

NECの秘密計算技術のご紹介

2018年11月

NEC セキュリティ研究所

Orchestrating a brighter world

未来に向かい、人が生きる、豊かに生きるために欠かせないもの。
それは「安全」「安心」「効率」「公平」という価値が実現された社会です。

NECは、ネットワーク技術とコンピューティング技術をあわせ持つ
類のないインテグレーターとしてリーダーシップを発揮し、
卓越した技術とさまざまな知見やアイデアを融合することで、
世界の国々や地域の人々と協奏しながら、
明るく希望に満ちた暮らしと社会を実現し、未来につなげていきます。

目次

1. 複数組織のデータ結合・分析の社会価値
2. 秘密計算技術のご紹介
3. ユースケースについて
4. NECの秘密計算技術

1.複数組織のデータ結合・分析の社会価値

組織間でのデータ利活用への期待

データを組織間で共有し、データを結合・分析することは、社会的な価値を創出できると期待されているが、共有には課題がある

データの共有分析による価値創出

- 例：米国の医療は、病院・介護者・製薬会社などがデータを共有して活用できれば、**毎年3000億ドル以上**の価値を生み出せる。[1][2]

データ共有の狙い	創出価値(億ドル)
生活習慣の改善	700-1000
医療・介護の連携	900-1100
最適な医療の選定	500-700
費用対効果の検証	500-1000
創薬・実証の加速	400-700
合計	3000-4500

データ共有の主な阻害要因

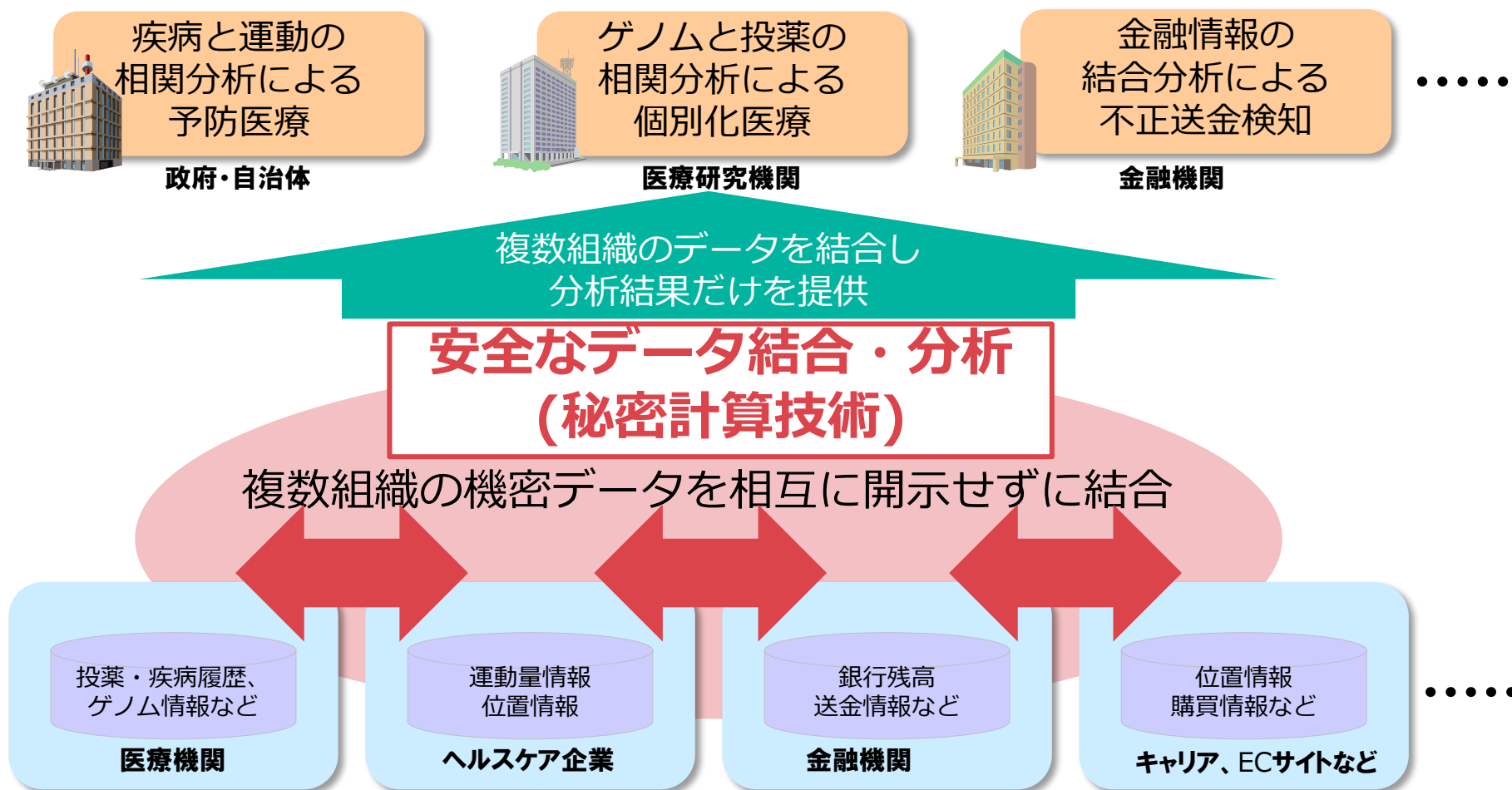
1. 個人のプライバシー保護：
個人の同意なく、個人情報の
第三者への提供は禁止(違法)
2. 競争力の源泉になる秘密情報：
企業や研究機関は、データを
競合に開示するのを嫌う

[1] McKinsey Global Institute, *Big data: The next frontier for innovation, competition, and productivity*, 2011年5月

[2] McKinsey Global Institute, *The 'big data' revolution in healthcare — Accelerating value and innovation*, 2013年1月. Exhibit 4.

安全なデータ結合・分析による社会価値創造

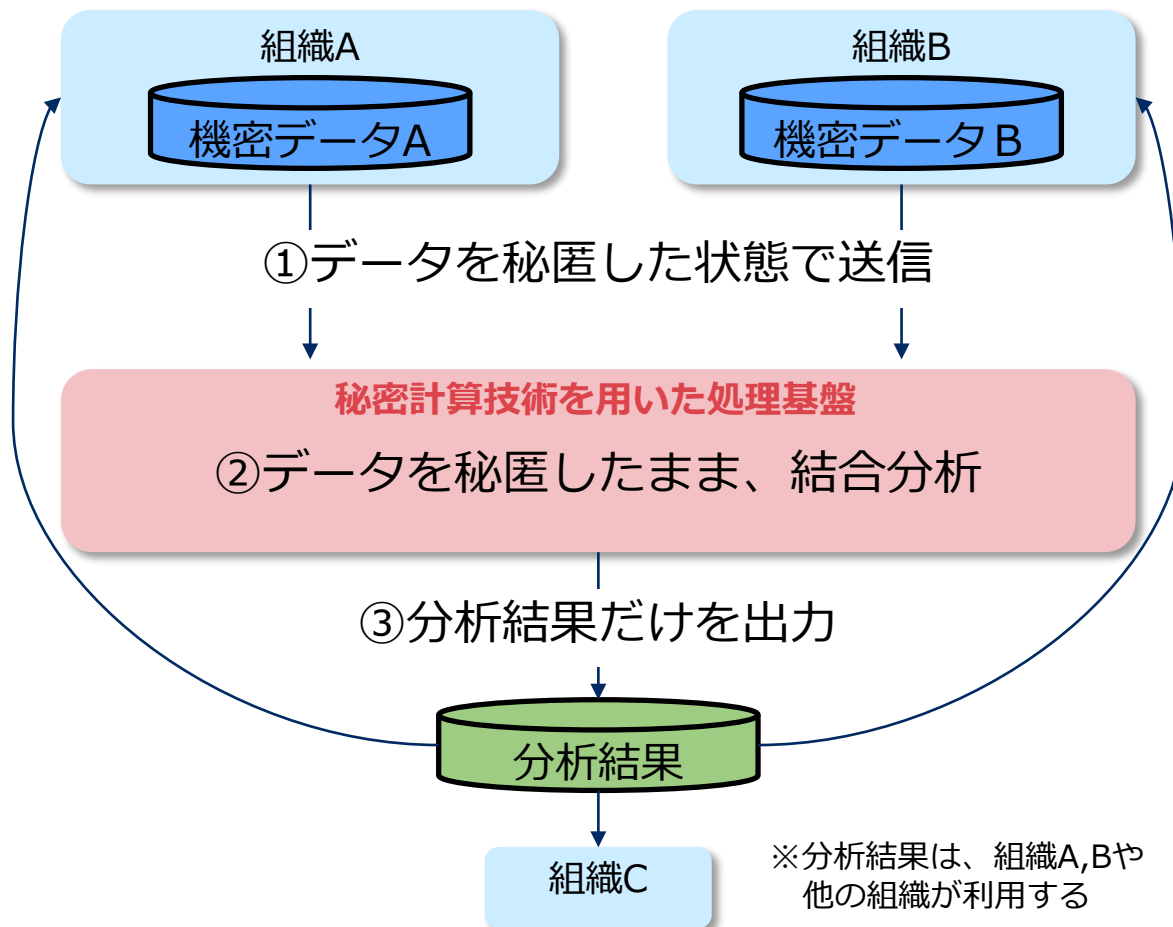
組織が保有する様々な機密データ(企業秘密、個人情報等)を、**相互に開示せず**に結合・分析し、組織間でのデータ活用による価値創出を促進



2. 秘密計算技術のご紹介

秘密計算技術の概要

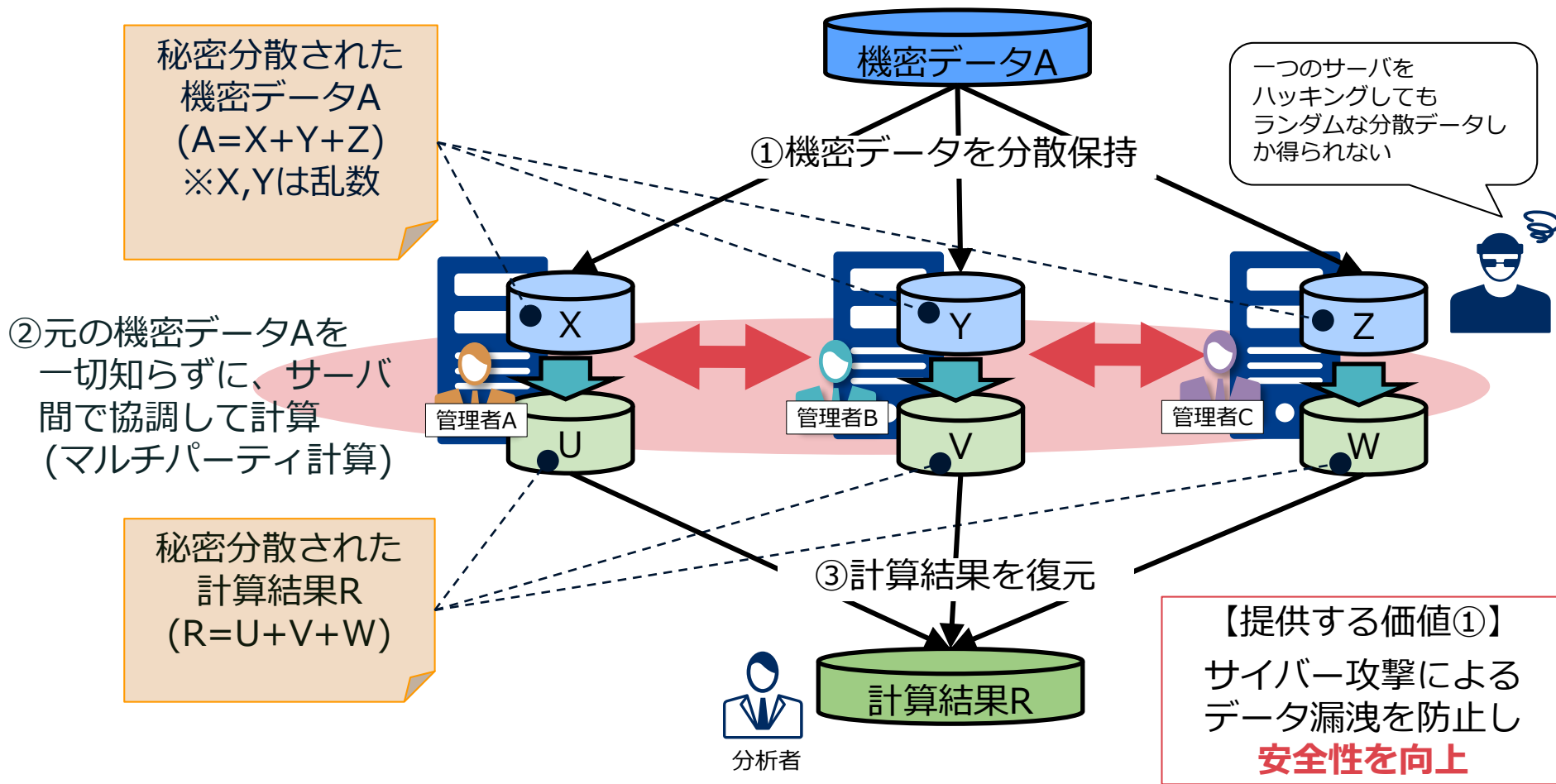
- 秘密計算技術とは、データを秘匿したまま処理できる技術
- 異なる組織のデータを、組織外に元データを一切開示せずに、データを結合した分析が可能



【ご参考】 秘密計算技術の説明(1/2)：概要

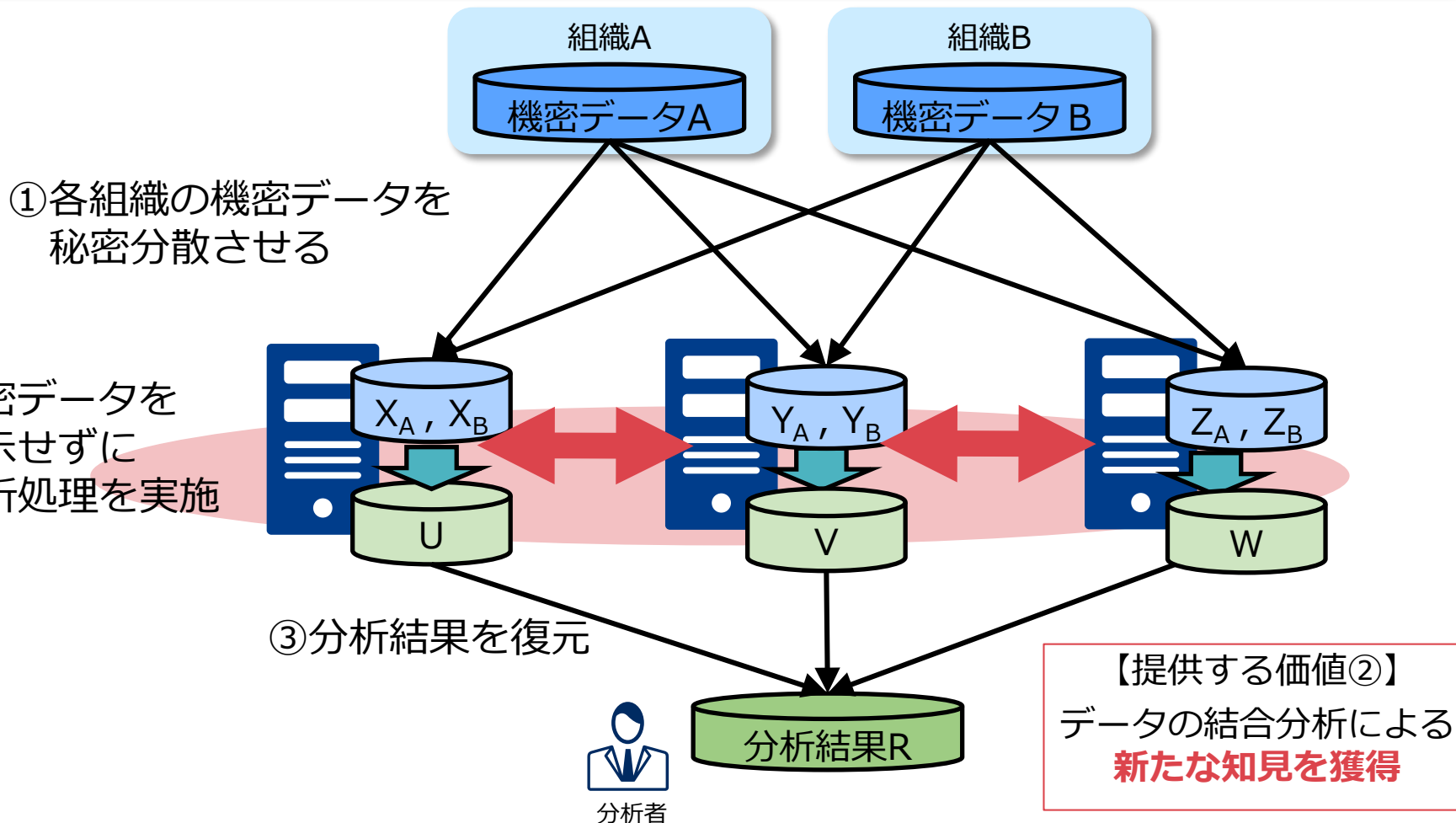
秘密計算技術（秘密分散型のマルチパーティ計算技術）によって、複数のサーバが秘匿したデータを分散保持し、**秘匿したままの任意の計算**を実行可能※1

※1 計算を論理回路として表現することで、理論上は任意の計算が可能



【ご参考】 秘密計算技術の説明(2/2)：組織間データ結合

異なる組織のデータを秘密分散したまま処理することで、組織外に元データを一切開示せずに、データを結合した分析結果を得る事が可能



秘密計算技術の特徴

秘密計算技術は情報を秘匿したまま処理できる技術であり、以下の価値を提供可能

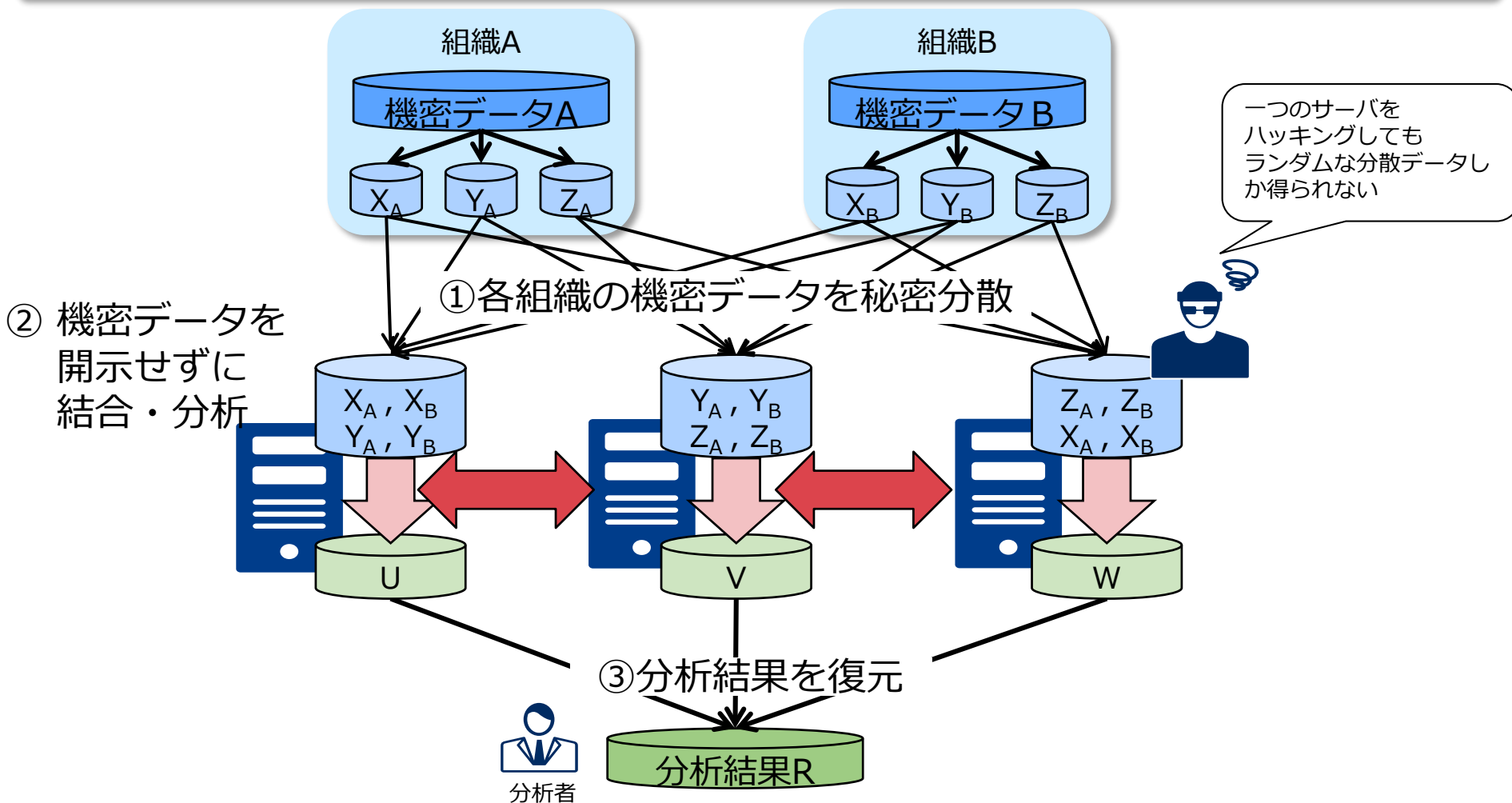
- ①サイバー攻撃によるデータ漏洩を防止し**安全性を向上**
- ②データの結合分析による**新たな知見を獲得**

制限事項

- 秘密計算技術では多くの通信を行うため、通常の処理よりも数10～数1000倍程度遅くなる
 - ・※補足：処理速度は、秘密計算を行うサーバの処理性能とネットワーク速度に依存します。

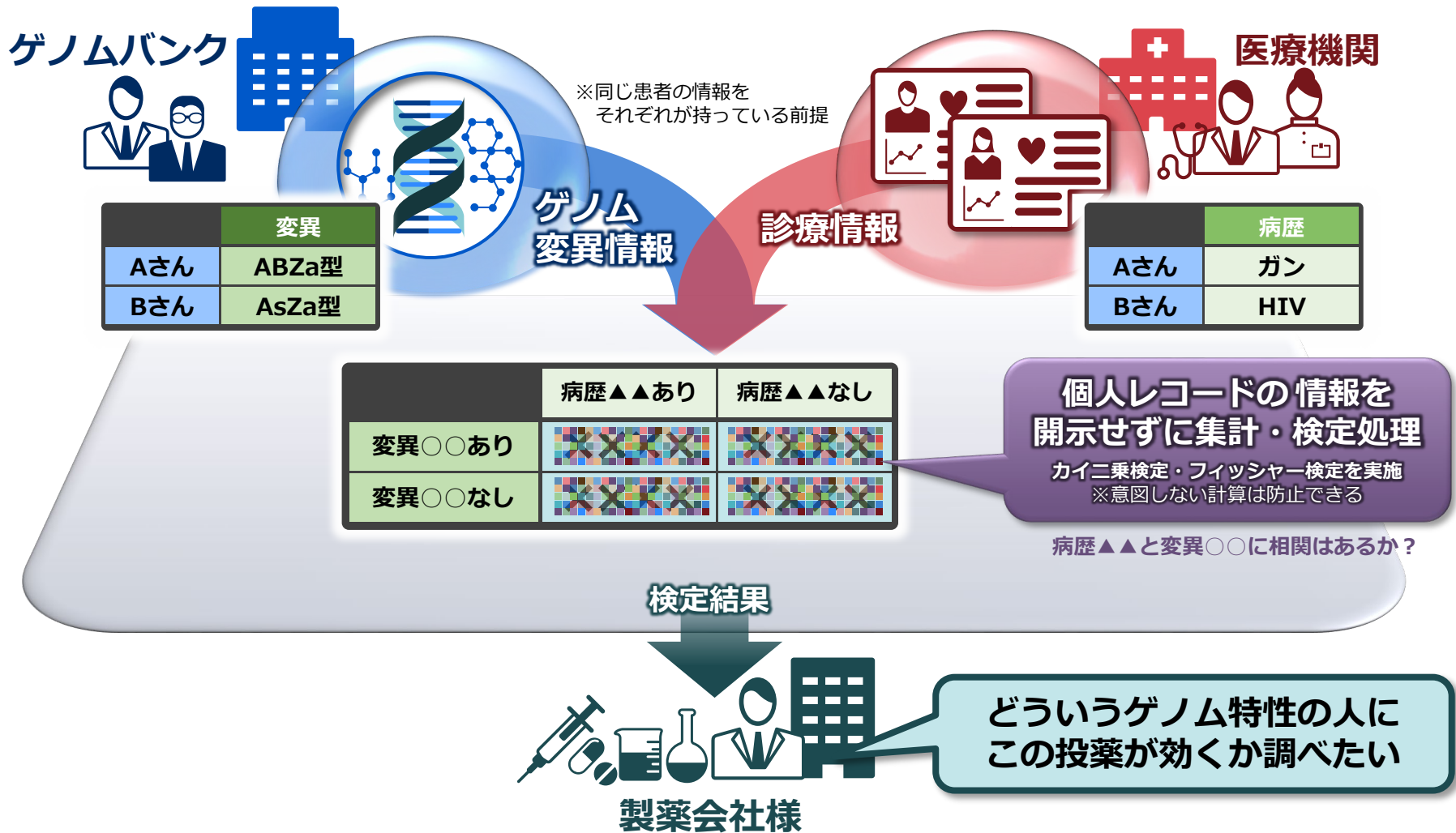
【ご参考】 NECの秘密計算技術

NECは高速化のため、3台のサーバに秘密分散した値のうち、2つの秘密分散値から元データを復元できる方式を採用



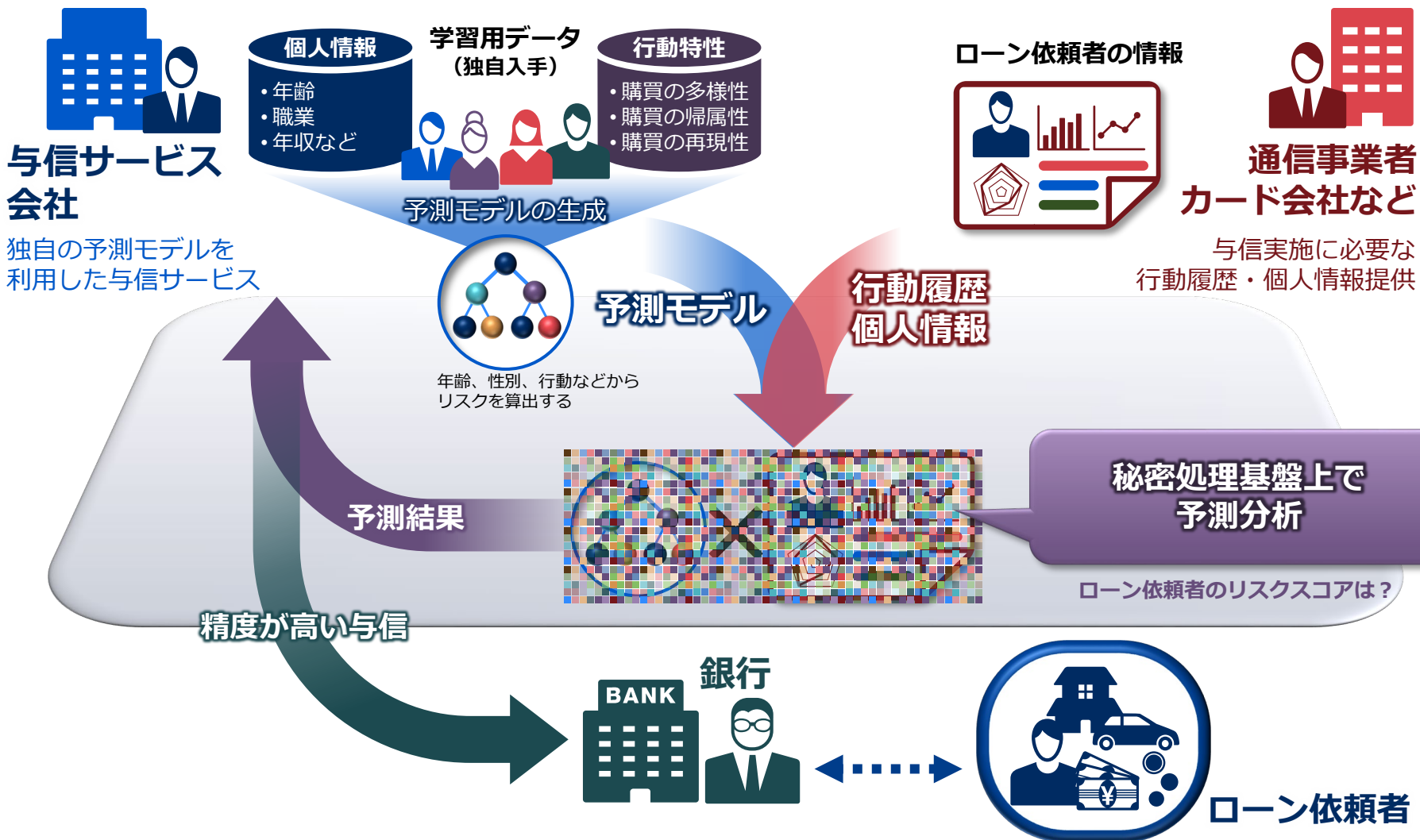
3. ユースケースについて

ゲノム特性に応じた薬の開発・オーダーメイド医療を後押し



ユースケース例：【金融分野】与信審査高精度化、貸し倒れリスクの予測分析

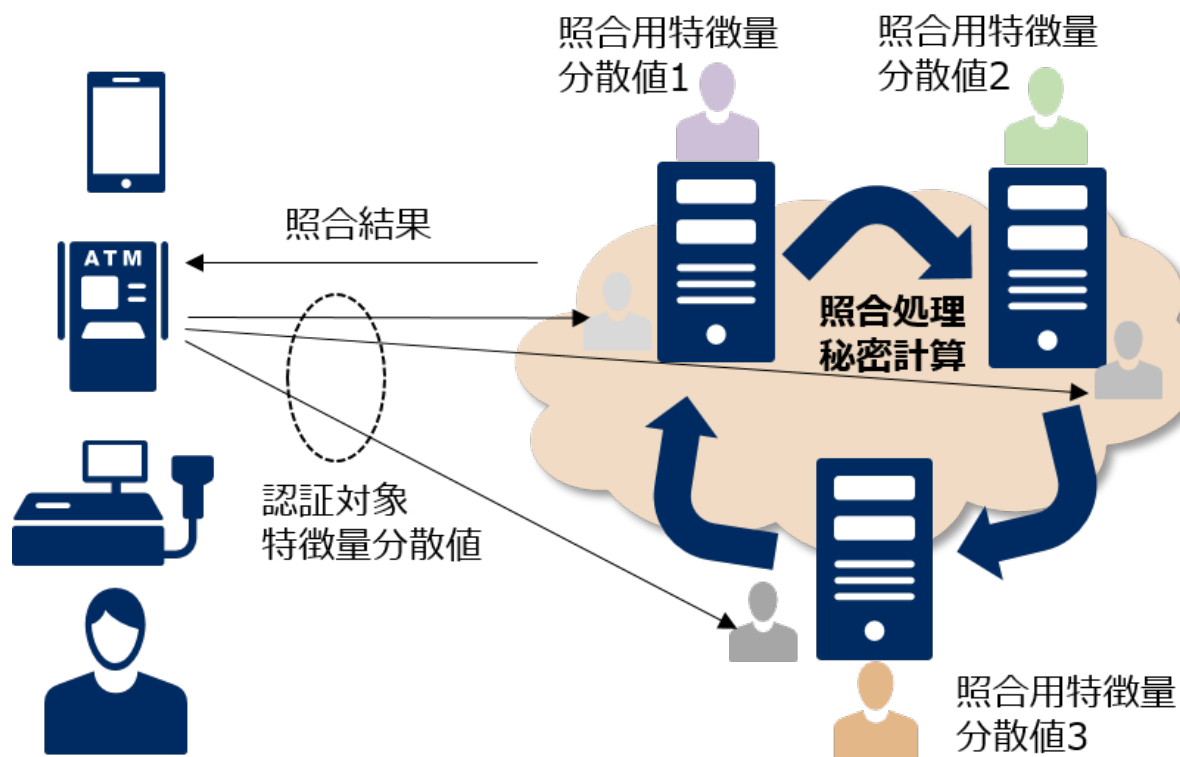
企業が保有するデータや独自の予測モデルを安全に活用した新事業を創出



ユースケース例：（顔生体認証）顔生体認証情報のクラウド上の安全管理

顔テンプレート情報を隠蔽しながらの、生体認証を実現することで、クラウド上での安全な生体情報の管理を実現。

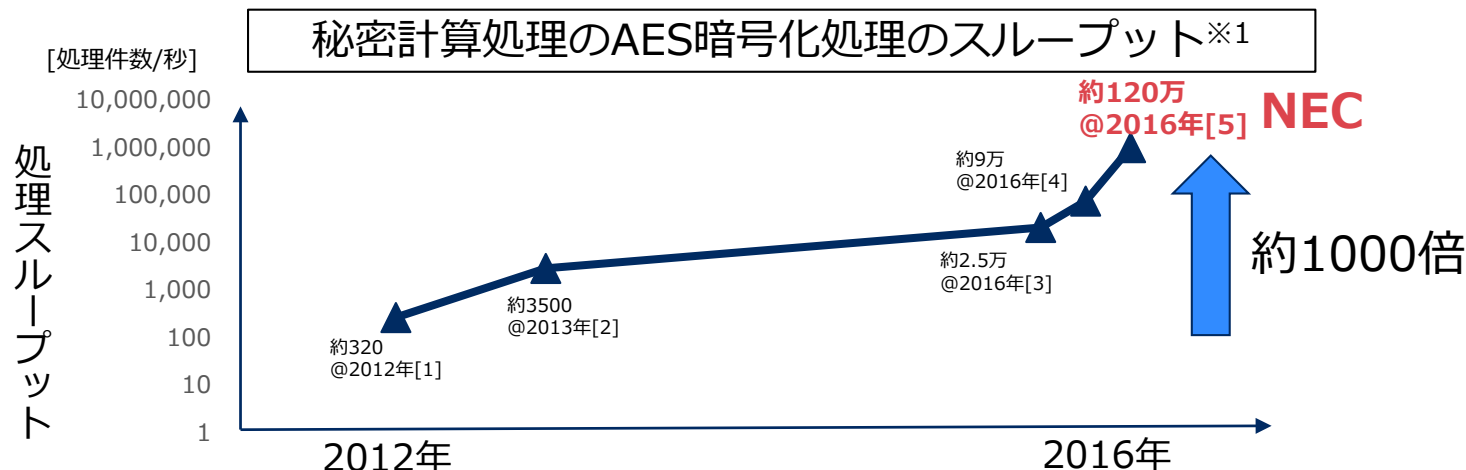
→ 物理的な保護が困難な環境での顔認証システムが導入可能となる



4. NECの秘密計算技術

NECは高速な秘密計算を実現し、一部処理で実用的な性能を達成

秘密計算は近年桁違いに高速化を実現（2012年以降で約1000倍）



例：顔特徴量の照合処理やDNA編集距離計算で、現実的な性能を達成※2

- 顔特徴量の照合処理：1000次元の特徴量の照合が約45,000件/秒※3
- DNA編集距離の計算処理：長さ100のDNA配列同士の距離計算：約145/秒

国際的に高い評価を得て、難関国際学会に採択。
CCS2016(Best Paper)、Eurocrypt2017、S&P2017、CCS2018

[1] J. Launchbury, I.S. Diatchki, T. DuBuisson and A. Adams-Moran. "Efficient lookup-table protocol in secure multiparty computation". ACM ICFP2012.

[2] S. Laur, R. Talviste and J. Willemson. "From Oblivious AES to Efficient 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 semi-honest安全な3パーティの秘密計算と比較。[5]論文のTable.1を参考にグラフ化

※2 詳細は「土田 他, "不正検知可能なマルチパーティ計算による生体情報と遺伝子情報の保護", SCIS2018」を参照

※3 VISAのピーク時トランザクション数：秒間47,000件

NECの秘密計算の開発支援ツール

NECが開発した秘密計算の「開発支援ツール」によって、一般のエンジニアでも容易に秘密計算のアプリケーション構築が可能

Before:

秘密計算の回路（論理回路、
算術回路）を設計し、記述



秘密計算の専門家

```
...  
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  
...
```

数万行

After:

Pythonに似た
プログラミング言語で
処理を記載



一般的なSE

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

数10～100行

※例：単純な集計処理の場合

開発支援ツール
("コンパイラ")

```
...  
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  
...
```

数万行 ※最適化処理によって、
処理の行数の削減も可能

記述の例：Pythonに似たプログラミング言語で記載

平均を計算する処理のProgram例

関数定義

```
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
```

sint: "secret integer"型
秘密にしたい整数

sfix: "secret fixed-point number"型
秘密にしたい固定小数点数

秘密にしたい固定小数点数との割り算

main部分

```
# === ここから main ===  
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' % result.reveal())
```

秘密にしたいsint型の配列

各配列に秘密のデータを格納

mean関数呼び出し

resultを復元(reveal)し、
結果を標準出力にprint。
(デバッグのため)

参考：単純なSQL程度の統計処理への対応例(1/2)

SQLで実行できる程度の単純な統計処理であれば、コード生成支援ツールを用いて、そのSQLに相当する処理を記載し、ある程度容易に実装が可能

● SQL処理の例

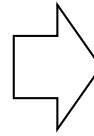
- `SELECT count(*), avg(mortgage debt), stddev(mortgage debt) FROM Census GROUP BY Income bucket, State;`
- `SELECT count(*), avg(credit limit) as avg, stddev(credit limit) as dev FROM Census Where State==Utah and Overall mortgage debt >100 GROUP BY Age, Sex Having 1.3 avg + 1.2 dev > avg(monthly housing cost) ;`

● データベース (table名 : 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

参考：単純なSQL程度の統計処理への対応例(2/2)

SQLの処理と同等な処理を行う アルゴリズムを検討



プログラムを記述

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 $[D_{\mu\nu}]^n$.

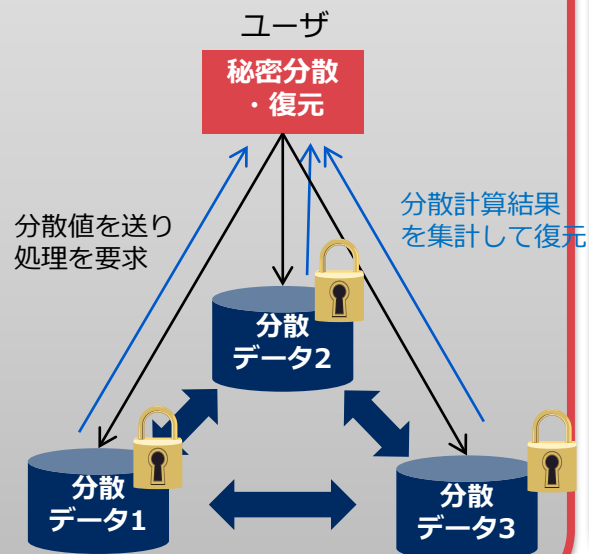
```
mpc/sql_3_py
1 #Based on "Algorithm 173 SQL Query"
2
3 #-----#
4 L = (20, 20, 20, 20, 20, 83, 20, 2, 10, 50)
5 a = 5 # 5 <= a <= 14
6 v = 0 # 0 <= v <= 4
7 N = 1000
8
9 #-----#
10 def read_row():
11     #
12     #
13     #
14     #
15     #
16     #
17     #
18     #
19     #
20     #
21     #
22     #
23     #
24     #
25     #
26     #
27     #
28     #
29     #
30     #
31     #
32     #
33     #
34     #
35     #
36     #
37     #
38     #
39     #
40     #
41     #
42     #
43     #
44     #
45     #
46     #
47     #
48     #
49     #
50     #
51     #
52     #
53     #
54     #
55     #
56     #
57     #
58     #
59     #
60     #
61     #
62     #
63     #
64     #
65     #
66     #
67     #
68     #
69     #
70     #
71     #
72     #
73     #
74     #
75     #
76     #
77     #
78     #
79     #
80     #
81     #
82     #
83     #
84     #
85     #
86     #
87     #
88     #
89     #
90     #
91     #
92     #
93     #
94     #
95     #
96     #
97     #
98     #
99     #
100    #
101    #
102    #
103    #
104    #
105    #
106    #
107    #
108    #
109    #
110    #
111    #
112    #
113    #
114    #
115    #
116    #
117    #
118    #
119    #
120    #
121    #
122    #
123    #
124    #
125    #
126    #
127    #
128    #
129    #
130    #
131    #
132    #
133    #
134    #
135    #
136    #
137    #
138    #
139    #
140    #
141    #
142    #
143    #
144    #
145    #
146    #
147    #
148    #
149    #
150    #
151    #
152    #
153    #
154    #
155    #
156    #
157    #
158    #
159    #
160    #
161    #
162    #
163    #
164    #
165    #
166    #
167    #
168    #
169    #
170    #
171    #
172    #
173    #
174    #
175    #
176    #
177    #
178    #
179    #
180    #
181    #
182    #
183    #
184    #
185    #
186    #
187    #
188    #
189    #
190    #
191    #
192    #
193    #
194    #
195    #
196    #
197    #
198    #
199    #
200    #
201    #
202    #
203    #
204    #
205    #
206    #
207    #
208    #
209    #
210    #
211    #
212    #
213    #
214    #
215    #
216    #
217    #
218    #
219    #
220    #
221    #
222    #
223    #
224    #
225    #
226    #
227    #
228    #
229    #
230    #
231    #
232    #
233    #
234    #
235    #
236    #
237    #
238    #
239    #
240    #
241    #
242    #
243    #
244    #
245    #
246    #
247    #
248    #
249    #
250    #
251    #
252    #
253    #
254    #
255    #
256    #
257    #
258    #
259    #
260    #
261    #
262    #
263    #
264    #
265    #
266    #
267    #
268    #
269    #
270    #
271    #
272    #
273    #
274    #
275    #
276    #
277    #
278    #
279    #
280    #
281    #
282    #
283    #
284    #
285    #
286    #
287    #
288    #
289    #
290    #
291    #
292    #
293    #
294    #
295    #
296    #
297    #
298    #
299    #
300    #
301    #
302    #
303    #
304    #
305    #
306    #
307    #
308    #
309    #
310    #
311    #
312    #
313    #
314    #
315    #
316    #
317    #
318    #
319    #
320    #
321    #
322    #
323    #
324    #
325    #
326    #
327    #
328    #
329    #
330    #
331    #
332    #
333    #
334    #
335    #
336    #
337    #
338    #
339    #
340    #
341    #
342    #
343    #
344    #
345    #
346    #
347    #
348    #
349    #
350    #
351    #
352    #
353    #
354    #
355    #
356    #
357    #
358    #
359    #
360    #
361    #
362    #
363    #
364    #
365    #
366    #
367    #
368    #
369    #
370    #
371    #
372    #
373    #
374    #
375    #
376    #
377    #
378    #
379    #
380    #
381    #
382    #
383    #
384    #
385    #
386    #
387    #
388    #
389    #
390    #
391    #
392    #
393    #
394    #
395    #
396    #
397    #
398    #
399    #
400    #
401    #
402    #
403    #
404    #
405    #
406    #
407    #
408    #
409    #
410    #
411    #
412    #
413    #
414    #
415    #
416    #
417    #
418    #
419    #
420    #
421    #
422    #
423    #
424    #
425    #
426    #
427    #
428    #
429    #
430    #
431    #
432    #
433    #
434    #
435    #
436    #
437    #
438    #
439    #
440    #
441    #
442    #
443    #
444    #
445    #
446    #
447    #
448    #
449    #
450    #
451    #
452    #
453    #
454    #
455    #
456    #
457    #
458    #
459    #
460    #
461    #
462    #
463    #
464    #
465    #
466    #
467    #
468    #
469    #
470    #
471    #
472    #
473    #
474    #
475    #
476    #
477    #
478    #
479    #
480    #
481    #
482    #
483    #
484    #
485    #
486    #
487    #
488    #
489    #
490    #
491    #
492    #
493    #
494    #
495    #
496    #
497    #
498    #
499    #
500    #
501    #
502    #
503    #
504    #
505    #
506    #
507    #
508    #
509    #
510    #
511    #
512    #
513    #
514    #
515    #
516    #
517    #
518    #
519    #
520    #
521    #
522    #
523    #
524    #
525    #
526    #
527    #
528    #
529    #
530    #
531    #
532    #
533    #
534    #
535    #
536    #
537    #
538    #
539    #
540    #
541    #
542    #
543    #
544    #
545    #
546    #
547    #
548    #
549    #
550    #
551    #
552    #
553    #
554    #
555    #
556    #
557    #
558    #
559    #
560    #
561    #
562    #
563    #
564    #
565    #
566    #
567    #
568    #
569    #
570    #
571    #
572    #
573    #
574    #
575    #
576    #
577    #
578    #
579    #
580    #
581    #
582    #
583    #
584    #
585    #
586    #
587    #
588    #
589    #
590    #
591    #
592    #
593    #
594    #
595    #
596    #
597    #
598    #
599    #
600    #
601    #
602    #
603    #
604    #
605    #
606    #
607    #
608    #
609    #
610    #
611    #
612    #
613    #
614    #
615    #
616    #
617    #
618    #
619    #
620    #
621    #
622    #
623    #
624    #
625    #
626    #
627    #
628    #
629    #
630    #
631    #
632    #
633    #
634    #
635    #
636    #
637    #
638    #
639    #
640    #
641    #
642    #
643    #
644    #
645    #
646    #
647    #
648    #
649    #
650    #
651    #
652    #
653    #
654    #
655    #
656    #
657    #
658    #
659    #
660    #
661    #
662    #
663    #
664    #
665    #
666    #
667    #
668    #
669    #
670    #
671    #
672    #
673    #
674    #
675    #
676    #
677    #
678    #
679    #
680    #
681    #
682    #
683    #
684    #
685    #
686    #
687    #
688    #
689    #
690    #
691    #
692    #
693    #
694    #
695    #
696    #
697    #
698    #
699    #
700    #
701    #
702    #
703    #
704    #
705    #
706    #
707    #
708    #
709    #
710    #
711    #
712    #
713    #
714    #
715    #
716    #
717    #
718    #
719    #
720    #
721    #
722    #
723    #
724    #
725    #
726    #
727    #
728    #
729    #
730    #
731    #
732    #
733    #
734    #
735    #
736    #
737    #
738    #
739    #
740    #
741    #
742    #
743    #
744    #
745    #
746    #
747    #
748    #
749    #
750    #
751    #
752    #
753    #
754    #
755    #
756    #
757    #
758    #
759    #
760    #
761    #
762    #
763    #
764    #
765    #
766    #
767    #
768    #
769    #
770    #
771    #
772    #
773    #
774    #
775    #
776    #
777    #
778    #
779    #
780    #
781    #
782    #
783    #
784    #
785    #
786    #
787    #
788    #
789    #
790    #
791    #
792    #
793    #
794    #
795    #
796    #
797    #
798    #
799    #
800    #
801    #
802    #
803    #
804    #
805    #
806    #
807    #
808    #
809    #
810    #
811    #
812    #
813    #
814    #
815    #
816    #
817    #
818    #
819    #
820    #
821    #
822    #
823    #
824    #
825    #
826    #
827    #
828    #
829    #
830    #
831    #
832    #
833    #
834    #
835    #
836    #
837    #
838    #
839    #
840    #
841    #
842    #
843    #
844    #
845    #
846    #
847    #
848    #
849    #
850    #
851    #
852    #
853    #
854    #
855    #
856    #
857    #
858    #
859    #
860    #
861    #
862    #
863    #
864    #
865    #
866    #
867    #
868    #
869    #
870    #
871    #
872    #
873    #
874    #
875    #
876    #
877    #
878    #
879    #
880    #
881    #
882    #
883    #
884    #
885    #
886    #
887    #
888    #
889    #
890    #
891    #
892    #
893    #
894    #
895    #
896    #
897    #
898    #
899    #
900    #
901    #
902    #
903    #
904    #
905    #
906    #
907    #
908    #
909    #
910    #
911    #
912    #
913    #
914    #
915    #
916    #
917    #
918    #
919    #
920    #
921    #
922    #
923    #
924    #
925    #
926    #
927    #
928    #
929    #
930    #
931    #
932    #
933    #
934    #
935    #
936    #
937    #
938    #
939    #
940    #
941    #
942    #
943    #
944    #
945    #
946    #
947    #
948    #
949    #
950    #
951    #
952    #
953    #
954    #
955    #
956    #
957    #
958    #
959    #
960    #
961    #
962    #
963    #
964    #
965    #
966    #
967    #
968    #
969    #
970    #
971    #
972    #
973    #
974    #
975    #
976    #
977    #
978    #
979    #
980    #
981    #
982    #
983    #
984    #
985    #
986    #
987    #
988    #
989    #
990    #
991    #
992    #
993    #
994    #
995    #
996    #
997    #
998    #
999    #
1000   #
1001   #
1002   #
1003   #
1004   #
1005   #
1006   #
1007   #
1008   #
1009   #
1010   #
1011   #
1012   #
1013   #
1014   #
1015   #
1016   #
1017   #
1018   #
1019   #
1020   #
1021   #
1022   #
1023   #
1024   #
1025   #
1026   #
1027   #
1028   #
1029   #
1030   #
1031   #
1032   #
1033   #
1034   #
1035   #
1036   #
1037   #
1038   #
1039   #
1040   #
1041   #
1042   #
1043   #
1044   #
1045   #
1046   #
1047   #
1048   #
1049   #
1050   #
1051   #
1052   #
1053   #
1054   #
1055   #
1056   #
1057   #
1058   #
1059   #
1060   #
1061   #
1062   #
1063   #
1064   #
1065   #
1066   #
1067   #
1068   #
1069   #
1070   #
1071   #
1072   #
1073   #
1074   #
1075   #
1076   #
1077   #
1078   #
1079   #
1080   #
1081   #
1082   #
1083   #
1084   #
1085   #
1086   #
1087   #
1088   #
1089   #
1090   #
1091   #
1092   #
1093   #
1094   #
1095   #
1096   #
1097   #
1098   #
1099   #
1100   #
1101   #
1102   #
1103   #
1104   #
1105   #
1106   #
1107   #
1108   #
1109   #
1110   #
1111   #
1112   #
1113   #
1114   #
1115   #
1116   #
1117   #
1118   #
1119   #
1120   #
1121   #
1122   #
1123   #
1124   #
1125   #
1126   #
1127   #
1128   #
1129   #
1130   #
1131   #
1132   #
1133   #
1134   #
1135   #
1136   #
1137   #
1138   #
1139   #
1140   #
1141   #
1142   #
1143   #
1144   #
1145   #
1146   #
1147   #
1148   #
1149   #
1150   #
1151   #
1152   #
1153   #
1154   #
1155   #
1156   #
1157   #
1158   #
1159   #
1160   #
1161   #
1162   #
1163   #
1164   #
1165   #
1166   #
1167   #
1168   #
1169   #
1170   #
1171   #
1172   #
1173   #
1174   #
1175   #
1176   #
1177   #
1178   #
1179   #
1180   #
1181   #
1182   #
1183   #
1184   #
1185   #
1186   #
1187   #
1188   #
1189   #
1190   #
1191   #
1192   #
1193   #
1194   #
1195   #
1196   #
1197   #
1198   #
1199   #
1200   #
1201   #
1202   #
1203   #
1204   #
1205   #
1206   #
1207   #
1208   #
1209   #
1210   #
1211   #
1212   #
1213   #
1214   #
1215   #
1216   #
1217   #
1218   #
1219   #
1220   #
1221   #
1222   #
1223   #
1224   #
1225   #
1226   #
1227   #
1228   #
1229   #
1230   #
1231   #
1232   #
1233   #
1234   #
1235   #
1236   #
1237   #
1238   #
1239   #
1240   #
1241   #
1242   #
1243   #
1244   #
1245   #
1246   #
1247   #
1248   #
1249   #
1250   #
1251   #
1252   #
1253   #
1254   #
1255   #
1256   #
1257   #
1258   #
1259   #
1260   #
1261   #
1262   #
1263   #
1264   #
1265   #
1266   #
1267   #
1268   #
1269   #
1270   #
1271   #
1272   #
1273   #
1274   #
1275   #
1276   #
1277   #
1278   #
1279   #
1280   #
1281   #
1282   #
1283   #
1284   #
1285   #
1286   #
1287   #
1288   #
1289   #
1290   #
1291   #
1292   #
1293   #
1294   #
1295   #
1296   #
1297   #
1298   #
1299   #
1300   #
1301   #
1302   #
1303   #
1304   #
1305   #
1306   #
1307   #
1308   #
1309   #
1310   #
1311   #
1312   #
1313   #
1314   #
1315   #
1316   #
1317   #
1318   #
1319   #
1320   #
1321   #
1322   #
1323   #
1324   #
1325   #
1326   #
1327   #
1328   #
1329   #
1330   #
1331   #
1332   #
1333   #
1334   #
1335   #
1336   #
1337   #
1338   #
1339   #
1340   #
1341   #
1342   #
1343   #
1344   #
1345   #
1346   #
1347   #
1348   #
1349   #
1350   #
1351   #
1352   #
1353   #
1354   #
1355   #
1356   #
1357   #
1358   #
1359   #
1360   #
1361   #
1362   #
1363   #
1364   #
1365   #
1366   #
1367   #
1368   #
1369   #
1370   #
1371   #
1372   #
1373   #
1374   #
1375   #
1376   #
1377   #
1378   #
1379   #
1380   #
1381   #
1382   #
1383   #
1384   #
1385   #
1386   #
1387   #
1388   #
1389   #
1390   #
1391   #
1392   #
1393   #
1394   #
1395   #
1396   #
1397   #
1398   #
1399   #
1400   #
1401   #
1402   #
1403   #
1404   #
1405   #
1406   #
1407   #
1408   #
1409   #
1410   #
1411   #
1412   #
1413   #
1414   #
1415   #
1416   #
1417   #
1418   #
1419   #
1420   #
1421   #
1422   #
1423   #
1424   #
1425   #
1426   #
1427   #
1428   #
1429   #
1430   #
1431   #
1432   #
1433   #
1434   #
1435   #
1436   #
1437   #
1438   #
1439   #
1440   #
1441   #
1442   #
1443   #
1444   #
1445   #
1446   #
1447   #
1448   #
1449   #
1450   #
1451   #
1452   #
1453   #
1454   #
1455   #
1456   #
1457   #
1458   #
1459   #
1460   #
1461   #
1462   #
1463   #
1464   #
1465   #
1466   #
1467   #
1468   #
1469   #
1470   #
1471   #
1472   #
1473   #
1474   #
1475   #
1476   #
1477   #
1478   #
1479   #
1480   #
1481   #
1482   #
1483   #
1484   #
1485   #
1486   #
1487   #
1488   #
1489   #
1490   #
1491   #
1492   #
1493   #
1494   #
1495   #
1496   #
1497   #
1498   #
1499   #
1500   #
1501   #
1502   #
1503   #
1504   #
1505   #
1506   #
1507   #
1508   #
1509   #
1510   #
1511   #
1512   #
1513   #
1514   #
1515   #
1516   #
1517   #
1518   #
1519   #
1520   #
1521   #
1522   #
1523   #
1524   #
1525   #
1526   #
1527   #
1528   #
1529   #
1530   #
1531   #
1532   #
1533   #
1534   #
1535   #
1536   #
1537   #
1538   #
1539   #
1540   #
1541   #
1542   #
1543   #
1544   #
1545   #
1546   #
1547   #
1548   #
1549   #
1550   #
1551   #
1552   #
1553   #
1554   #
1555   #
1556   #
1557   #
1558   #
1559   #
1560   #
1561   #
1562   #
1563   #
1564   #
1565   #
1566   #
1567   #
1568   #
1569   #
1570   #
1571   #
1572   #
1573   #
1574   #
1575   #
1576   #
1577   #
1578   #
1579   #
1580   #
1581   #
1582   #
1583   #
1584   #
1585   #
1586   #
1587   #
1588   #
1589   #
1590   #
1591   #
1592   #
1593   #
1594   #
1595   #
1596   #
1597   #
1598   #
1599   #
1600   #
1601   #
1602   #
1603   #
1604   #
1605   #
1606   #
1607   #
1608   #
1609   #
1610   #
1611   #
1612   #
1613   #
1614   #
1615   #
1616   #
1617   #
1618   #
1619   #
1620   #
1621   #
1622   #
1623   #
1624   #
1625   #
1626   #
1627   #
1628   #
1629   #
1630   #
1631   #
1632   #
1633   #
1634   #
1635   #
1636   #
1637   #
1638   #
1639   #
1640   #
1641   #
1642   #
1643   #
1644   #
1645   #
1646   #
1647   #
1648   #
1649   #
1650   #
1651   #
1652   #
1653   #
1654   #
1655   #
1656   #
1657   #
1658   #
1659   #
1660   #
1661   #
1662   #
1663   #
1664   #
1665   #
1666   #
1667   #
1668   #
1669   #
1670   #
1671  
```

付録

秘密計算

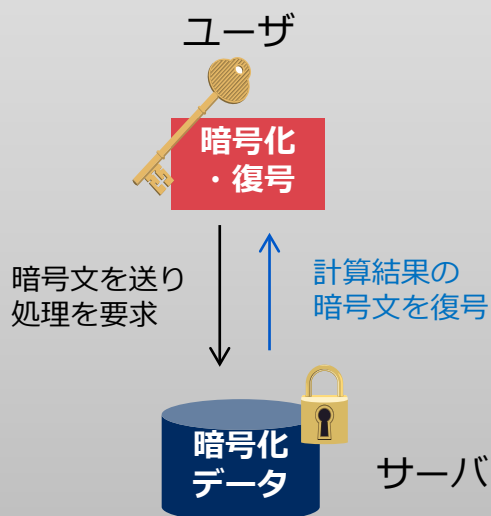
秘密分散を利用した方式

データを秘密分散したまま処理



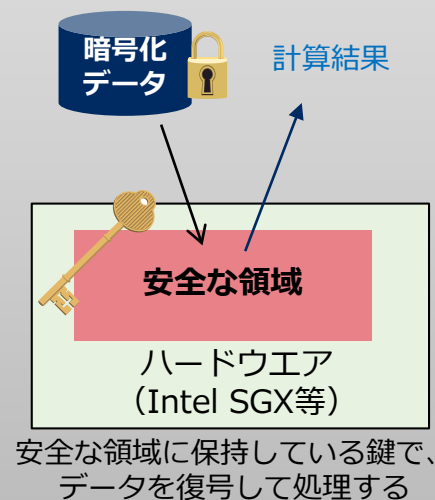
準同型暗号を利用した方式

データを暗号化したまま処理



ハードウェアを利用した方式 (Trusted Execution Environment等)

ハードウェア上の安全な領域で処理



■■■
其他方式も
存在

 **Orchestrating** a brighter world

NEC