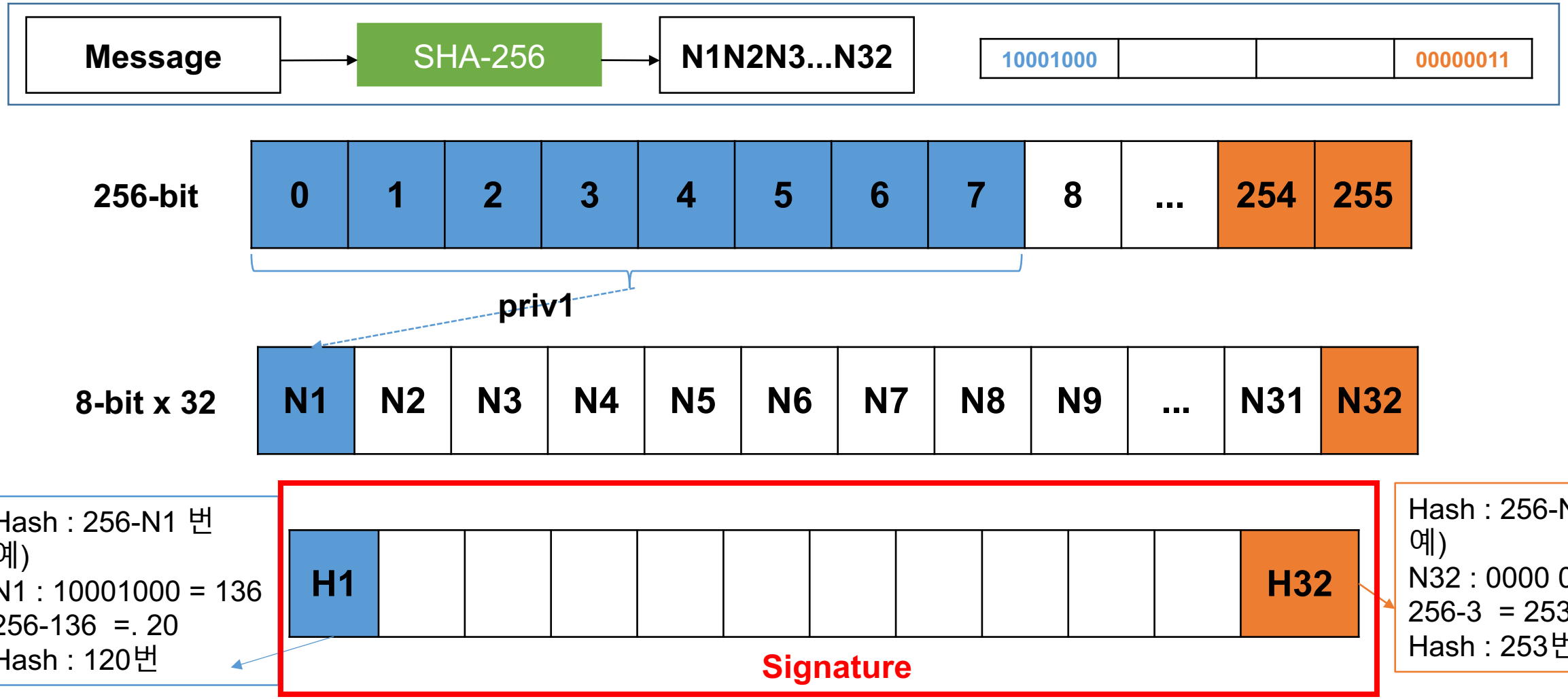
- 키 생성




총 256번
Hash Function 수행

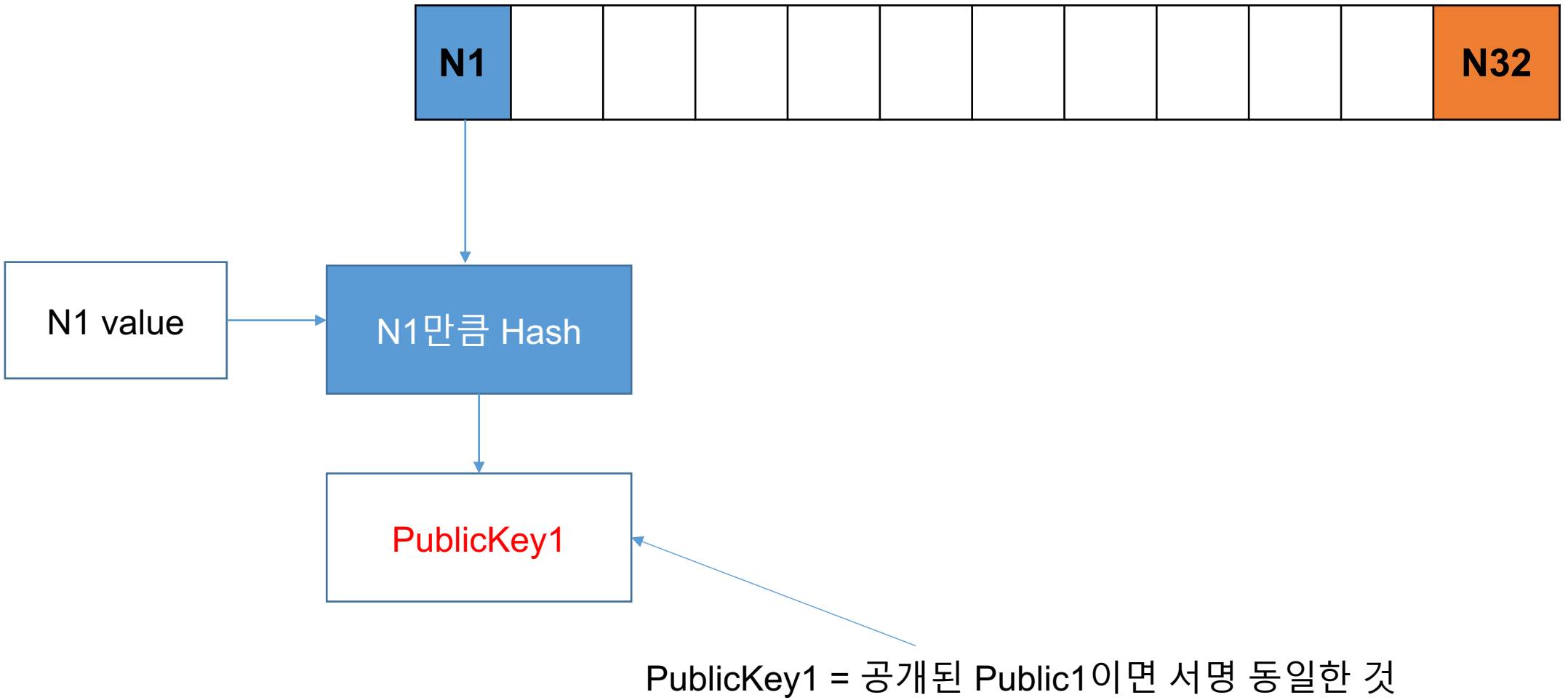
# Winternitz Signature Scheme

- 서명 생성



# Winternitz Signature Scheme

- **서명 검증**



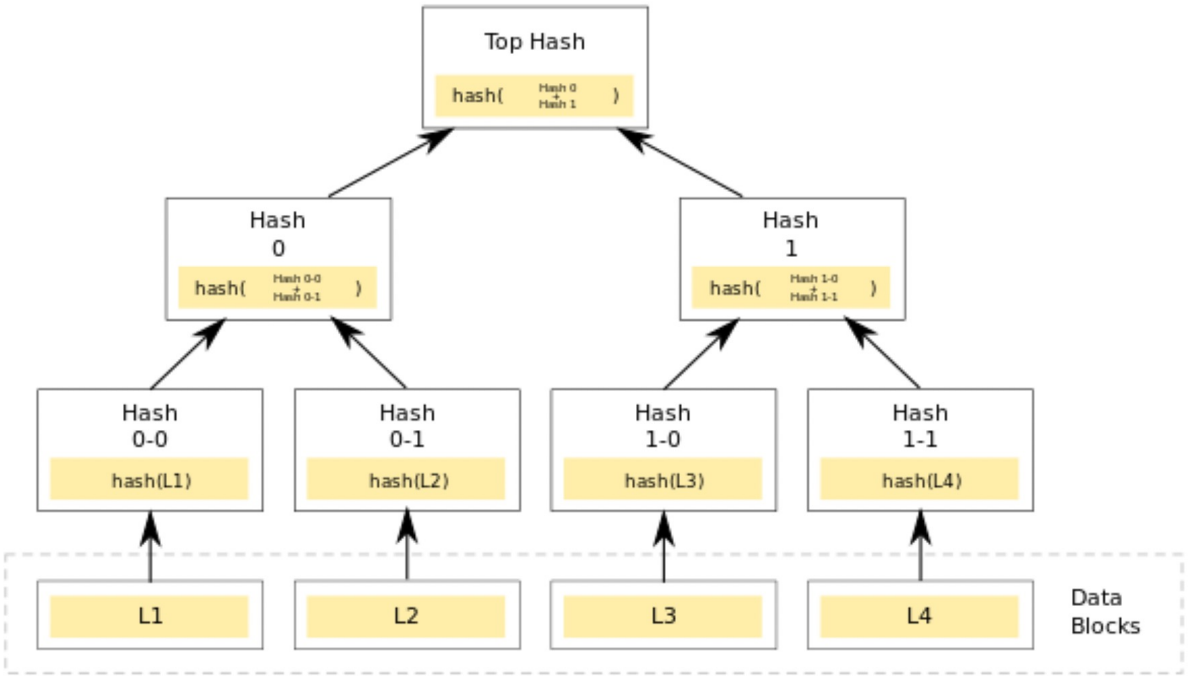
# Merkle signature scheme

- DSA(Digital Signature Algorithm)이나 RSA와 같은 서명 알고리즘
- **Hash trees(= Merkel trees)와 one-time(Winternitz)기반 서명 기법**

https://en.wikipedia.org/wiki/Merkle_signature_scheme

# Merkle Tree

- 머클 트리(= 해시 트리)
- Leap 노드 : data (L1, L2, L3, L4 ,,)
- 데이터 검증 방법
  - 루트 노드의 해시 값만 알면 가능
- 데이터 전체가 아니라 일부만 검증
  - 자식 노드 가운데 하나의 해시값 알면,
  그 노드의 모든 자식 노드에 대한 데이터 검증 가능

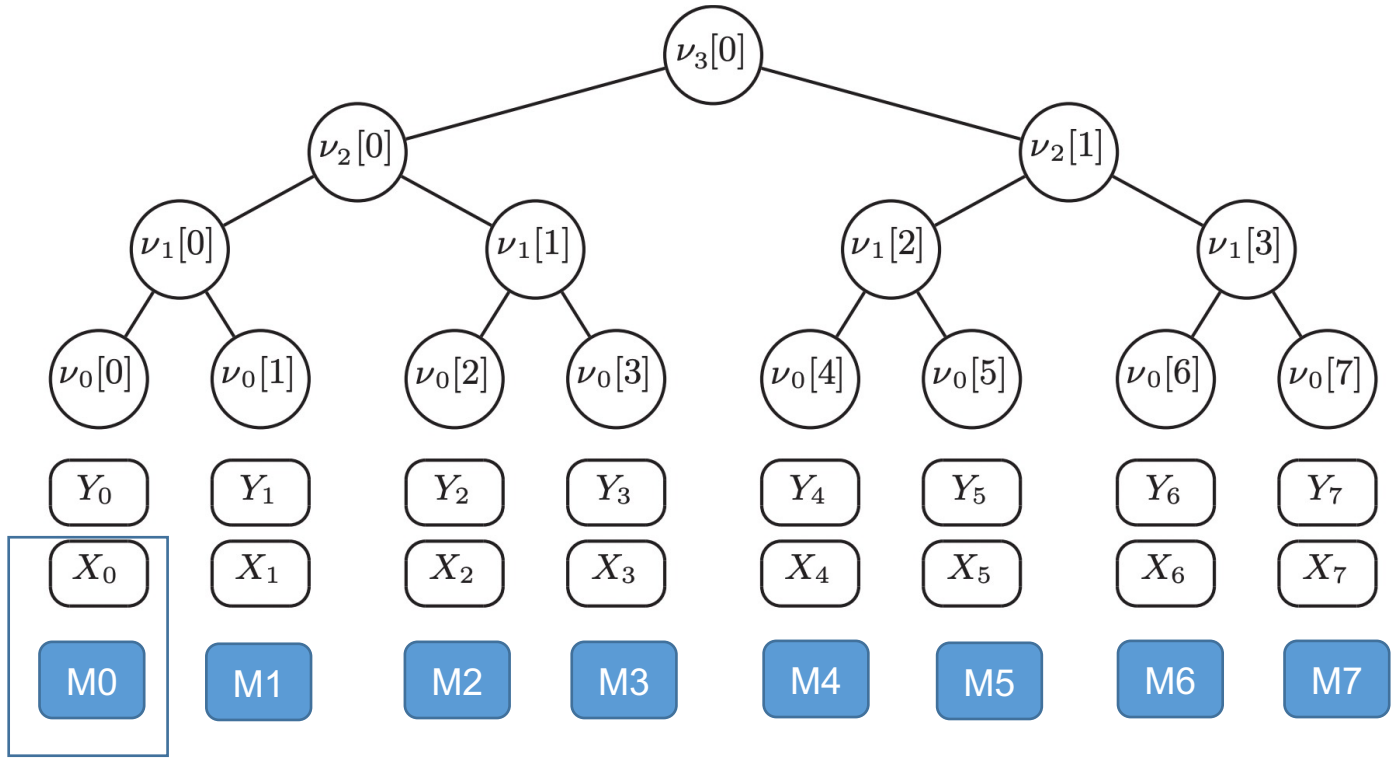



https://ko.wikipedia.org/wiki/%ED%95%B4%EC%8B%9C_%ED%8A%B8%EB%A6%AC

# Merkle signature scheme(MSS)

- 하나의 공개키로 제한된 메시지 서명 가능
- 가능한 메시지의 수 : $2^n$
- 공개키 : 머클트리의 root 노드 존재
- 서명키(X), 검증키(Y): leaf 노드 존재
- Merkle 서명
  - One-time 서명 + 검증키

$$\nu_h[j] = g(\nu_{h-1}[2j] \| \nu_{h-1}[2j+1]), \quad 1 \le h \le H, 0 \le j < 2^{H-h}.$$




https://www.researchgate.net/profile/Nicolas-Sendrier-2/publication/226115302_Code-Based_Cryptography/links/540d62d50cf2df04e7549388/Code-Based-Cryptography.pdf

# XMSS(eXtended Merkle Signature Scheme)

- 머클 트리와 기본 구조는 동일

- Stateful 방식
  - 서명을 하기 위해 비밀키 업데이트
  - 서명에 사용된 일회용 키의 상태를 저장하고 **절대 재사용되지 않게 관리**

- 기존 머클 트리
  - 두 자식 노드의 연결 값을 해시 함수에 넣는 방식으로 부모 노드 값 구하는 방법

https://link.springer.com/chapter/10.1007/978-3-642-25405-5_8

# XMSS(eXtended Merkle Signature Scheme)

- XMSS 트리
  - 두 자식 노드의 연결 값을 해시 함수에 넣을 때 마다 각 레벨(H)의 고유한 랜덤 비트 마스크 값을 XOR하는 방식

$$\text{NODE}_{i,j} = h_K((\text{NODE}_{2i,j-1} \oplus b_{l,j})||(\text{NODE}_{2i+1,j-1} \oplus b_{r,j}))$$

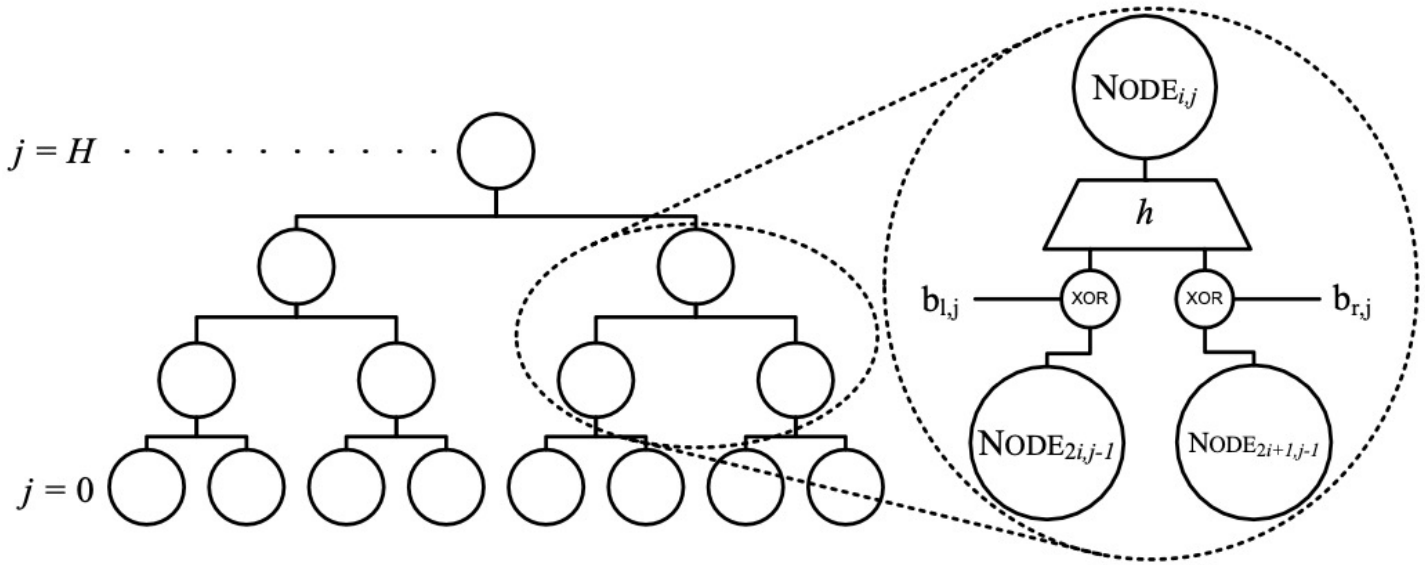



**Fig. 1.** The XMSS tree construction

International Workshop on Post-Quantum Cryptography

PQCrypto 2011: Post-Quantum Cryptography pp 117-129 | Cite as

XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions

Authors | Authors and affiliations

Johannes Buchmann, Erik Dahmen, Andreas Hülsing

https://link.springer.com/chapter/10.1007/978-3-642-25405-5_8

# Q & A