Orchestrating a brighter world

NEC

Introduction of NEC's Secure Computing Technology

November 2018

NEC Security Research Laboratories

Orchestrating a brighter world

NEC brings together and integrates technology and expertise to create the ICT-enabled society of tomorrow.
We collaborate closely with partners and customers around the world, orchestrating each project to ensure all its parts are fine-tuned to local needs.

Every day, our innovative solutions for society contribute to greater safety, security, efficiency and equality, and enable people to live brighter lives.

Contents

1. The Social Value of Consolidating and Analyzing Data from Multiple Organizations
2. What is Secure Computing Technology?
3. Use Cases
4. NEC's Secure Computing Technology

1. The Social Value of Consolidating and Analyzing Data from Multiple Organizations

Expectations for Utilizing Data among Organizations

Sharing data among organizations, and consolidating and analyzing that data, are expected to create social value, but there are issues in sharing.

Value creation by data sharing and analysis

- Example: If in the U.S. hospitals, care givers, and pharmaceutical companies were able to share data, it would create **over $300B** of value in a year. [1][2]

Data sharing goals	Value creation ($100M)
Lifestyle habit improvement	700-1000
Medical care, nursing coordination	900-1100
Optimal medical care selection	500-700
Cost effectiveness verification	500-1000
Accelerate drug development, verification	400-700
Total	3000-4500

[1] McKinsey Global Institute, *Big data: The next frontier for innovation, competition, and productivity*, May 2011

[2] McKinsey Global Institute, *The 'big data' revolution in healthcare — Accelerating value and innovation*, January 2013. Exhibit 4.

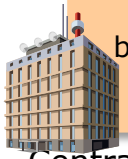
Major hindrances to data sharing

1. Personal privacy protection: **Providing personal information to a third party without the individual's consent is prohibited (illegal)**

2. Confidential information as source of competitiveness: Enterprises and research institutes **dislike disclosing their data to competitors**

Creating Social Value by Secure Data Consolidation and Analysis

Consolidate and analyze different confidential data (trade secrets, personal information, etc.) held by organizations **without mutual disclosure** to promote value creation by data utilization among organizations



Preventive medicine by correlation analysis of disease and exercise

Central and local governments

Individualized healthcare based on correlation analysis of genome and medication

Medical research institutions

Detection of fraudulent money transfers by consolidated analysis of financial information

Financial institutions

• • • • •

Consolidate data from multiple organizations and provide only the analysis results

Secure data consolidation and analysis (Secure computing technology)

Consolidate confidential data from multiple organizations without mutual disclosure

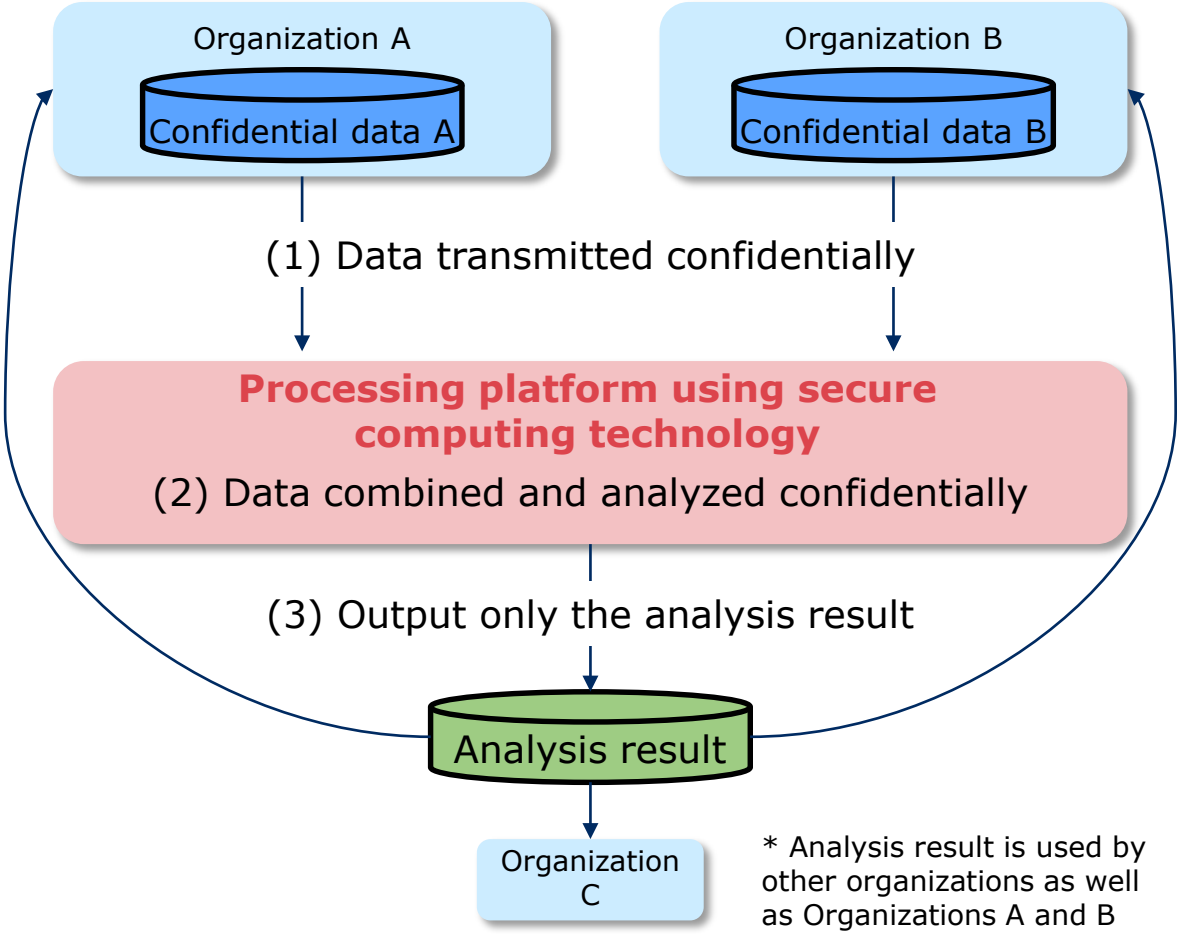
Medical institutions	Healthcare businesses	Financial institutions	Carriers, EC sites, etc.
Medication/Disease history, genome information, etc.	Exercise amount information Location information	Bank account balance Transfer information, etc.	Location information Purchase information, etc.

• • • • •

2. What is Secure Computing Technology?

Overview of Secure Computing Technology

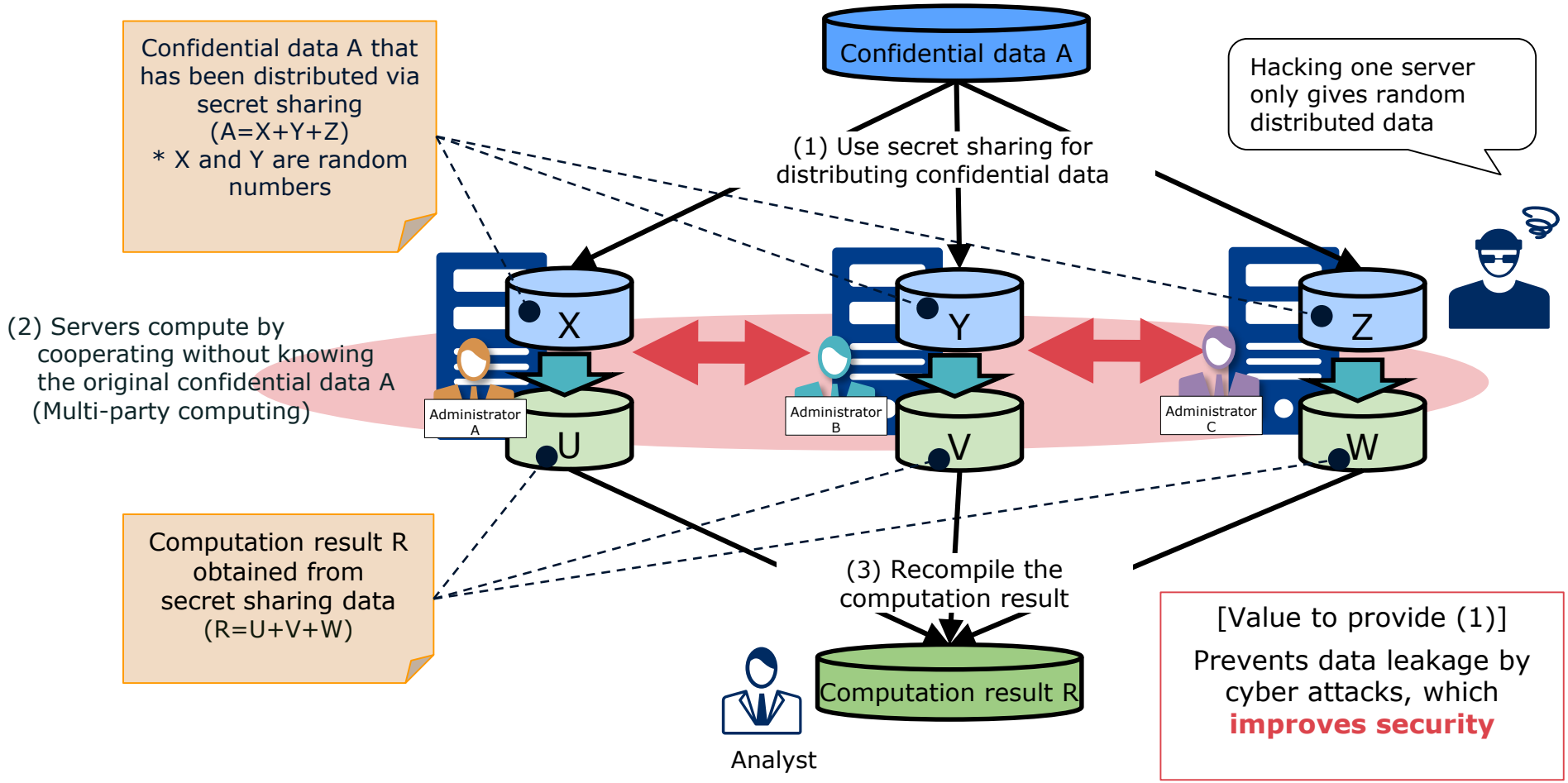
- Secure computing is a technology where data is processed confidentially
- Able to perform analysis based on data collected and consolidated from different organizations without disclosing the original data to entities outside of the respective organizations



[Reference] Explanation of Secure Computing Technology (1/2): Overview

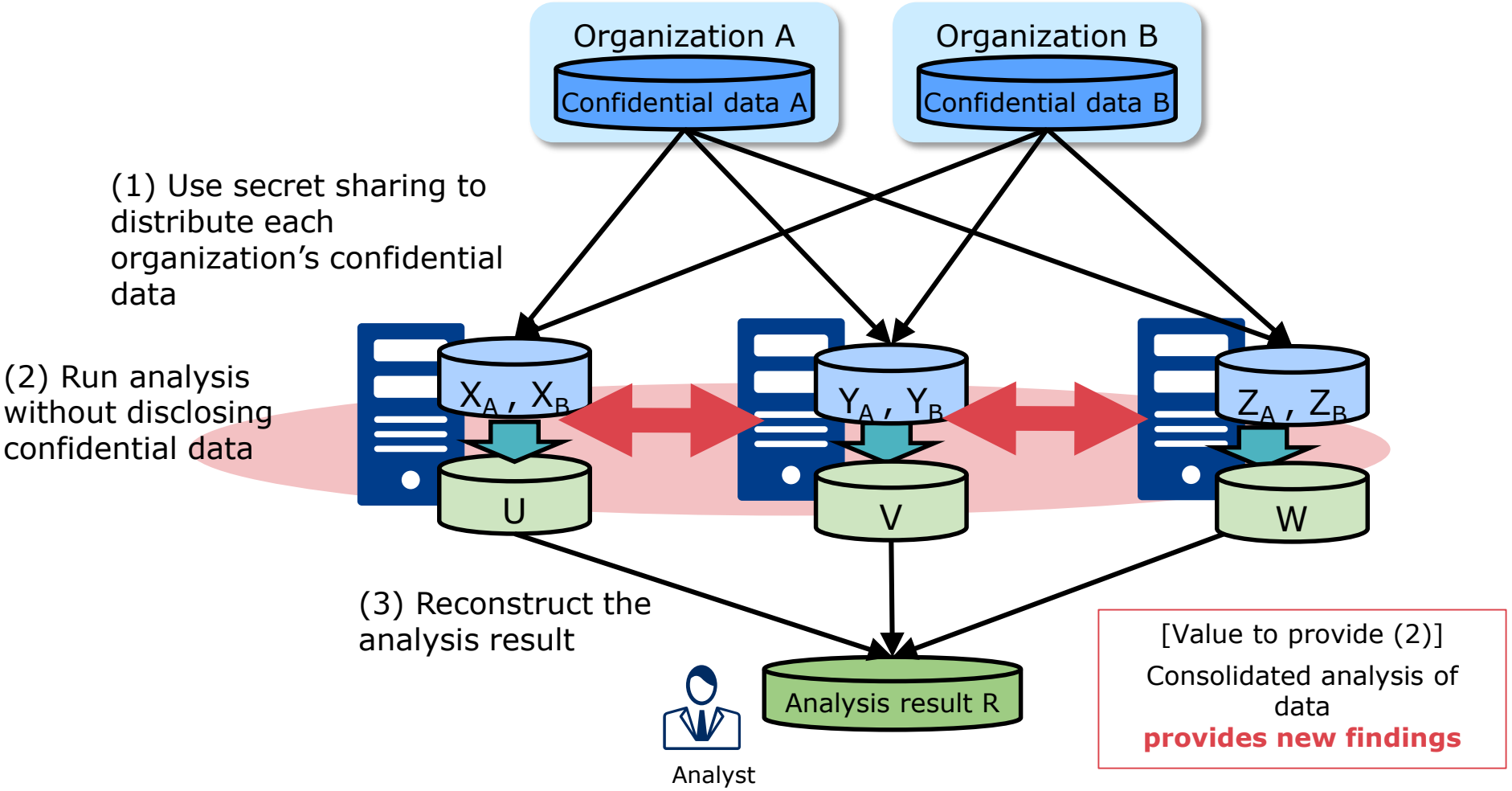
Secure computing technology (secret-sharing, multi-party computation) allows for computation using confidential data distributed across multiple servers **while keeping the original data confidential**.*1

*1: Theoretically any computation is possible by expressing computing as a logical circuit.



[Reference] Explanation of Secure Computing Technology (2/2): Inter-organization Data Consolidation

Able to obtain the consolidated analysis result by processing data distributed across multiple organizations without disclosing the original data to entities outside of the respective organizations



Characteristics of Secure Computing Technology

Secure computing technology is data processing where data is processed confidentially and can offer the following values:

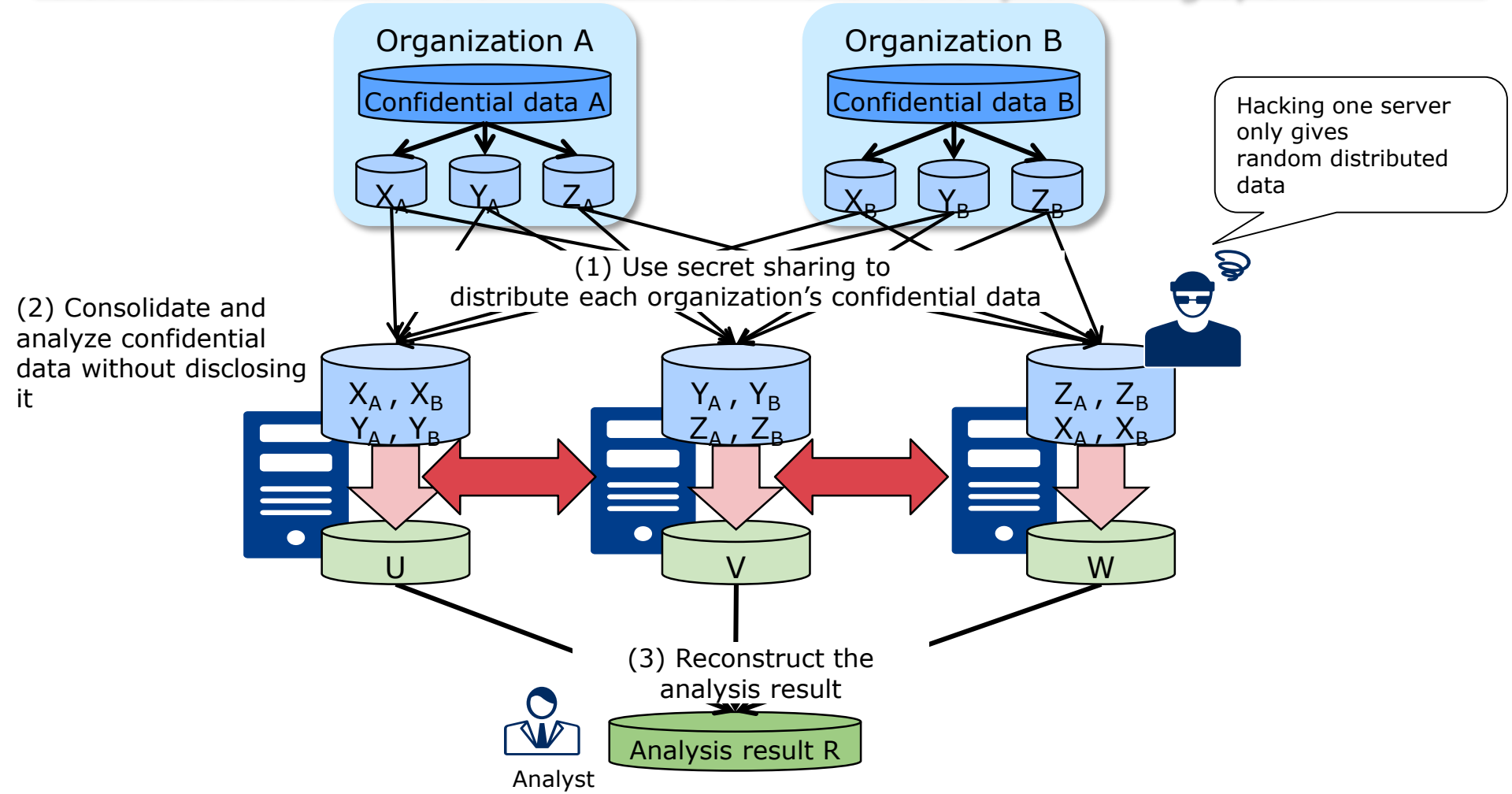
- (1) Prevents data leakage by cyber attacks, which **improves security**
- (2) **Provides new findings** through consolidated data analysis

Restrictions

- Secure computation involves a large amount of data communication, which makes processing several orders of magnitude slower than ordinary data processing
  - * Notes: Processing speed depends on the processing performance of the servers that perform secure computation and the network speed.

[Reference] NEC's Secure Computing Technology

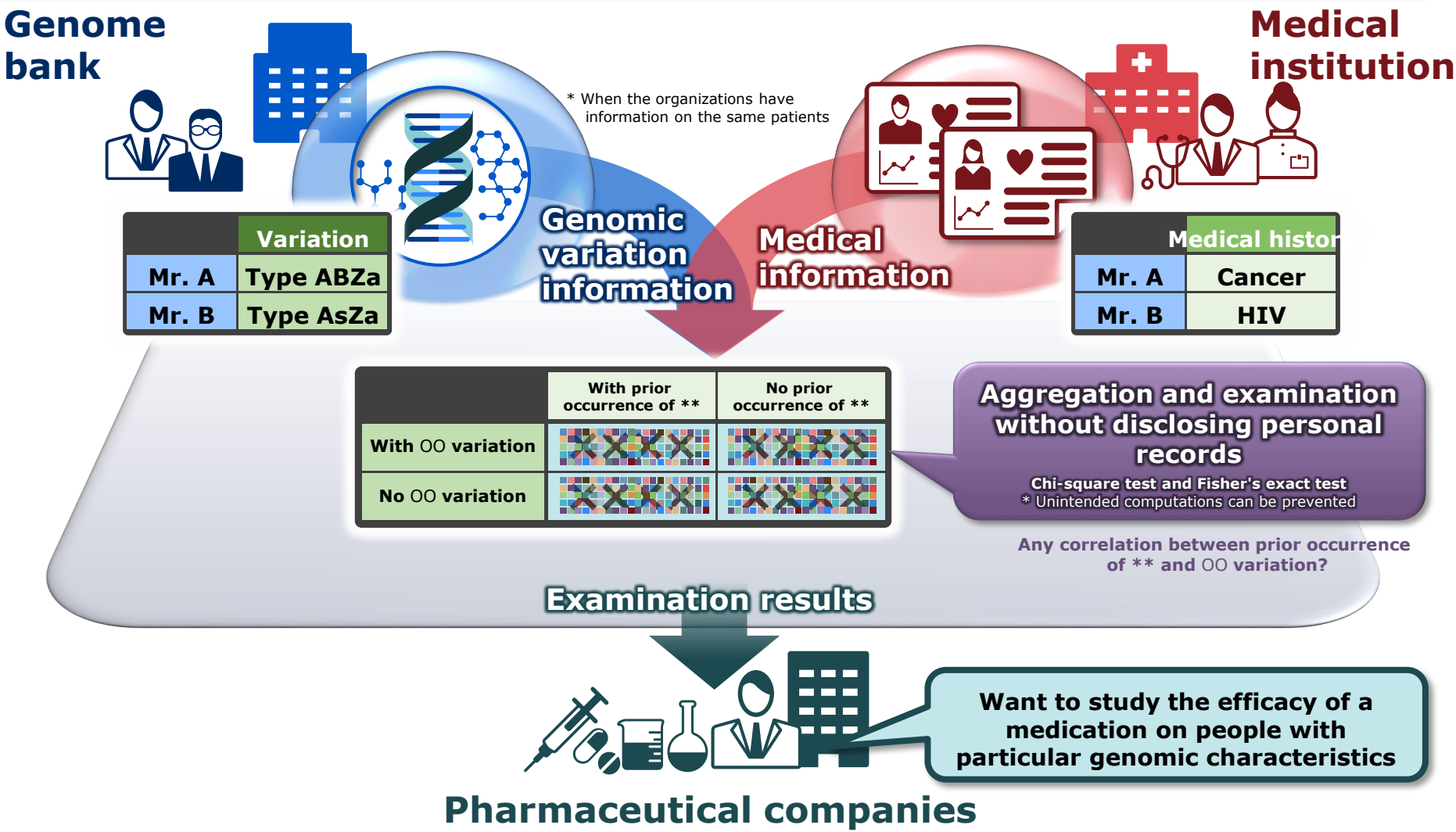
NEC adopts a secure computing approach where the original data is reconstructed from 2 secret-shared values out of the secret-shared values distributed to 3 servers in order to achieve faster processing speed



3. Use Cases

Use Case Example: Medical Area

Supports the development of medications according to genomic characteristics and custom-made medical treatment



Use Case Example: [Financial Area] Higher-precision Credit Examination and Predictive Analysis of Irrecoverable Loan Risks

Create a new business that securely uses each company's data and proprietary predictive models

Credit service companies

Credit service using proprietary prediction model

Prediction modeling

Prediction model

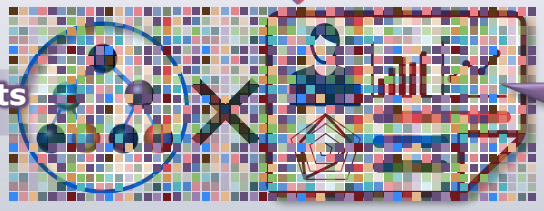
Calculate risks from age, gender, behavior, etc.

Loan applicant's information

Telecommunication carriers
Credit card companies, etc.

Provision of action history and personal information necessary for credit examination

Action history
Personal information



Prediction results

Predictive analysis on secure computing platform

What is the risk score of the loan applicant?

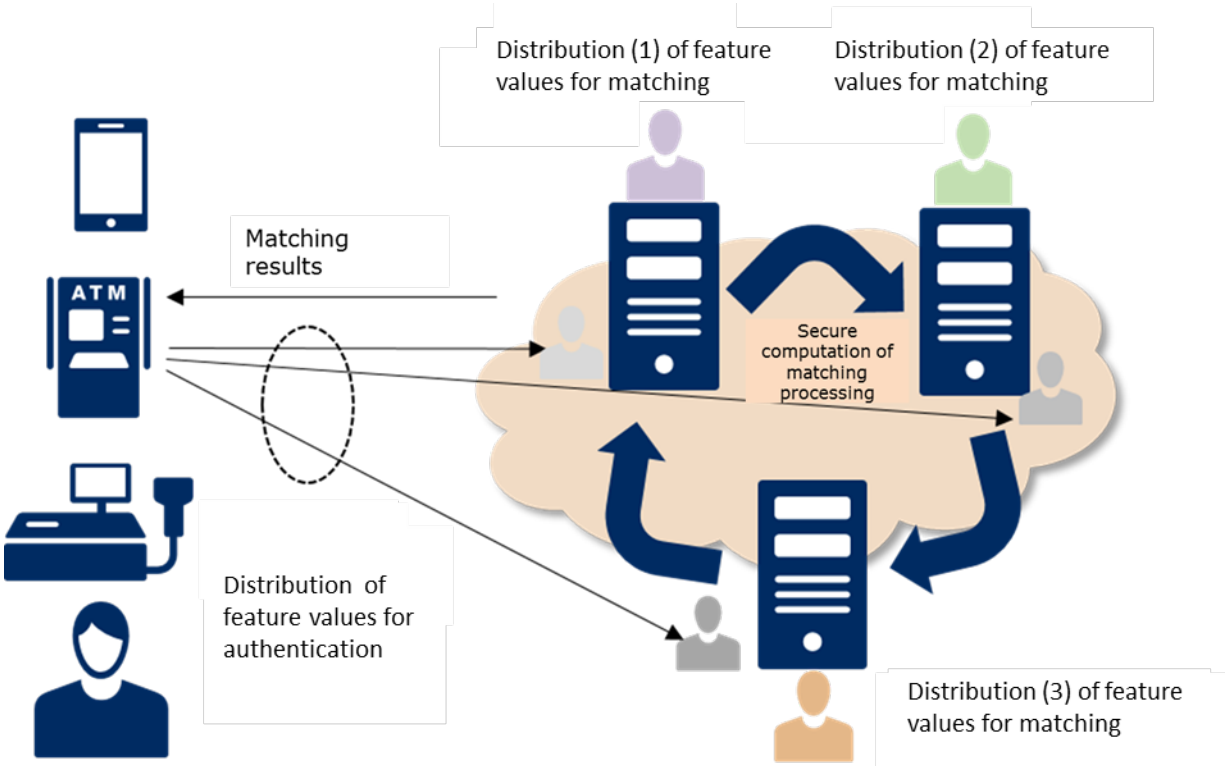
Higher-precision credit examination

Bank

Loan applicant

Use Case Example: (Face Biometric Authentication) Secure Management of Face Biometric Authentication Information on the Cloud

Realize secure management of biometric information on the cloud by developing biometric authentication that can keep face template information secret.
→ Face recognition systems can be introduced in environments where physical protection is difficult.



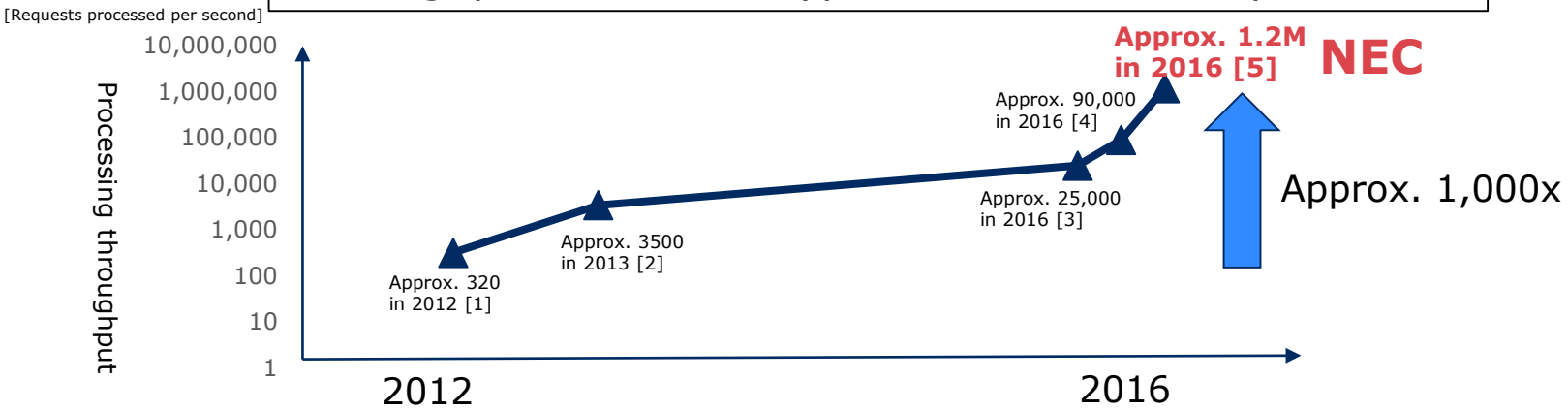
4. NEC's Secure Computing Technology

Characteristics of NEC's Secure Computing Technology

NEC established **high-speed secure computation** and achieved practical performance level for certain processing

- Secure computation got faster by orders of magnitude in recent years (Approx. 1,000x since 2012)

Throughput of AES encryption for secure computation*1



- Example: Realistic performance attained in matching of face feature values and DNA Levenshtein distance computation*2
  - Matching of Face feature values: Approx. 45,000 requests for 1000D feature values are processed per second*3
  - DNA Levenshtein distance computation: Levenshtein distance computation between DNA sequences with a length of 100: Approx. 145 per second

Research papers by NEC's researchers earned international recognition and were accepted by top international conferences
CCS2016 (Best Paper), **Eurocrypt2017, S&P2017, CCS2018**

[1] J. Launchbury, I.S. Diatchki, T. DuBuisson and A. Adams-Moran. "Effcient lookup-table protocol in secure multiparty computation". ACM ICFP2012.
[2] S. Laur, R. Talviste and J. Willemson. "From Oblivious AES to Effcient and Secure Database Join in the Multiparty Setting", ACNS2013.
[3] R. Talviste. "Applying Secure Multi-Party Computation in Practice", Ph.D dissertation, Univ. of Tartu, 2016.
[4] J. Randmets. Personal comm. AES performance on the new Sharemind cluster. May, 2016.
[5] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, Kazuma Ohara,"High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority", ACM CCS2016.

*1: Compared with secure computation among three semi-honest, secure parties. Graph created based on Table 1 of the paper [5] .
*2: For details, refer to "Tsuchida et al, "Protection of Biometric Information and Genetic Information by Fraud-detecting Multi-party Computation", SCIS2018."
*3: VISA's peak transaction volume: 47,000 transactions per second

NEC's Development Support Tool for Secure Computation

The development support tool for secure computation that NEC developed enables ordinary engineers to easily build an application using secure computation

Before:

Design a secure computation circuit (logical circuit, arithmetic circuit) and write a script

Secure computation expert

```
…
ldsi                    s100, 0
adds                    s102, s101, s100
movs  s102, s100
…
ge_startmult 2, sg5, sg7
…
ge_stopmult 1, sg506
gadds   sg505, sg2, sg503
gaddsi  sg496, sg505, 1
…
```

Several tens of thousands of lines

After:

Write the script
in a programming language
similar to Python

Ordinary SE

```
def  mean(value, num):
sum = sint(0)
for i in range(num):

      sum = sum + value[i]


      …
```

Several tens to hundreds of lines

* Example: Simple aggregate processing

Development tool
("Compiler")

```
…
ldsi                    s100, 0
adds                    s102, s101, s100
movs  s102, s100
…
ge_startmult 2, sg5, sg7
…
ge_stopmult 1, sg506
gadds   sg505, sg2, sg503
gaddsi  sg496, sg505, 1
…
```

Several tens of thousands of lines

* Number of processing lines can be reduced by optimization

Script example: Write the script in a programming language similar to Python

Sample program that calculates the mean

Function definition

Main part

```
def mean(value, num):
    sum = sint(0)    #sum is formatted as concealed
    for i in range(num):
        sum = sum + value[i]
    Sf = sfix(0)
    Sf.load_int(sum)    #cast to fixed point num
    Nf = sfix(0)
    Nf.load_int(num)    #cast to fixed point num
    mean = Sf / Nf

    return mean

# === Main processing starts here ====
num_input = 1000

input_values = Array (num_input, sint)

for i in range(num_input):
    input_values[i] = sint.get_input_from(2)

result = mean(input_values, num_input)

print_ln('mean = %s'¥n, result.reveal())
```

sint: "secret integer" type
Confidential integer

sfix: "secret fixed-point number" type
Confidential fixed-point number

Division with confidential fixed-point number

Confidential sint type array

Store confidential data in each array

Call mean function

Reveal the result, and print it to standard output. (For debugging)

Reference: Example of Implementing SQL-level Simple Statistical Processing (1/2)

A simple statistical processing that can be executed by SQL can be relatively easily implemented using the code generation support tool to write the processing corresponding to that SQL processing.

- SQL processing example
  - SELECT count(*), avg(mortage debt), stddev(mortage debt) FROM Census GROUP BY Incme bucket, State;
  - SELECT count(*), avg(credit limit) as avg, stddev(credit limit) as dev FROM Census Where State==Utah and Overall mortage debt >100 GROUP BY Age, Sex Having 1.3 avg + 1.2 dev > avg(monthly housing cost) ;
- Database (Table name: Census)

Column Name	data type	constraint	comment
Monthly housing cost	integer	$0 \leq x \leq 1,000,000$	USD
Overall mortgage debt	integer	$0 \leq x \leq 1,000,000,000$	USD
Mortgage percentage over property	integer	$0 \leq x \leq 100$	%
Credit Limit	integer	$0 \leq x \leq 1,000,000$	USD
Total other debt	integer	$0 \leq x \leq 1,000,000,000$	USD
Monthly housing cost in bucket	bit string	20 bits, all are 0 except one is 1	dependency
Overall mortgage debt in bucket	bit string	20 bits, all are 0 except one is 1	dependency
Mortgage percentage over property in bucket	bit string	20 bits, all are 0 except one is 1	dependency
Credit Limit in bucket	bit string	20 bits, all are 0 except one is 1	dependency
Total other deb in bucket	bit string	20 bits, all are 0 except one is 1	dependency
Age	bit string	83 bits, all are 0 except one is 1	$18 \leq x \leq 100$
Income in bucket	bit string	20 bits, all are 0 except one is 1	
Sex	bit string	2 bits, all are 0 except one is 1	M/F
Family size	bit string	10 bits, all are 0 except one is 1	$1-9$
State	bit string	50 bits, all are 0 except one is 1	50 states

Reference: Example of Implementing SQL-level Simple Statistical Processing (2/2)

Think up an algorithm that runs the same processing as the SQL processing

Write the program

Algorithm 173 SQL Query

Query Notation:

Select count(*), avg($\{[c_k]^n\}_{k|g_{k,\mu\nu}=1,w_k=1}$) as $[m_{\mu\nu}]^n$, stddev($\{[c_k]^n\}_{k|g_{k,\mu\nu}=1,w_k=1}$) as $[d_{\mu\nu}]^n$
From Census
Where $[w_k] := [([f_{k,u}] == [1])] \cdot [[o_k]^n > [100]^n]$
GROUP BY $[g_{k,\mu\nu}] := ([a_{k,\mu}] \otimes [s_{k,\nu}])$
HAVING $h'_{\mu\nu} := (1.3 \cdot [m_{\mu\nu}]^n + 1.2[d_{\mu\nu}]^n > \text{avg}(\{[m_k]^n\}_{k|g_{k,\mu\nu}=1,w_k=1})$;

Where Clause Computation:

For each k-th row, MPC $[w_k] = [f_{k,u} == [1]] \cdot [[o_k]^n > [100]^n]$ where $f_{k,u}$ represents u-th bits in the array $f_{k,*}$.

Group By Clause Computation (conceptual):

For each k-th row, MPC $[g_{k,\mu\nu}] = [a_{k,\mu}] \otimes [s_{k,\nu}]$.

Aggregation

For each μ, ν, MPC $[C_{\mu\nu}]^n = \sum_k \{[\text{inj}(a_{k,\mu})]^n \otimes [\text{inj}([s_{k,\nu}] \cdot [w_k])]^n\}$
For each μ, ν, MPC $[S_{\mu\nu}]^n = \sum_k \{([c_k]^n \cdot [\text{inj}(a_{k,\mu})]^n) \otimes ([\text{inj}(s_{k,\nu})]^n \cdot [\text{inj}(w_k)]^n)\}$
For each μ, ν, MPC $[S'_{\mu\nu}]^n = \sum_k \{([m_k]^n \cdot [\text{inj}(a_{k,\mu})]^n) \otimes ([\text{inj}(s_{k,\nu})]^n \cdot [\text{inj}(w_k)]^n)\}$
For each μ, ν, MPC $[D'_{\mu\nu}]^n = \sum_k \{[c_k]^n \cdot [c_k]^n \cdot [\text{inj}(a_{k,\mu})]^n) \otimes ([\text{inj}(s_{k,\nu})]^n \cdot [\text{inj}(w_k)]^n)\} - [S_{\mu\nu}]^n \cdot [S_{\mu\nu}]^n$
For each μ, ν, MPC $[D_{\mu\nu}]^n = \text{sqrt}([D'_{\mu\nu}]^n / [C_{\mu\nu}]^n)$

Having Clause Computation:

For each μ, ν, MPC $[h_{\mu\nu}]^n = 1.3 \cdot [S'_{\mu\nu}]^n / [C_{\mu\nu}]^n + 1.2 \cdot [D_{\mu\nu}]^n / [C_{\mu\nu}]^n - [S_{\mu\nu}]^n / [C_{\mu\nu}]^n$.
For each μ, ν, $[h'_{\mu\nu}] = ([h_{\mu\nu}]^n > 0)$.

Query Result: For each μ, ν, open all $[h'_{\mu\nu}]$. For those that are 1, open $[C_{\mu\nu}]^n$, $[S_{\mu\nu}]^n / [C_{\mu\nu}]^n$, and $[D_{\mu\nu}]^n$.

mpc/sql_3_.py

```
#Based on "Algorithm 173 SQL Query"

#------------------------------------------------------------------------------#
L = (20, 20, 20, 20, 20, 83, 20, 2, 10, 50)
a = 5                # 5 <= a <= 14
v = 0                # 0 <= v <= 4
N = 1000

#------------------------------------------------------------------------------#
def read_row():

#------------------------------------------------------------------------------#
def print_tup(t):
    print_ln('showing 5 integers')
    for i in range(5):
        print_ln('int %s = %s', i, t[i].reveal())

    for i in range(5, 15):
        print_ln('showing bit array %s', i)
        for j in range(L[i-5]):
            print_ln('bit %s = %s', j, t[i][j].reveal())

#------------------------------------------------------------------------------#

S = Array(L[a], sint)
C = Array(L[a], sint)

@for_range(N)
def perform_row(i):
    Ti = read_row()
    #print_tup(Ti)
    for j in range(L[a-5]):
        S[j] = S[j] + Ti[v]*Ti[a][j]
        C[j] = C[j] + Ti[a][j]

R = Array(L[a-5], sfix)
for j in range(L[a-5]):
    sfSj = sfix(0)
    sfSj.load_int(S[j])
    sfCj = sfix(0)
    sfCj.load_int(C[j])
    R[j] =sfSj/sfCj
    print_ln('R[%s] = %s; C[%s] = %s;', j, R[j].reveal(), j, C[j].reveal())

#------------------------------------------------------------------------------#
print_ln('Algorithm 173 SQL Query test done')
```

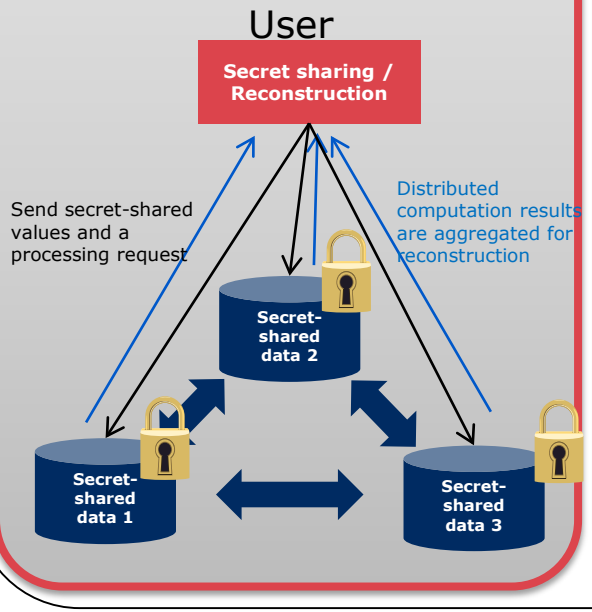
Appendix

Appendix: Different Approaches to Secure Computation

Secure Computation

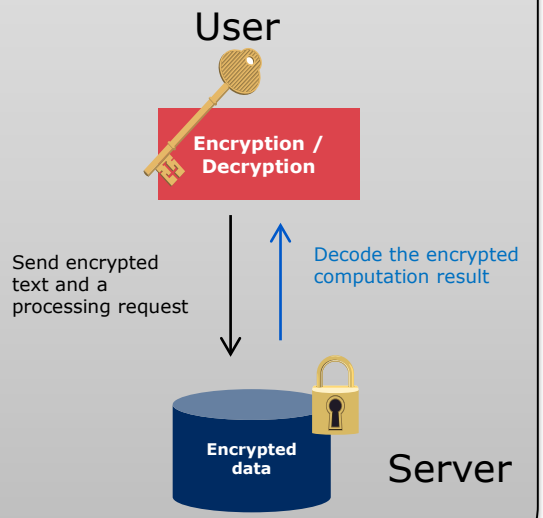
Using secret sharing

Secret-shared data processed individually



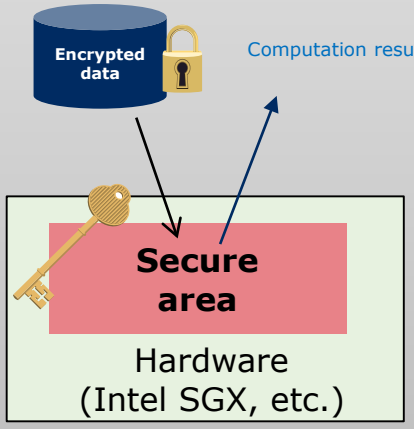
Using homomorphic encryption

Data processed without decoding



Using hardware (Trusted Execution Environment, etc.)

Data processed in secure area on hardware



Use a key stored in the secure area to decode and process data

. . .

There are also other approaches


Orchestrating a brighter world
NEC