SNIA NSF | NETWORKING STORAGE

Storage Networking Security Series: Applied Cryptography

Live Webcast

August 5, 2020

10:00 am PT

Today's Presenters

Alex McDonald
Vice Chair, SNIA NSF
NetApp

John Kim
Chair, SNIA NSF
NVIDIA

Eric Hibbard, CISSP, CIPT, CISA
Chair, SNIA Security TWG
Chair, INCITS TC CS1 Cyber Security
Chair, IEEE CS Cybersecurity & Privacy Standards Committee

Olga Buchonina
Chair, Blockchain Storage TWG
ActionSpot

SNIA Legal Notice

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations
- This presentation is a project of the SNIA
- Neither the authors nor the presenters are attorneys and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel
  - If you need legal advice or a legal opinion please contact your attorney
- The information presented herein represents the authors' personal opinion and current understanding of the relevant issues involved
  - The authors, the presenters, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information

    NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK

SNIA-At-A-Glance

SNIA-at-a-Glance

185
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

Learn more: **snia.org/technical** **@SNIA**

Technologies We Cover

- Ethernet
- iSCSI
- NVMe-oF
- InfiniBand
- Fibre Channel, FCoE
- Hyperconverged (HCI)
- Storage protocols (block, file, object)
- Virtualized storage
- Software-defined storage

SNIA NSF | NETWORKING STORAGE

Agenda

- Authenticating Users
- Encrypting Data
- Hashing
- Blockchain

Authenticating Users

Authenticating Users, Agencies, or Applications

- How do we know someone is who he/she claims to be?
- Certificates
  - Verify authenticity of a server or web site
  - Relies on trusted certificate authorities (CA)
- User Credentials
  - Username and password, biometrics, and/or two-factor authentication
- Often starts with asymmetric encryption
  - Then user credentials transmitted with symmetric encryption
  - Can also use hashes or blockchain

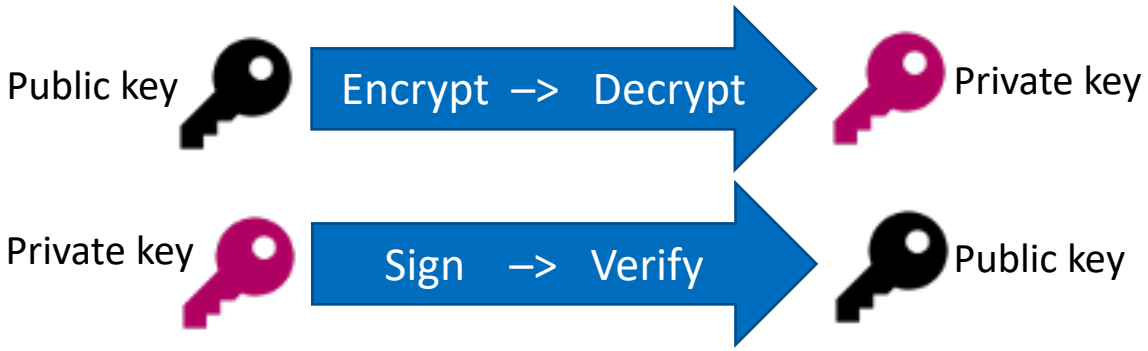
Symmetric vs. Asymmetric Keys

Symmetric

- Encrypt/decrypt with same key
- Server/app/user holds the key
- Share key with all users
- Mostly for limited audience

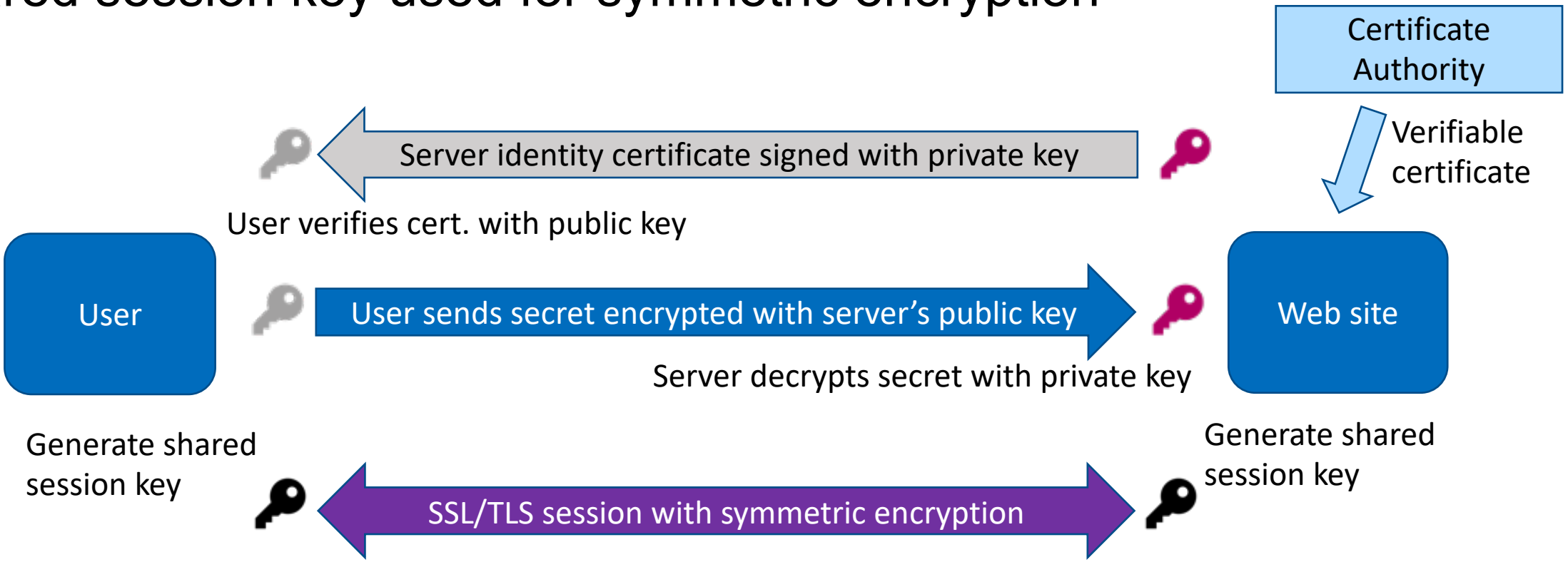
Asymmetric

- Uses public and private key pair
- Encrypt with public, decrypt with private key (one-way encryption)
- Encrypt with private, decrypt with public key (digital signing)



Authentication and Data Encryption on the Wire

- Asymmetric encryption to verify identity and establish session key
- Shared session key used for symmetric encryption



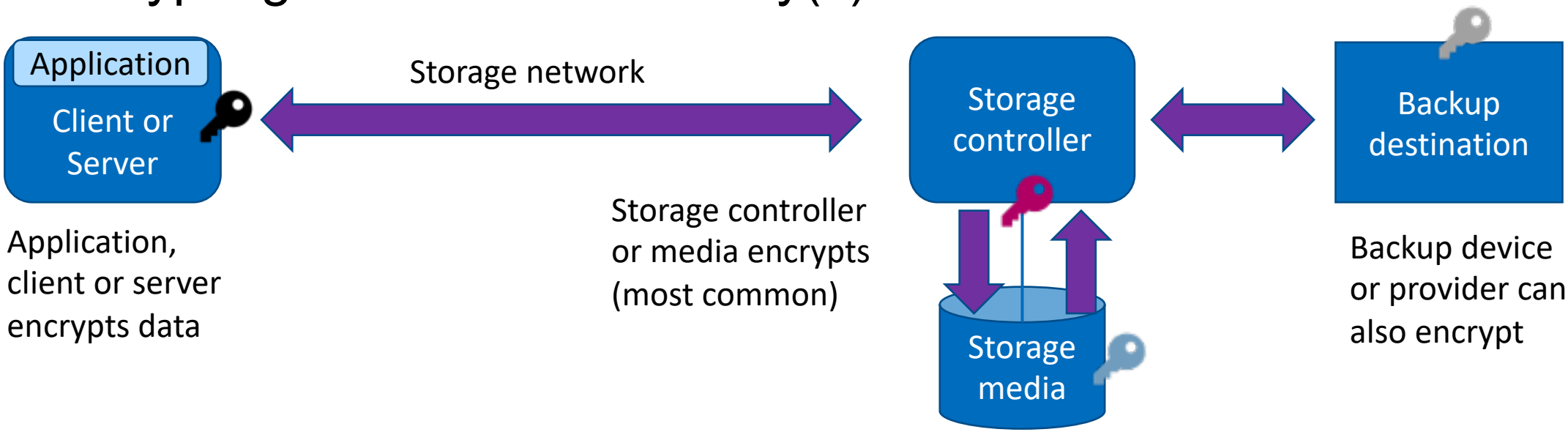
Encrypting Data

Encrypting Data

- Protects from unauthorized access
- Helps with regulatory compliance
  - Security and/or privacy
- Data at rest and/or on the wire
- Symmetric or asymmetric keys

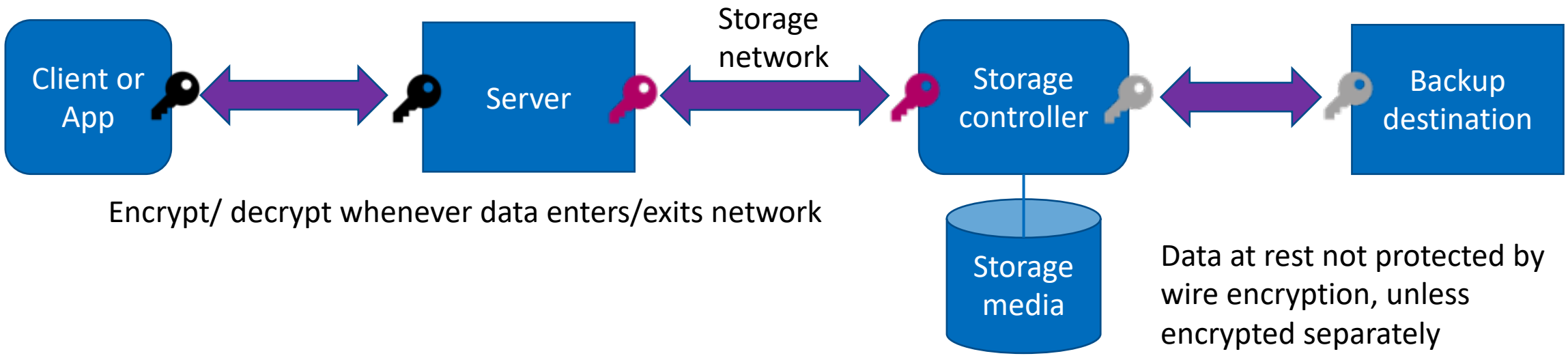
Encrypting Data at Rest

- Client, server, or storage media encrypts data
- Protects against theft/reuse of storage/media
- Does not protect data in transit (usually)
- The encrypting device holds the key(s)



Encrypting Data in Motion

- Client, server, and/or storage encrypt data while moving
- Protects against interception on the wire (not data at rest)
- Can offload encryption to SmartNIC, software library, or FPGA
- Uses symmetric keys

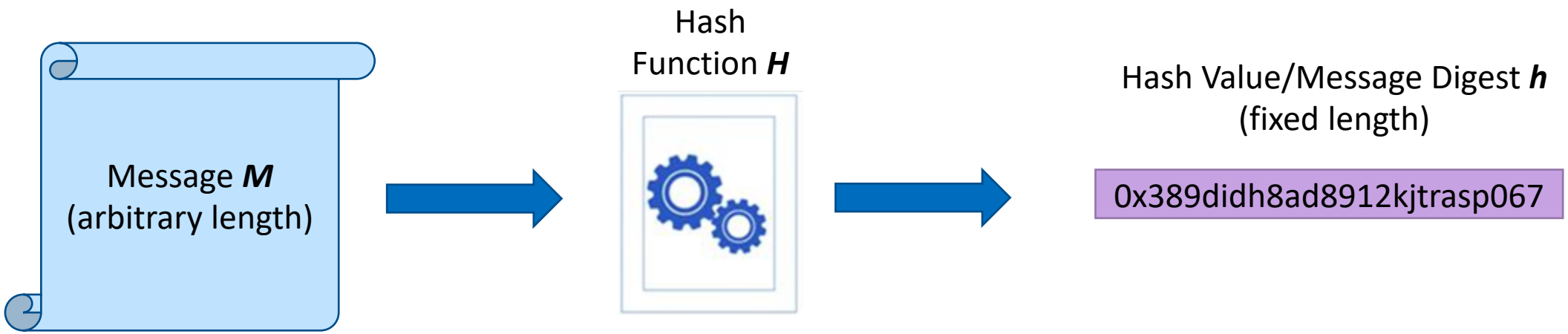


Hashing

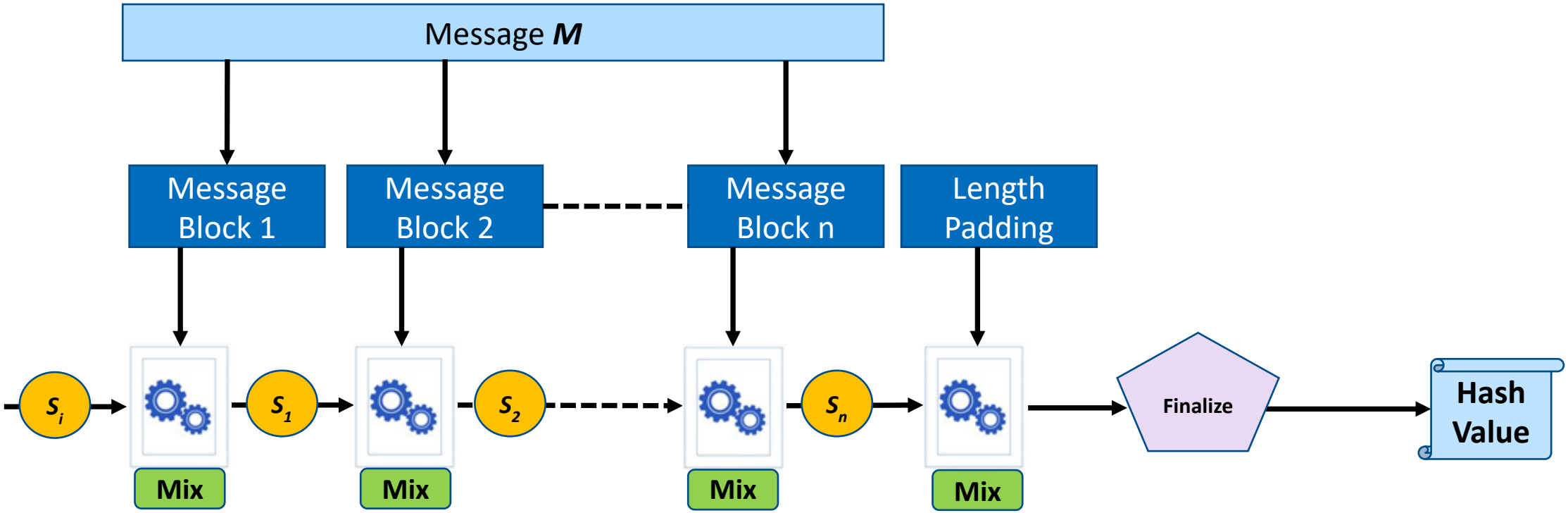
Attributes of Cryptographic Hash Functions

- Mathematical function that converts a numerical input value into another compressed numerical value
- Input to the hash function is of arbitrary length but output is always of fixed length
- It should be computationally hard to reverse a hash function (i.e., if a hash function ***H*** produced a hash value ***h***, then it should be a difficult process to find any input value ***x*** that hashes to ***h***.)
- It should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function

Abstract View of a Hash Function



Mechanics of a Hash Function



Popular Hash Functions

- Message Digest (MD)
  - A 128-bit hash function; block size 512 bits
  - In 2004, collisions were found in MD5, so MD5 is not recommended
- Secure Hash Function (SHA)
  - SHA-1 is a 160-bit hash function; block size 512 bits; collisions found
  - SHA-2 has multiple outputs (224, 256, 384, 512); block size 512/1024 bits
  - SHA-3 has multiple outputs (224, 256, 384, 512); block size 1152/1088/832/512
- RIPEMD-160, RIPEMD-256, and RIPEMD-320
- Whirlpool
- Shabal

Applications – Protecting Passwords

- Passwords are not stored in plaintext, but rather as hash values
- The password hashes are typically calculated using a *salt* (seed)
- Common authentication process:
  - User supplied ID is used to retrieve the password hash
  - User supplied password is hashed exactly as password file entries
  - The calculated password has is compared to the stored password hash
    - A match means authentication succeeds
    - A mismatch means authentication fails

- NOTE: Plaintext passwords should never be transmitted without encryption.

Applications – Data Integrity Check

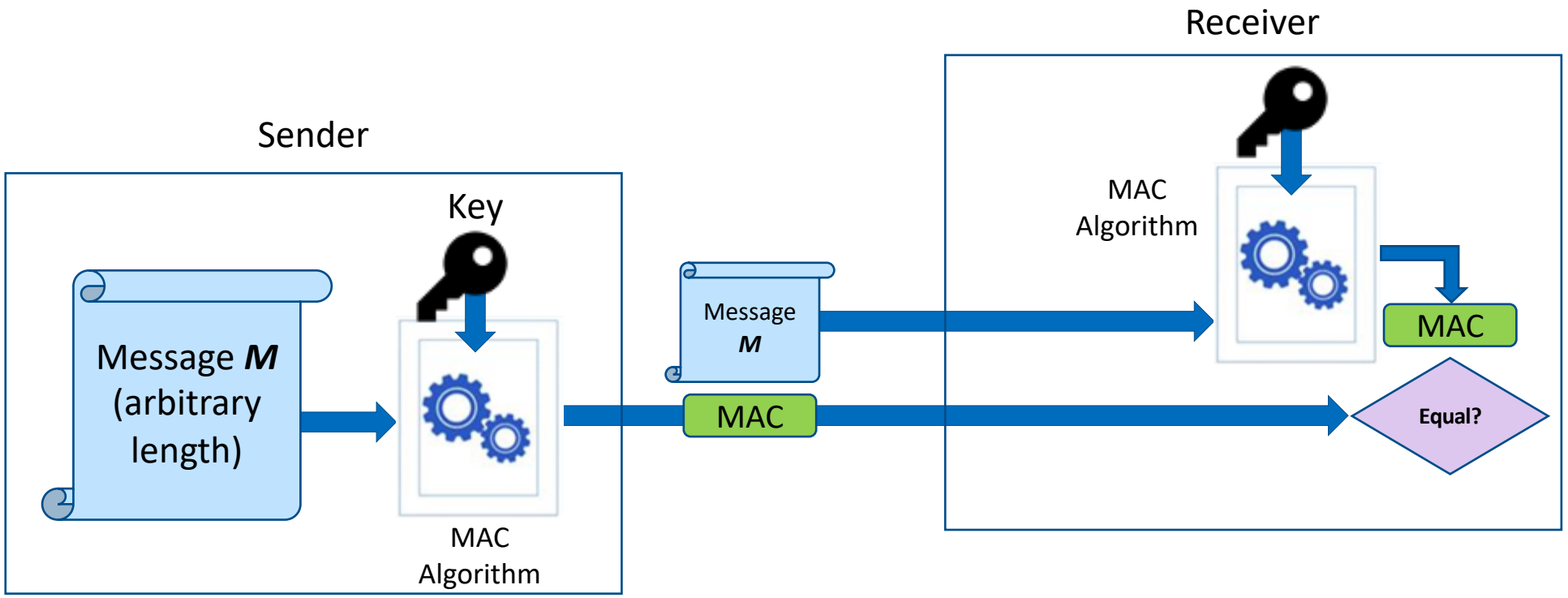
- Data integrity checking is the most common use of hash functions
- CRCs produce *checksums* for error checking; hashes produce *message digests* for data integrity
- Does not provide any assurance about originality (source)
  - The attacker, instead of modifying the data, can change the entire file and compute all together new hash and send to the receiver.
- Common data integrity check process:
  - Original data is input to hash function, creating hash value
  - Original data and hash value are “transferred”
  - To perform verification:
    - Target data (e.g., original data) is input to same hash function
    - If original hash value and calculated hash value match, the data is unchanged

Applications – Message Authentication

- Message Authentication Code (MAC) algorithm is a symmetric key cryptographic technique to provide message authentication; MAC algorithm are different from hash algorithms
- Input to the MAC algorithm is the message and a secret key (also known to the receiver)
- Output from the MAC algorithm is a MAC (sometimes known as a *tag*)
- Both the message and MAC are provided to receiver
- Authentication is based on the secret key, which is known only to the sender and receiver
- Data integrity is also verified because the message is used as input
- Requires establishment of shared secret prior to use of MAC

- MAC techniques do not provide non-repudiation services

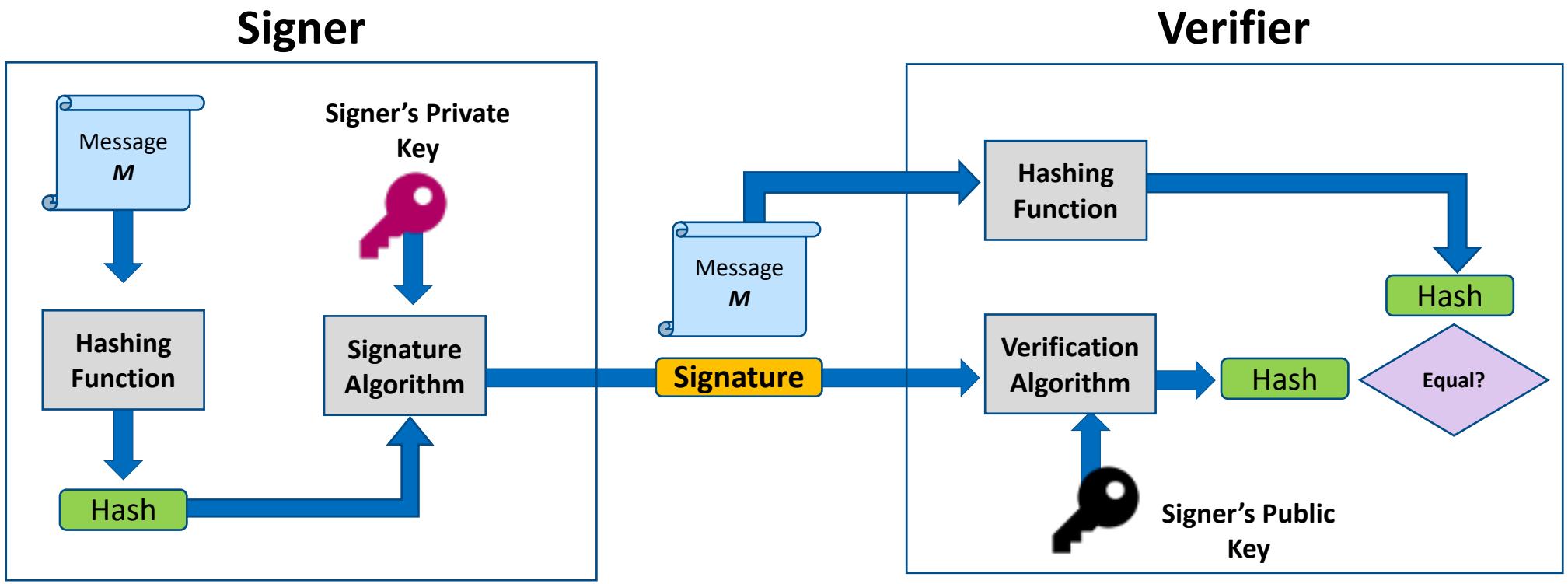
Using a MAC for Authentication



Applications – Digital Signatures

- Used when the receiver of message needs assurance that the message belongs to the sender and sender should not be able to repudiate the origination of that message
- The digital signature scheme is based on public key cryptography
- Digital signatures are the public-key primitives of message authentication; they are used to bind a signatory to the message
- A digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer
- Since the hash of data is a unique representation of data, it is sufficient to sign the hash in place of data (efficient)
- Adding public-key encryption to digital signature scheme, provides Confidentiality, Authentication, Integrity, and Non-repudiation

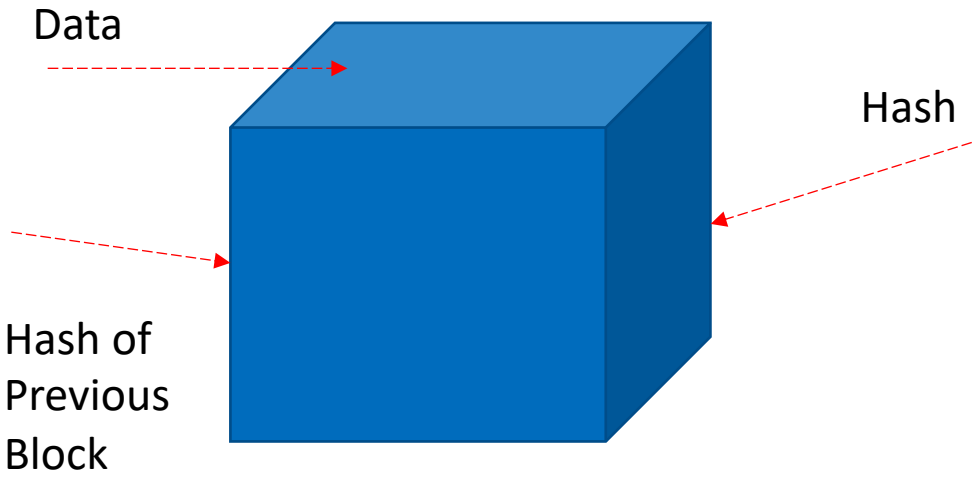
Using Digital Signatures



NOTE: When it is desirable to exchange an encrypted, signed messages, a process that uses an encrypt-then-sign approach is recommended. The receiver encrypts using the verifier's public key and the verifier decrypts using the corresponding private key.

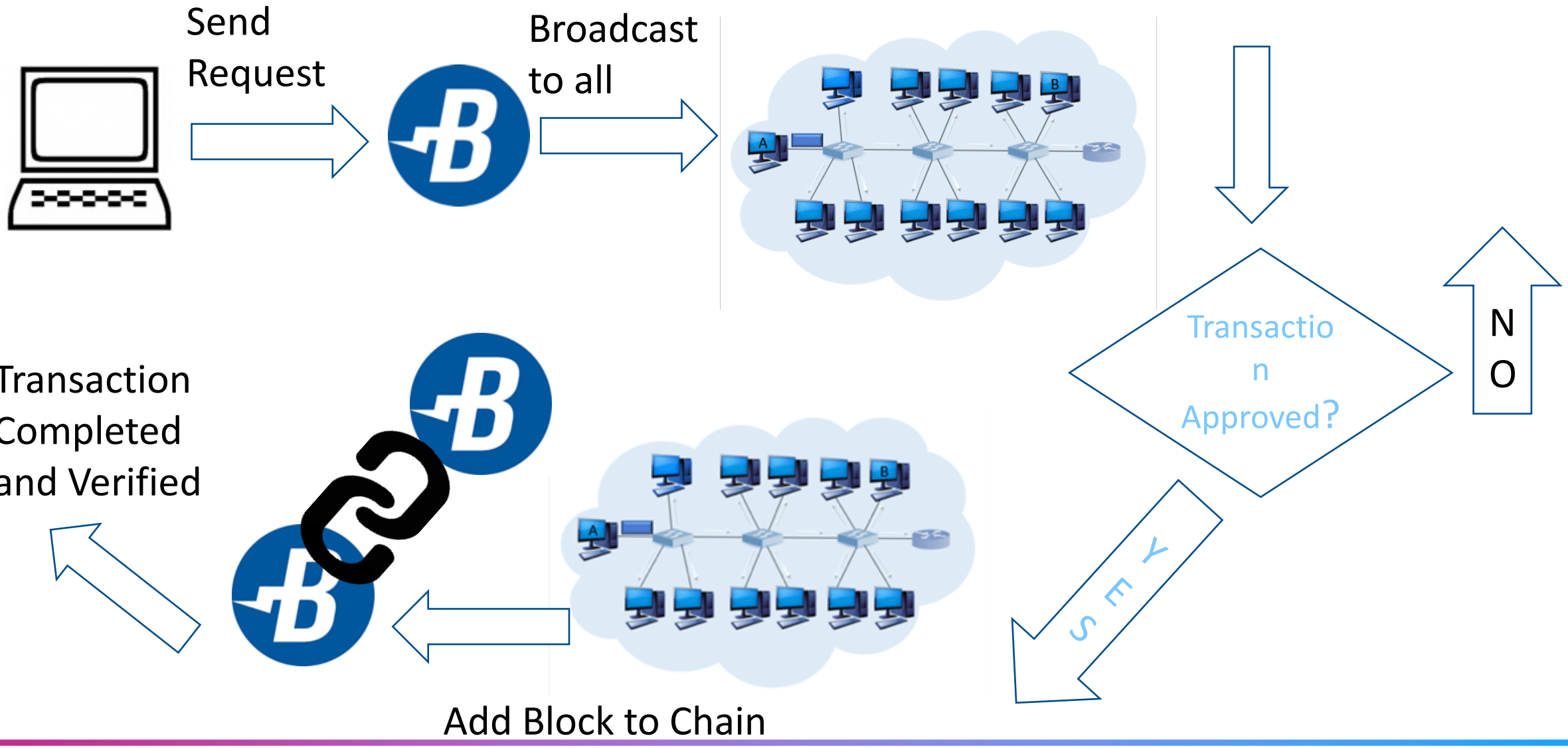
Blockchain

What is Blockchain?



- Blockchain is a distributed database of records stored in blocks.
- Blockchain is secured using peer validation in cryptography.
- Blockchain as a technology has several facets that directly or indirectly can impact user depending on implementation.

How Does Blockchain Work?



Network Limitations for Data Processing

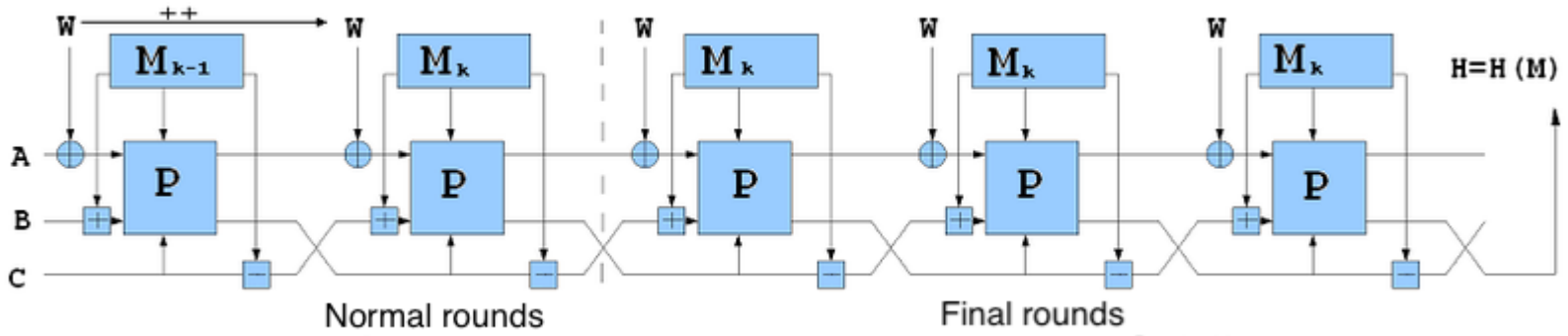
Blockchain technology is a chain of “blocks” that contain data. It is a distributed database system that accommodates a continuously growing list of immutable blocks. Blockchain allows consumers and suppliers to have a transparent mode of communication and transaction to connect directly without the need of middlemen.

- **KEY ELEMENTS :**
  - Signature verification
  - Redundancy.
  - Attaining consensus
  - Complexity
  - Energy and resource consumption
  - Security flaws

Storage Based Blockchain Protocols

What is Proof of Capacity

- Proof of Capacity uses the outputs of the shabal-256 cryptographic function to validate capacity to be used in mining.
- Shabal-256 currently is ASIC-resistant due to the IO requirements (as it requires writes).
- One time hashing process(plotting) versus continuous hashing.
- Mining process only involves reading the plots every new block(~ 4 min. average) and submitting the answers plus deadline(time to read to actual nonce).
- Power requirements for reading the plots greatly reduce overall energy consumed by the burstcoin blockchain.



Shabal Algorithm

Proof of Space

- **Proof of space** (**PoSpace**), also called **Proof-of-capacity** (**PoC**), is a means of showing that one has a legitimate interest in a service (such as sending an email) by allocating a non-trivial amount of memory or disk space to solve a challenge presented by the service provider.
- Proofs of space are very similar to proofs of work, except that instead of computation, storage is used. Proof-of-space is related to, but also considerably different from, memory-hard functions and proofs of retrievability.
- After the release of Bitcoin, alternatives to its PoW mining mechanism were researched and PoSpace was studied in the context of cryptocurrencies.
- Proofs of space are seen as a fairer and greener alternative due to the general-purpose nature of storage and the lower energy cost required by storage.
  - Example : BurstCoin vs Bitcoin

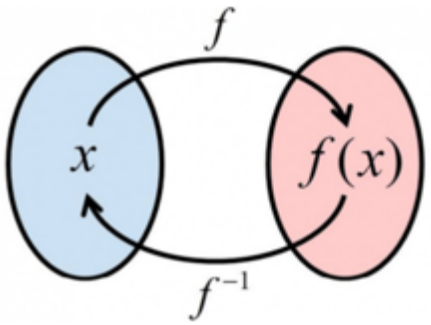
How it Works - Proof of Space ?

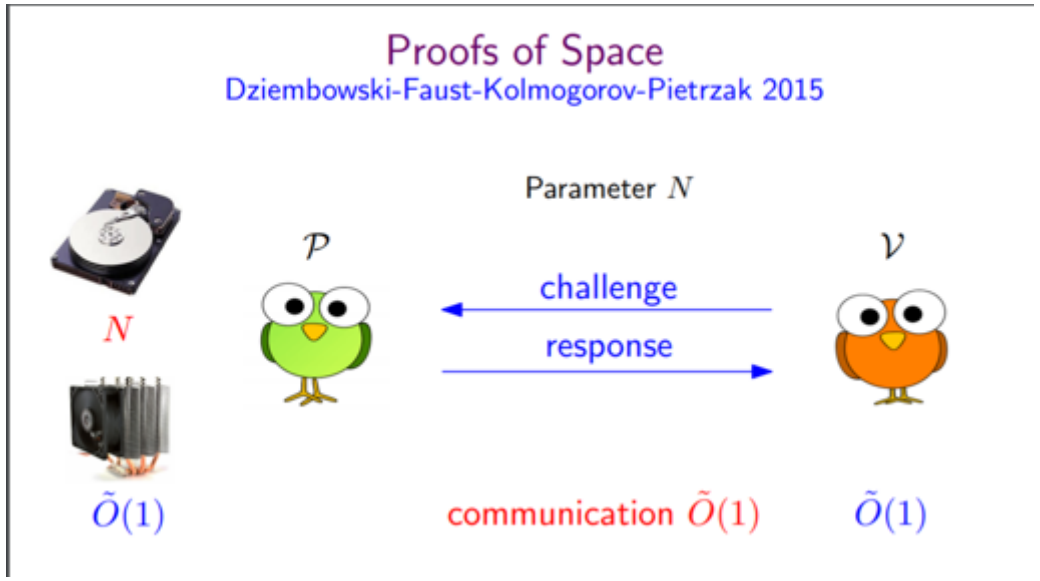
- A proof-of-space is a piece of data that a prover sends to a verifier to prove that the prover has reserved a certain amount of space.
- For practicality, the verification process needs to be efficient, namely, consume a small amount of space and time.
- For soundness, it should be hard for the prover to pass the verification if it does not actually reserve the claimed amount of space.

Way to implement:

One way of implementing PoSpace is by using hard-to-pebble graphs.

- The verifier asks the prover to build a labeling of a hard-to-pebble graph.
- The prover commits to the labeling.
- The verifier then asks the prover to open several random locations in the commitment.

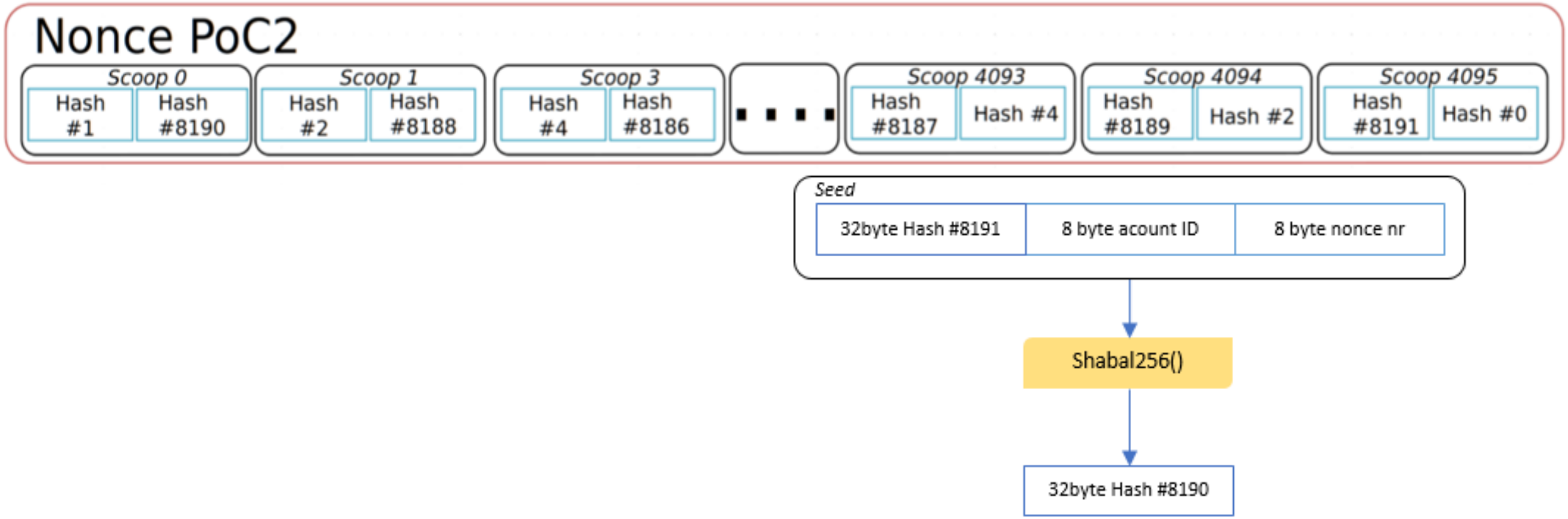




Krzysztof Pietrzak presenting at IST Austria
https://spotniq.files.wordpress.com/2018/08/spotniqbertinoro.pdf

Blockchain and Proof of Capacity

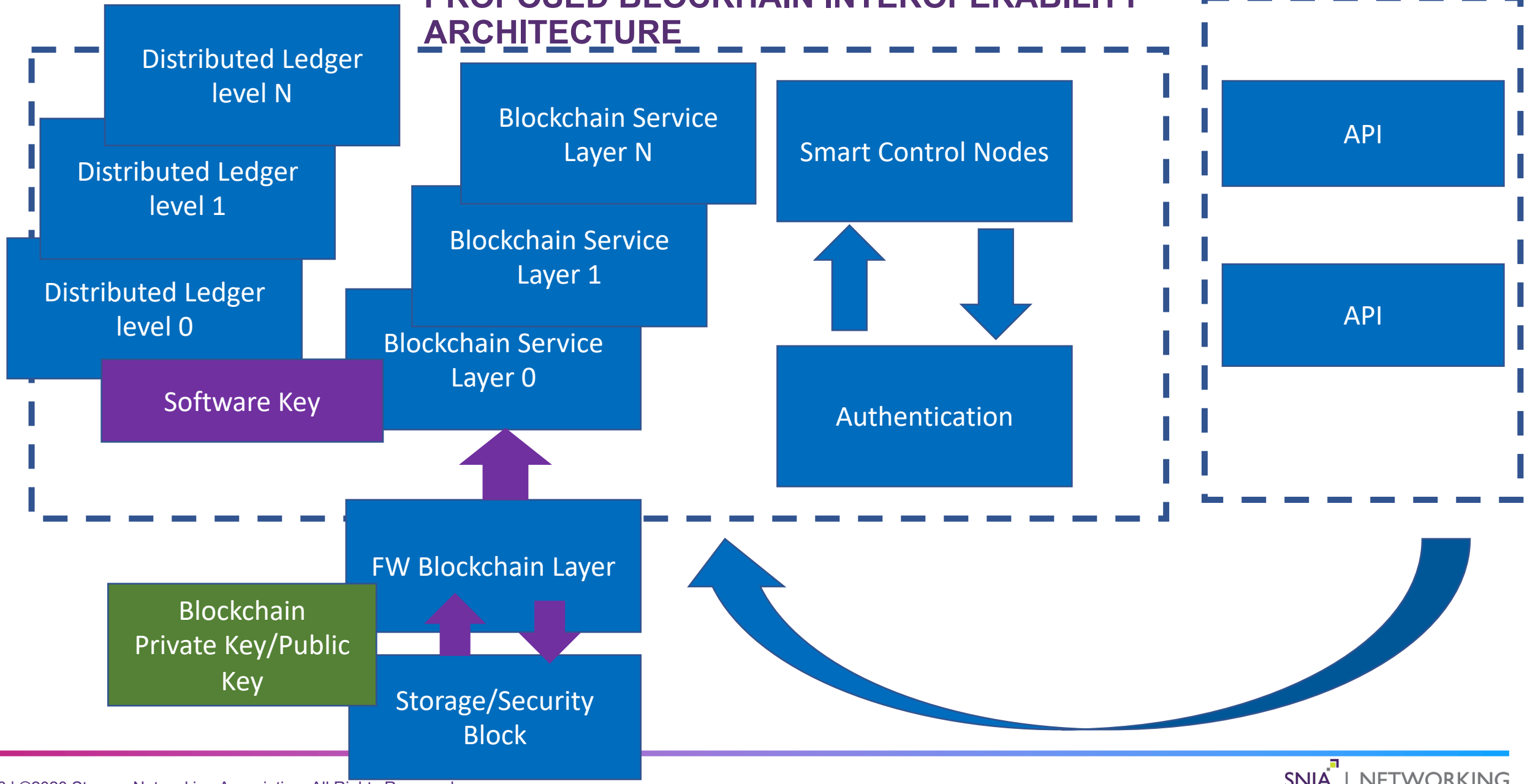
- Plotting is the process of generating plot files, which are just files storing a large number of pre-computed hashes. Each *plot* file contains one of more groups of 8192 hashes, these groups are called *nonces.* A nonce is exactly 256KiB in size (8192 x 32 bytes per hash). Additionally, each nonce is divided into 4096 pairs of hashes, the pairs are referred to as *scoops*. Each nonce can also be identified by its index number, ranging from 0 to 2^64.



Role Of Cryptography in Blockchain

- Cryptography role can be described as a way to protect certain information by keeping it private and confidential. It also allows to keep information intact and original when it's delivered from source to target
- Cryptography allows to maintain the anonymity where it's required
- Blockchain uses cryptography to protect it's users and create a trustless communication
- Blockchain uses digital signatures to authenticate it's users

PROPOSED BLOCKHAIN INTEROPERABILITY ARCHITECTURE



More SNIA Security Resources

- Storage Networking Security Webcast Series: On-demand at the SNIA Educational Library: snia.org/educational-library
  - Understanding Storage Security and Threats
  - Protecting Data at Rest
  - Encryption 101
  - Key Management 101
  - Security & Privacy Regulations
- Follow us on Twitter @SNIANSF for dates and times of others planned:
  - Protecting Data in Transit
  - Securing the Protocol
  - Securing the System: Hardening Methods

After this Webcast

- Please rate this webcast and provide us with your feedback
- This webcast and a copy of the slides will be available at the SNIA Educational Library https://www.snia.org/educational-library
- A Q&A from this webcast, including answers to questions we couldn't get to today, will be posted on our blog at https://sniansfblog.org/
- Follow us on Twitter @SNIANSF

Thank you!

