東京理科大学
TOKYO UNIVERSITY OF SCIENCE

The 12th Workshop among Asian Information Security Labs (WAIS 2020)

Computationally Secure Verifiable Secret Sharing Scheme

Imai Takato[1] **and Iwamura Keiichi**[1]

[1]Department of Electrical Engineering
Tokyo University of Science
Tokyo, JAPAN

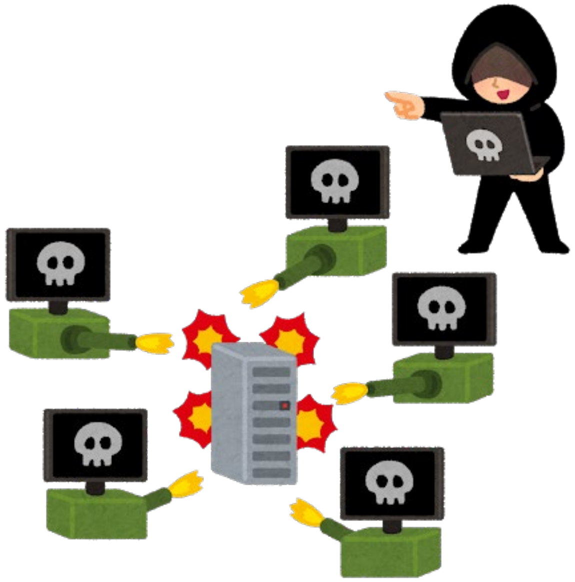
2020/2/21

Contents

1. Background
2. Secret Sharing Scheme
3. Proposed Scheme
4. Security of Proposed Scheme

Background①

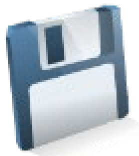
Information Leakage



Cloud

User

Personal Information



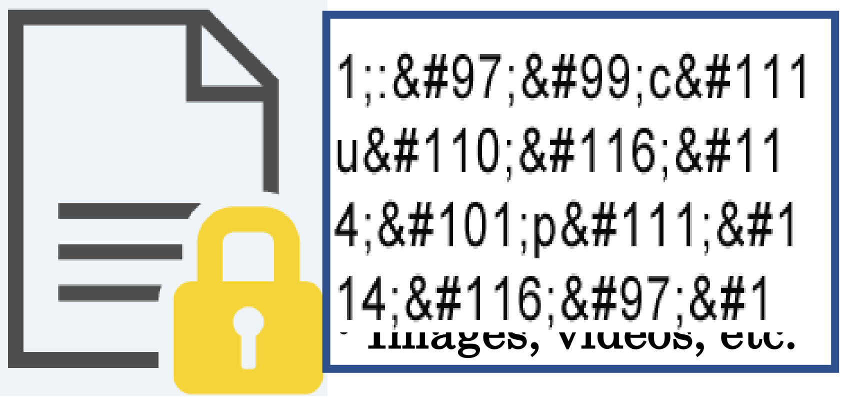
- Purchase history
- Credit card number
- App backup
- Images, videos, etc. …

Background①

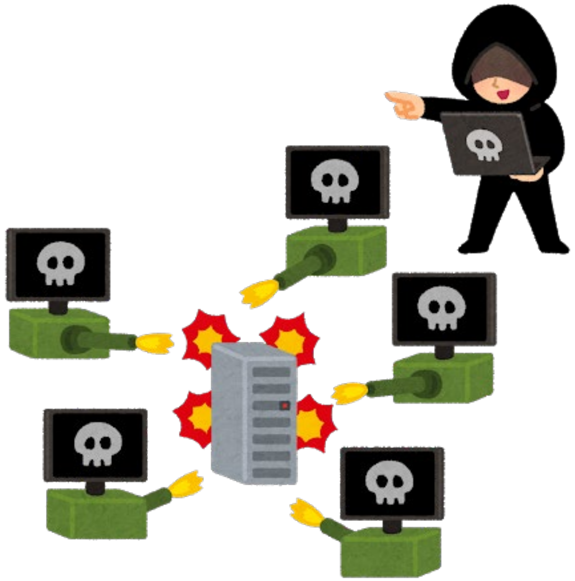
Cloud

User

Encrypted Data
Fragmented Data



Encryption



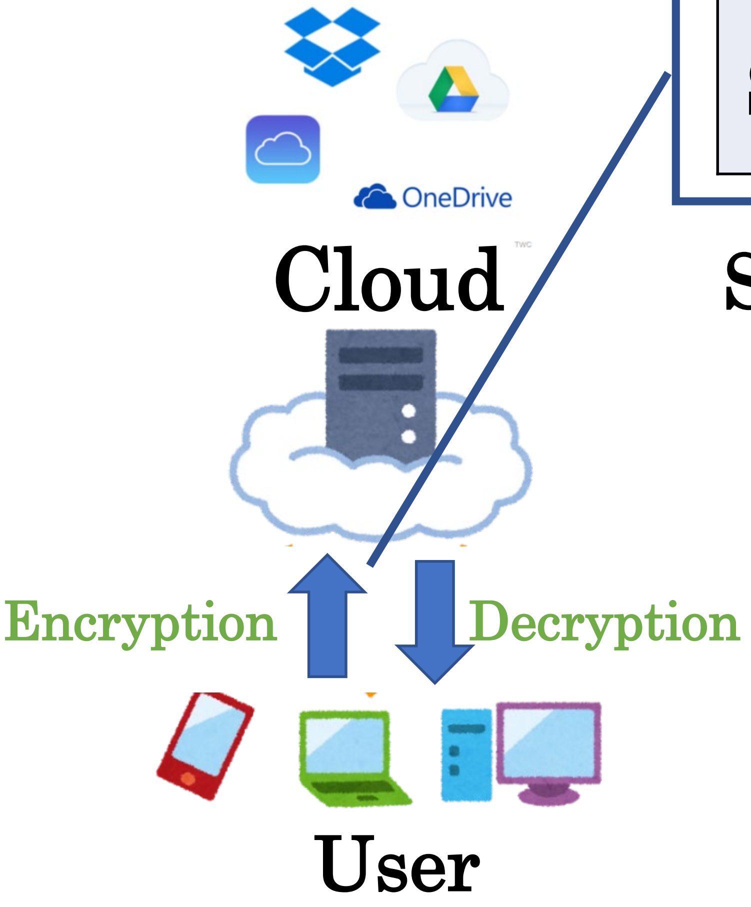
Background②

Data Modification

Cloud

Validation of Reconstructed Data

Purpose



Cloud

Encryption

Decryption

User

Encryption / Decryption technology between User and Cloud

Secret Sharing Scheme(SSS)	Homomorphic Encryption

SSS ＋ Encryption using Key

= **Asymmetric Secret Sharing Scheme (A-SSS)** [1]

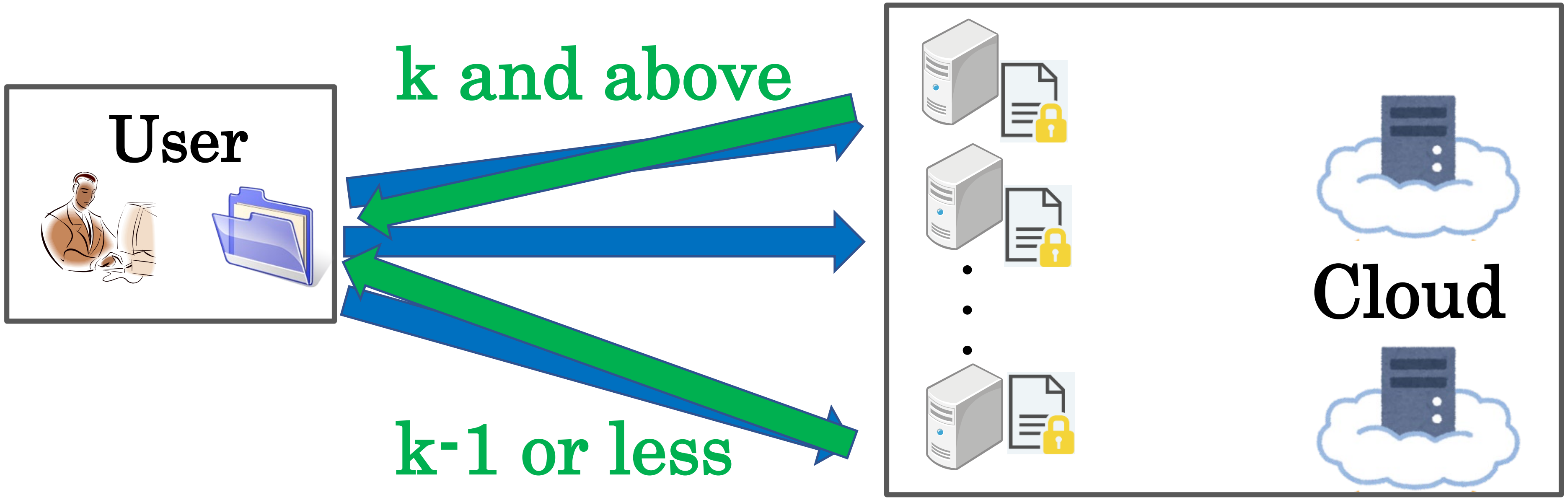
- Cheat Detect(Verify recnstructed Secret)
- Identify Dishonest Server

Dishonest Server：Server that outputs forged share

[1] Satoshi Takahashi, Hyunho Kang, Keiichi Iwamura, Asymmetric secret sharing scheme suitable for cloud systems, 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)

Secret Sharing Scheme (SSS)

- Shamir's k out of n Secret Sharing Scheme [2]
- Secret data is converted into n number of different values (shares) and distributed to n number of servers to be stored.

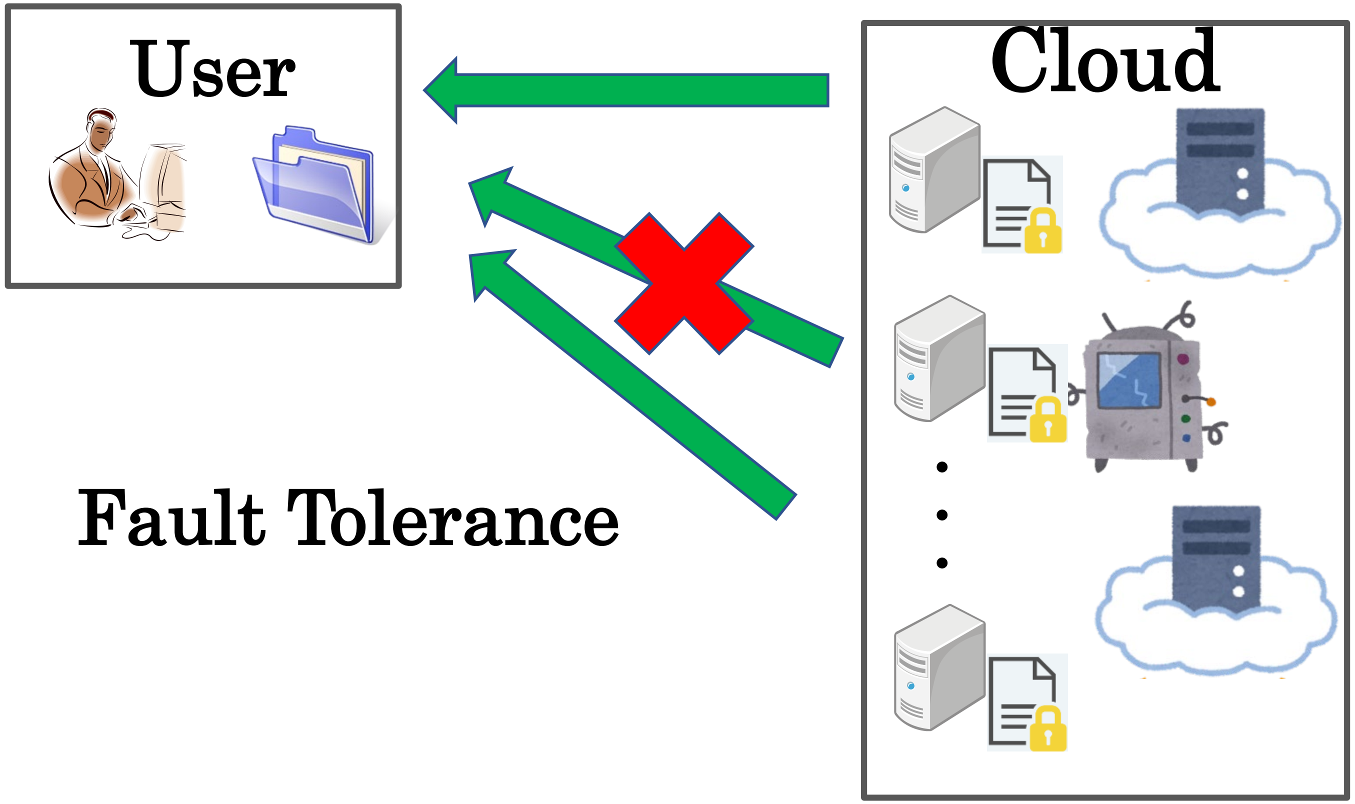


[2]A.Shamir . How to Share a Secret.Communications of the ACM,vol,22,no.11,pp.612-613,1979

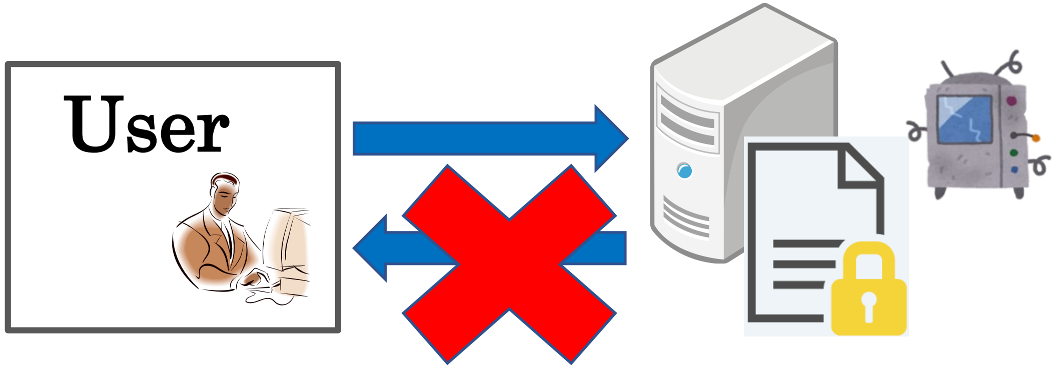
Secret Sharing Scheme(SSS)

Shamir's k out of n SSS[2]

- Convert one data into *n* pieces(shares),Distribute *n* shares, Collect k shares and Reconstruct



Fault Tolerance



No Fault Tolerance

- Multiple servers (At least 2 servers)

→Suitable for the Cloud.

[2]A.Shamir . How to Share a Secret.Communications of the ACM,vol,22,no.11,pp.612-613,1979

Distribution Process of SSS $(n = 3, k = 2)$

Secret is converted into **three shares**

x_i : ServerID(Public)

y_i : Share(Secrecy)

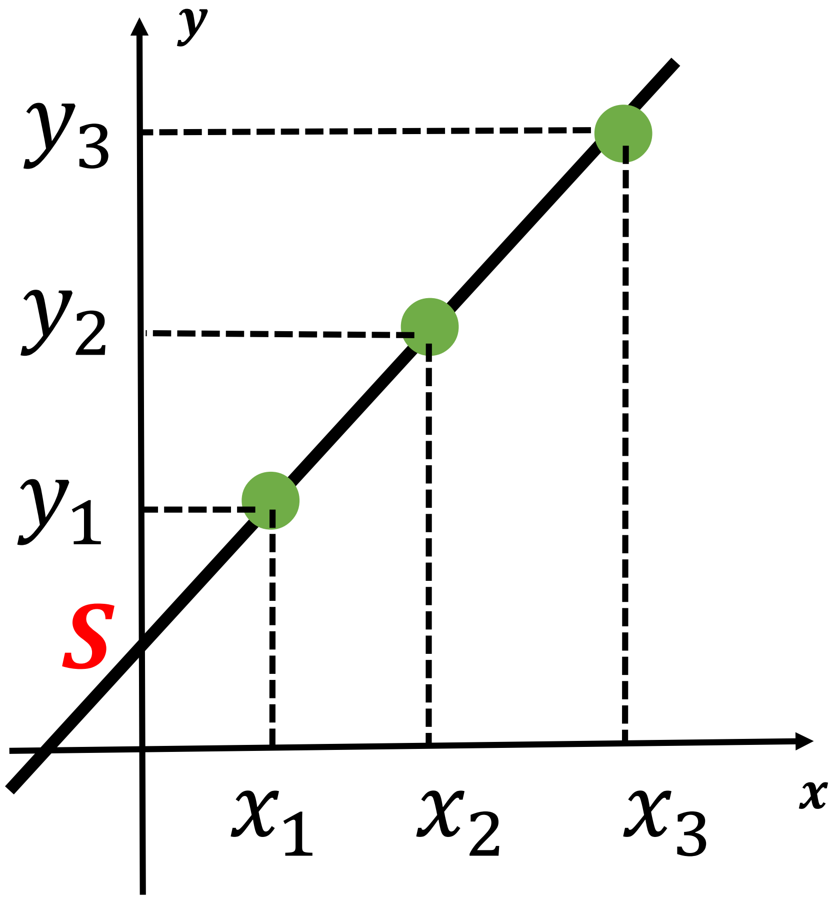
User

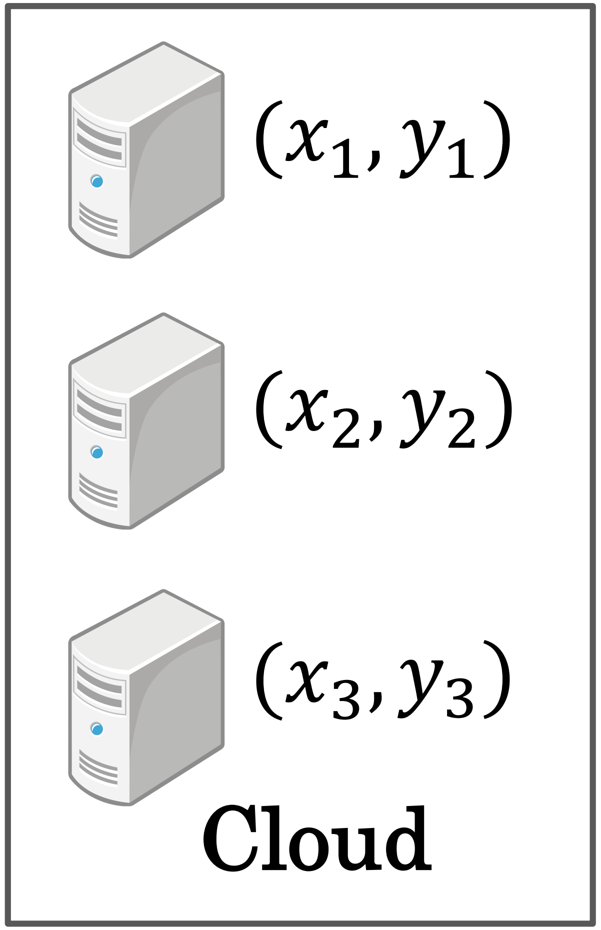
Secret S

k-1Polynomial through S

↓

Linear Polynomial

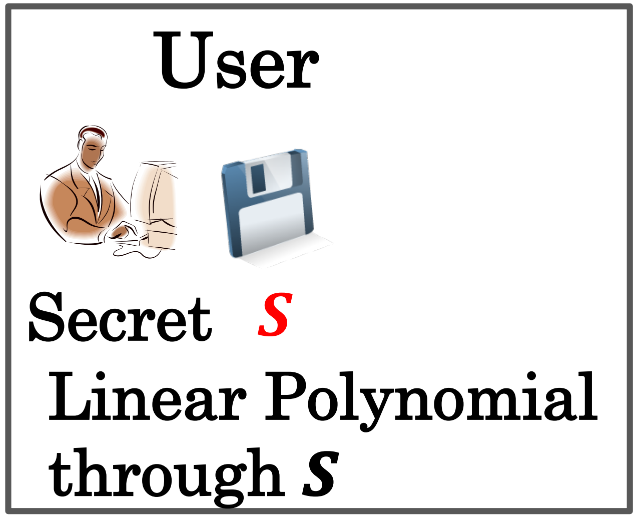




Reconstruction Process of SSS$(n = 3, k = 2)$

User collects any two (x_i, y_i)
and reconstruct secret s.

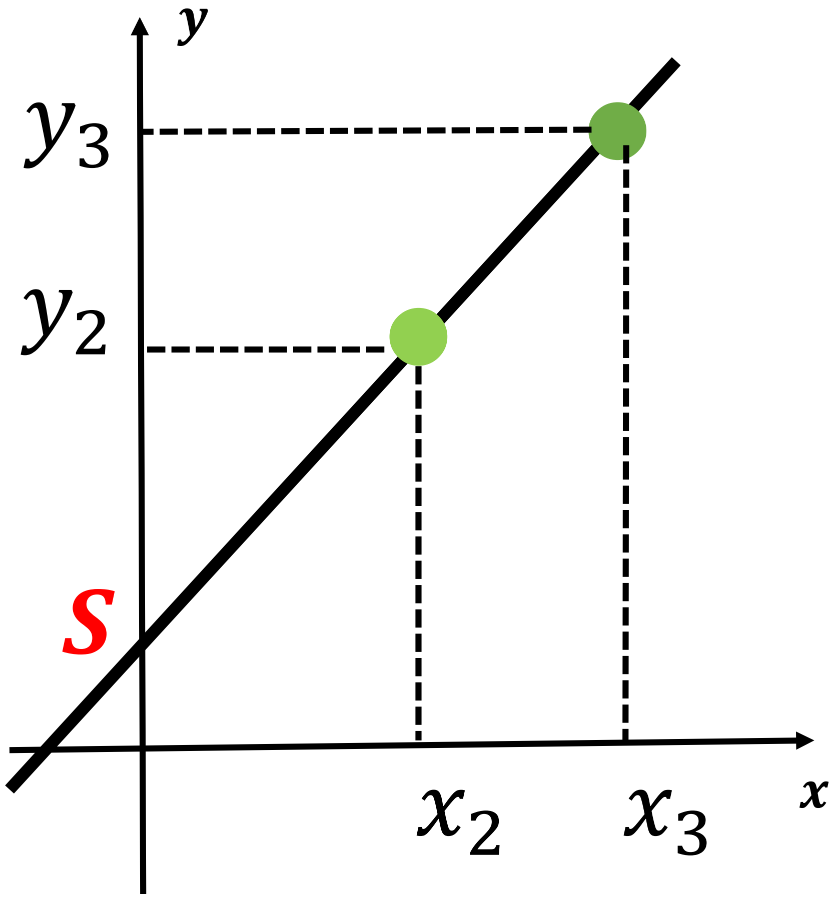
x_i : ServerID(Public)
y_i : Share(Secrecy)

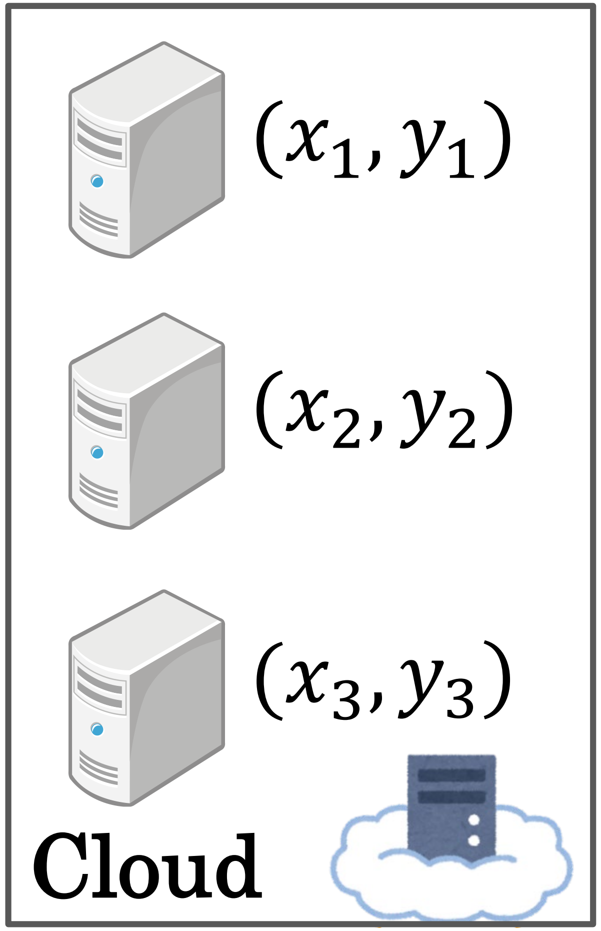


User

Secret S

Linear Polynomial through S





(x_1, y_1)

(x_2, y_2)

(x_3, y_3)

Cloud

Asymmetric Secret Sharing Scheme (A-SSS) [1]

- Combination of **Secret Sharing Scheme** and **Key Cryptography**

  - SSS is **Information-theoretic Secure.**
  - A-SSS partially has encryption using Key, so it results in secure using key.

→A-SSS is Computationally Secure.

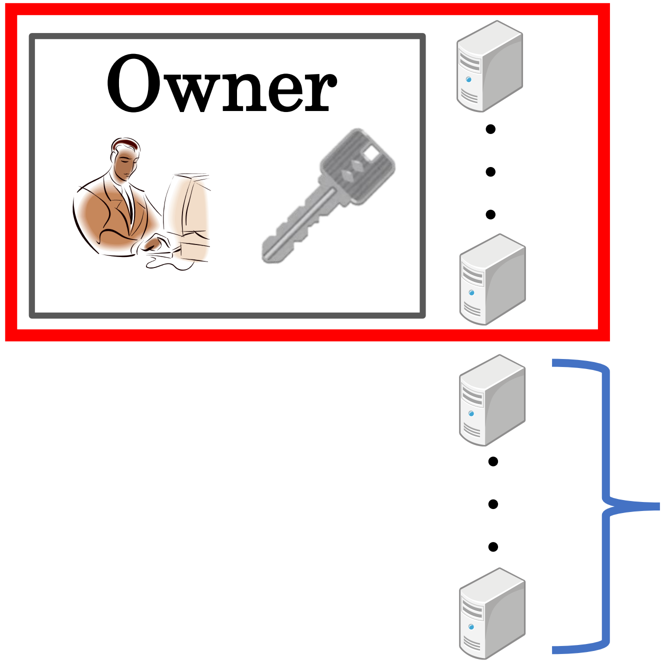
[1] Satoshi Takahashi, Hyunho Kang, Keiichi Iwamura, Asymmetric secret sharing scheme suitable for cloud systems, 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)

Asymmetric Secret Sharing Scheme (A-SSS) [1]

- A-SSS can limit external data servers to less than k
- A-SSS reconstruction requires permission from the key owner

⬇

No possibility of information leakage from external data servers without owner's permission



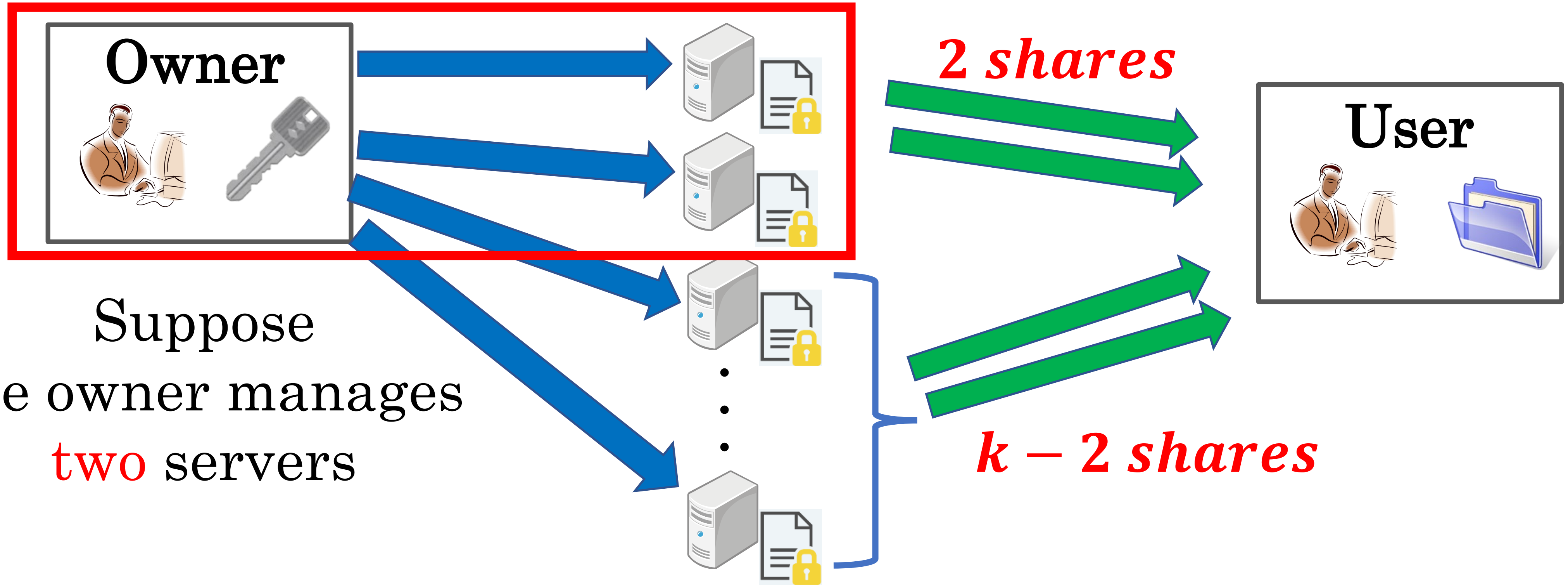
Key Servers less than k

Data Servers less than k

[1] Satoshi Takahashi, Hyunho Kang, Keiichi Iwamura, Asymmetric secret sharing scheme suitable for cloud systems, 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)

Asymmetric Secret Sharing Scheme (A-SSS) [1]

- <u>**In reconstruction, key server's share is must be used.**</u>



Suppose
the owner manages
two servers

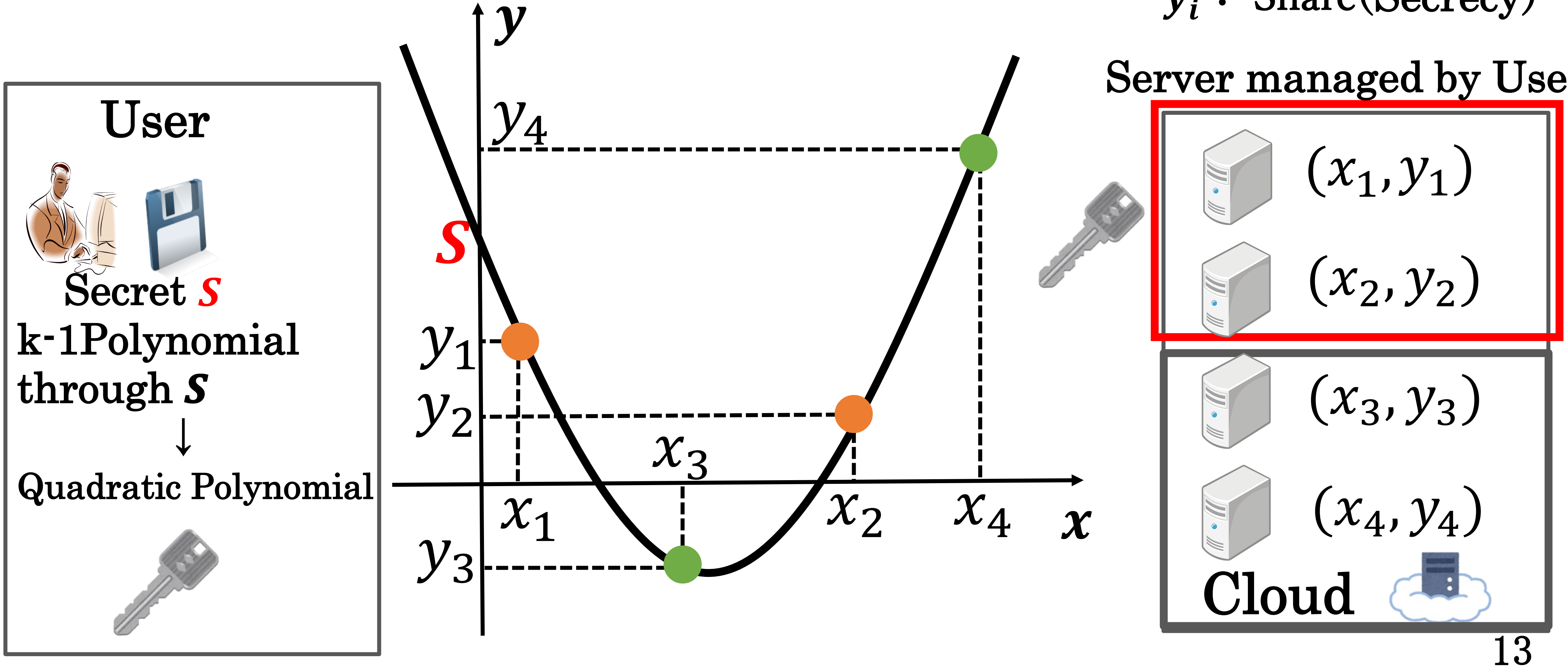
[1] Satoshi Takahashi, Hyunho Kang, Keiichi Iwamura, Asymmetric secret sharing scheme suitable for cloud systems, 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)

Distribution Process of A-SSS $(n = 4, k = 3)$

Create remaining shares from owner's share

x_i : ServerID(Public)

y_i : Share(Secrecy)



User

Secret S

k-1Polynomial through S

↓

Quadratic Polynomial

Server managed by User

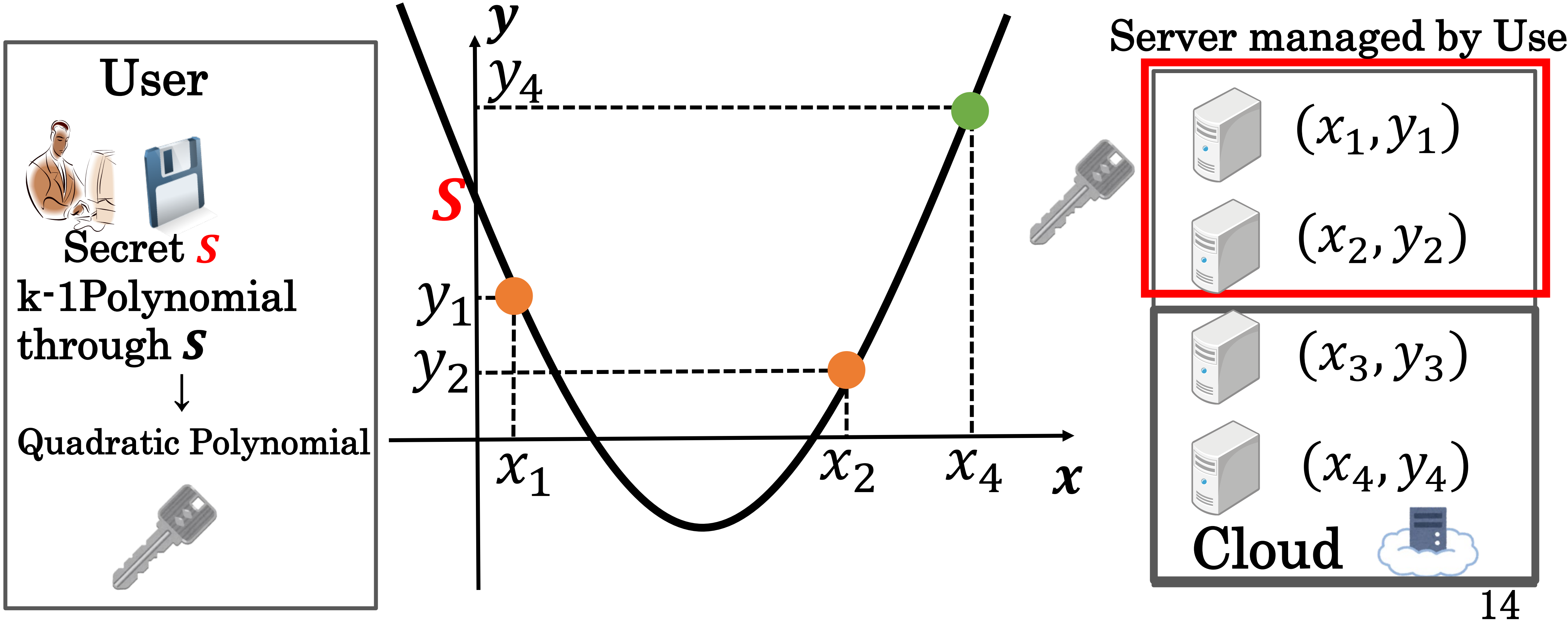
Cloud

Reconstruction Process of A-SSS $(n = 4, k = 3)$

Key Server's shares are must be used

x_i : ServerID(Public)

y_i : Share(Secrecy)



User

Secret s

k-1Polynomial through s

↓

Quadratic Polynomial

Server managed by User

(x_1, y_1)

(x_2, y_2)

(x_3, y_3)

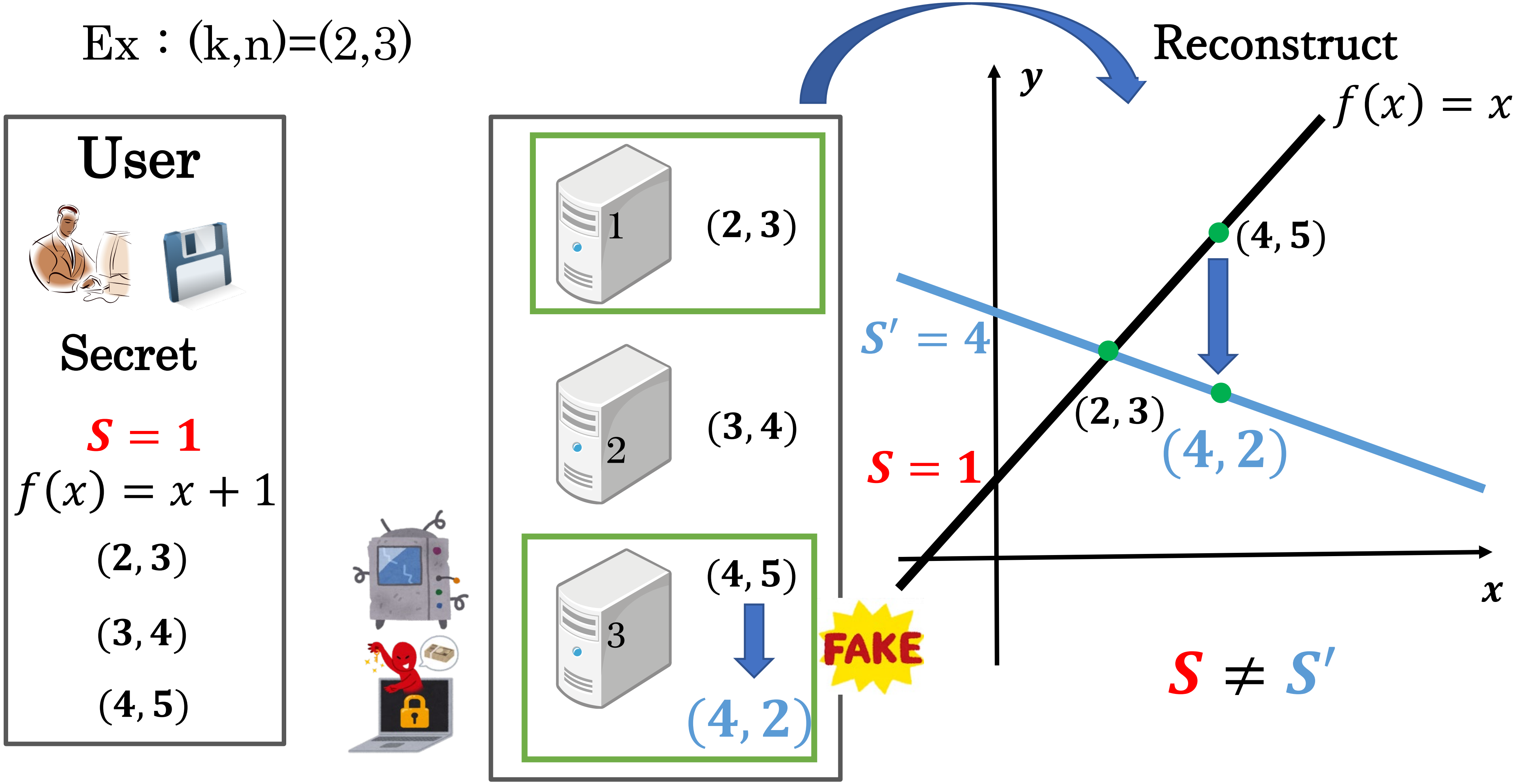
(x_4, y_4)

Cloud

Problem 1 of SSS and A-SSS

Correct Secret cannot be reconstructed if at least one forged share is collected

Ex：(k,n)=(2,3)



User

Secret

$S = 1$

$f(x) = x + 1$

$(2, 3)$

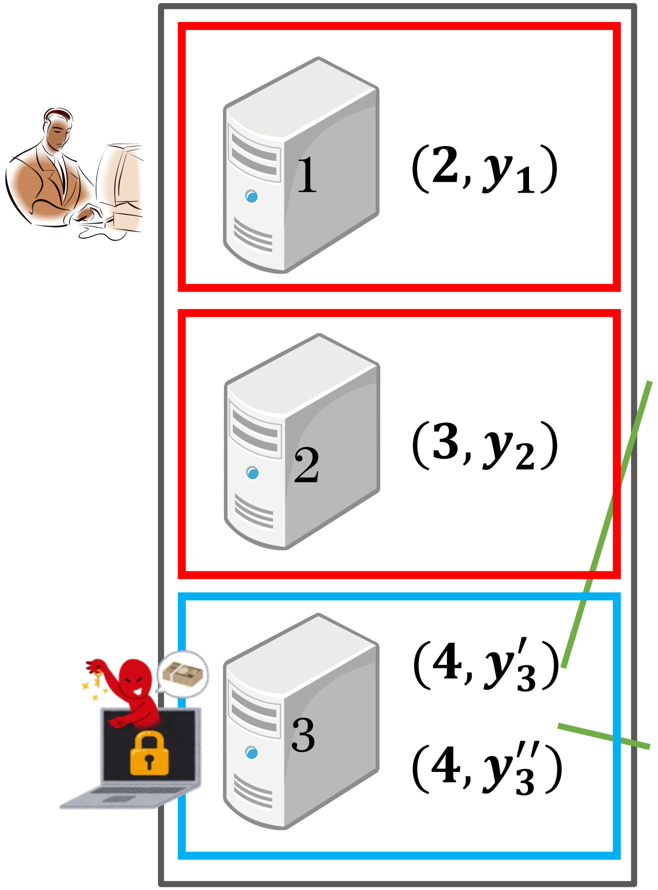
$(3, 4)$

$(4, 5)$

$S \neq S'$

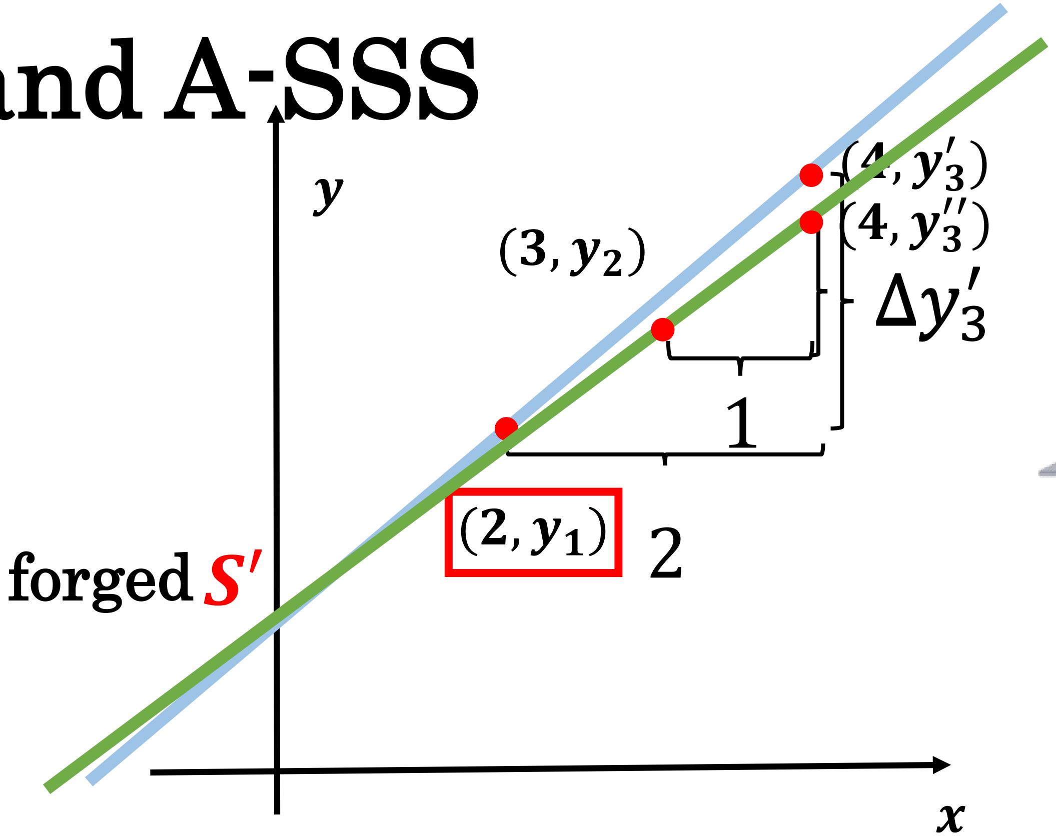
Problem 2 of SSS and A-SSS

ServerID$(x_1, x_2, \cdots, x_n)$ is public

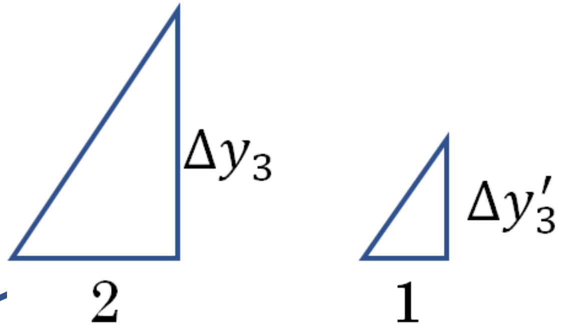


Difference
$\Delta x: 2, \Delta y_3$

Difference
$\Delta x: 1, \Delta y_3'$



$$\Delta y_3' = \frac{1}{2}\Delta y_3$$



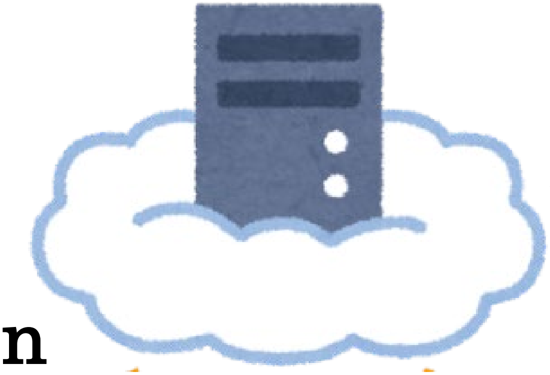
: ?

Continue outputting forged S'

Normal SSS

User

Cloud



Distribution

7

Reconstruction

Reconstruction

Forged Secret

Forged Secret

Forged Secret

If attackers are involved

7

8 Forgery

Furthermore,
if the server ID
is made public

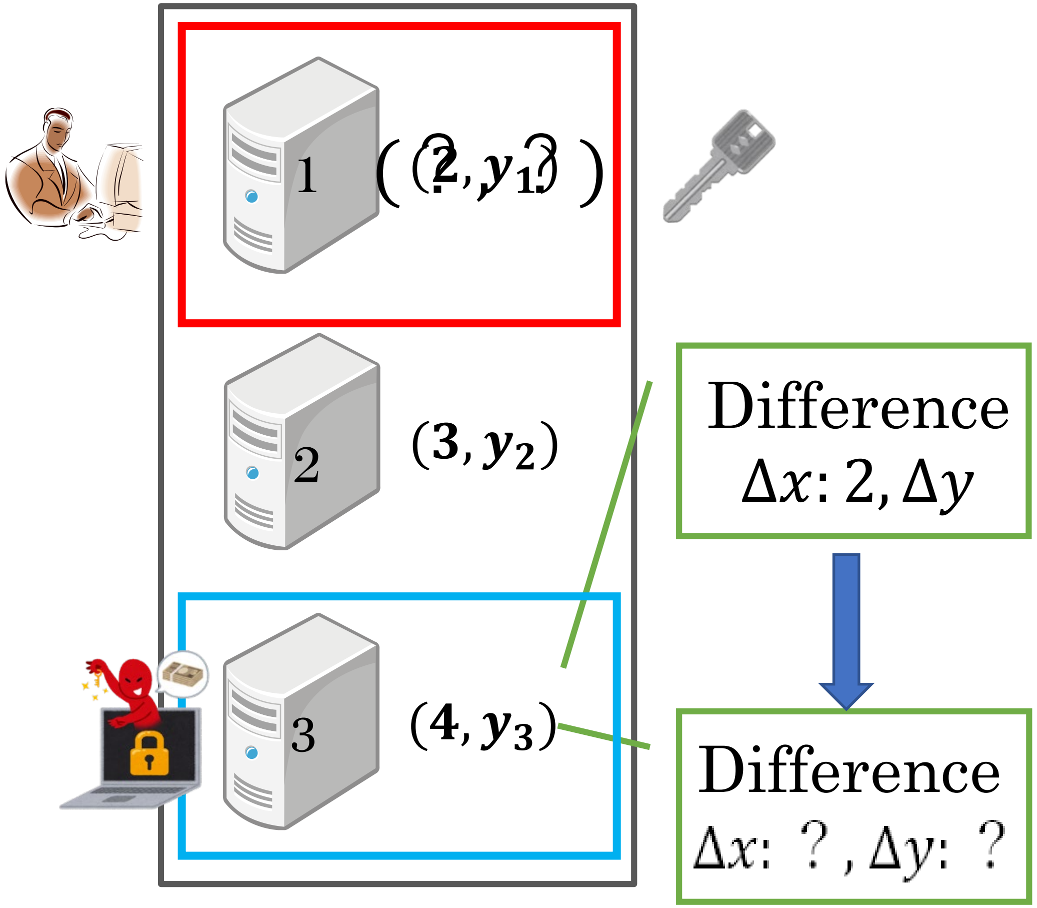
8

8

Solution to the Second Problem

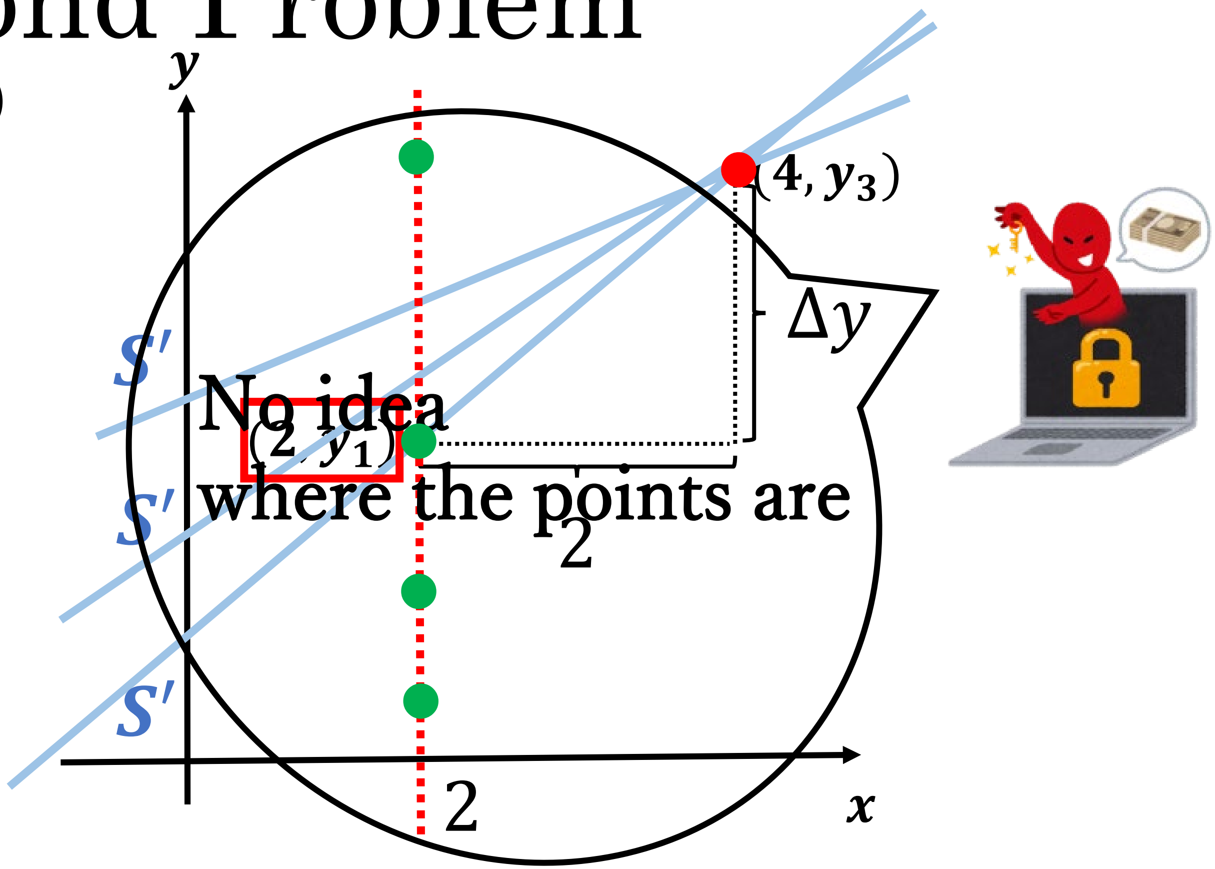
(Concealment of Server's ID)

→Keep key server's ID private



Difference
Δx: 2, Δy

Difference
Δx: ? , Δy: ?



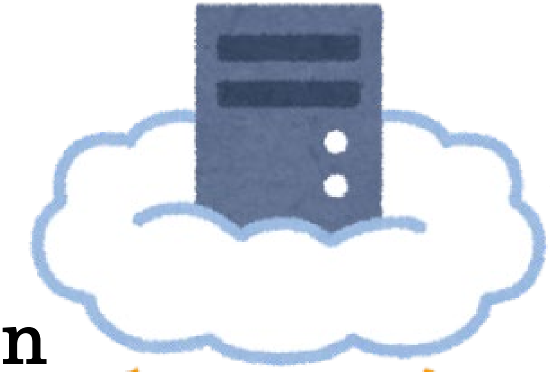
→ Unable to Specify function uniquely

⇒ Unable to Match with false Secret S'

Prevent matching with false Secret S'

Extended A-SSS

User



Cloud

Distribution

7

Reconstruction

Prevent continuous outputting 8

Forged Secret

✖

If attackers are involved

7

↓

8 Forgery

However,
Key's server ID
is Private

8
8

Proposed Scheme

- Extension of A-SSS → Concealment of Key Server's ID

  →Prevent continuous generation of false Secret S'

  →Data involved and generated by the attacker is different each time

→ **Able to verify correctness of reconstructed secret only by repeating reconstruction and just comparing the outputted values.**

Proposed Scheme

Data involved and generated by the attacker is different each time

→ That matched are considered correct

That did not matched are considered incorrect.

Ex) Perform Reconstruction Process 4 times →(7,9,7,8)

$$\boxed{7 = 7}$$

7 : Correct
(match)

$$\boxed{8 \neq (7,9), 9 \neq (7,8)}$$

8,9 : Cheating
(not match)

- Perform the same reconstruction multiple times

→ Cheat Detection

Conditions for identifying dishonest servers

Specific identification possible：$\boldsymbol{k - t < u - e}$

ex)$n = 5, k = 3, t = 2, u = 3, e = 1$

$(\boldsymbol{k - t = 1 < u - e = 2})$

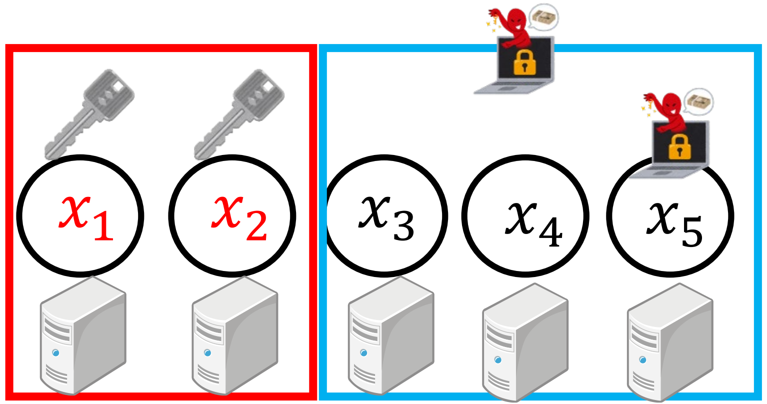
n ：Total number of servers $(n = t + u)$

k ：Threshold number of servers

t ： Number of Key servers $(0 < t < k)$

u ： Number of DS $(0 < u < k)$

e ： Number of dishonest servers in DS $(0 < e < u)$



$t = 2$ $\quad u = 3, e = 1$

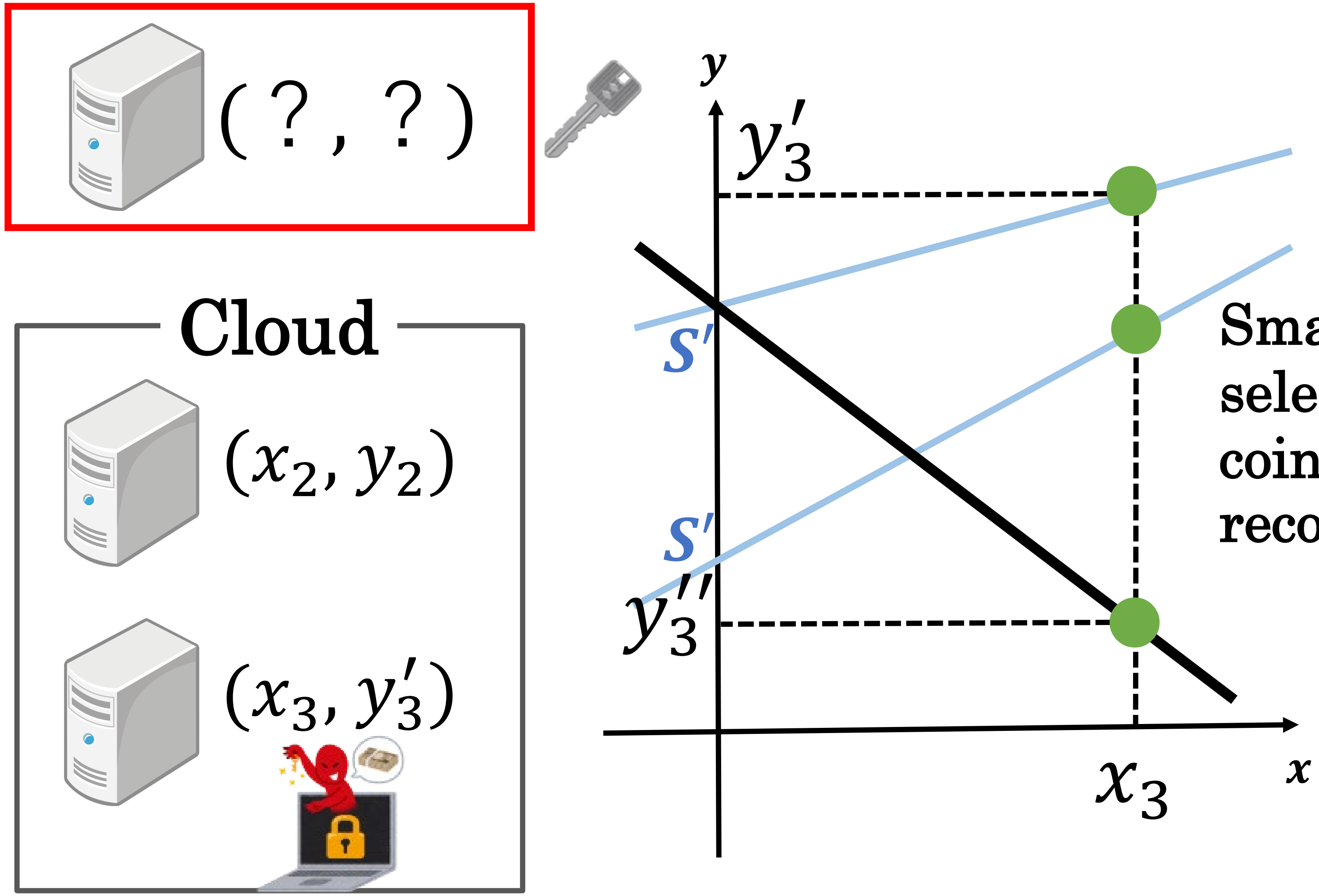
$(x_1, x_2, x_3) \rightarrow 7$

$(x_1, x_2, x_4) \rightarrow 7 \qquad (7 = 7)$

$(x_1, x_2, x_5) \rightarrow 8 \qquad (8 \neq 7)$

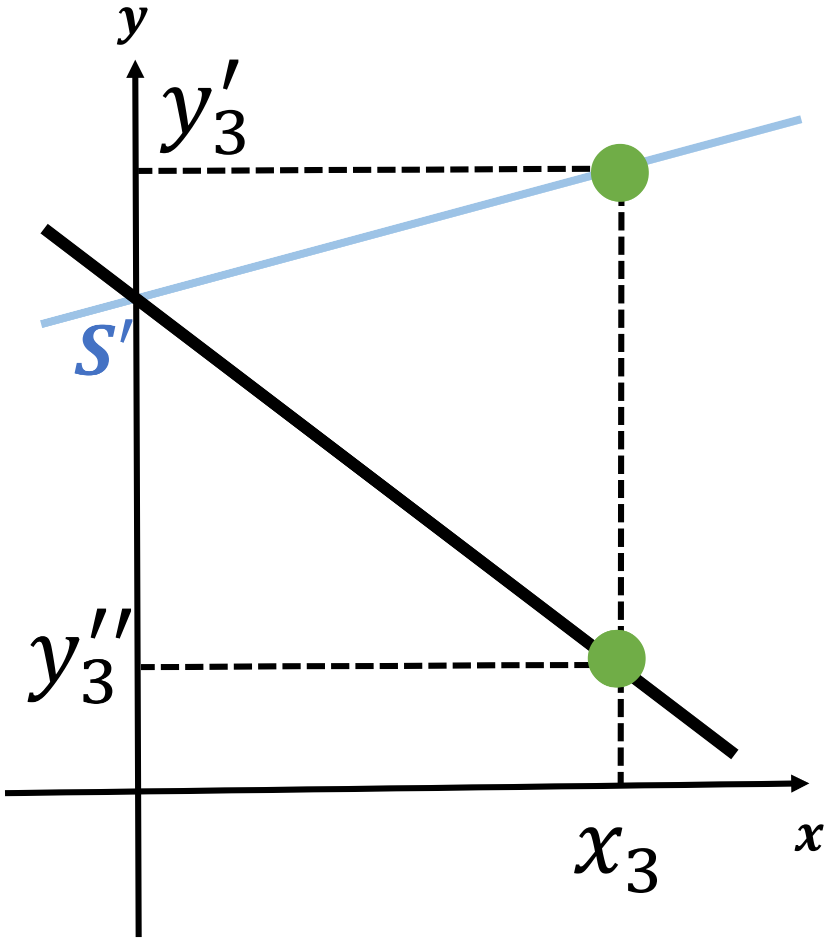
7 matchs,
→Adopt **7**,$(x_3, x_4)$is honest
→ **8** is forged Secret
→ x_5 is dishonest

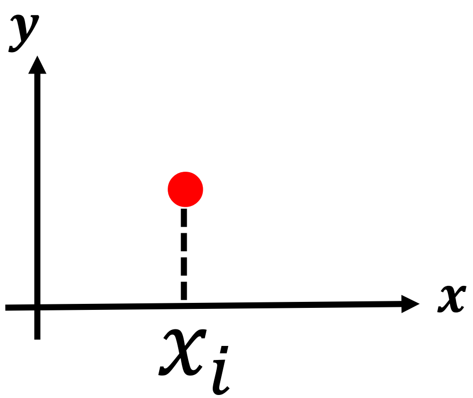
Security of Proposed A-SSS



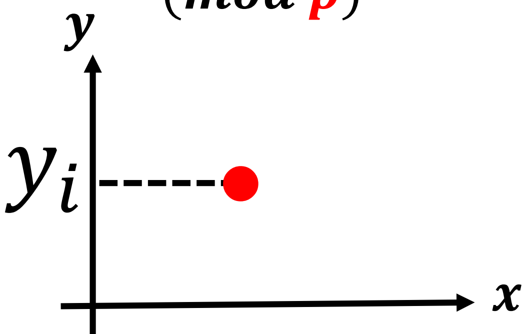
Small probability that (x, y) selected by attackers by coincidence with false reconstruction Secret S'

Security of Proposed A-SSS





(mod p)



(mod p)

x_i : p Points
y_i : p Points

Equations
(x_i, y_i) : p Pairs

Small probability by coincidence with false reconstruction Secret S'

$$\frac{p}{p \times p} = \frac{1}{p}$$

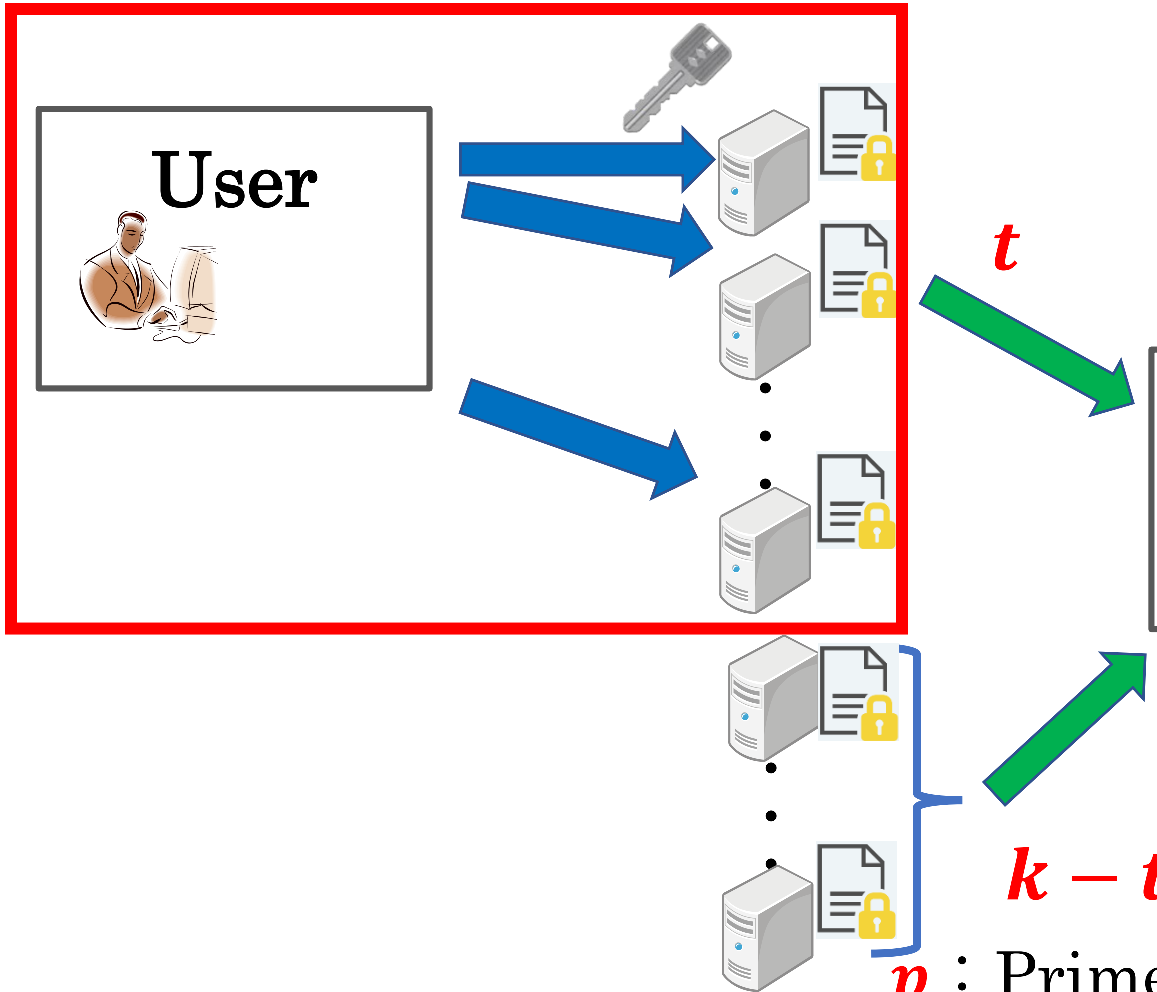
(Other methods have the same probability)

p : Prime numbers that can be represented by 128*bit* or more

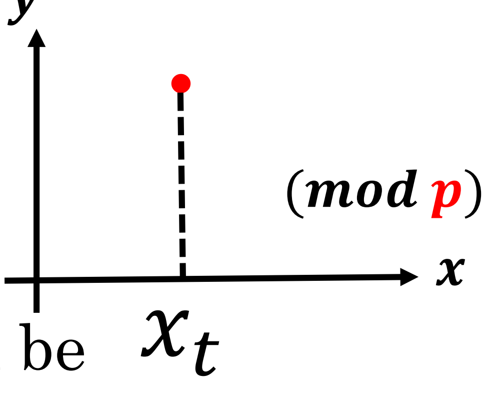
Security of Proposed A-SSS

Probability that guess and match key server ID

$$\frac{1}{p^t} = \left(\frac{1}{p}\right)^t$$



Probability of knowing one key server ID ： $\frac{1}{p}$



p ： Prime numbers that can be represented by 128*bit* or more

CONCLUSION

- Extension of A-SSS(server ID concealment)
  - Secret can be verified with only two reconstructions
    (Able to check whether it has been forged)
  - Possible to identify dishonest server
    - Up to ${}_{u}C_{k-t}$ reconstructions
    - $k-t<u-e$

$(u$: date servers, e : dishonest servers, t : key servers$)$

Thank you for your attention