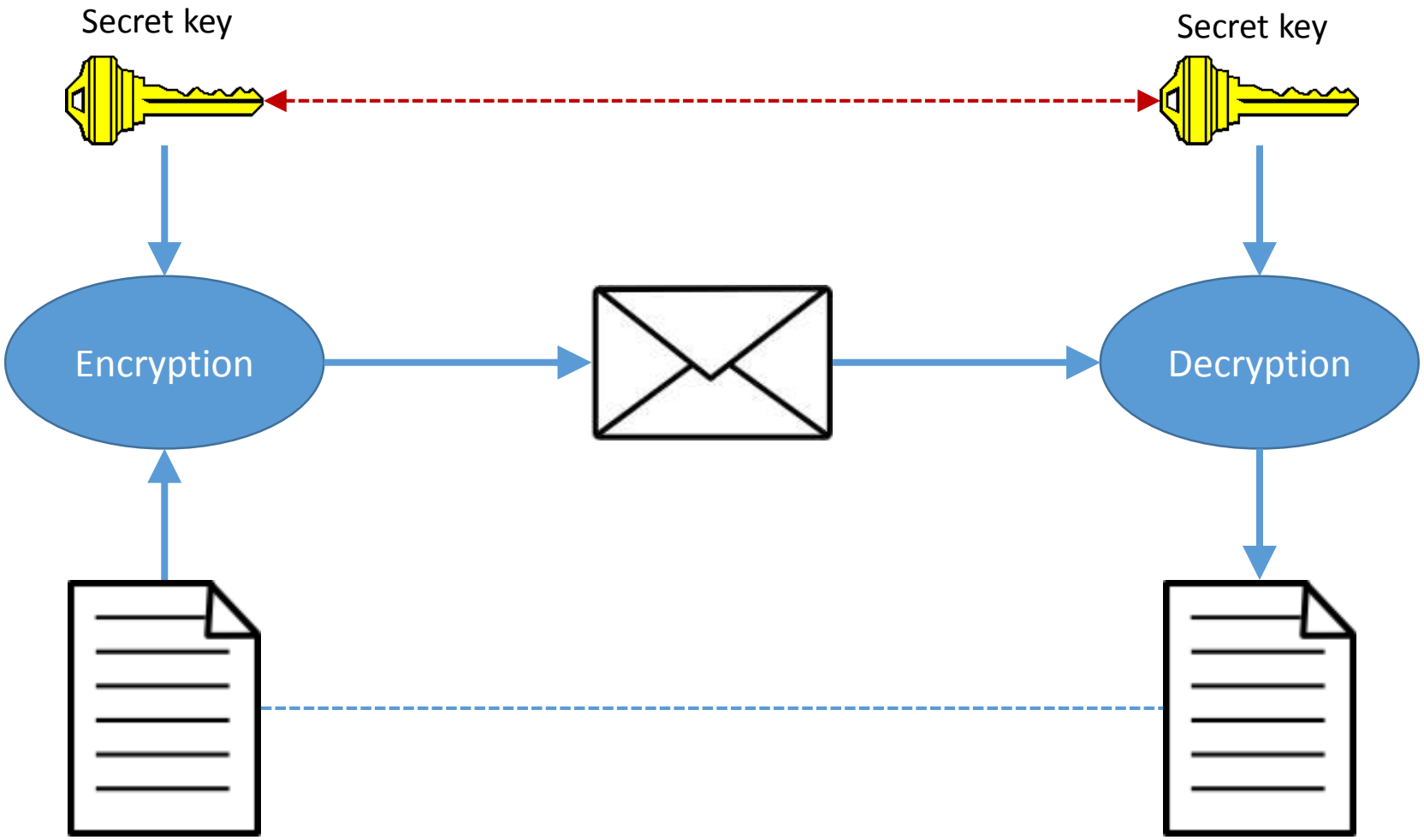
Cryptographic Applications of One-Way Functions

Ahto Truu, Guardtime, ahto.truu@guardtime.com

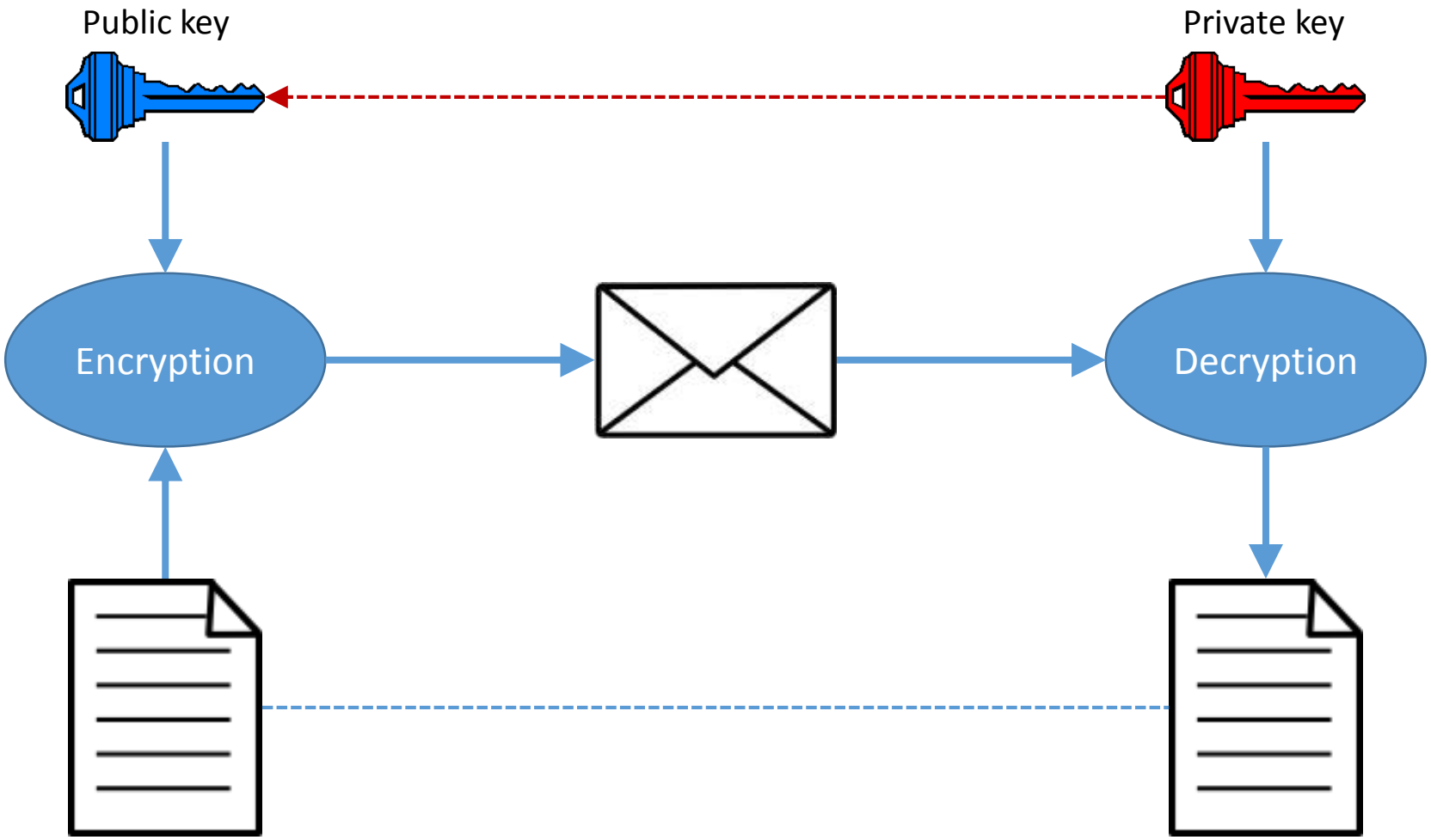
Three Pillars of Data Security

- Confidentiality
  - Deniability
- Integrity
  - Authentication
  - Non-repudiation
- Availability

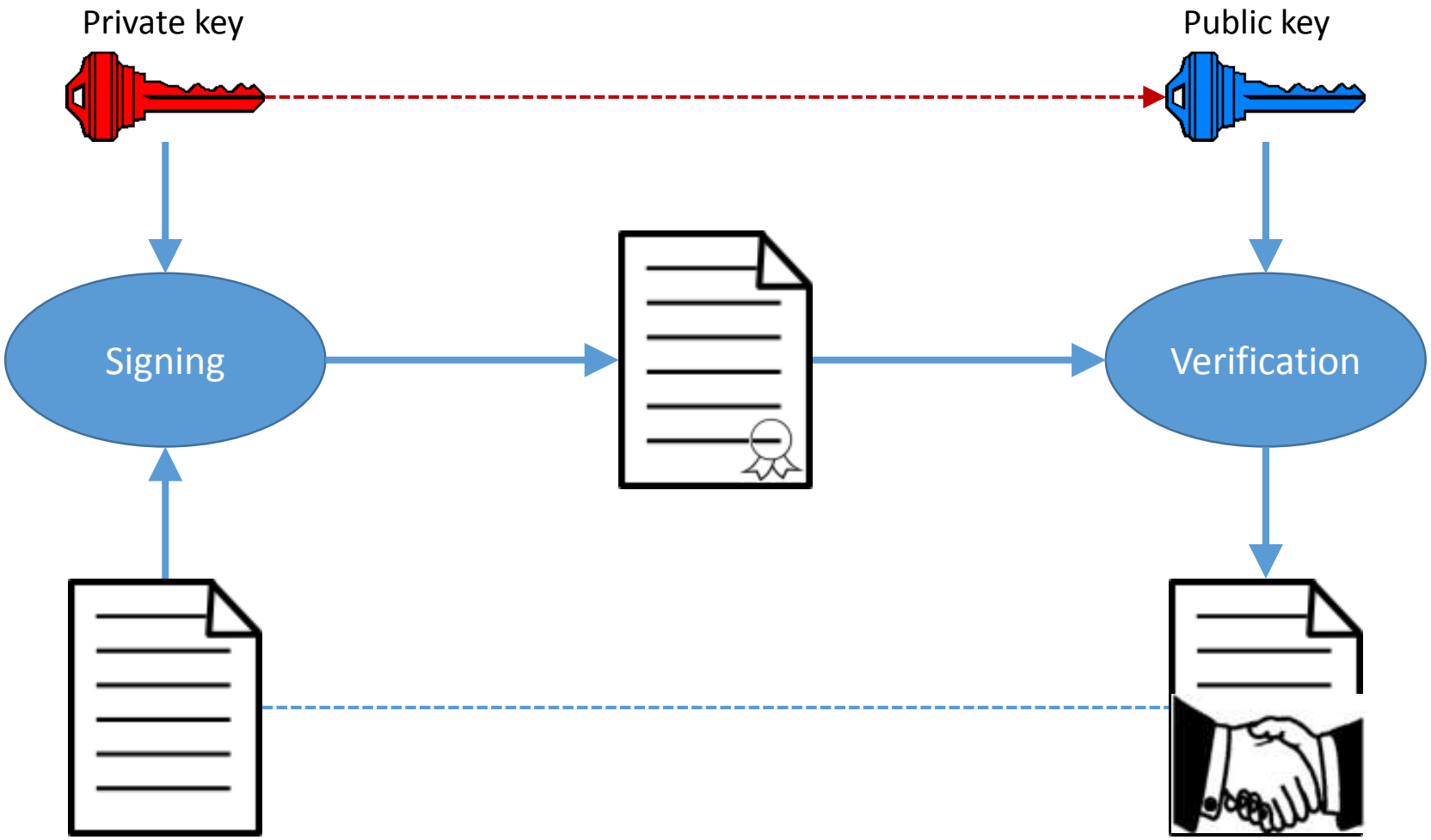
Symmetric Encryption



Asymmetric Encryption



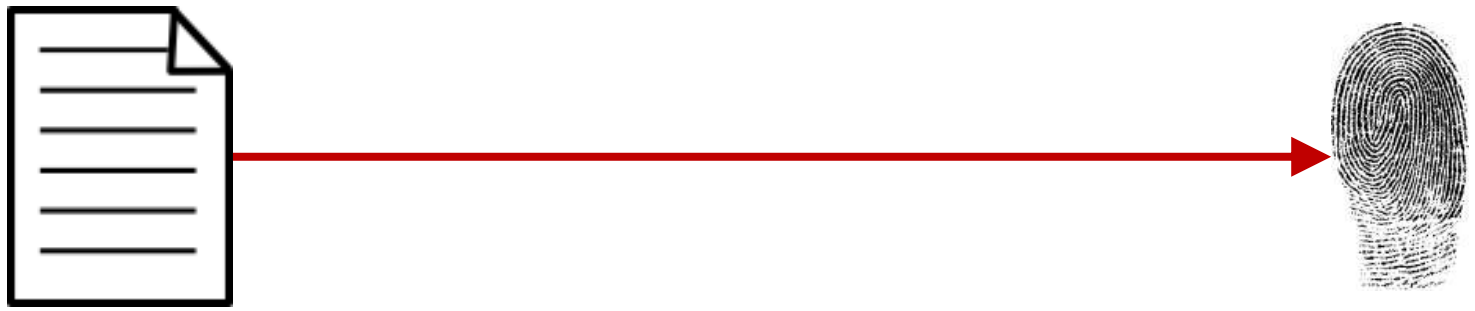
Digital Signatures



Some Ancient History

- Galileo Galilei 1610
- Haec immatura a me iam frustra leguntur o.y.
- These are at present too young to be read by me
- Cynthiae figuras aemulatur mater amorum
- The mother of love imitates the shapes of Cynthia
- Venus has the same phases as the Moon

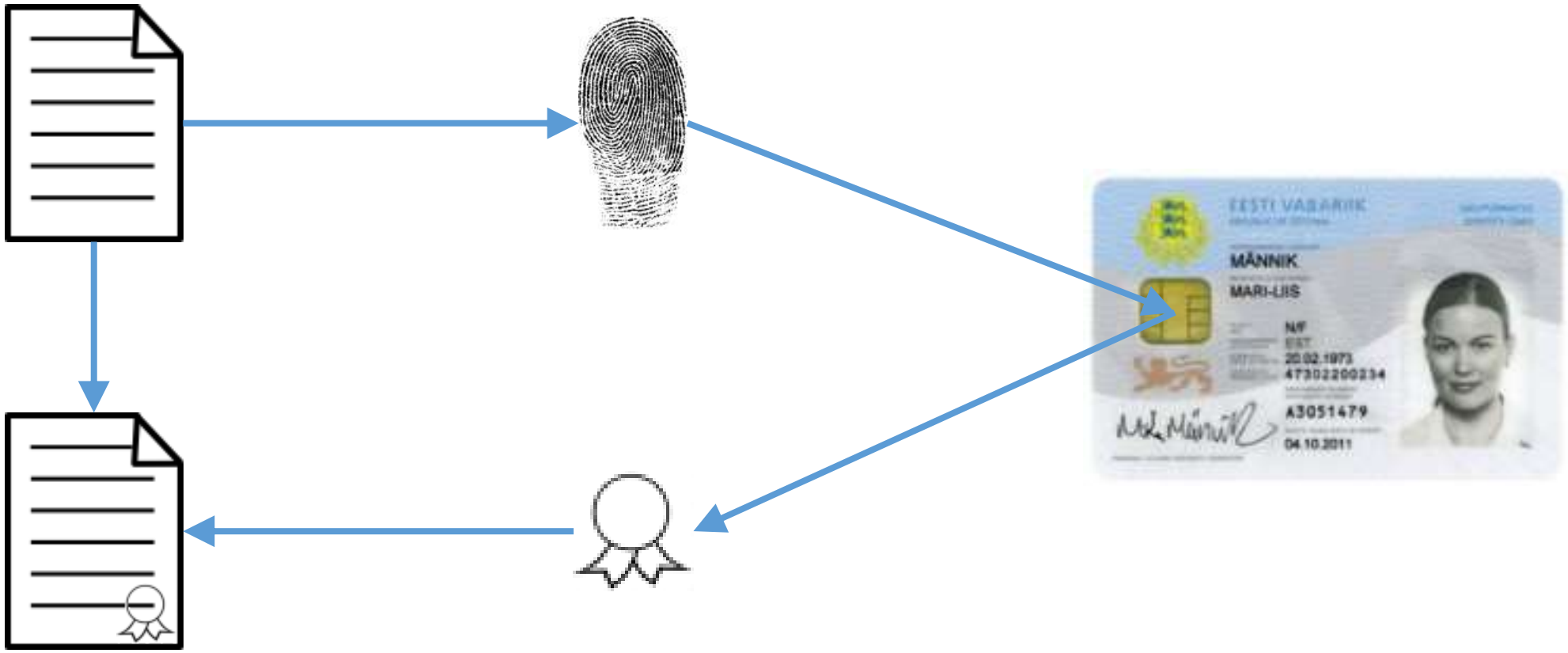
Hash Functions



Properties of Hash Functions

- Efficiently computable
  - Given x, easy to compute y = f(x)
- Pre-image resistant
  - Given y, infeasible to find x such that f(x) = y
- Second pre-image resistant
  - Given x, infeasible to find x’ ≠ x such that f(x’) = f(x)
- Collision resistant
  - Infeasible to find $x_1 \neq x_2$ such that $f(x_1) = f(x_2)$

Digital Signatures in Practice



A Bit of Theory

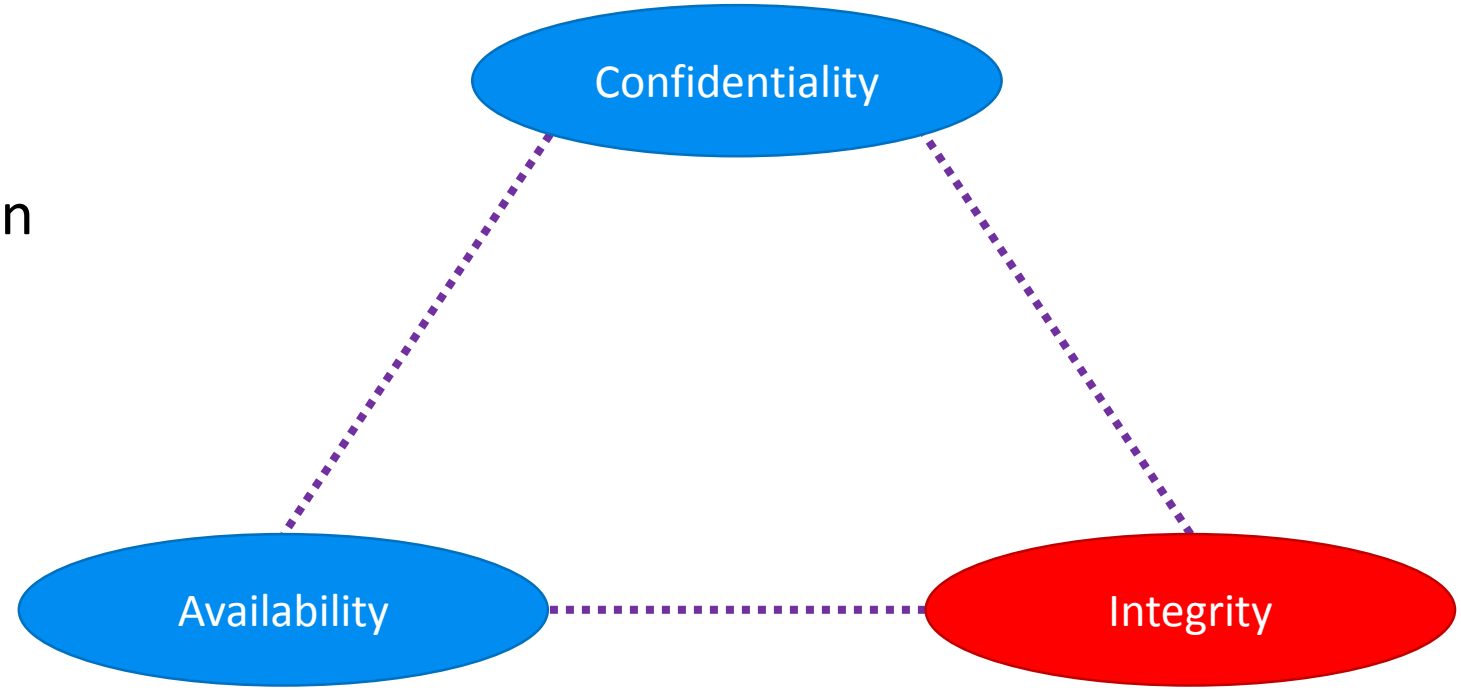
- Do one-way functions exist?

- In theory, nobody knows
  - Does P = NP?

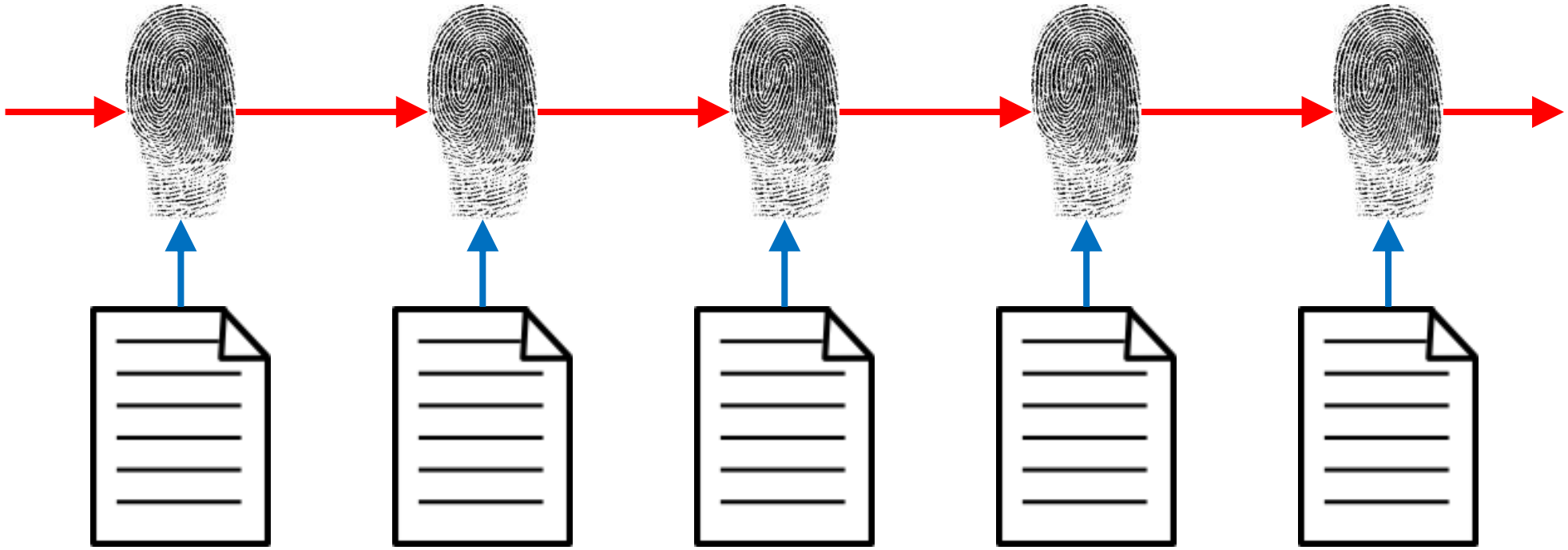
- In practice, several families
  - SHA-1, SHA-2, SHA-3, RIPEMD, …

Applications

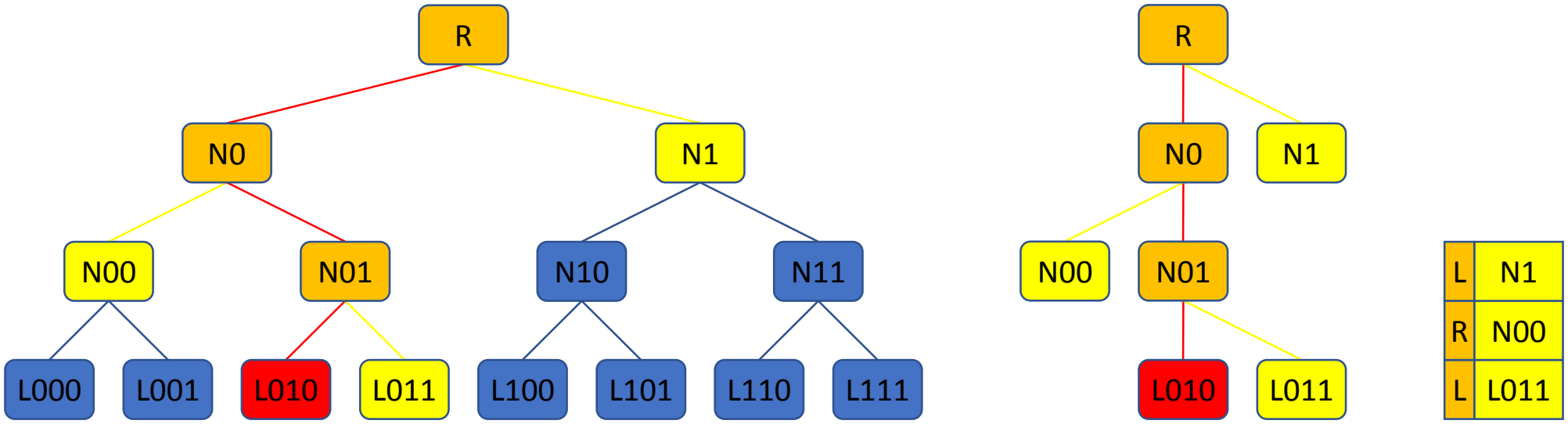
- Integrity
  - Time-stamping
- Authentication
  - Message authentication
- Non-repudiation
  - MAC + time-stamping



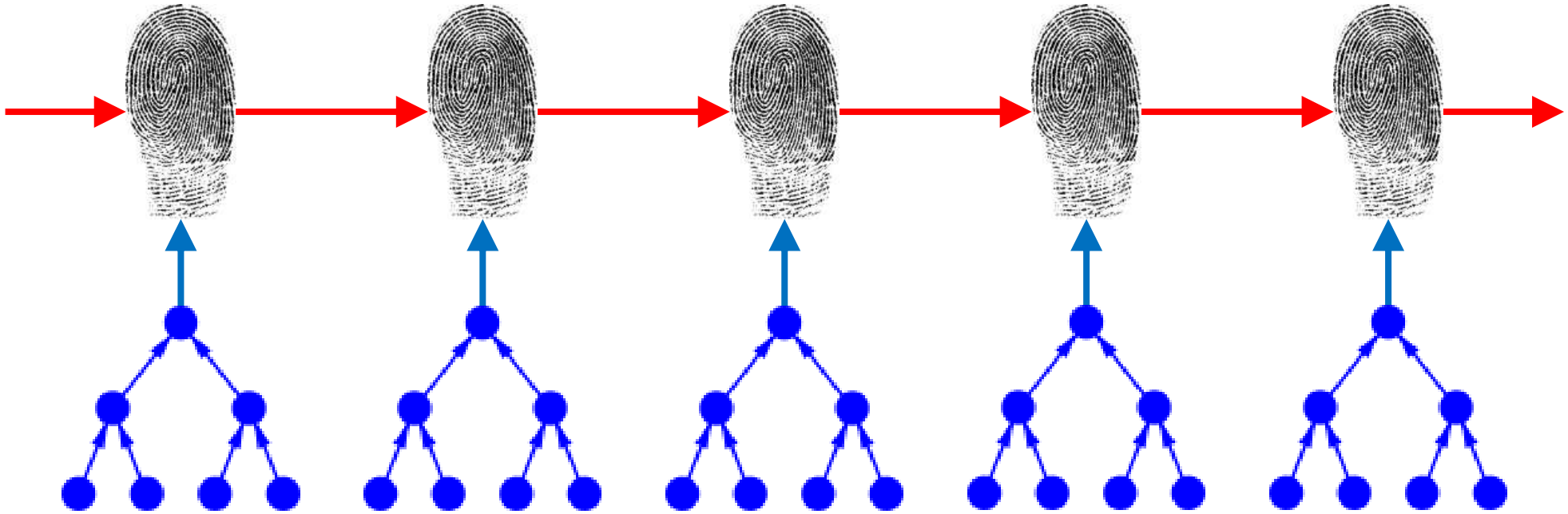
Time-Stamping with Hash-Chain



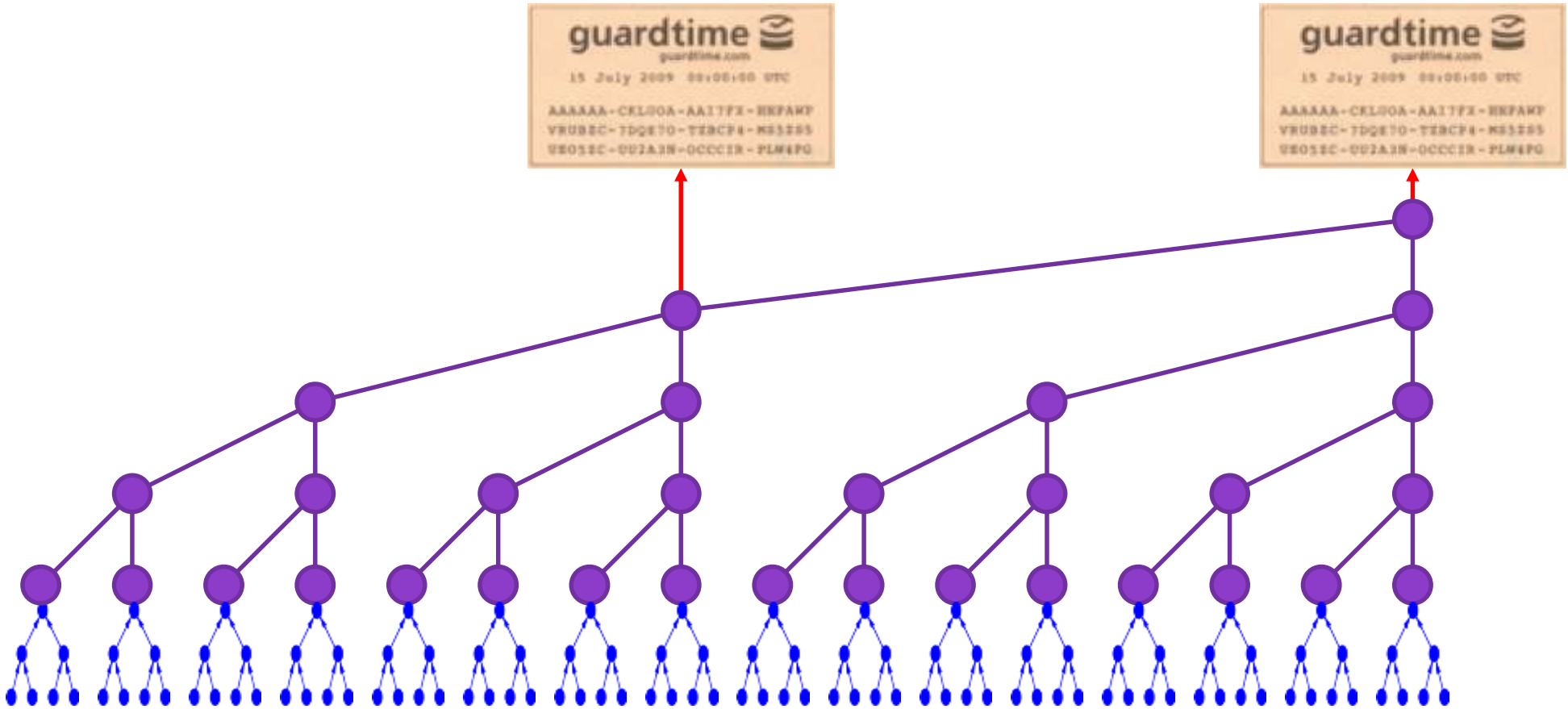
Merkle Trees



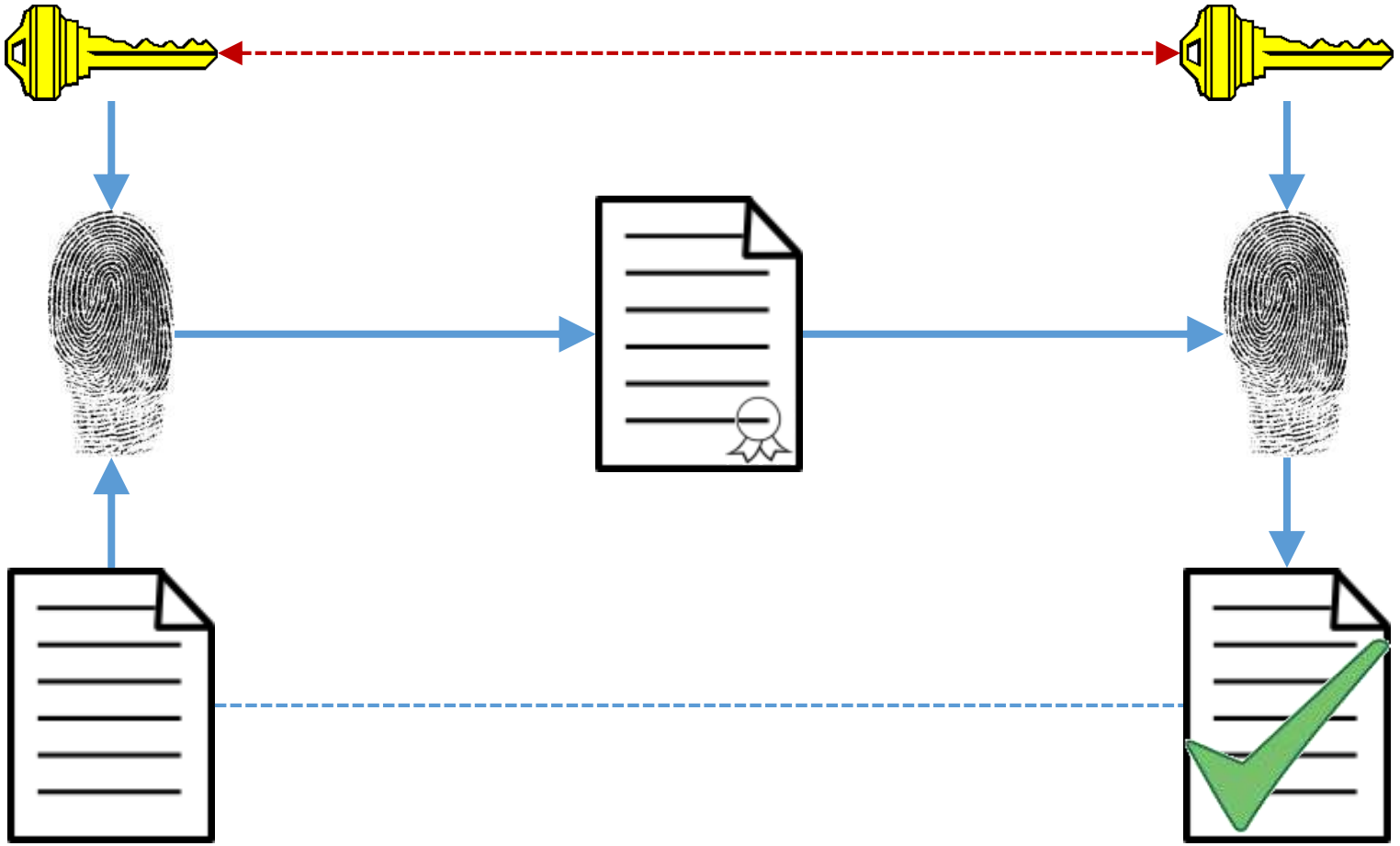
Time-Stamping with Block-Chain



Time-Stamping with Hash Calendar



Message Authentication



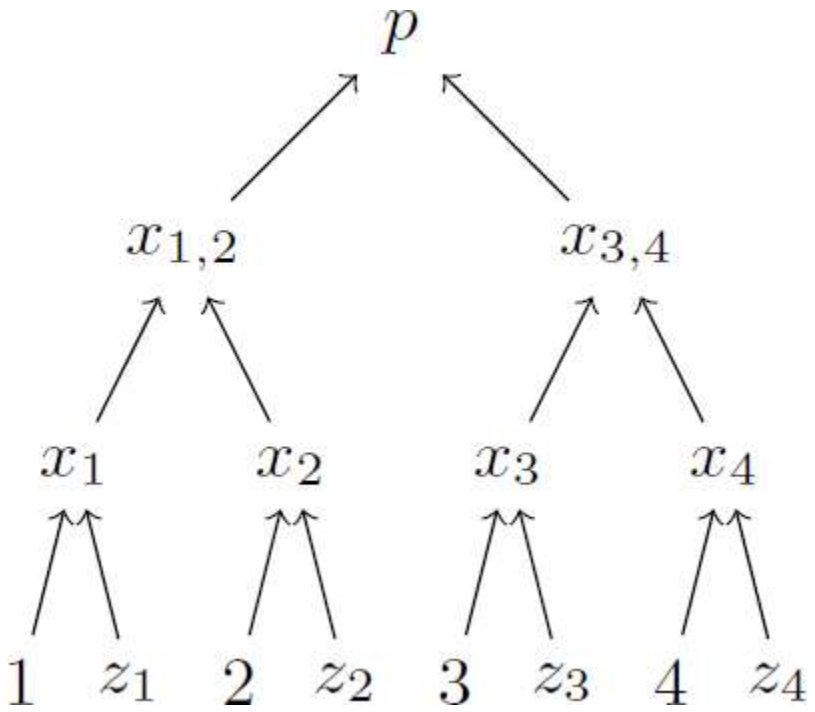
Digital Signatures

- Message authentication
  - Receiver can verify that the message came from sender
  - But can’t prove this to anyone else

- Digital signature
  - Receiver could not have created the message/signature pair by themselves
  - Can prove this to anyone, thus achieving non-repudiation

BLT – Key Generation

- To sign messages at times 1..T
  - Generate T keys $z_1..z_T$
  - Bind each key to a time slot $x_t = \text{hash}(t, z_t)$
  - Aggregate the bindings into a Merkle tree
  - Publish the root of the tree



BLT – Signing & Verification

- To sign message m at time t
  - Authenticate message with correct key: y = hash(m, z_t)
  - Time-stamp the authenticator to prove time of use: a_t = stamp(y)
  - Send m, (t, z_t, c_t, a_t)
- To verify signature s = (t, z, c, a) on message m with public key p
  - Check that z is indeed correct key for time t: hash(t, z) must link to p via the hash chain c
  - Check that z was used at correct time: hash(m, z) must link to hash calendar via the hash chain a

Questions?

Ahto Truu, ahto.truu@guardtime.com