Information theoretically secure distributed storage system with quantum key distribution network and password authenticated secret sharing scheme

M. Fujiwara[1*], A. Waseda[1], R. Nojima[1], S. Moriai[1],W. Ogata[2] and M. Sasaki[1]
[1]National Institute of Information and Communications Technology (NICT)
[2]Tokyo Institute of Technology

Tokyo Tech

[*]Email: fujiwara@nict.go.jp

NICT 独立行政法人 情報通信研究機構

Contents

NICT

Requests for storage system

Requests

We want a system which can transmit, store, and process critical data with information theoretical security.

Requirements for the system

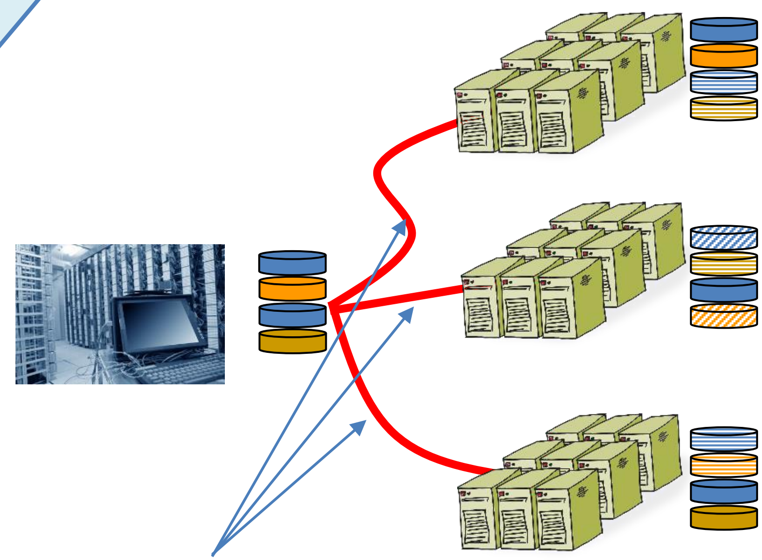
1. Confidentiality :
   The data should be accessible only to authorized parties.
   ➡ Information theoretically secure encryption.
2. Integrity :
   The data should remain unchanged.
   ➡ The data owner can check integrity of the data.
3. Availability :
   The data should be available whenever required.
   ➡ Redundant data backup, fail safe mechanism.
4. Functionality :
   The data can be processed without decryption.
   ➡ Full homomorphic encryption.

Combination SS and QKD

1-1 Secret sharing scheme (Shamir's (k, n) threshold secret sharing) with QKD

-Confidentiality (without authentication, transmission) of storage
-Integrity (checked by owner)
-Availability
-Functionality
with the assumption of (*k*, *n*)-threshold scheme.

Secret sharing
(*k*, *n*)-threshold scheme

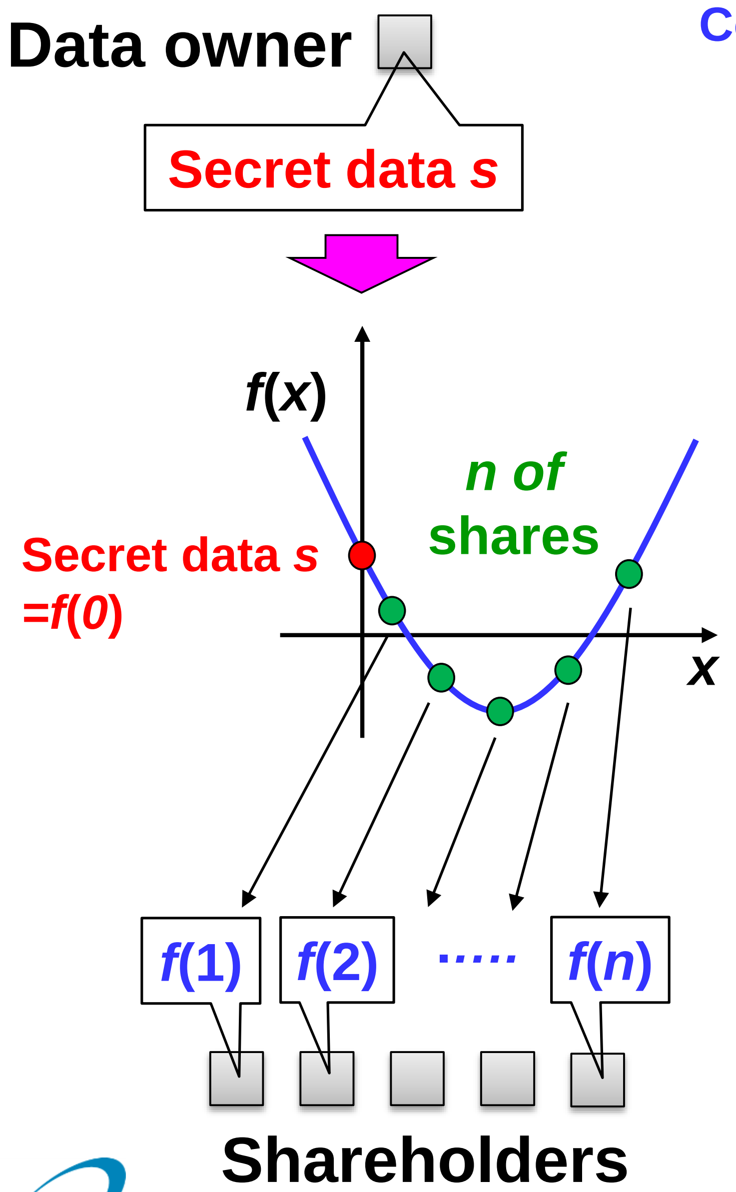


Confidentiality and integrity of transmitted data

QKD+ one time pad encryption with Wegman-Carter authentication

Implementation detail;
M. F. et al., *Int. J. Network security* 17, 34-39 (2015).

Shamir's (k, n) threshold secret sharing



Communications of the ACM, 22, 612-613 (1979).

For a given **secret data s**,

- Generate a polynomial of order k-1

$$f(x) = s+a_1x+\ldots+a_{k-1}x^{k-1}$$

with **secret data "s"** a constant term.
- Create n of coordinates **"shares"**

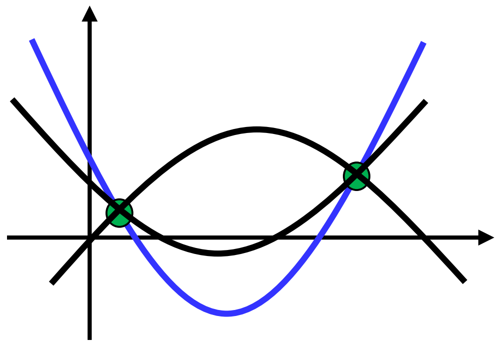
$[1, f(1)], \cdots, [n, f(n)]$

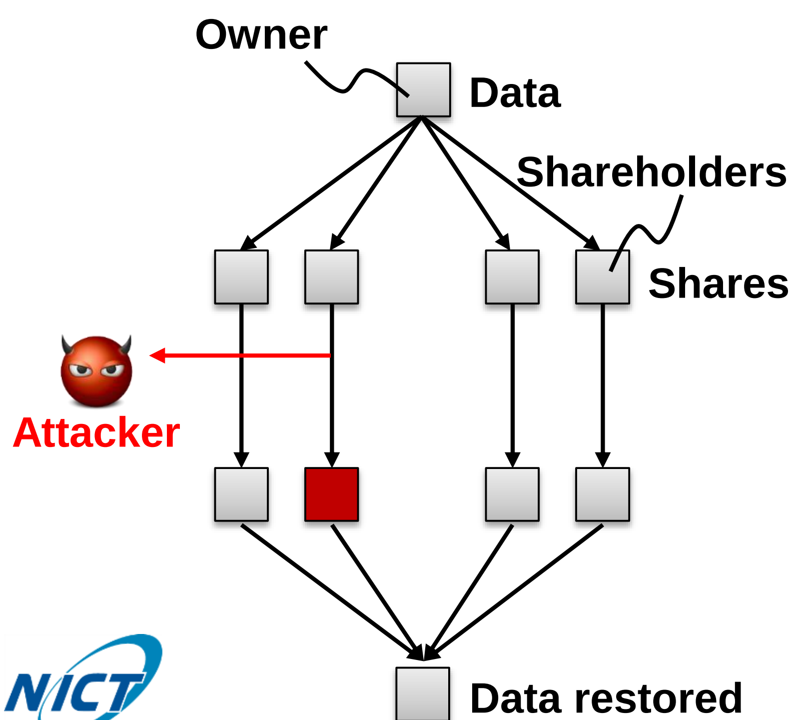
- Store the shares in data servers, called "shareholders".
- The shares themselves need not necessarily be encrypted.
- The coefficients "*a*"s are deleted covertly.

Shamir's (*k*, *n*) threshold secret sharing

Communications of the ACM, 22, 612-613 (1979).

Ex. (3,4)-threshold scheme





With shares less than *k*,
the original data can never be re-constructed.
There remain infinitely many possibilities of polynomial.

Information theoretic confidentiality

With more than *k* of shares,
the polynomial *f*(*x*) can be specified.

Even if *n*-*k* of shares are lost,
the data can be re-constructed.

Availability

Shares can be added and multiplied.

Functionality (Full homomorphism)

Shamir's (*k*, *n*) threshold secret sharing

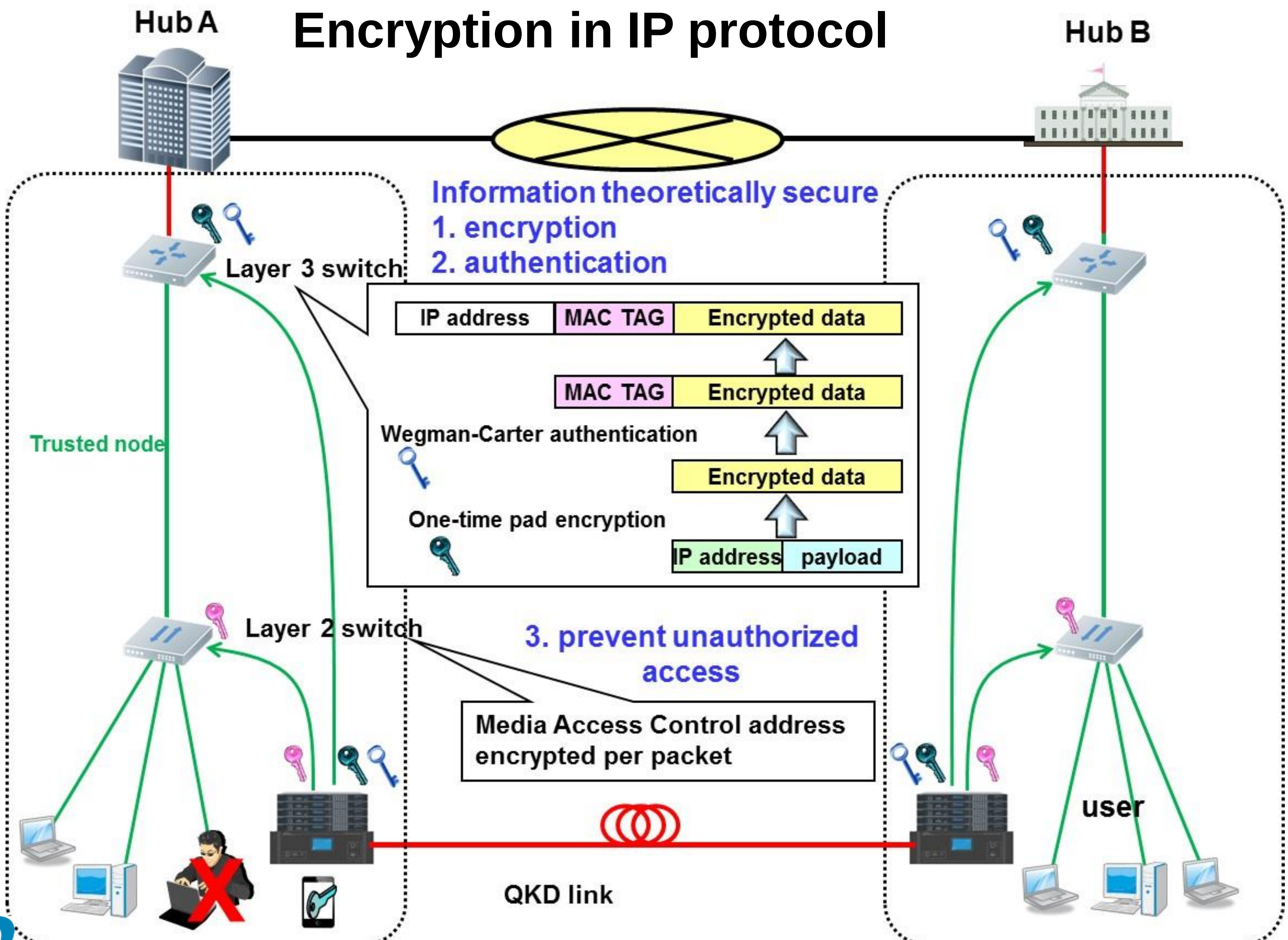
Shamir's secret sharing scheme itself cannot realize integrity.

➡ Message authentication code (MAC) embedded (linked) in secret data;
The data owner can check integrity himself/herself at data re-construction phase.

Security of channels for data-transmission is just assumed.

➡ Networked QKD link + one time pad encryption scheme can provide the information theoretical security in transmission.

Information theoretically secure communication



M. F. et al., *Int. J. Network security* 17, 34-39 (2015).

QKD

1. It works in a point-to-point link, not in a multi-party link.
2. Speed and distance of a direct link are limited.

1M bits/s at 50km (TV conference data) [1,2]

~10k bits/s at 100km (Voice data) [3]

(for standard optical fiber with loss rate of 0.2dB/km)

(1) K. Yoshino et. al., *Opt. Express* 21, 31395-31401 (2013).
(2) J. F. Dynes et al., *Opt. Express* 20, 16339-16347 (2012).
(3) K. Shimizu et al,. *IEEE J. Lightwave tech.* 32, 141-151 (2013).

NICT

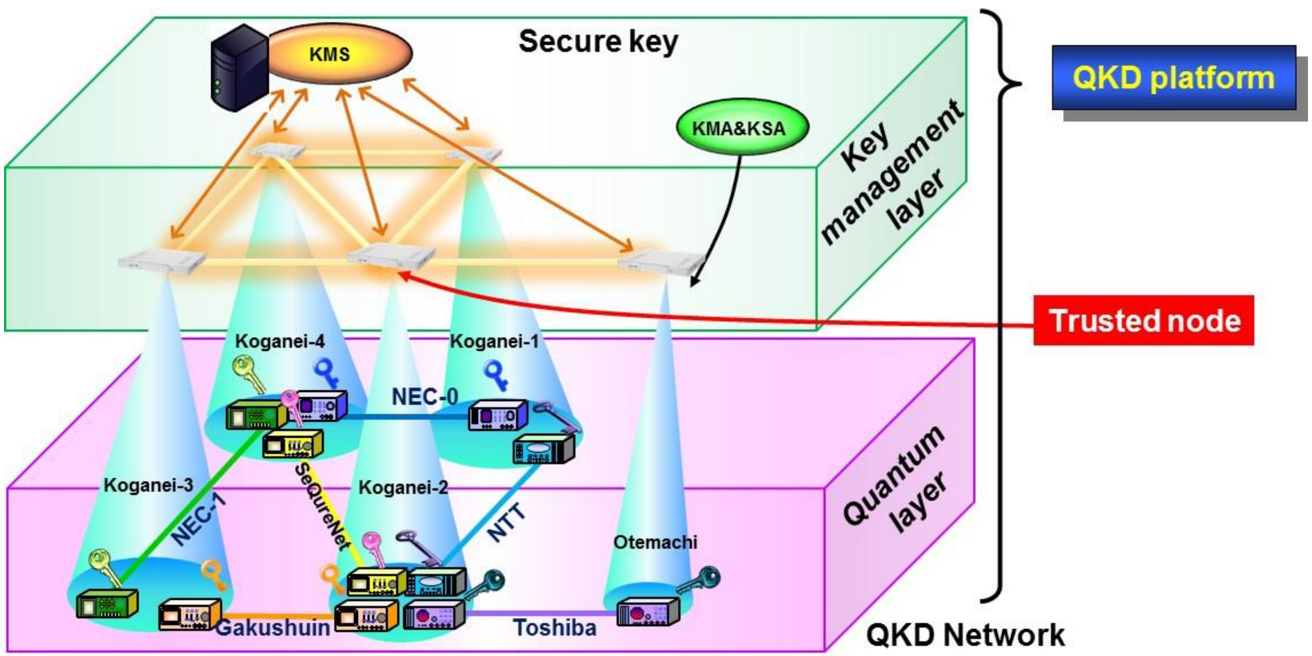
QKD Network

1. Networking is made by introducing the trusted nodes, and by relaying a key via the nodes.

2. Rerouting function must be installed.

Appropriate redundancy

M. Sasaki et al., *J. Selected Topics in Quant. Elec.*, 21, 6400313 (2015).



Contents

1. Our scheme
    - 1-1 Secret sharing scheme with QKD Network
    - 1-2 Information theoretically secure authentication with a single password
2. Experiment of password-authenticated secret sharing scheme
    - 2-1 Tokyo QKD Network
    - 2-2 Performance of password authenticated secret sharing scheme
3. Summary and perspectives

NICT

Single-password SS authentication

Authentication (identification) methods are classified into three types;

1. Something you know
Password authentication
➡ **Security is computational one.**

2. Something you have
IC cards, token devices
➡ **Risk of duplicating data is unavoidable.**

3. Something you are
Biometrics information
➡ **Risk of duplicating data is unavoidable.**
➡ **One cannot change his/her biometrics information.**

NICT

Single-password SS authentication

Password has been used in many cases because it is simple, convenient and with low cost implementation. 80% on-line certifications are combination of ID numbers and passwords. (2013)
(http://internet.watch.impress.co.jp/docs/news/621665.html)

When owner communicates with shareholders, using many passwords leads to human error in security aspects.
(using simple passwords, memorizing them on the paper)

We desire authentication method using a single password with information theoretic security.

We make shares of a single password, and store them in multiple shareholders.

Fujiwara, Waseda, Nojima, Moriai, Ogata and Sasaki, Scientific Reports, 6:28988 (2016). On-line

Single-password SS authentication

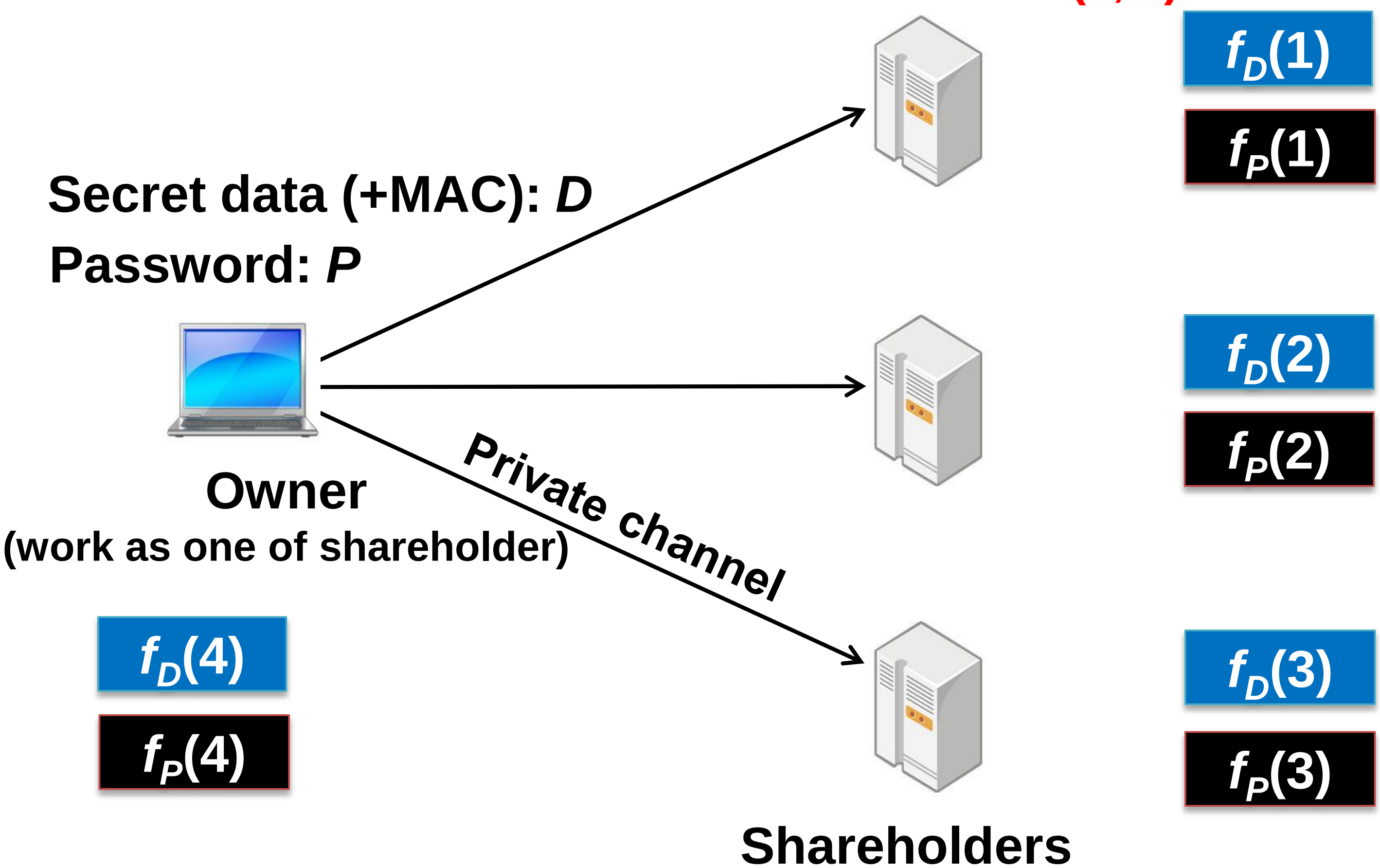
Registration of data and password sharing phase

(1) Owner creates and sends shares of *D* and *P* by using

2nd order polynomial $f_D(x) = D + a_D^{(1)} x + a_D^{(2)} x^2$

1st order polynomial $f_P(x) = P + a_P^{(1)} x$

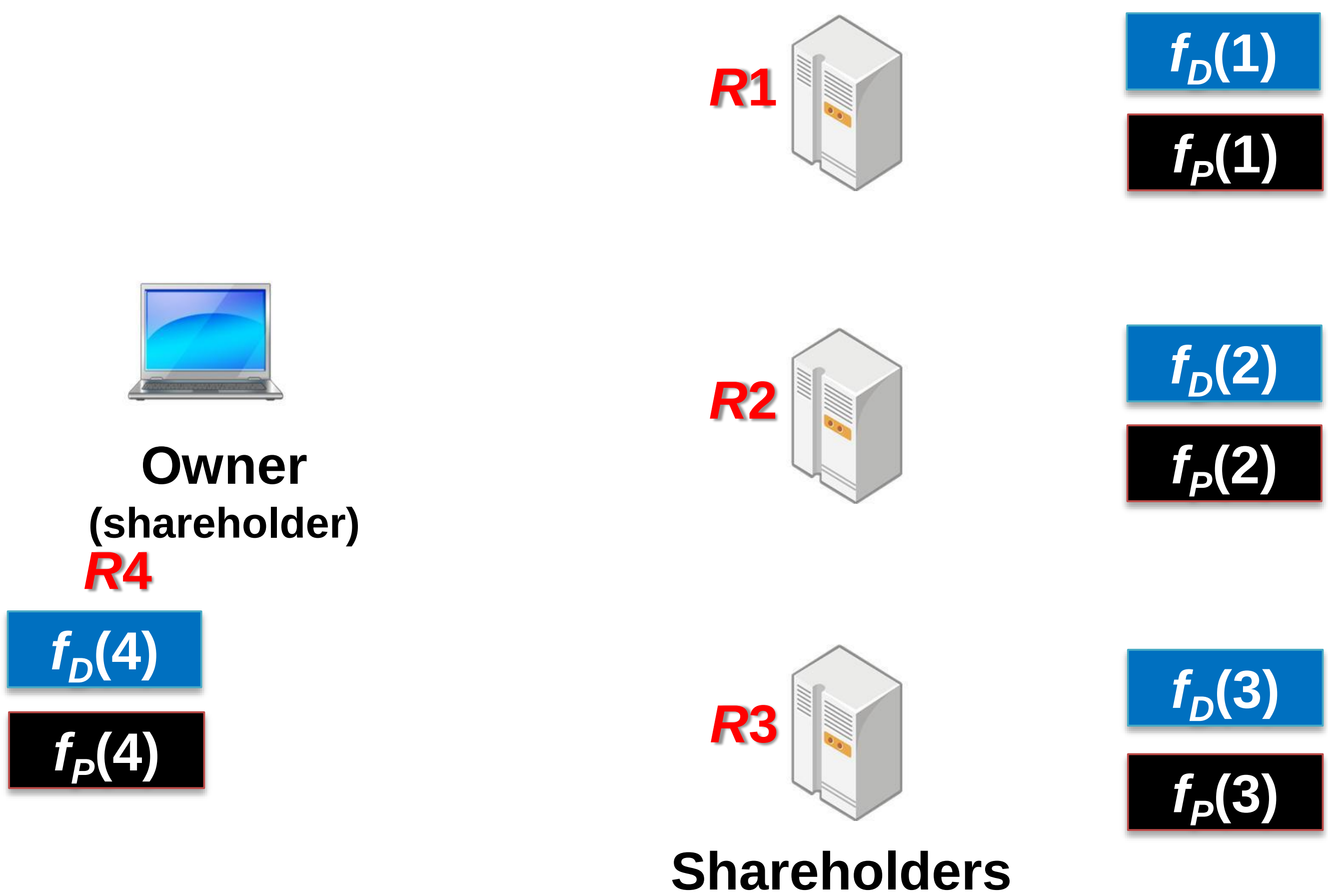
(3, 4) threshold for secret data



Single-password SS authentication

Pre-computation and communication among servers phase

(2) Each shareholder generates a random number *Rj*

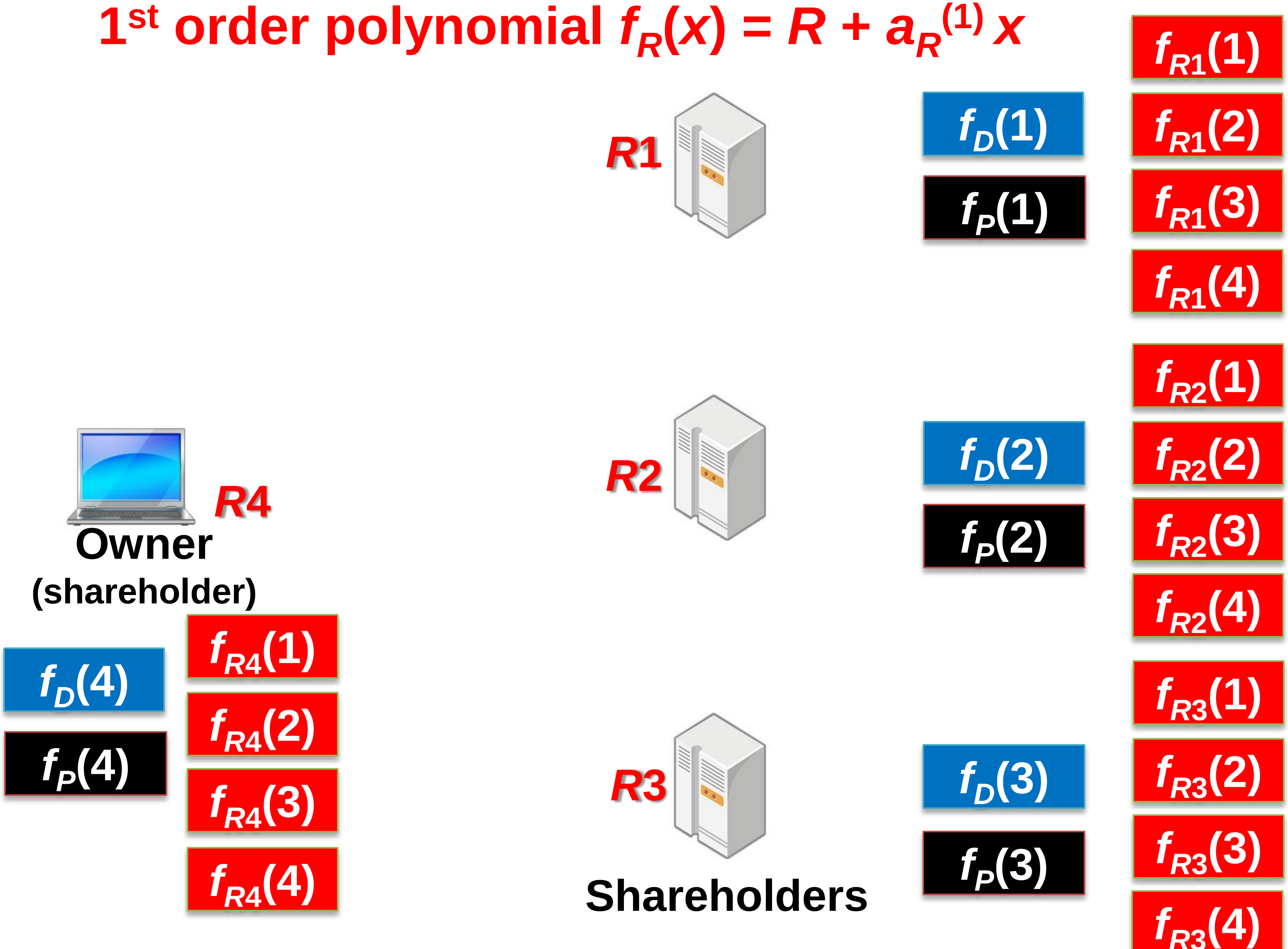


Single-password SS authentication

Pre-computation and communication among servers phase

(3) Each shareholder makes shares of *Rj* by using

1st order polynomial $f_R(x) = R + a_R^{(1)} x$



Single-password SS authentication

Pre-computation and communication among servers phase

(4) Each shareholder generates shares of "0" by using 2nd order polynomial $f_0(x) = a_0^{(1)} x + a_0^{(2)} x^2$ such that $f_{0j}(0) = 0$ (constant term=0).

To mask secret data shares $f_D(j)$ in the re-construction phase.

$f_D(1)$ $f_P(1)$

$f_{R1}(1)$ $f_{01}(1)$
$f_{R1}(2)$ $f_{01}(2)$
$f_{R1}(3)$ $f_{01}(3)$
$f_{R1}(4)$ $f_{01}(4)$

Owner (shareholder)

$f_D(4)$ $f_P(4)$

$f_{R4}(1)$ $f_{04}(1)$
$f_{R4}(2)$ $f_{04}(2)$
$f_{R4}(3)$ $f_{04}(3)$
$f_{R4}(4)$ $f_{04}(4)$

$f_D(2)$ $f_P(2)$

$f_{R2}(1)$ $f_{02}(1)$
$f_{R2}(2)$ $f_{02}(2)$
$f_{R2}(3)$ $f_{02}(3)$
$f_{R2}(4)$ $f_{02}(4)$

$f_D(3)$ $f_P(3)$

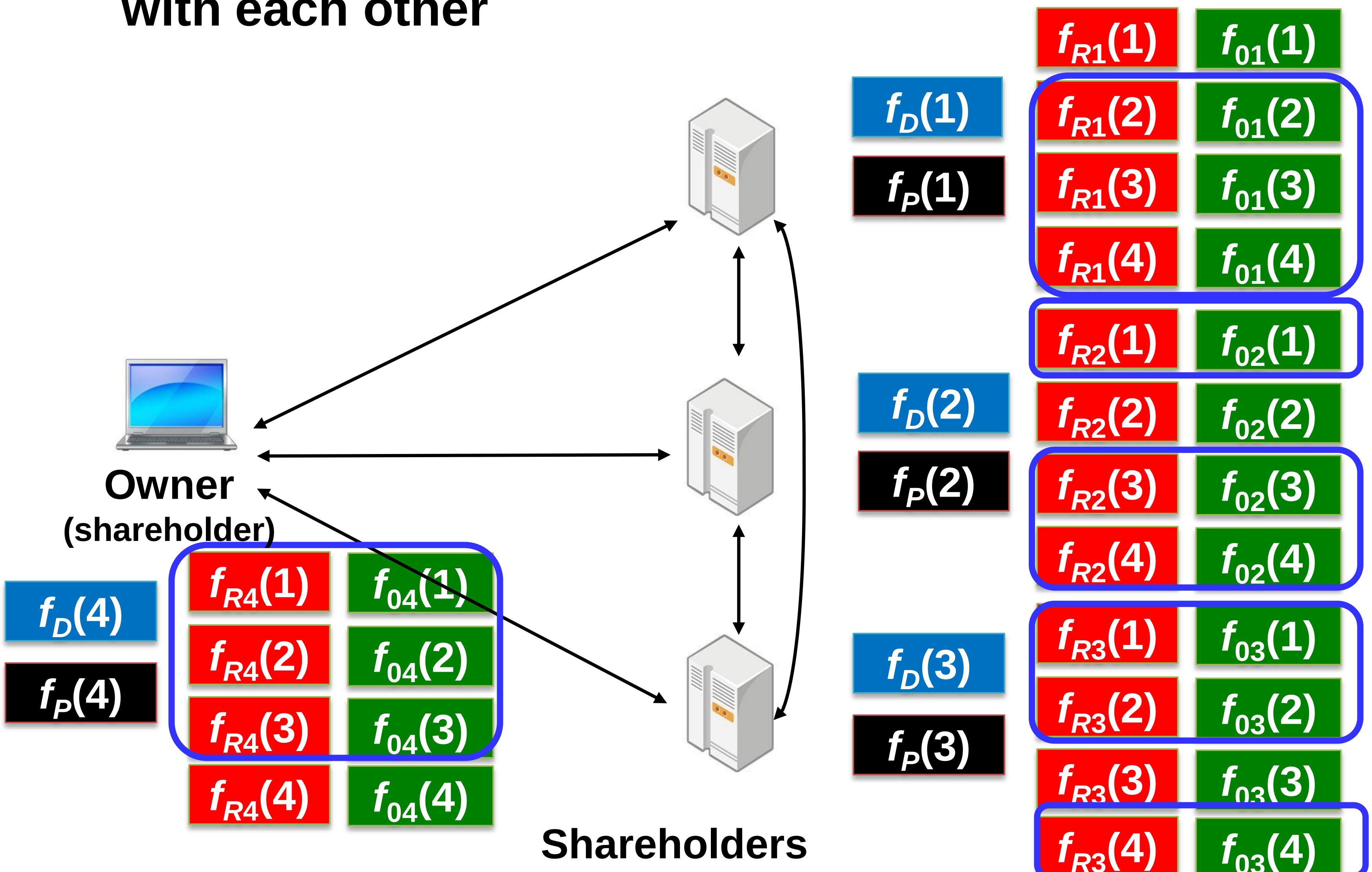
$f_{R3}(1)$ $f_{03}(1)$
$f_{R3}(2)$ $f_{03}(2)$
$f_{R3}(3)$ $f_{03}(3)$
$f_{R3}(4)$ $f_{03}(4)$

Shareholders

Single-password SS authentication

Pre-computation and communication among servers phase

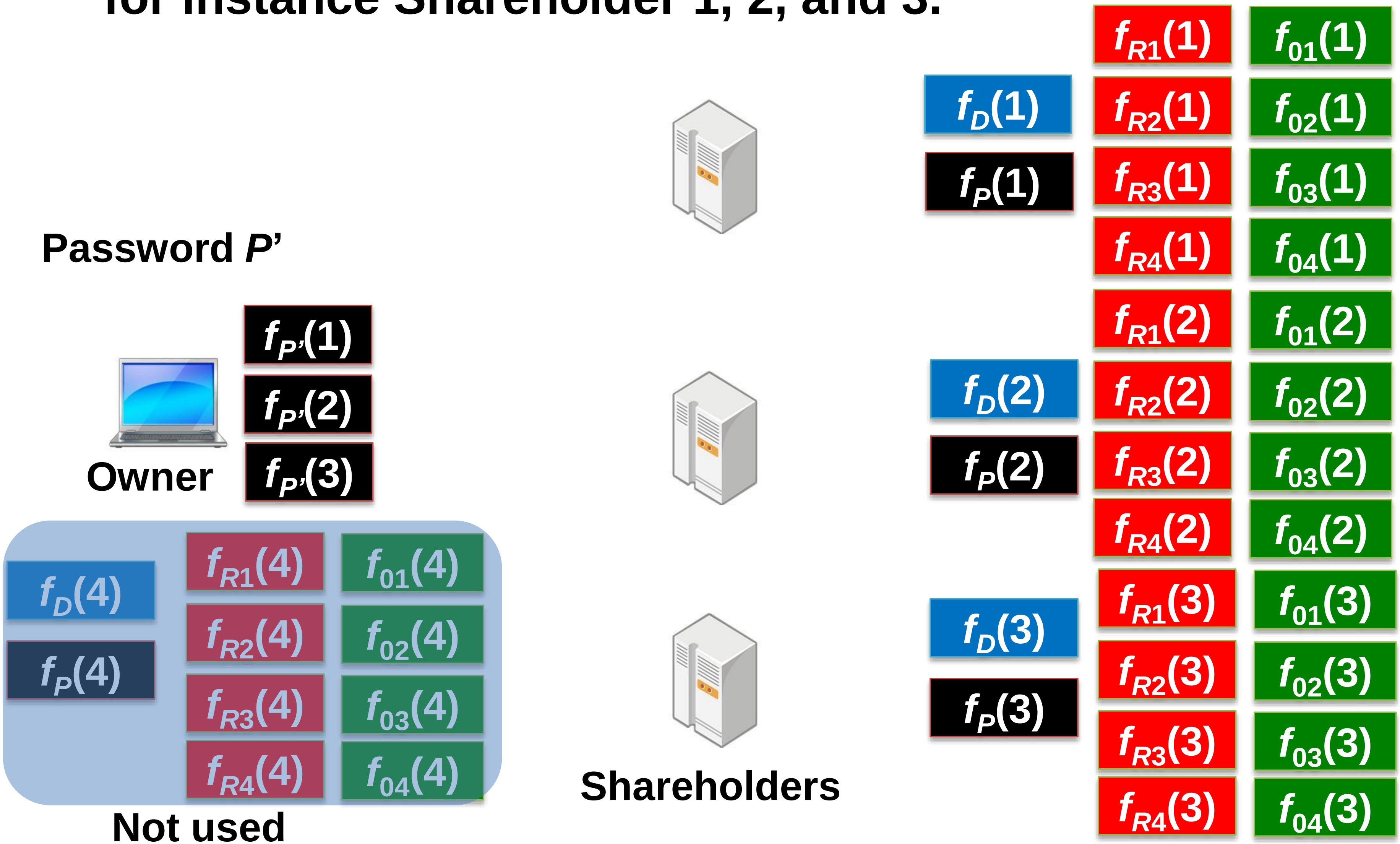
(5) Shareholders exchange shares of *Rj* and “0” with each other



Single-password SS authentication

Data re-construction phase

(6) Owner selects three shareholders, for instance Shareholder 1, 2, and 3.



Single-password SS authentication

Data re-construction phase

(7) Owner remembers the password, which is P', and generates shares of P' by using 1^{st} order polynomial $f_{P'}(x) = P' + a_{P'}^{(1)}x$.

$f_{R1}(1)$ $f_{01}(1)$

$f_D(1)$ $f_{R2}(1)$ $f_{02}(1)$

$f_P(1)$ $f_{R3}(1)$ $f_{03}(1)$

$f_{R4}(1)$ $f_{04}(1)$

Password P'

$f_{P'}(1)$

$f_{P'}(2)$

Owner $f_{P'}(3)$

$f_{R1}(2)$ $f_{01}(2)$

$f_D(2)$ $f_{R2}(2)$ $f_{02}(2)$

$f_P(2)$ $f_{R3}(2)$ $f_{03}(2)$

$f_{R4}(2)$ $f_{04}(2)$

$f_{R1}(3)$ $f_{01}(3)$

$f_D(3)$ $f_{R2}(3)$ $f_{02}(3)$

$f_P(3)$ $f_{R3}(3)$ $f_{03}(3)$

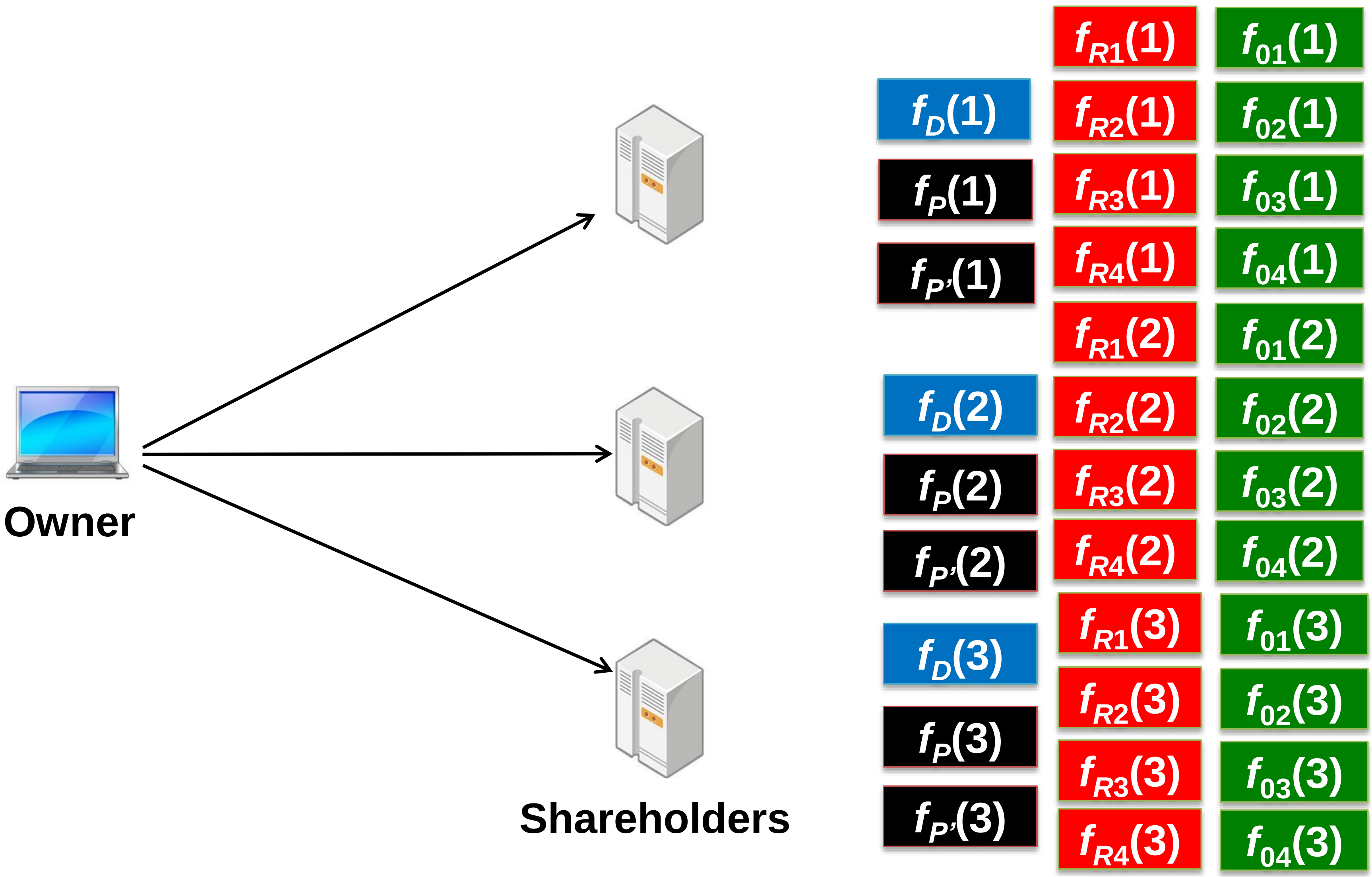
$f_{R4}(3)$ $f_{04}(3)$

Shareholders

Single-password SS authentication

Data reconstruction phase

(8) Owner sends the password shares to the shareholders.



Single-password SS authentication

Data re-construction phase

(9) The shareholders compute the three quantities, R_j, Z_j, and F_j.

Owner

Shareholders

$R_1=f_{R1}(1)+f_{R2}(1)+f_{R3}(1)$

$Z_1=f_{01}(1)+f_{02}(1)+f_{03}(1)$

$F_1 = \left[f_P(1)-f_{P'}(1)\right]R_1+ Z_1+ f_D(1)$

$R_2=f_{R1}(2)+f_{R2}(2)+f_{R3}(2)$

$Z_2=f_{01}(2)+f_{02}(2)+f_{03}(2)$

$F_2 = \left[f_P(2)-f_{P'}(2)\right]R_2+ Z_2+ f_D(2)$

$R_3=f_{R1}(3)+f_{R2}(3)+f_{R3}(3)$

$Z_3=f_{01}(3)+f_{02}(3)+f_{03}(3)$

$F_3 = \left[f_P(3)-f_{P'}(3)\right]R_3+ Z_3+ f_D(3)$

Single-password SS authentication

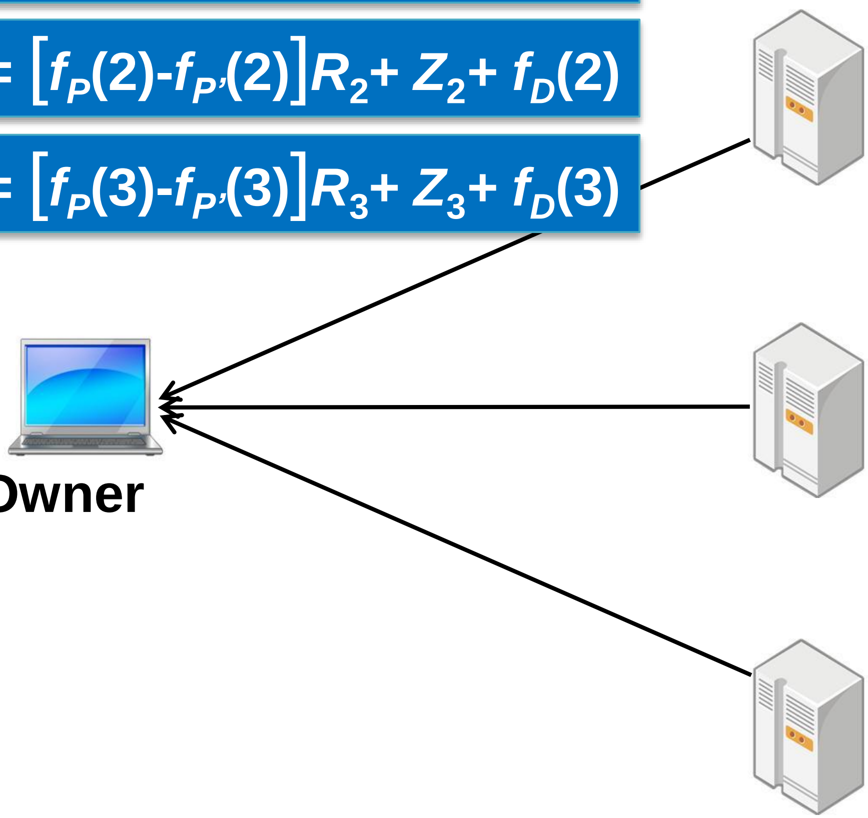
Data re-construction phase

(10) Shares F_1, F_2 and F_3 are sent back to the owner.

$F_1 = [f_P(1)\text{-}f_{P'}(1)]R_1 + Z_1 + f_D(1)$

$F_2 = [f_P(2)\text{-}f_{P'}(2)]R_2 + Z_2 + f_D(2)$

$F_3 = [f_P(3)\text{-}f_{P'}(3)]R_3 + Z_3 + f_D(3)$



$R_1 = f_{R1}(1) + f_{R2}(1) + f_{R3}(1)$

$Z_1 = f_{01}(1) + f_{02}(1) + f_{03}(1)$

$F_1 = [f_P(1)\text{-}f_{P'}(1)]R_1 + Z_1 + f_D(1)$

$R_2 = f_{R1}(2) + f_{R2}(2) + f_{R3}(2)$

$Z_2 = f_{01}(2) + f_{02}(2) + f_{03}(2)$

$F_2 = [f_P(2)\text{-}f_{P'}(2)]R_2 + Z_2 + f_D(2)$

$R_3 = f_{R1}(3) + f_{R2}(3) + f_{R3}(3)$

$Z_3 = f_{01}(3) + f_{02}(3) + f_{03}(3)$

$F_3 = [f_P(3)\text{-}f_{P'}(3)]R_3 + Z_3 + f_D(3)$

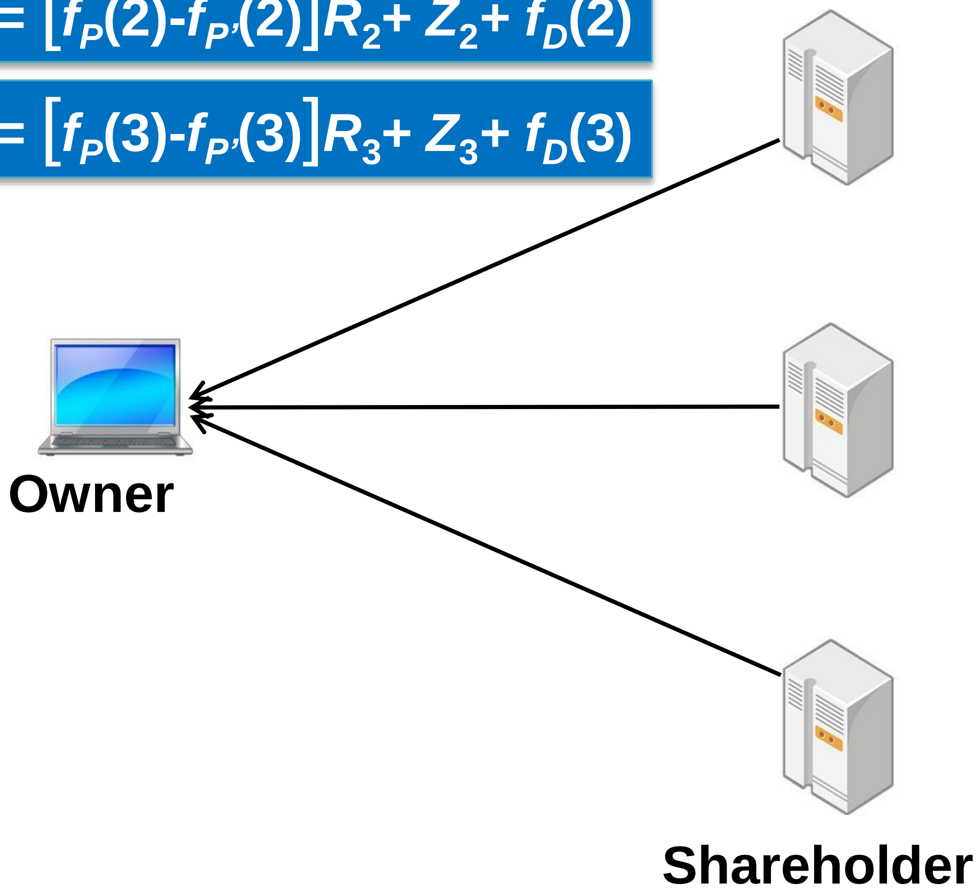
Single-password SS authentication

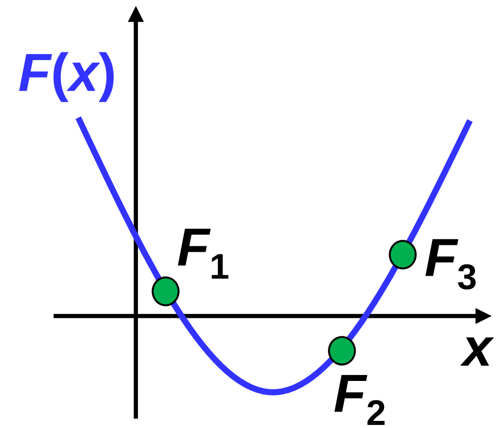
(11) The owner finds a polynomial $F(x)$ with F_1, F_2 and F_3 by interpolation.

$$F_1 = [f_P(1)-f_{P'}(1)]R_1 + Z_1 + f_D(1)$$

$$F_2 = [f_P(2)-f_{P'}(2)]R_2 + Z_2 + f_D(2)$$

$$F_3 = [f_P(3)-f_{P'}(3)]R_3 + Z_3 + f_D(3)$$





Single-password SS authentication

Data reconstruction phase

(12) If the password is wrong, $P' \neq P$, then shares $f_D(1)$, $f_D(2)$ and $f_D(3)$ are masked by R_1, R_2, R_3, Z_1, Z_2 and Z_3. Therefore no information on D is leaked.

$$F_1 = \left[f_P(1)-f_{P'}(1)\right]R_1 + Z_1 + f_D(1)$$

$$F_2 = \left[f_P(2)-f_{P'}(2)\right]R_2 + Z_2 + f_D(2)$$

$$F_3 = \left[f_P(3)-f_{P'}(3)\right]R_3 + Z_3 + f_D(3)$$

Owner

Single-password SS authentication

Data re-construction phase

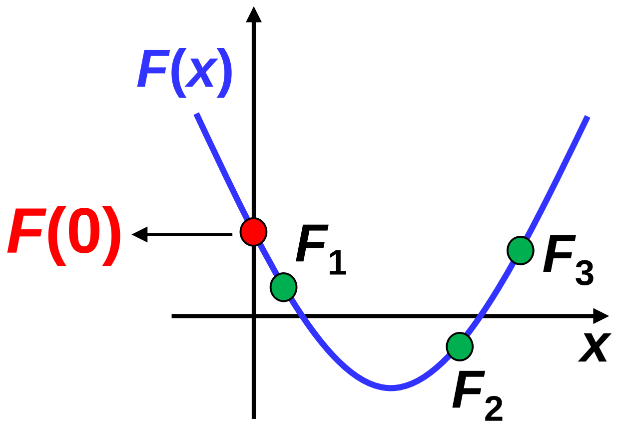
(13) If the password is correct, $P'=P$, then

$$F_j = [f_P(j)-f_{P'}(j)]R_j + Z_j + f_D(j)$$

$[f_P(j)-f_{P'}(j)]R_j$ terms canceled

Z_j terms → "0"

Owner



The owner re-constructs the original data as

$$F(0)=f_D(0)=D \text{ (secret data +MAC)}$$

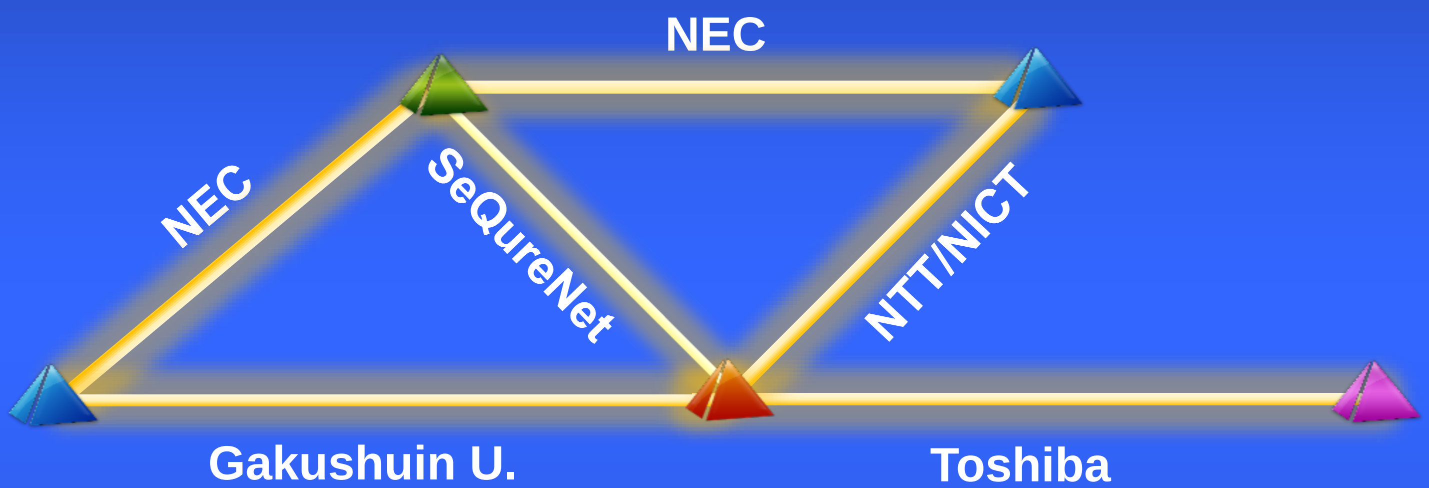
owner calculates MAC and checks integrity.

Contents

- QKD link distance is a metropolitan scale
- Networking is made by trusted nodes

Tokyo QKD Network

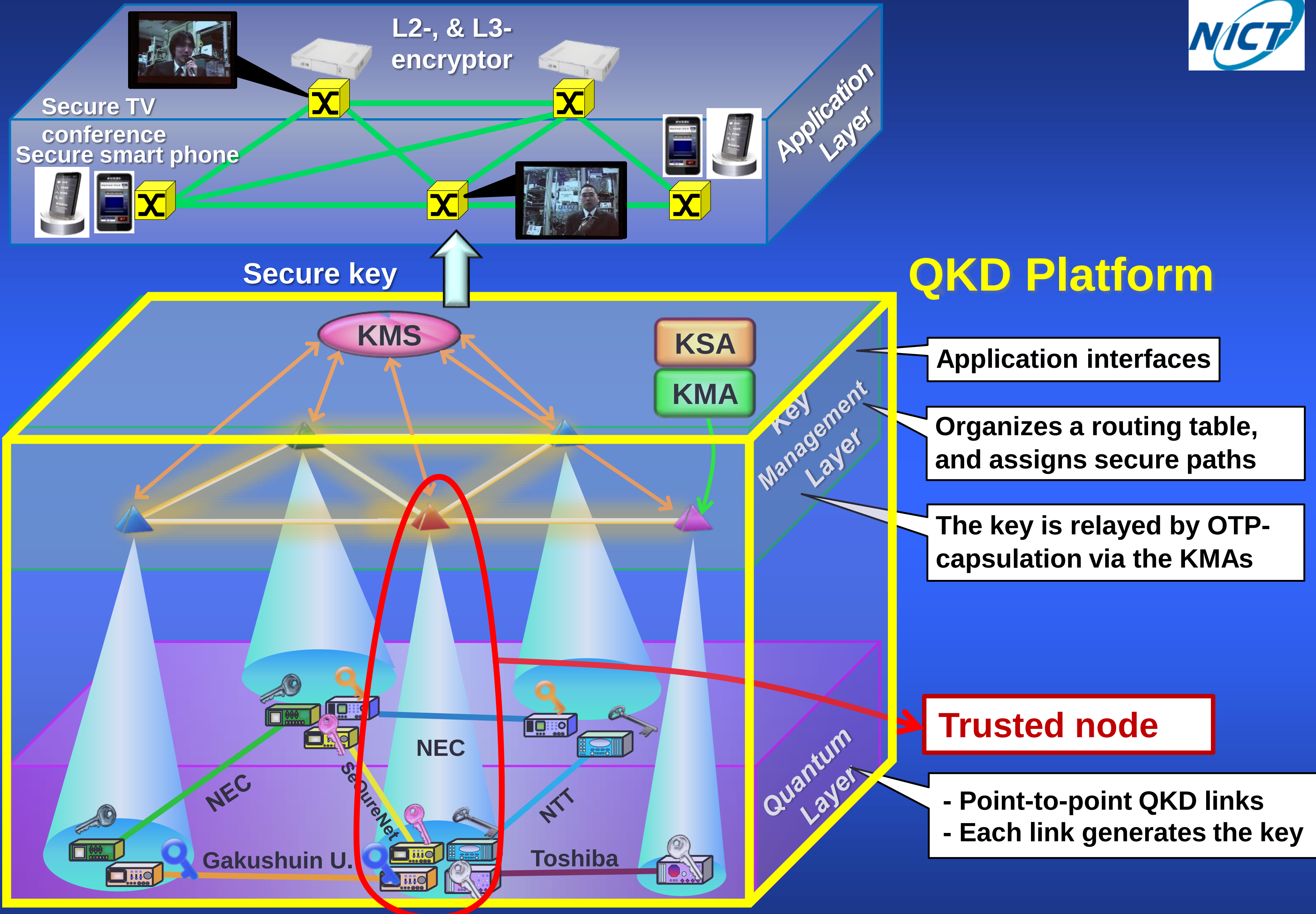
since 2010



(1) BB84 : NEC and Toshiba

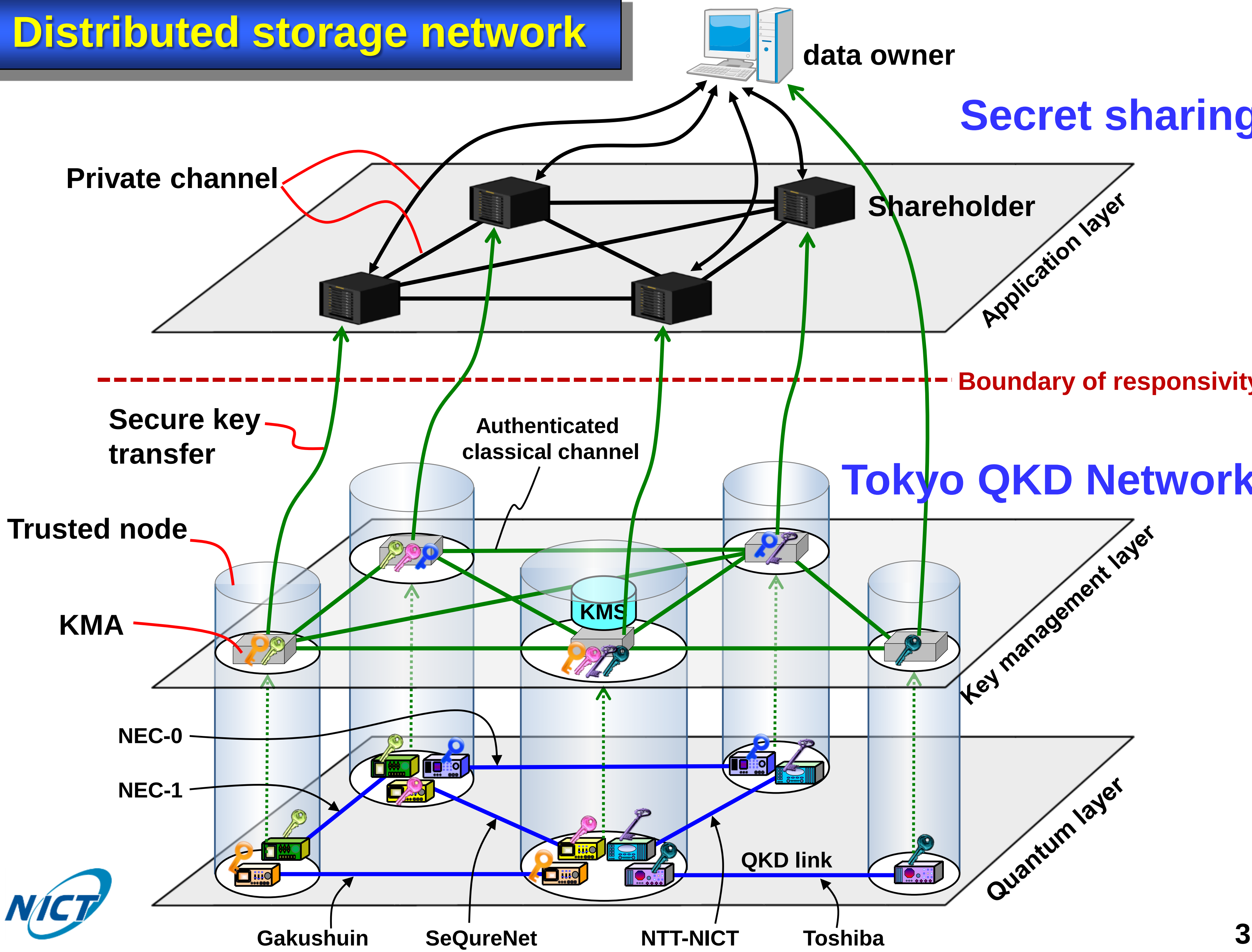
(2) Continuous variable-QKD : Gakushuin U. and SeQureNet

(3) DPS-QKD : NTT/NICT



L2-, & L3-
encryptor
Secure TV
conference
Secure smart phone
Application
Layer
Secure key
QKD Platform
KMS
KSA
KMA
Key
Management
Layer
Application interfaces
Organizes a routing table, and assigns secure paths
The key is relayed by OTP-capsulation via the KMAs
NEC
NEC
SeQureNet
NTT
Gakushuin U.
Toshiba
Quantum
Layer
Trusted node
- Point-to-point QKD links
- Each link generates the key

Distributed storage network



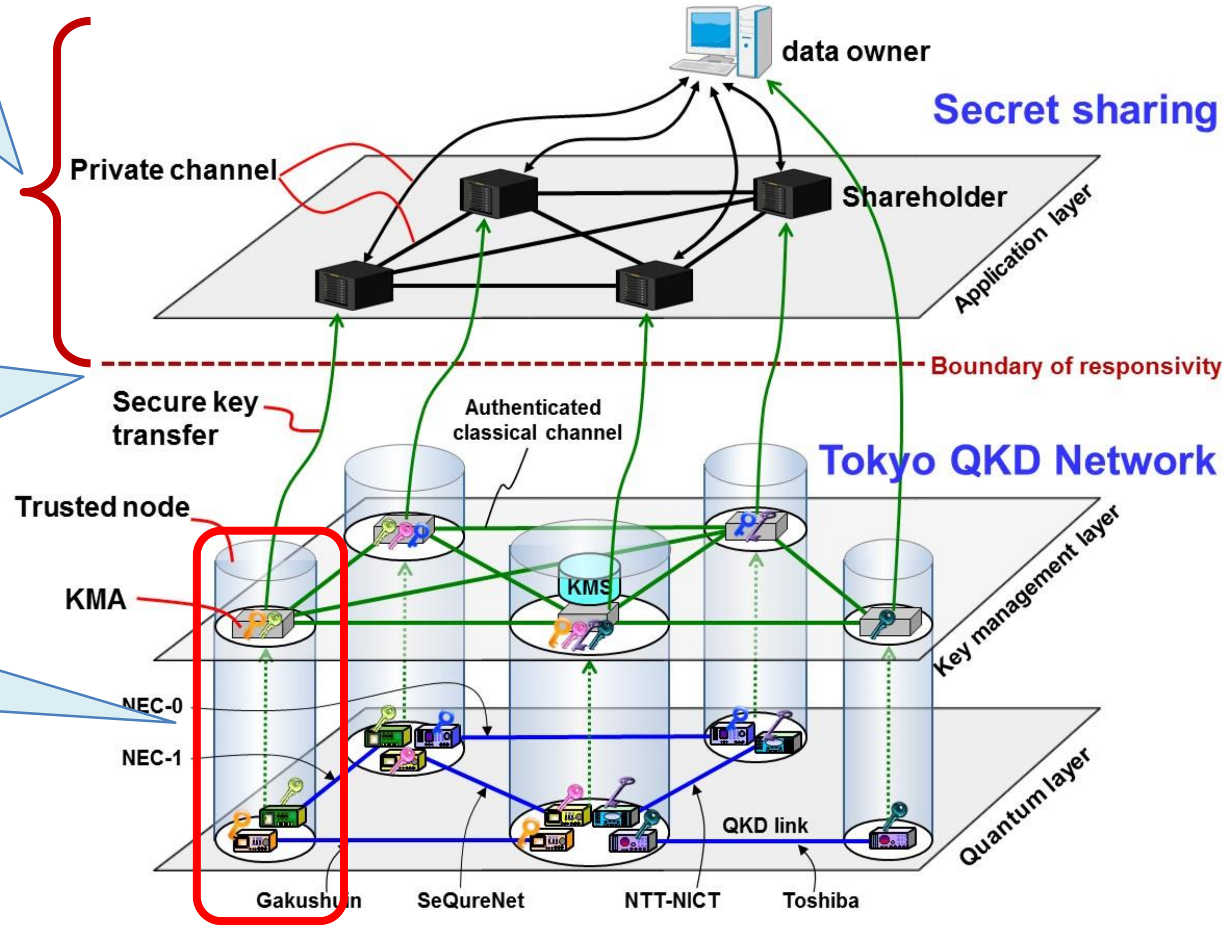
NICT

Assumed operation condition 1/3

The data owner and the shareholders are set in another admission managed area

Access rights to the QKD platform and the data owner/ the shareholders are completely separated

Need to be protected at the expense of necessary costs

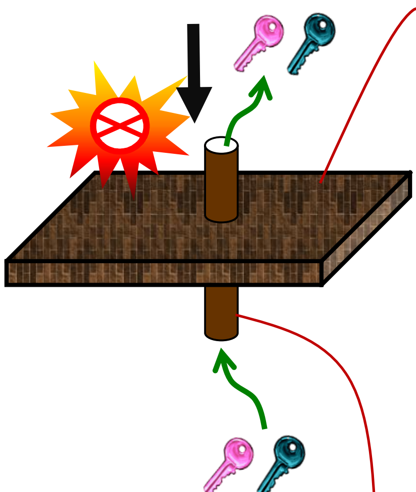


NICT

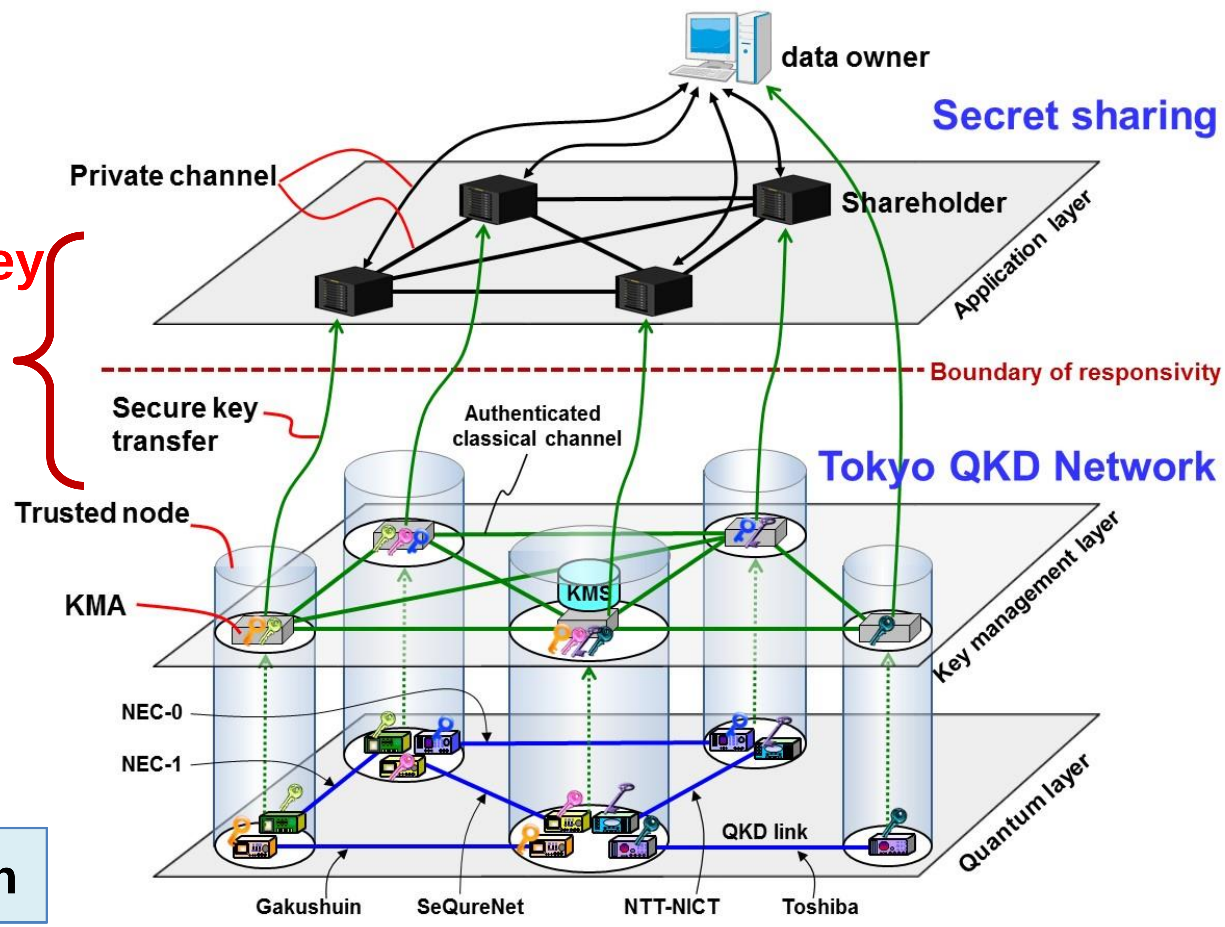
Assumed operation condition 2/3

Malicious attacker on the shareholder cannot crack the QKD platform

One-way firewall



Secure key transfer



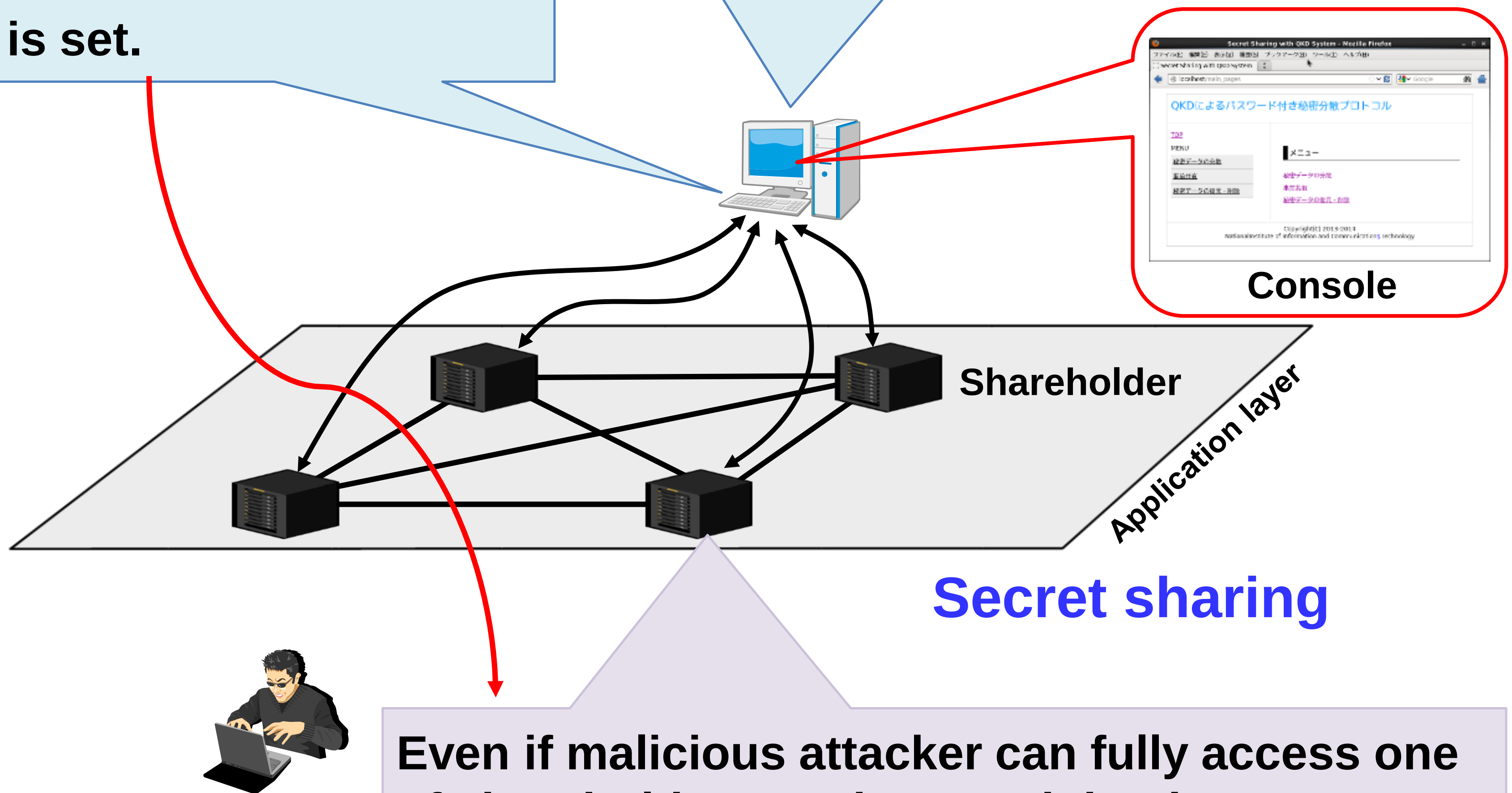
Tamper resistant metal cable of short distance

User authentication

Assumed operation condition 3/3

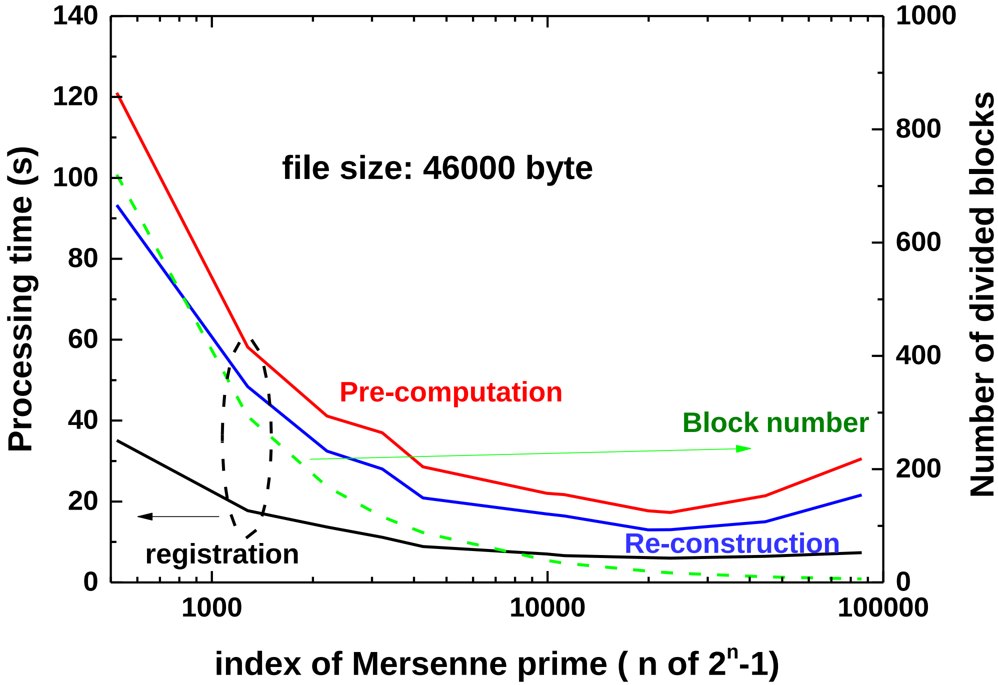
(3,4) threshold for secret data is set.

Data owner server works as a shareholder.



Even if malicious attacker can fully access one of shareholders, and pretend the data owner, this scheme can work.

Performance of our system



Processing time as functions of index of Mersenne prime for 46 kbyte data size.

All calculations are made in a finite Galois field with prime order *q*. Mersenne primes have suitable form $q=2^m-1$ for calculations.

The best performance can be found in the range of *q* with11213≤ m ≤ 23209.

Performance of our system

The total quantity of keys required to store and retrieve is about 30 times as large as the original secret data.

Performance depends on the size of *q*.
This is because (1) the computational time of the shares increases roughly in the square of bit length of *q* and (2) using a smaller prime *q* increases the number of blocks *l*, and hence a longer processing time is required for dividing/managing the blocks and sending IP packets.

current IP packet

IP address	MAC TAG	Encrypted data (1 block)

To improve the performance:

1. **Trucking more blocks in one IP packet for small *q***
2. **Parallel processing of IP packets**
3. **Simplifying key sorting and synchronization**

We expect to decrease processing time one-tenth.

Summary and perspectives

Proof-of-principle demonstration of Information theoretically secure distributed storage system

Confidentiality

- **Secret sharing + QKD Network**
- **Password secret sharing authentication**

Integrity

- **Using MAC in IP packet at transmission**
- **Embedding MAC in secret data in re-construction phase**

Future works

- **Improvement of QKD links and storage system**
- **Implementation of proactive secret sharing**
- **Data relay demonstration**

NICT

Thank you for your attention