Flash Memory Summit

# Proof of authenticity of general IoT information with sensors and blockchain

Kenji Saito, Waseda University
Hiroshi Watanabe, National Yang Ming Chiao Tung University
Shogo Watanabe, Beyond Blockchain
Katsuo Taniguchi, Beyond Blockchain

35 YEARS OF NAND FLASH MEMORY INNOVATION

# Abstract


- Risks of applying blockchain and digital signatures to IoT services
  - Falsified records may be written, which cannot be revised
  - Public key certificates expire
  - Private keys and/or signature algorithms can be compromised
- Our proposal for sensor/actuator data to be recorded in blockchain
  - We assume tamper-evident hardware that can digitally sign occasionally
  - Atomically writes to both caller and blockchain service, which can fail
    - Writes just evidences (digests, not their preimages) to blockchain service
- Anticipated merits
  - Integrity and confidentiality even if private keys or signature algorithms are compromised, or public key certificates expire

# Introduction



- Increasingly widespread use of sensors/actuators
  - e.g. surveillance cameras, smart speakers, smart home appliances, …
- Example questions
  - Are images obtained from a certain surveillance camera really being captured by that camera?
  - Can we make sure that without trusting intermediate services?
- Our contribution is an answer to such questions

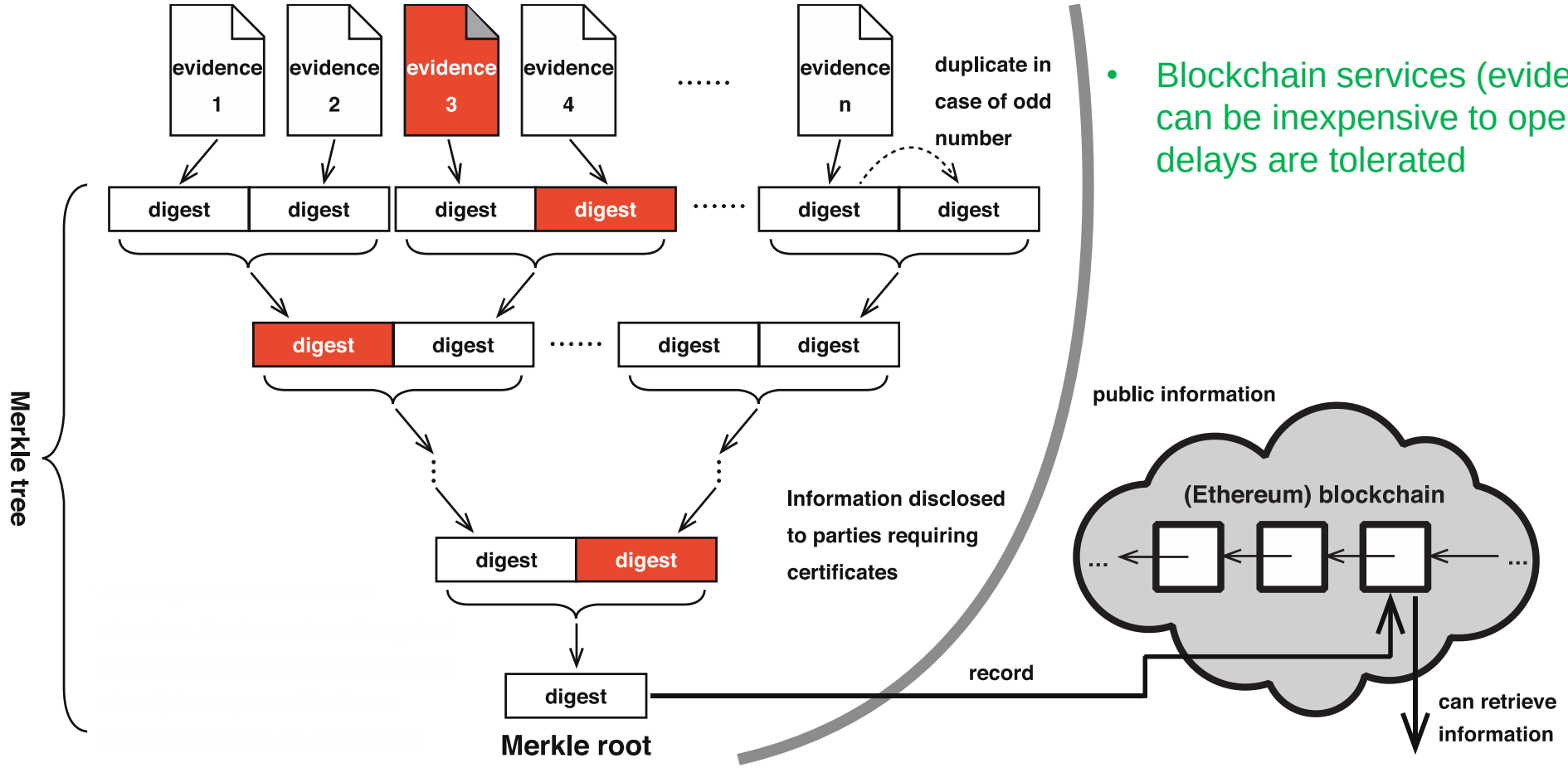
# Background – What Blockchain is really about



- Blockchain is a technology that aims to achieve censorship resistance in the broadest sense
  - 1) Self-sovereignty : Users have the right to make their own decisions, including the ability to create their own accounts
  - 2) Censorship resistance in the narrow sense : No one can stop users from recording data or verifying that data
  - 3) Fault tolerance : Recording or verifying data is not even stopped by a failure
  - 4) Tamper resistance : Once recorded, data cannot be erased or changed, nor can data that was not there be fabricated

# Background – Merkle proof for evidences




- Blockchain services (evidence services) can be inexpensive to operate if some delays are tolerated

*** Starting with the digest of evidence 3, verifier will know the series of digests to be concatenated, so they can reproduce the calculations down to the Markle root, and confirm that the root matches the value recorded in the (Ethereum blockchain) smart contract**

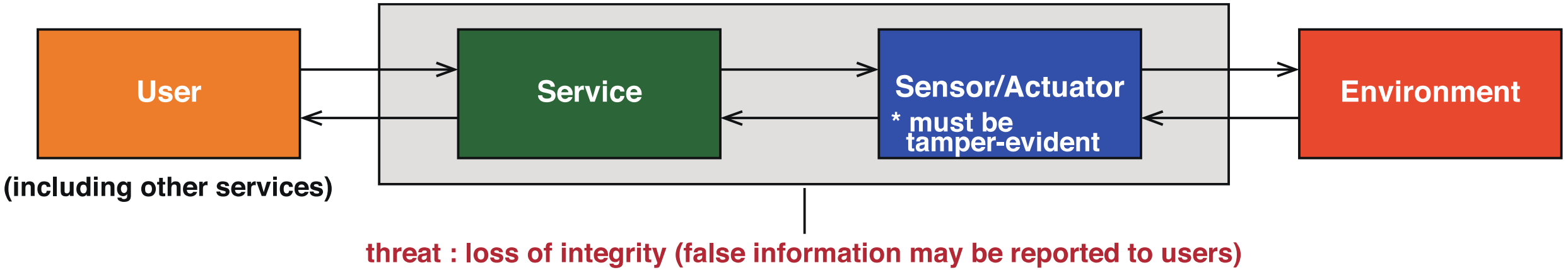
# Background – Smart contract for evidence service



```
contract BBcAnchor {
  mapping (uint256 => uint) public _digests;
  constructor () public {
  }
  function getStored(uint256 digest) public view returns (uint block_no) {
    return (_digests[digest]);
  }
  function isStored(uint256 digest) public view returns (bool isStored) {
    return (_digests[digest] > 0);
  }
  function store(uint256 digest) public returns (bool isAlreadyStored) {
    bool isRes = _digests[digest] > 0;
    if (!isRes) {
      _digests[digest] = block.number;
    }
    return (isRes);
  }
}      /* This contract saves the current block number for a stored digest */
```

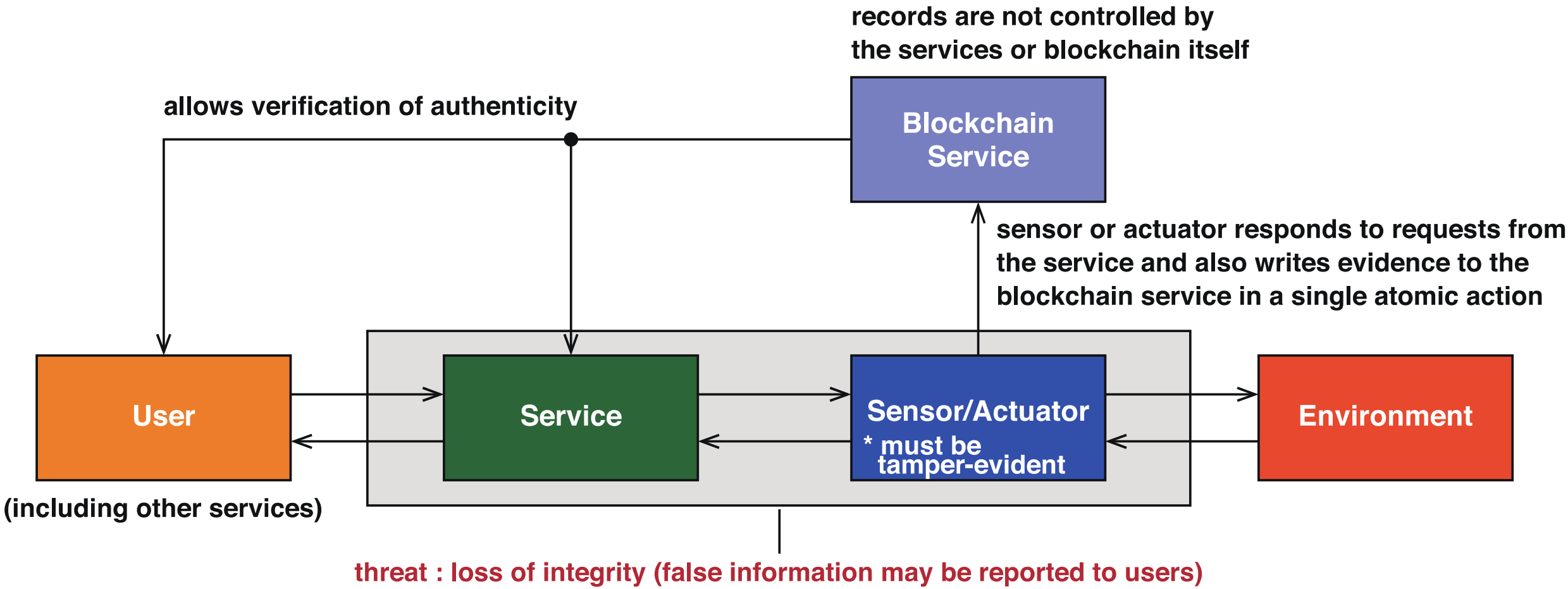
# Problem




- Make sure that genuine data produced by sensor/actuator is communicated to the user
  - On the assumption that sensor/actuator is tamper-evident, and is capable of digitally signing data
  - Data generated before the private key is compromised, the signature algorithm is compromised, or the public key certificate expires must be verifiable
  - Data can be sporadic or streamed, while we must assume that some data may be missing due to failure

# Proposed Solution – Overview




- This is a direct extension of our previous work on authenticity verification of RFID [Watanabe, et. al., 2021]

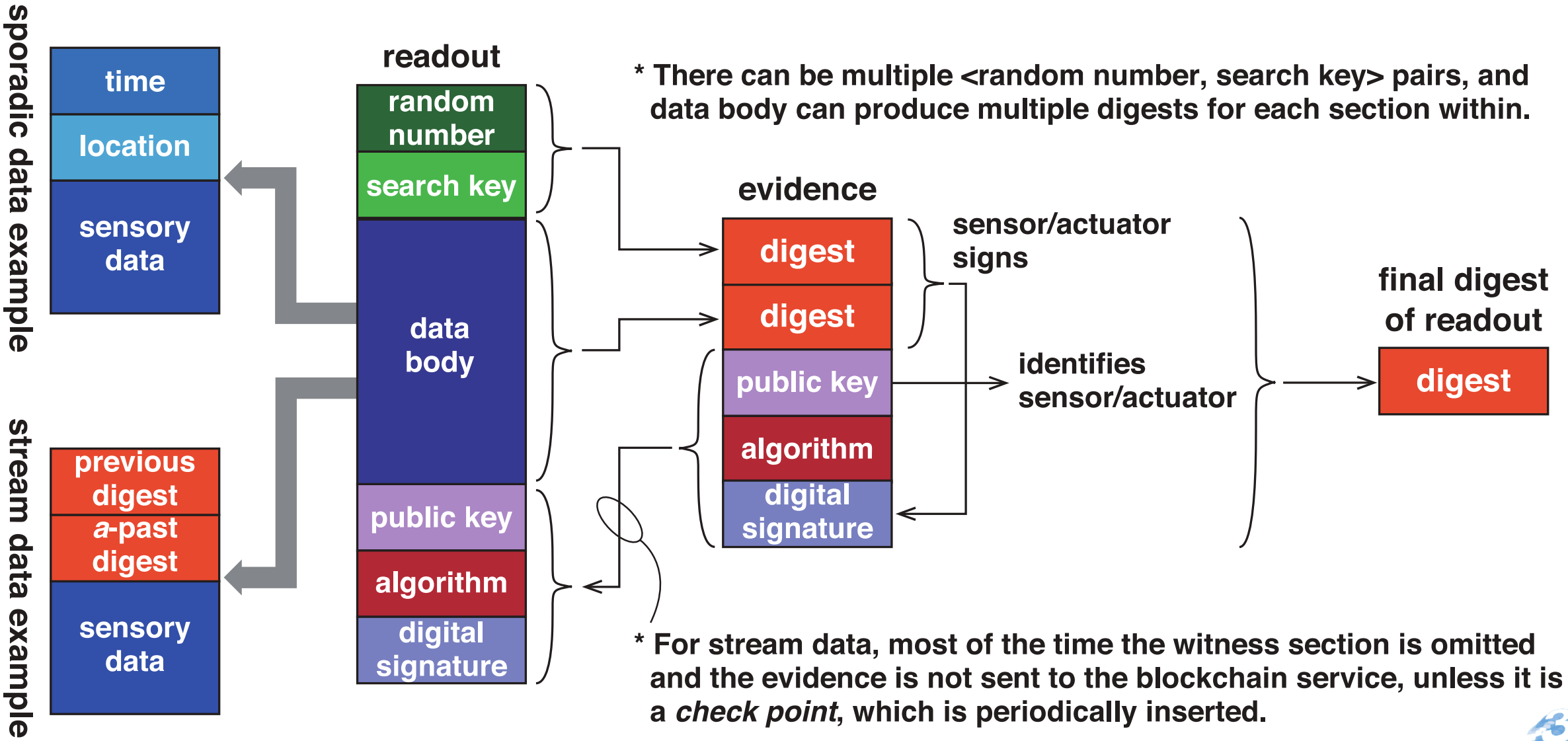
# Proposed Solution – How It Works



- Case 1 : sporadic data (e.g. RFID, event logging)
  - Sensor/actuator sends in an atomic action 1) signed readout to the IoT service and 2) its evidence that conceals actual data to blockchain service
  - Blockchain service builds a Merkle tree and writes its root to Ethereum blockchain periodically
- Case 2 : stream data (e.g. video from surveillance cameras)
  - Sensor/actuator sends hash-chained readouts to the IoT service, and only periodically signs a readout, which is handled the same as above

# Proposed Solution – Data structure

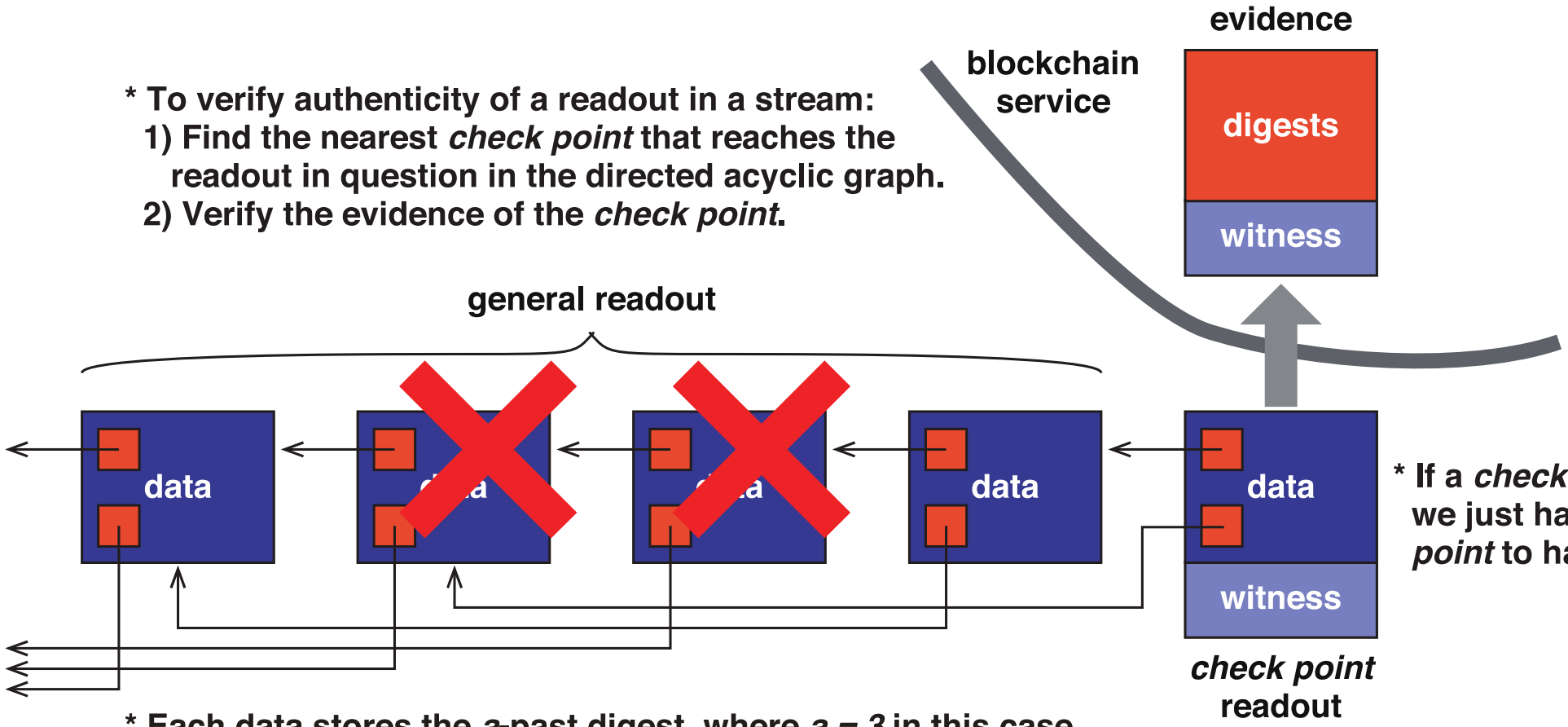


# Proposed Solution – Handling stream data w/ losses



* To verify authenticity of a readout in a stream:
  1) Find the nearest *check point* that reaches the readout in question in the directed acyclic graph.
  2) Verify the evidence of the *check point*.



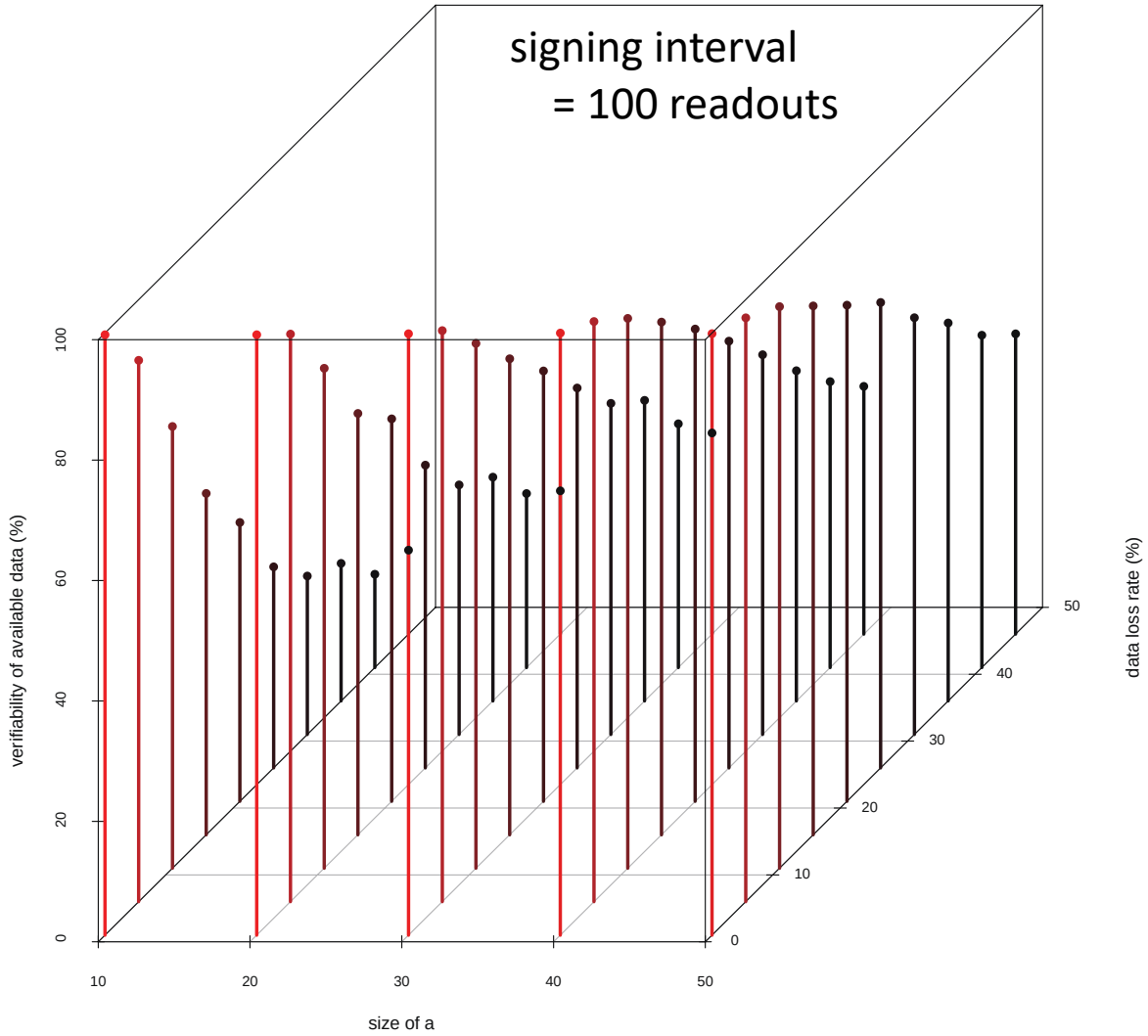

* If a *check point* or all it points are lost, we just have to wait for another *check point* to have recent data verifiable.

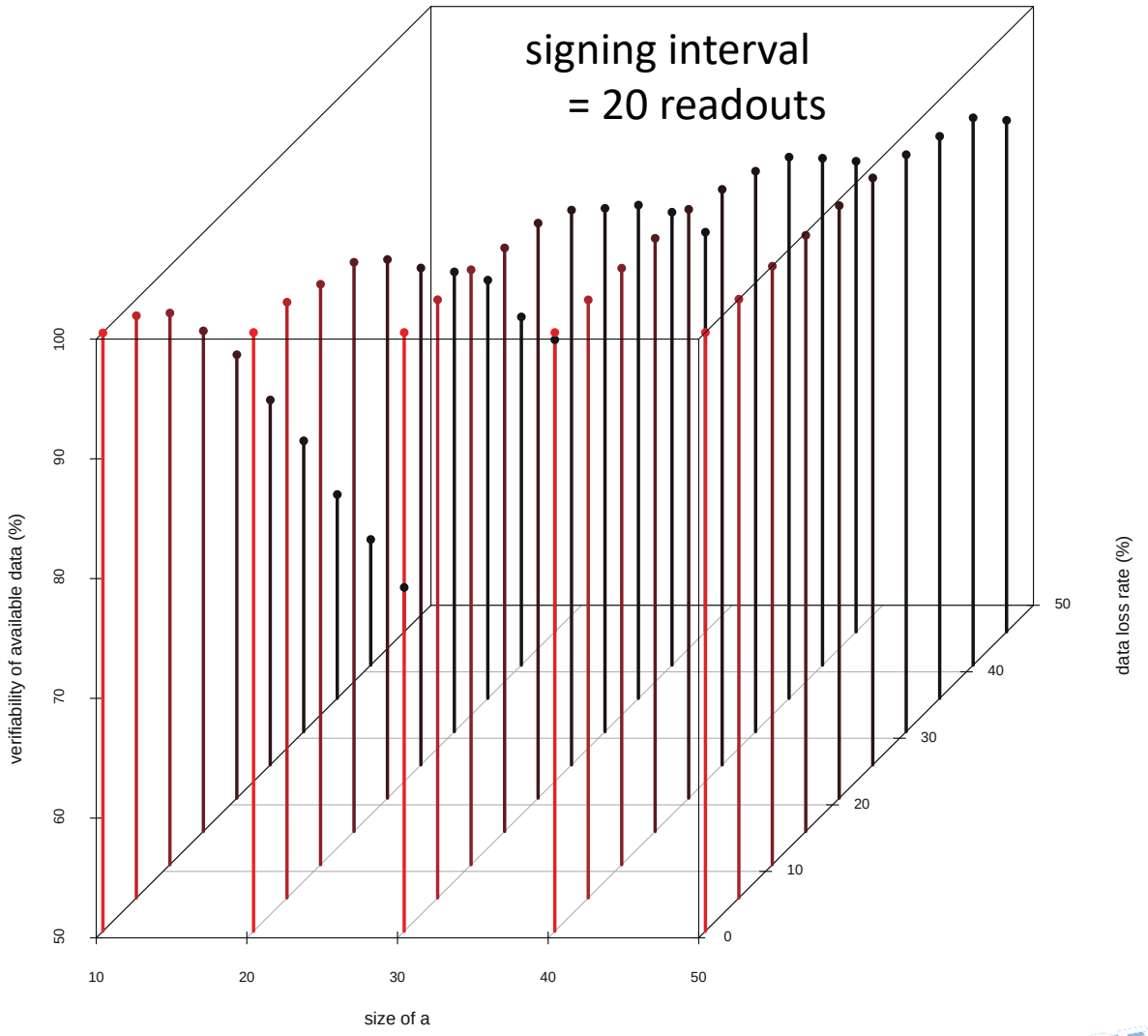
* Each data stores the *a*-past digest, where *a = 3* in this case, as well as the previous digest.
* This tolerates up to *a - 1* consecutive data losses.

This is a straightforward application of a technique introduced in [Golle and Modadugu, 2001].

# Simulation Results



# Discussion



- This proposal is a combination of established methods
  - Authenticity verification of (RFID) sensor data [Watanabe, et. al., 2021] (our previous work), and
  - Authenticating streamed data in the presence of losses [Golle and Modadugu, 2001]
- Public key certificates must be certified
  - This can be done by sensor/actuator vendors using the same proposed method, as described in our previous work [Watanabe, et. al., 2021]
- Stream data solution is also…
  - Useful for (frequent) sporadic data, as it can tolerate loss of data and failure of the atomic action
- If an atomic action fails, it is handled as either
  - 1) Data loss, or
  - 2) General readout without witness, and the sensor/actuator tries to sign the next available readout

# Conclusions



- We have proposed a technique to make sure that genuine data produced by general sensor/actuator is communicated to the user even through untrusted intermediate IoT services
  - By extending our previous work designed for RFID [Watanabe, et. al, 2021], and
  - By applying a technique introduced in [Golle and Modadugu, 2001] for handling data losses

- This solves some of the problems of our previous work, and provides a practical way to authenticate both sporadic and streamed sensor/actuator data

# References and Acknowledgment


- Gennaro, R., & Rohatgi, P. (2001). How to Sign Digital Streams. *Information and Computation*, *165*(1), 100–116. doi:10.1006/inco.2000.2916
- Golle, P., & Modadugu, N. (2001). Authenticating Streamed Data in the Presence of Random Packet Loss (Extended Abstract). *Network and Distributed System Security Symposium (NDSS) 2001*.
- Watanabe, H., Saito, K., Miyazaki, S., Okada, T., Fukuyama, H., Kato, T., & Taniguchi, K. (2021). Proof of Authenticity of Logistics Information with Passive RFID Tags and Blockchain. *Proceedings of 2021 IEEE International Conference on Electronic Communications, Internet of Things and Big Data*.



- This research has been supported by JSPS KAKENHI Grant Number JP21H04872