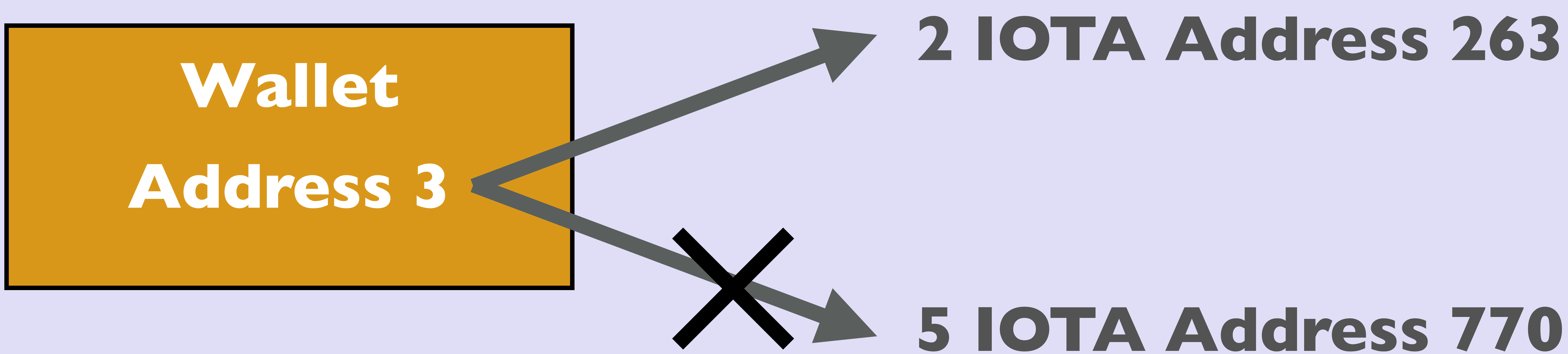
IOTA TUTORIAL 6

Why you should not reuse an address for outgoing txs



v1.0.0

INTRO

- In this video I will explain why you should not reuse an IOTA address for **outgoing** transactions.

DIGITAL SIGNATURES

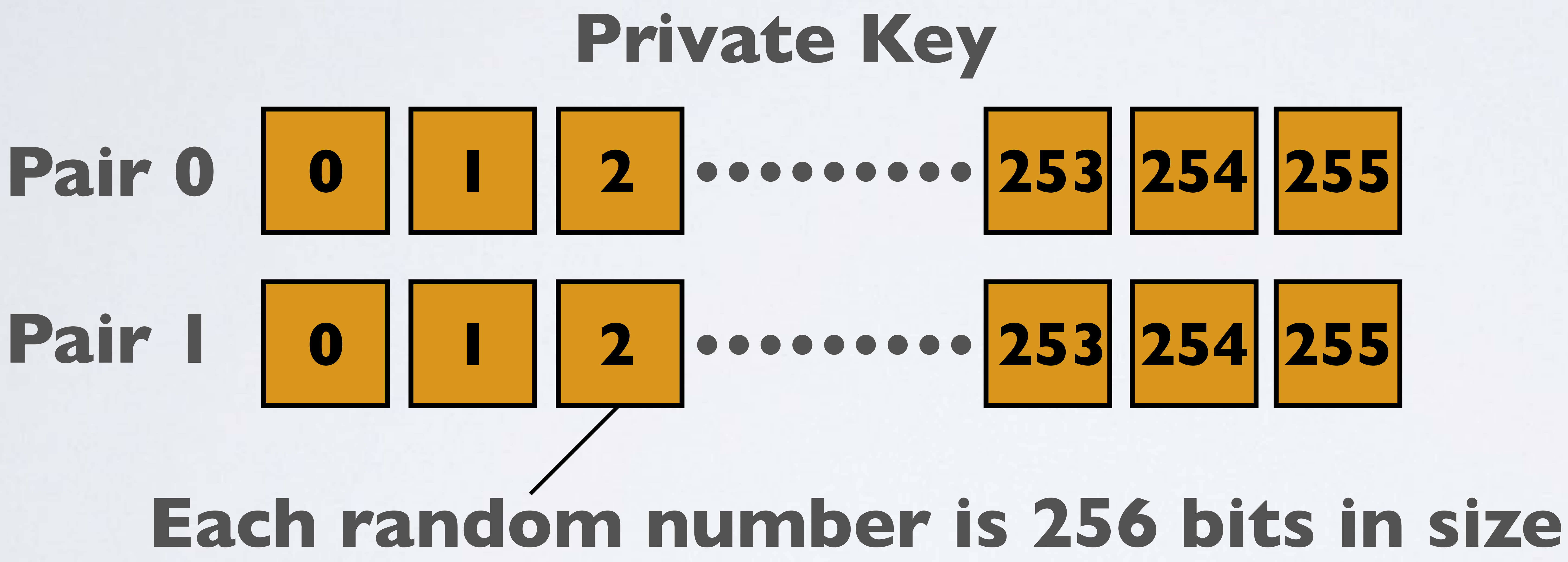
- Digital signatures are used for authentication, integrity checking and non-repudiation.
- Development of quantum computers threatens the security of currently used digital signature algorithms such as Rivest–Shamir–Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA).
- Cryptographers developed a variety of quantum-resistant alternatives of which hash based signatures are the most promising.
- Hash based signatures are based on so called **O**ne **T**ime **S**ignatures (OTS). The term implies that a single public/private key pair must only be used once. Otherwise, an attacker is able to reveal more parts of the private key and spoof signatures.

LAMPORT ONE TIME SIGNATURES SCHEME

- In 1979 Leslie Lamport created a method to construct digital signatures using only cryptographically secure one way hash functions.
- This method is called the Lamport signature or Lamport One Time Signature (OTS) scheme.
- Other One Time Signature schemes are the Merkle OTS and Winternitz OTS.
- The Lamport One Time Signature scheme is very easy to understand and is **VERY LOOSELY** comparable to Winternitz OTS.
- For simplicity's sake I will be using the Lamport One Time Signature scheme explaining why you should never reuse an IOTA address for outgoing transactions.

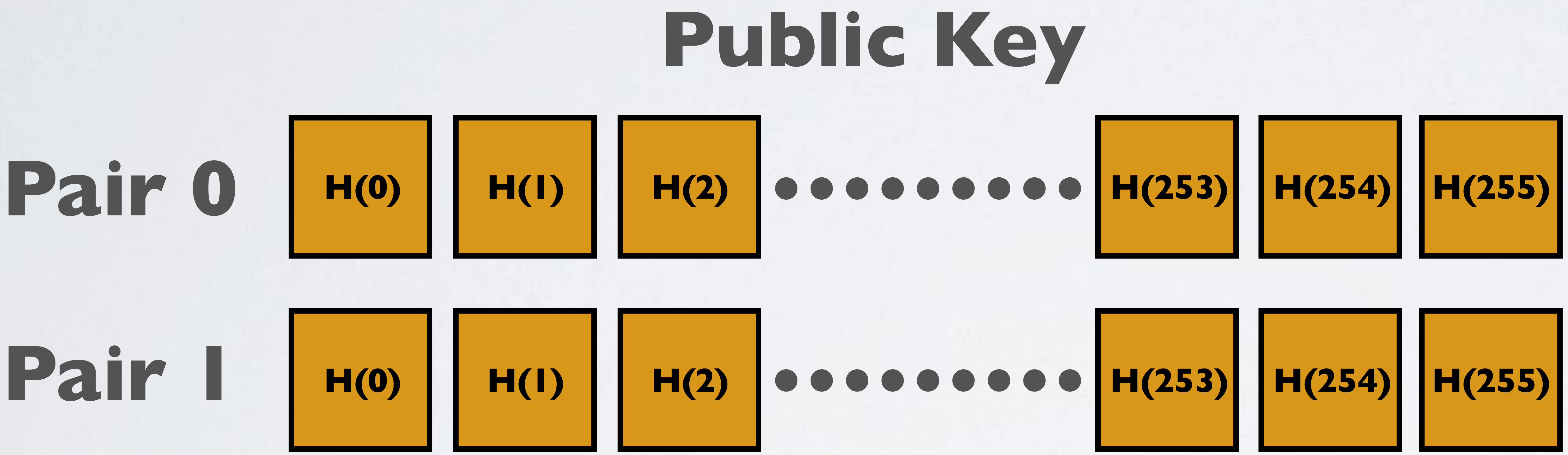
LAMPORT ONE TIME SIGNATURES SCHEME

- Alice uses a random number generator and produces two pairs of 256 random numbers, total 512 numbers. Each random number is 256 bits in size. These random numbers forms the private key.



LAMPORT ONE TIME SIGNATURES SCHEME

- Each of the 512 random numbers are separately hashed, using for example SHA-256. These hashed random numbers forms the public key.



LAMPORT ONE TIME SIGNATURES SCHEME

- Alice has a document (or transaction data) which is hashed using SHA-256.
  The document hash is of course 256 bits long: **`101..011`**

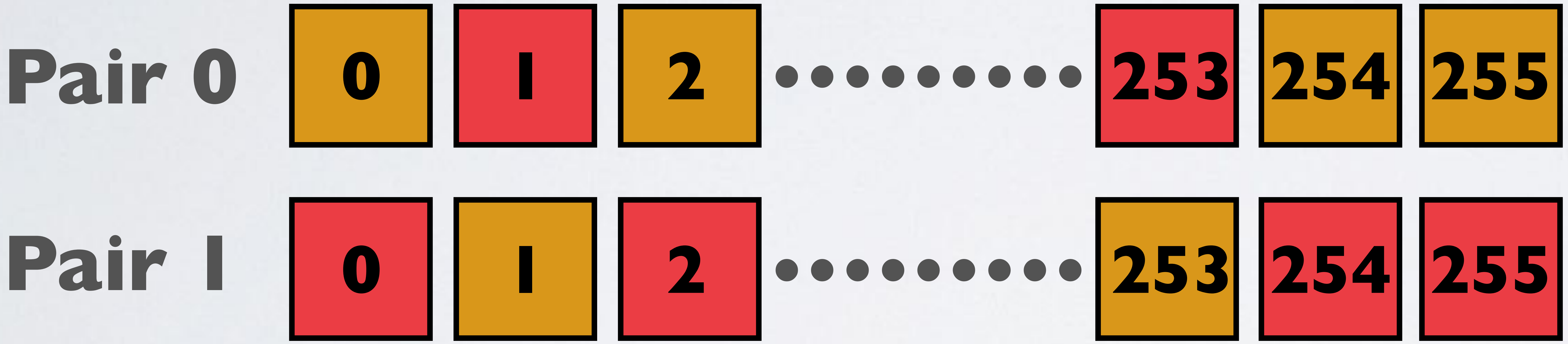
- Alice wants to create a digital signature for her document.
  She applies the following procedure:
  - Loop thru each bit (n) of the hash from 0-255.
  - If the bit is a 0, publish the n^{th} number from pair 0.
  - If the bit is a 1, publish the n^{th} number from pair 1.
  - When all bits are looped, destroy all unused numbers from pair 0 and 1.

- This produces a sequence of 256 random numbers.

LAMPORT ONE TIME SIGNATURES SCHEME

- The document hash: **101..011**

Private Key

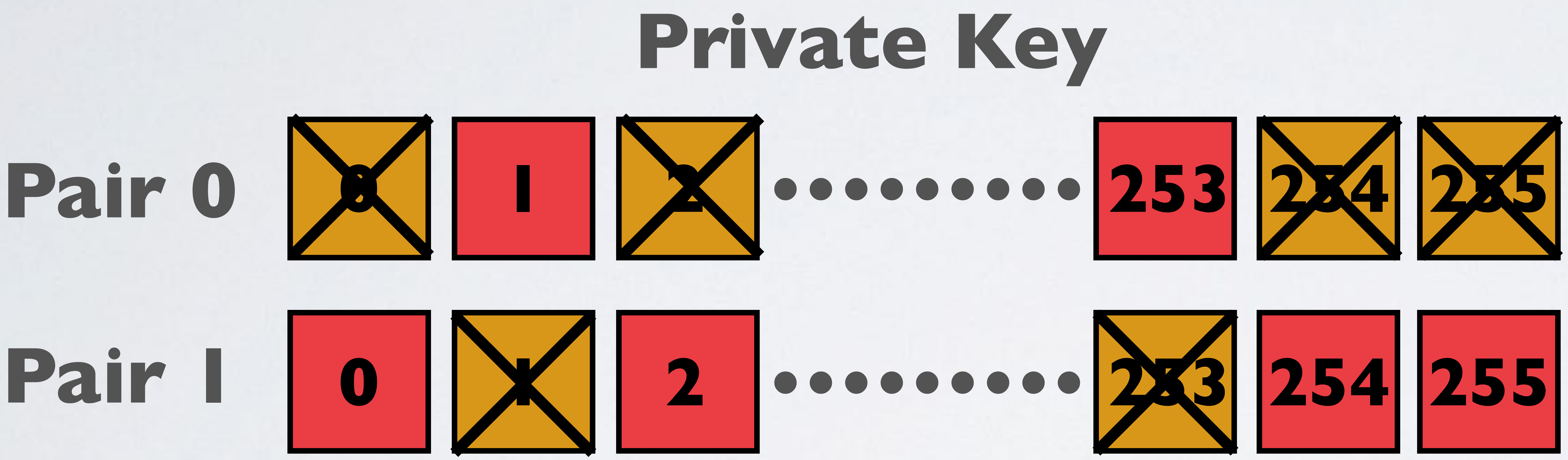


- The digital signature is a sequence of 256 random numbers.

Digital signature

LAMPORT ONE TIME SIGNATURES SCHEME

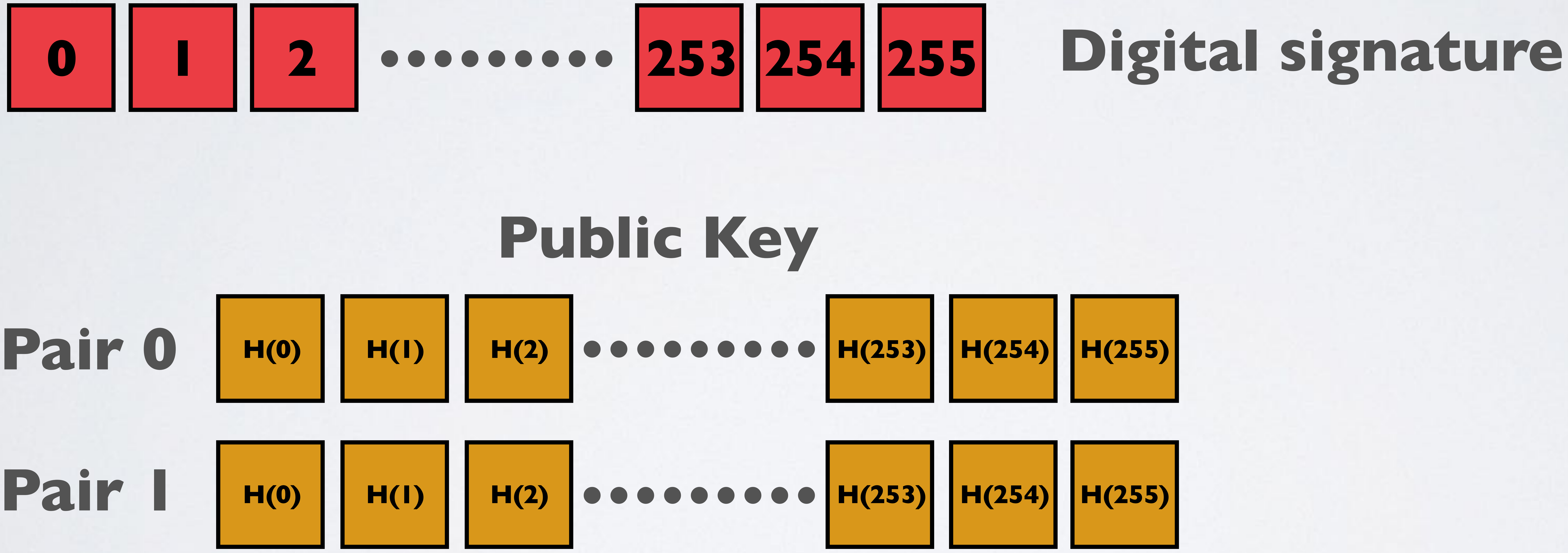
- After the digital signature is created, delete all unused numbers from the private key.



- The digital signature consist half of the private key, the other 256 random numbers are still unknown and thus nobody can create signatures that fit other message hashes.

LAMPORT ONE TIME SIGNATURES SCHEME

- Alice sends her document, together with the corresponding digital signature and public key to Bob.

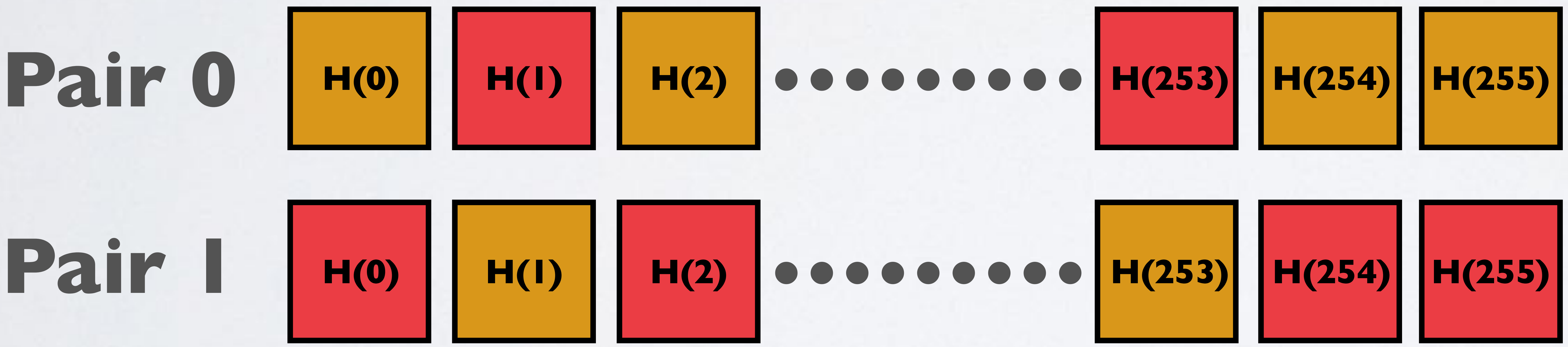


LAMPORT ONE TIME SIGNATURES SCHEME

- Bob wants to verify Alice's document signature.
  He first hashes the document using SHA-256.
  The document hash is again: **101..011**

- Bob follows the same steps when Alice created the digital signature, but instead uses the public key.

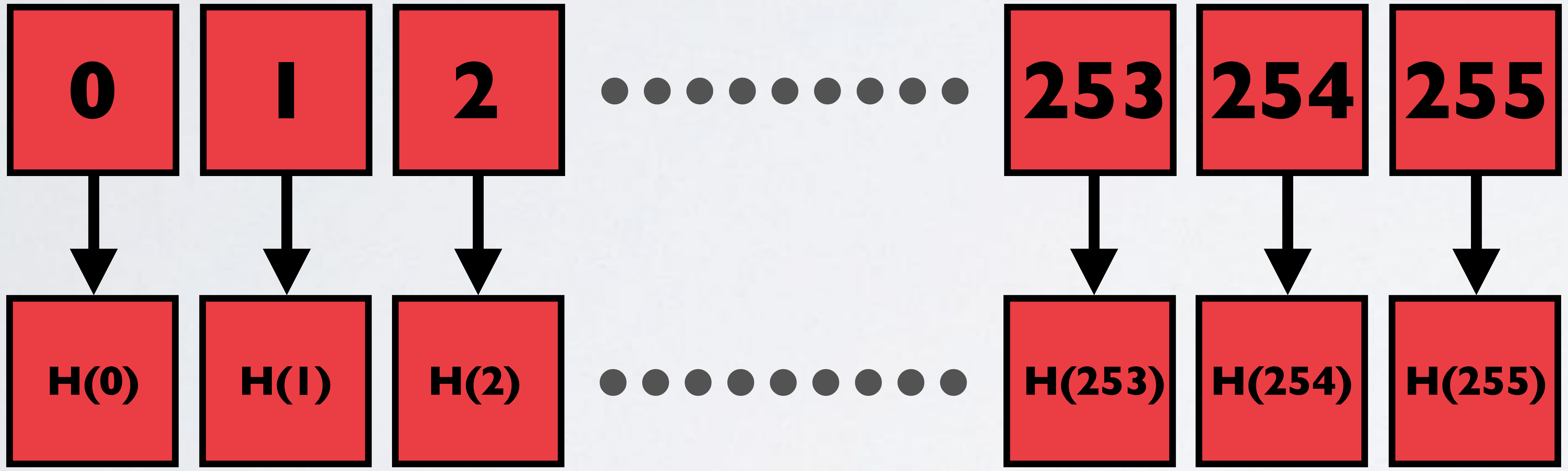
Public Key



LAMPORT ONE TIME SIGNATURES SCHEME

- Bob produces a sequence of 256 hashes picked from Alice's public key.

- Bob now hashes each of the random number in the digital signature.



Digital signature

- If both sequence of hash numbers match then the signature is ok.

LAMPORT ONE TIME SIGNATURES SCHEME

- The Lamport signature creates a digital signature which reveals part of the private key.

- The private key has 512 numbers and using it once will reveal 256 numbers.

- Using the private key twice weakens the security of the scheme again by half.
  The probability of an attacker being able to successfully forge a signature for a given message increases from $1/2^{256}$ to $1/2^{128}$.

- A third signature using the same key would increase the probability of a successful forgery to $1/2^{64}$ and a fourth signature to $1/2^{32}$, and so on.

LAMPORT ONE TIME SIGNATURES SCHEME

- Please note IOTA's signature scheme is based on the Winternitz One Time Signature (WOTS) scheme and is **NOT** the same as the Lamport signature scheme.
- However by using the Lamport One Time Signature scheme I am trying to give you a very simplistic understanding why you should never reuse an IOTA address for outgoing transactions.