C10 プライバシーを保護した疫学調査のための確率検定プロトコル

1BDRM015 佐藤智貴

指導教員 菊池浩明

疫学調査

喫煙者

名前	年齢	喫煙
佐藤智貴	27	有
菊池浩明	32	有
佐久間淳	30	有

死亡者

名前	死因
佐藤智貴	胃がん
田中真二	肺がん
鈴木太郎	肺がん

組織A　　組織B

プライバシーを考慮する必要

プライバシー保護疫学調査

1. 放射線疫学調査(CSEC54)

2. 相対危険度の検定(ICSS2012)

3. 傾向性の検定(SCIS2013)

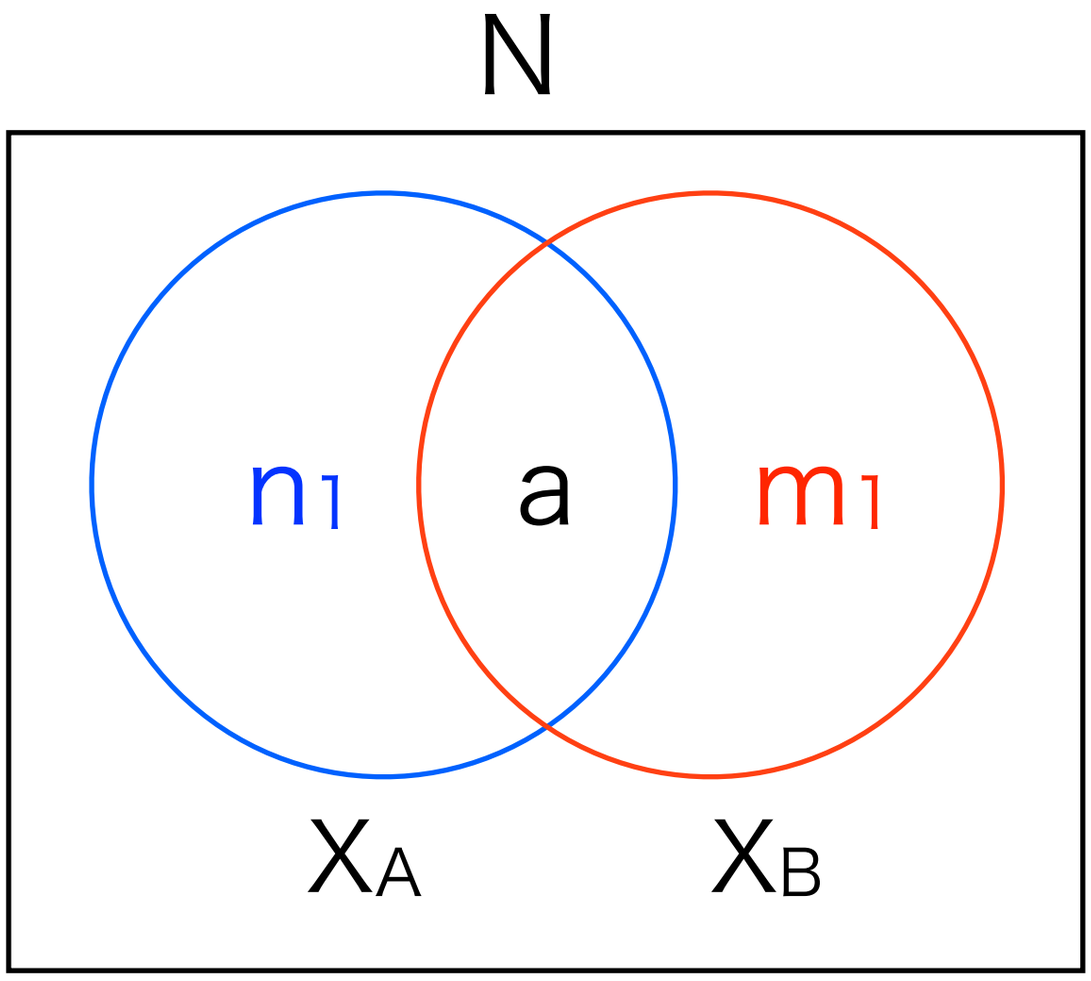
プライバシー保護疫学調査

1. 放射線疫学調査(CSEC54)　中間発表

2. 相対危険度の検定(ICSS2012)　本発表

3. 傾向性の検定(SCIS2013)

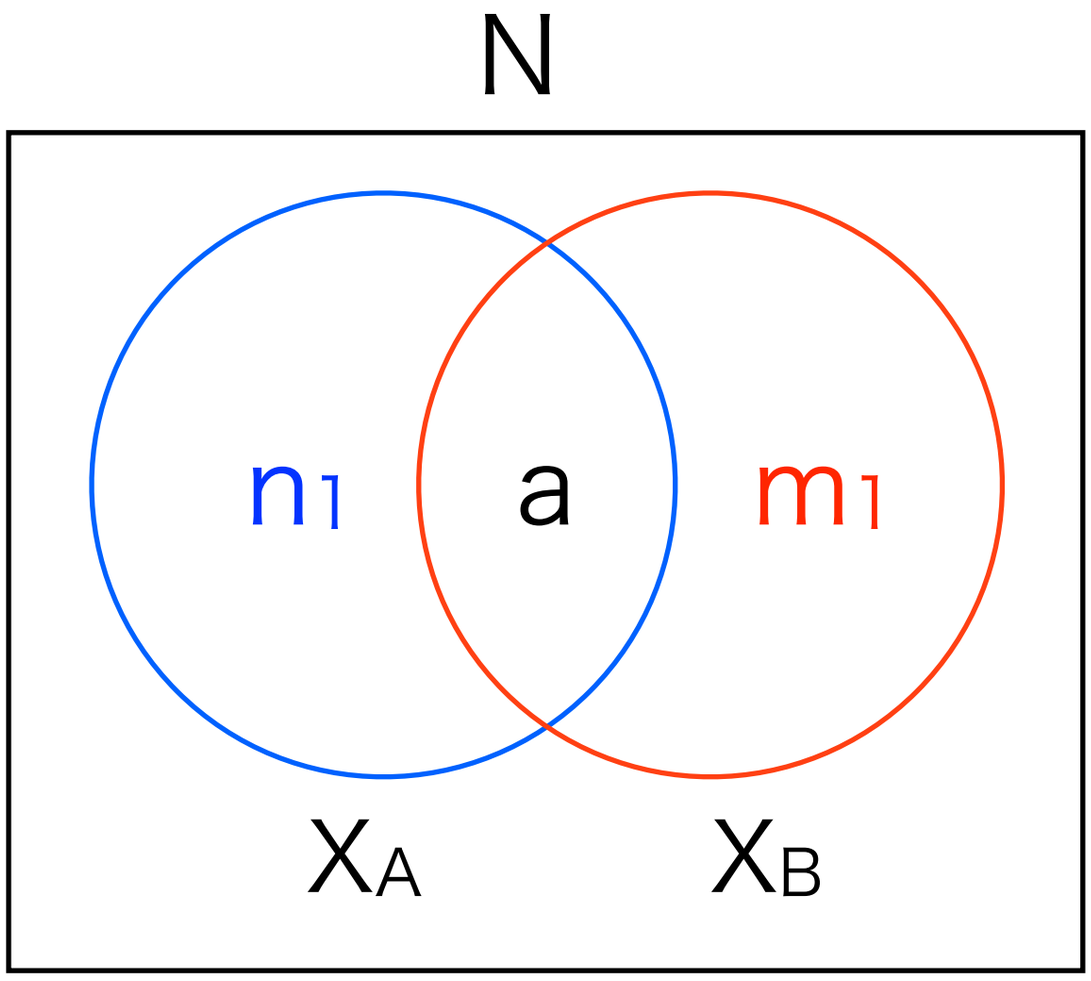
相対危険度RR



	死亡	生存	計
喫煙	a	b	n_1
非喫煙	c	d	n_2
計	m_1	m_2	N

$$RR = \frac{a}{n_1} / \frac{c}{n_2} = \frac{a(c+d)}{(a+b)c} \approx \frac{ad}{bc}$$

相対危険度RR

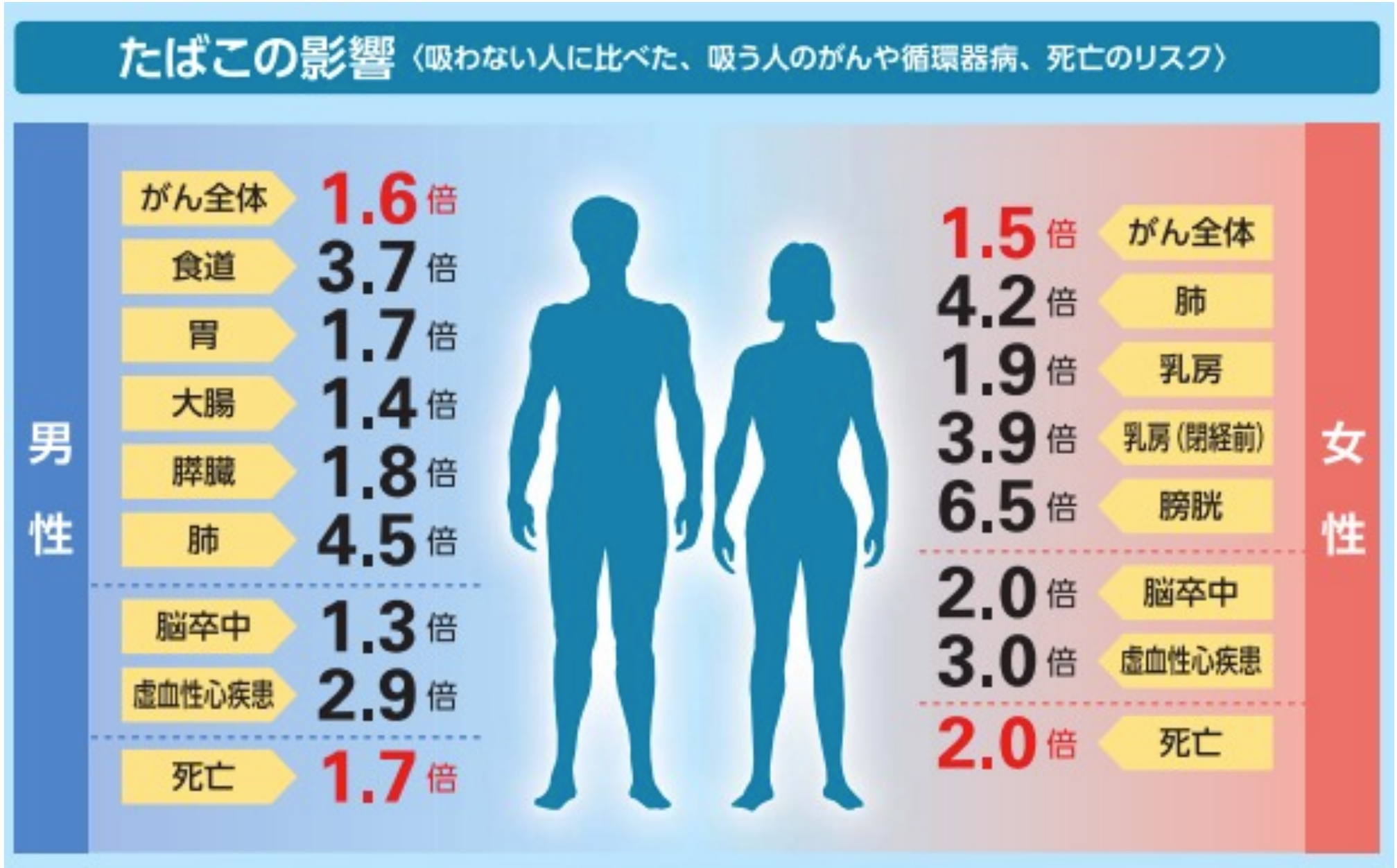


	死亡	生存	計
喫煙	a	b	n_1
非喫煙	c	d	n_2
計	m_1	m_2	N

$$RR = \frac{a}{n_1} / \frac{c}{n_2} = \frac{a(c+d)}{(a+b)c} \approx \frac{ad}{bc}$$

喫煙は非喫煙に比べて何倍危険か

多目的コホート研究*



*国立がん研究センター "多目的コホート研究の成果"より引用
http://epi.ncc.go.jp/jphc/

相対危険度の有意性検定

帰無仮説：RR = 1

対立仮説：RR ≠ 1

$$\chi = \frac{\sqrt{N-1}\{(ad-bc) \pm N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$$

$$\chi > 1.960$$

目的：暗号技術を用いて，二つの組織のデータを秘匿したままRRの有意性を検定する

秘匿内積プロトコル[1]

データを互いに開示することなく内積が可能

普通に照合

喫煙(組織A)	死亡 (組織B)
0	1
1	1
1	1
1	0

$a = 2$

秘匿内積

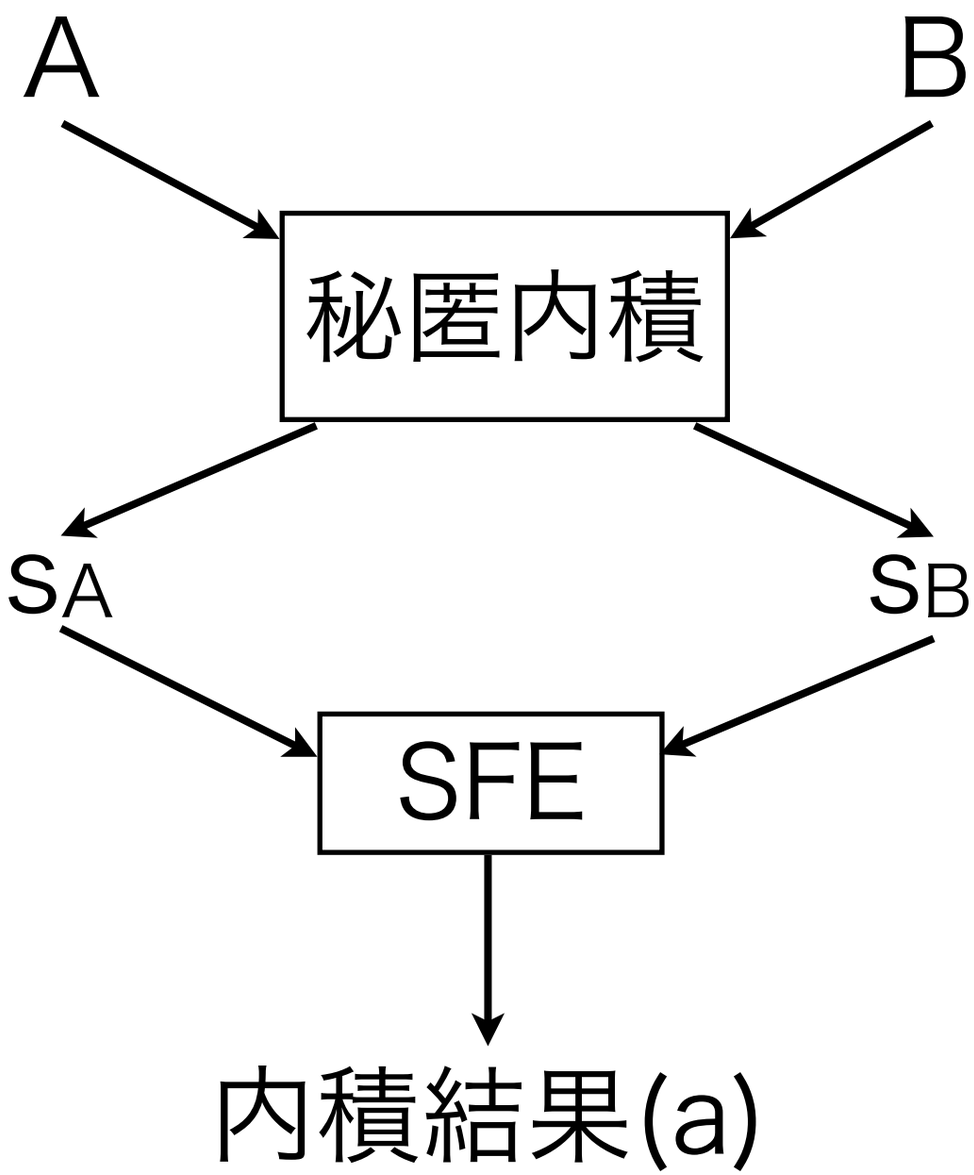
喫煙(組織A)	死亡 (組織B)
E(0)	E(1)
E(1)	E(1)
E(1)	E(1)
E(1)	E(0)

$S_A + S_B = a = 2$

秘密関数計算(SFE)[2]

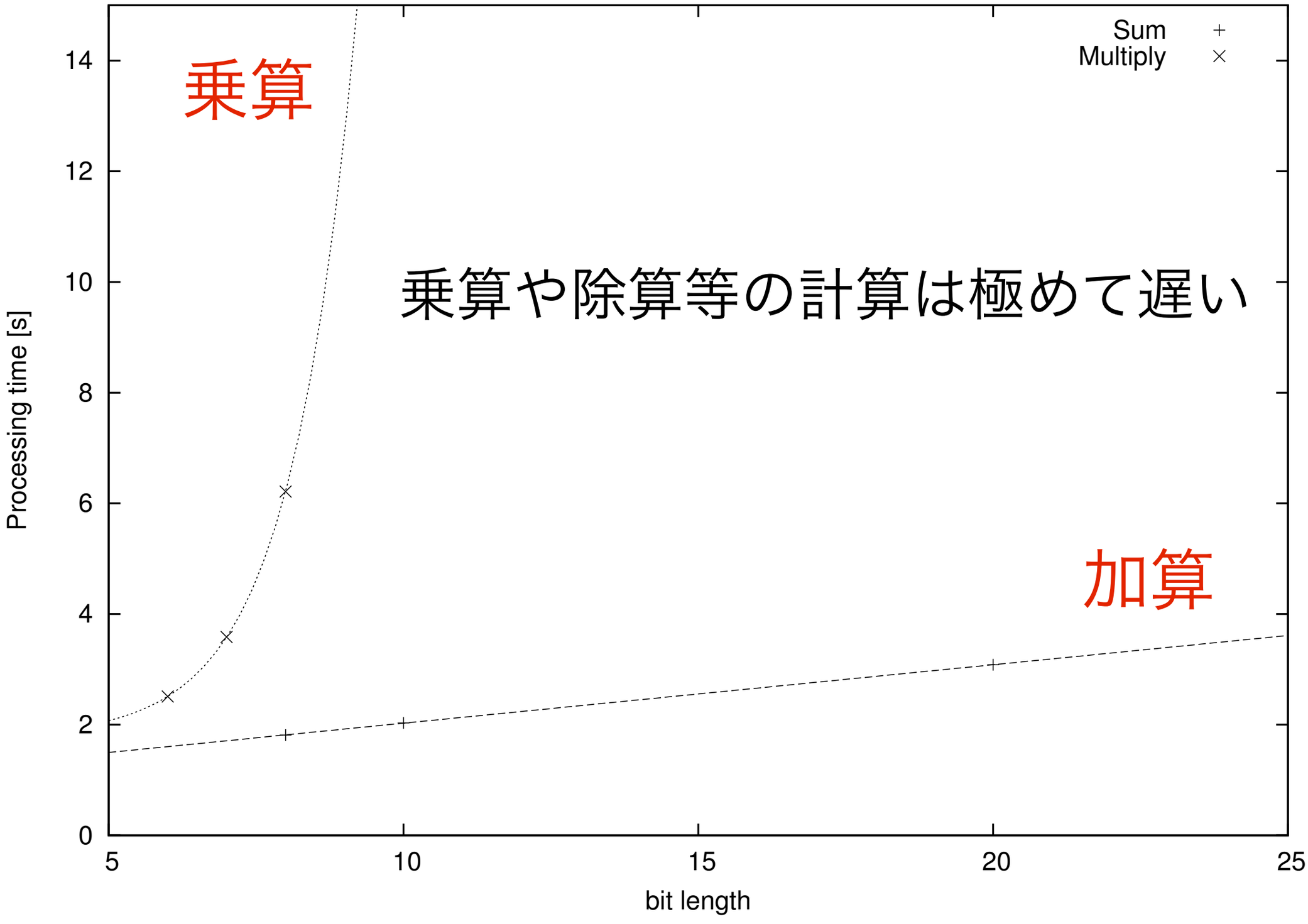
- 秘匿内積プロトコルで得たs_Aとs_Bを開示せずに任意の計算を行う

- $s_A + s_B = a$



	死亡	生存	計
喫煙	a	b	n_1
非喫煙	c	d	n_2
計	m_1	m_2	N

SFE：Fairplayの制約



問題定義

	死亡	生存	計
喫煙	a	b	n_1
非喫煙	c	d	n_2
計	m_1	m_2	N

誰も知らない情報

Aのみが知っている情報

Bのみが知っている情報

- 対象の特定要因の相対危険度が有意か否かを判定
- 出力は判定結果のみ

問題点

1. 大きな計算量

$$\chi = \frac{\sqrt{N-1}\{(ad-bc) \pm N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$$

2. 分散値の定義域の大きさ

- $s_A = a - s_B$

我々の提案

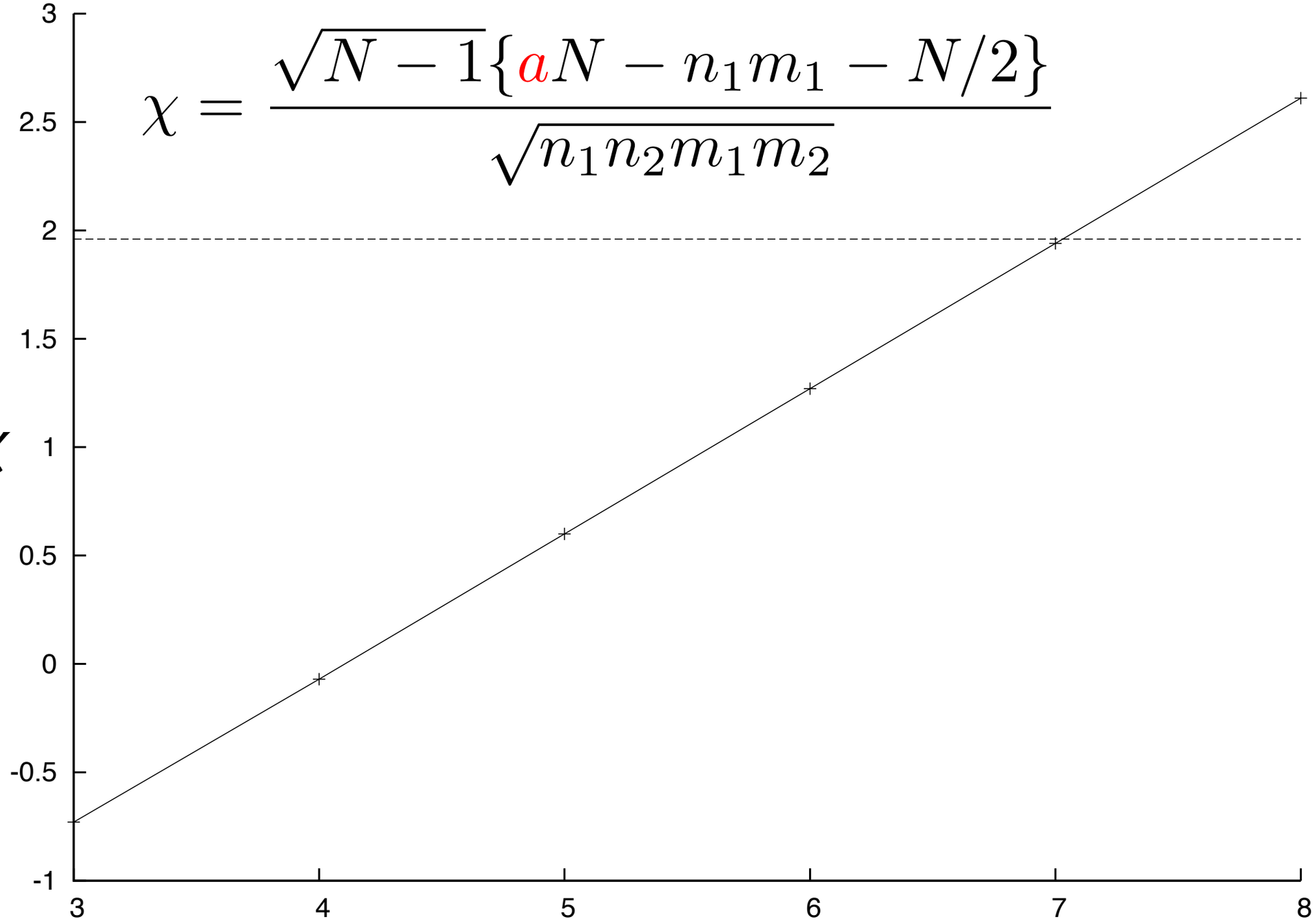
	従来	本提案
問題1	$\frac{\sqrt{N-1}\{(ad-bc) \pm N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$	$(s_A + s_B) > (t_A + t_B) + (u_A + u_B)$
問題2	$s_B \in Z_n$	$s_B \in [0, \mu\text{-}1]$

我々の提案

	従来	本提案
問題1	$\frac{\sqrt{N-1}\{(ad-bc)\pm N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$	$(s_A+s_B)>$ $(t_A+t_B)+(u_A+u_B)$
問題2	$s_B \in Z_n$	$s_B \in [0,\ \mu\text{-}1]$

図2. aを変化させた時の統計量 χ

$$\chi = \frac{\sqrt{N-1}\{aN - n_1 m_1 - N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$$

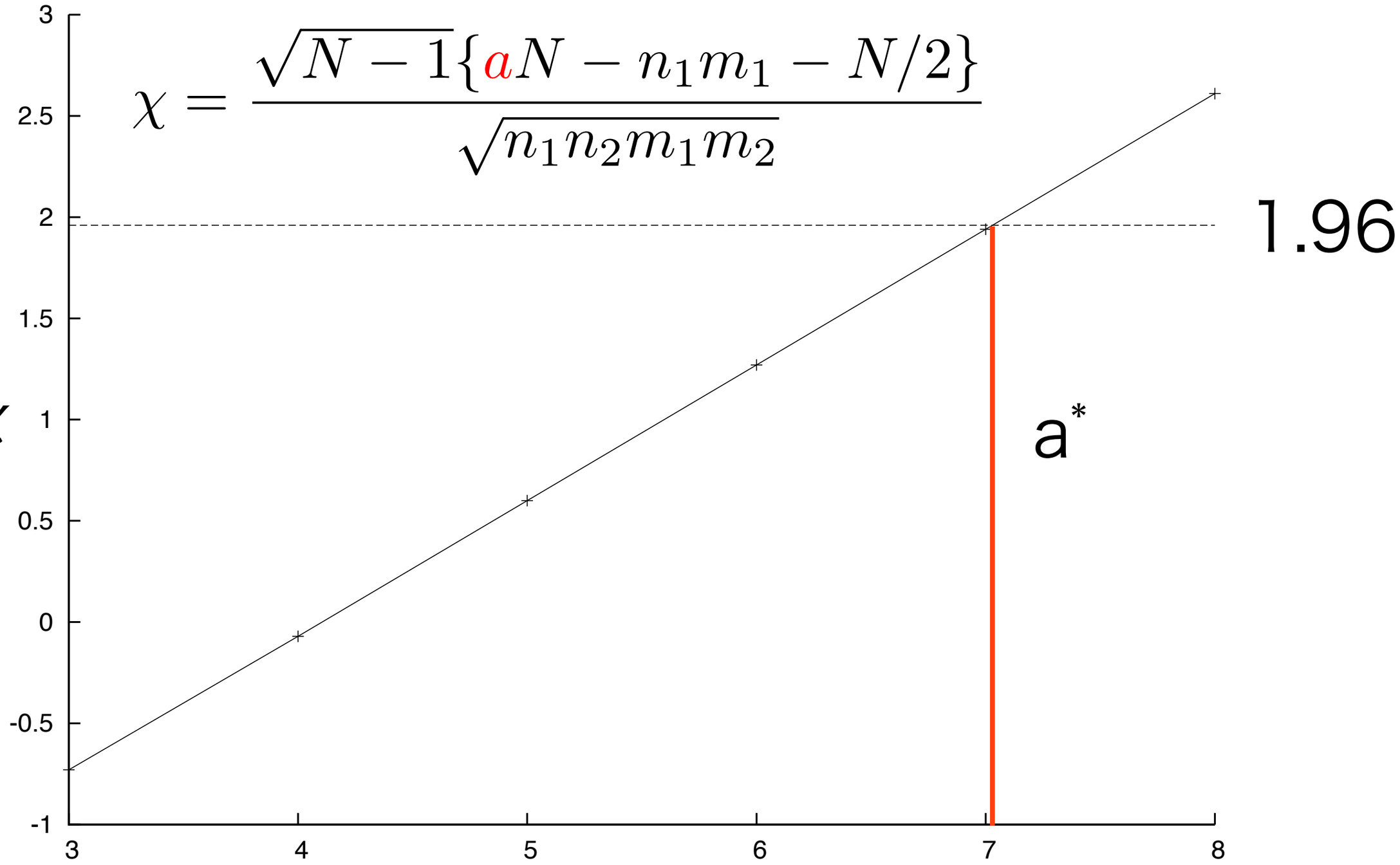


統計量 χ

$|X_A \cap X_B| = a$

図2. aを変化させた時の統計量 χ

$$\chi = \frac{\sqrt{N-1}\{aN - n_1 m_1 - N/2\}}{\sqrt{n_1 n_2 m_1 m_2}}$$



$$|X_A \cap X_B| = a$$

問題1へのアプローチ

$$a^* = \left(\frac{\chi \cdot \sqrt{n_1 n_2 m_1 m_2}}{\sqrt{N-1}} + n_1 m_1 + \frac{N}{2} \right) \cdot \frac{1}{N}$$

$$a^* N = \frac{1.960 \cdot \sqrt{n_1 n_2 m_1 m_2}}{\sqrt{N-1}} + n_1 m_1 + \frac{N}{2}$$

- χ = 1.960の時, $|X_A \cap X_B|$ = a がa^*を上回っているか否かで判定

問題1へのアプローチ

秘匿内積で計算，分散する

$$s_A + s_B = aN = |X_A \cap X_B|N$$

$$t_A + t_B = \left(\frac{\chi\sqrt{n_1 n_2}}{\sqrt{N-1}}\right) \cdot \sqrt{m_1 m_2}$$

$$u_A + u_B = n_1 m_1 + \frac{N}{2}$$

Fairplayで評価を行う

$$aN > \frac{\chi \cdot \sqrt{n_1 n_2 m_1 m_2}}{\sqrt{N-1}} + n_1 m_1 + \frac{N}{2}$$

$$(s_A + s_B) > (t_A + t_B) + (u_A + u_B)$$

問題1へのアプローチ

秘匿内積で計算，分散する

$$s_A + s_B = aN = |X_A \cap X_B|N$$

$$t_A + t_B = \left(\frac{\chi\sqrt{n_1 n_2}}{\sqrt{N-1}}\right) \cdot \sqrt{m_1 m_2}$$

$$u_A + u_B = n_1 m_1 + \frac{N}{2}$$

$$aN > \frac{\chi \cdot \sqrt{n_1 n_2 m_1 m_2}}{\sqrt{N-1}} + n_1 m_1 + \frac{N}{2}$$

Fairplayで評価を行う

$$(s_A + s_B) > (t_A + t_B) + (u_A + u_B)$$

問題1へのアプローチ

秘匿内積で計算，分散する

$$s_A + s_B = aN = |X_A \cap X_B|N$$

$$t_A + t_B = \left(\frac{\chi\sqrt{n_1 n_2}}{\sqrt{N-1}}\right) \cdot \sqrt{m_1 m_2}$$

$$u_A + u_B = n_1 m_1 + \frac{N}{2}$$

$$aN > \frac{\chi \cdot \sqrt{n_1 n_2 m_1 m_2}}{\sqrt{N-1}} + n_1 m_1 + \frac{N}{2}$$

Fairplayで評価を行う

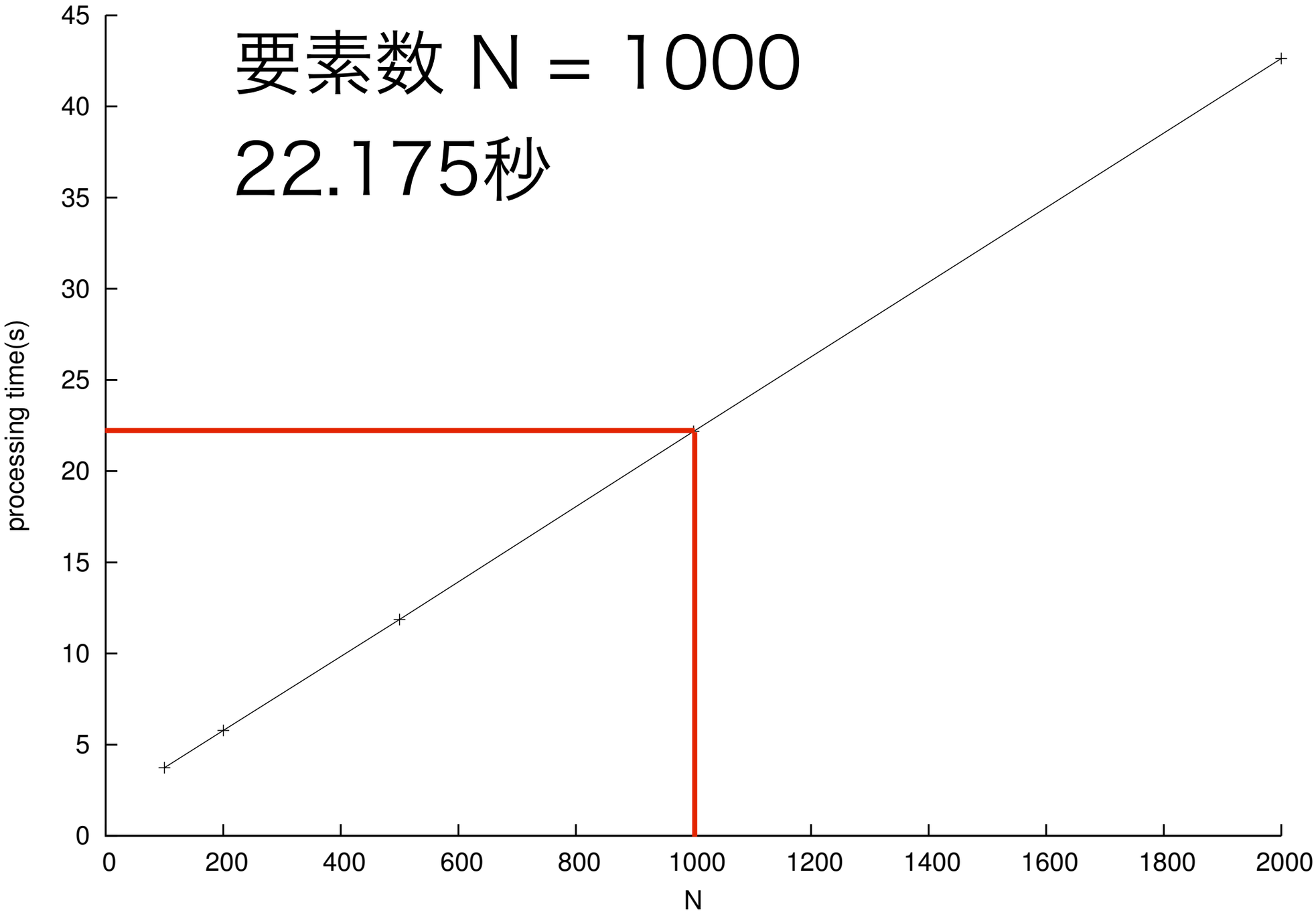
$$(s_A + s_B) > (t_A + t_B) + (u_A + u_B)$$

Fairplayでの計算は加算と比較のみ

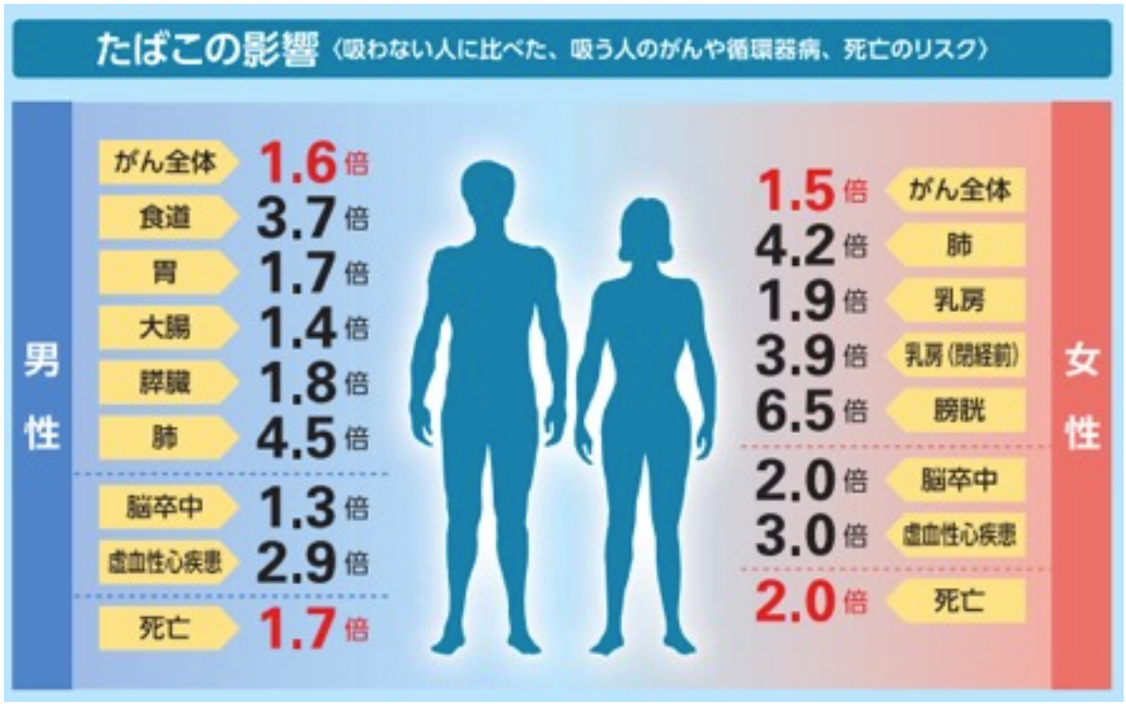
実装したプログラムの処理時間

要素数 N = 1000

22.175秒



実際の調査への適用



多目的コホート研究

140,240名のデータセット

2855.5秒(約48分)

実用的な時間で処理可能

Mac OS X 10.6.8, CPU 2.4GHz

メモリ4GB, Java, BigIntegerクラス

まとめ

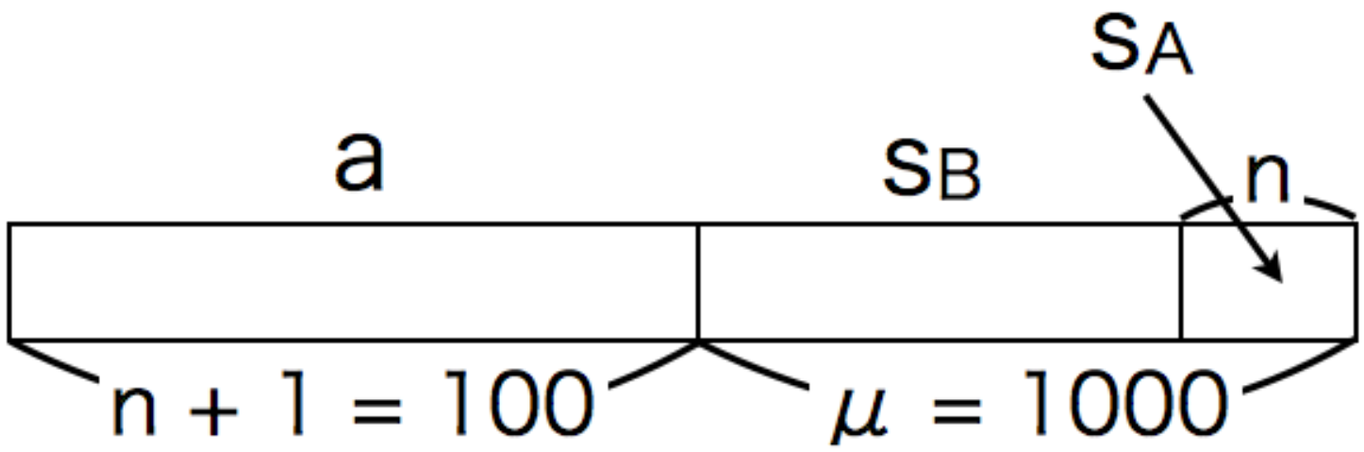
- 秘匿内積プロトコルとFairplayを用いて，２つのデータセットを秘匿した確率検定を行うシステムを実装した
  - 問題1に対して式を変形し加算,比較のみを用いて処理を行った
  - 問題2に対して乱数の大きさを抑えることで低下する安全性について評価を行った
- 今後の課題は多値や連続値なども扱えるシステムへの拡張

(2)のための変更点

従来[3]	本提案
$s_B \in Z_n$	$s_B \in [0,\ \mu\text{-}1]$
Fairplay 68.2sec (1024bit)	Fairplay 2sec (30bit)
$c = E(x_1)^{y_1} \cdot E(x_2)^{y_2} / E(s_B)$	$c = E(x_1)^{y_1} \cdot E(x_2)^{y_2} \odot E(s_B)$
危険な領域になる確率 なし	$\frac{n-1}{2\mu}$

危険な領域

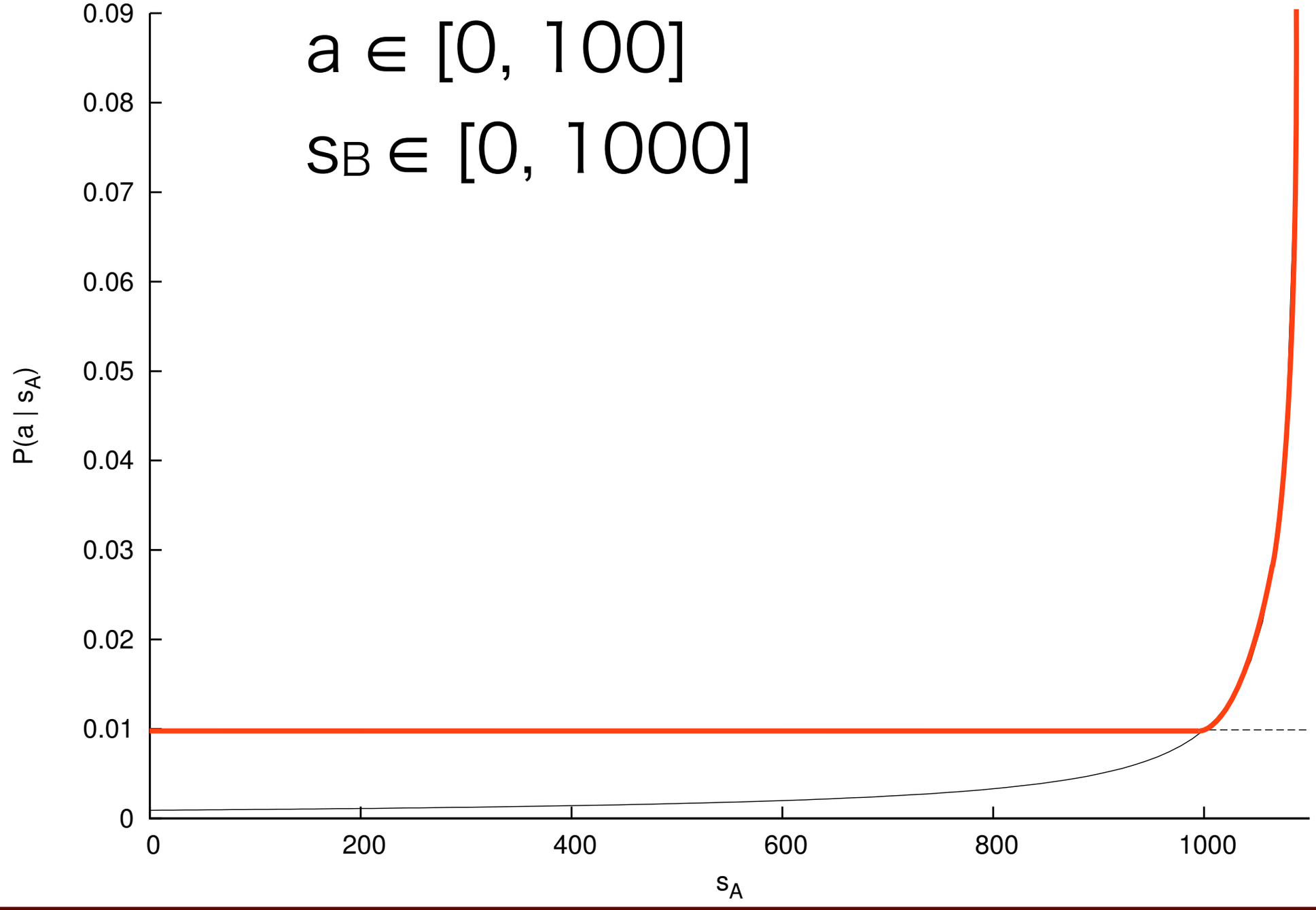
- $s_A = a + s_B$
- 要素数 $n + 1 = 100$, 乱数の最大値$\mu = 1000$の時
- $a \in [0, 100]$, $s_B \in [0, 1000]$
- $s_A = 1050$だった場合，aは少なくとも50以上



損なわれる条件付き確率P(a|s_A)の変化

$a \in [0, 100]$

$s_B \in [0, 1000]$



定理1

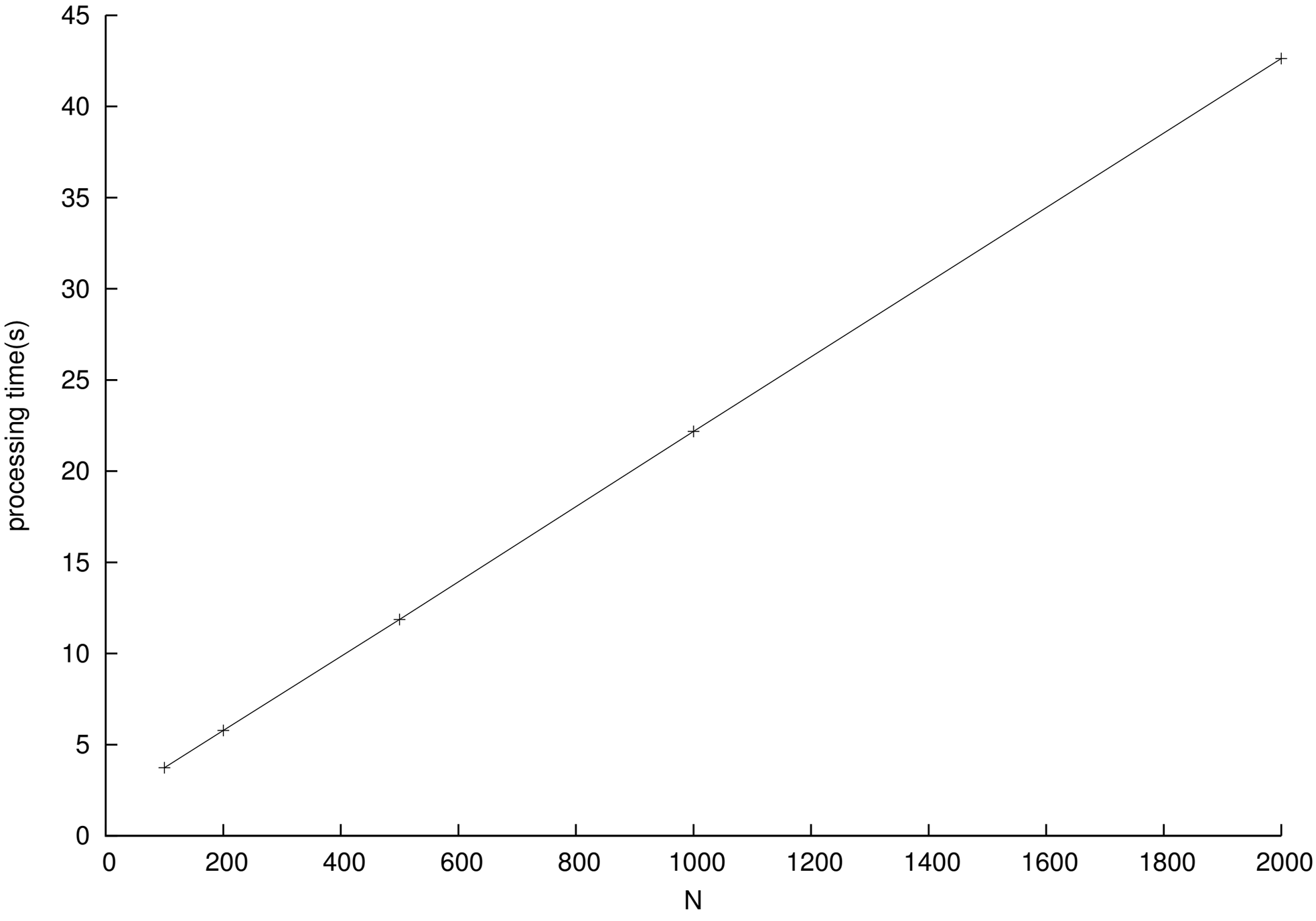
- $a \in [0, n]$, $s_B \in [0, \mu\text{-}1]$の一様分布から選んだ値
- $s_A = s_B + a > \mu$となる確率は

$$P(\mu < s_A) = \frac{n-1}{2\mu}$$

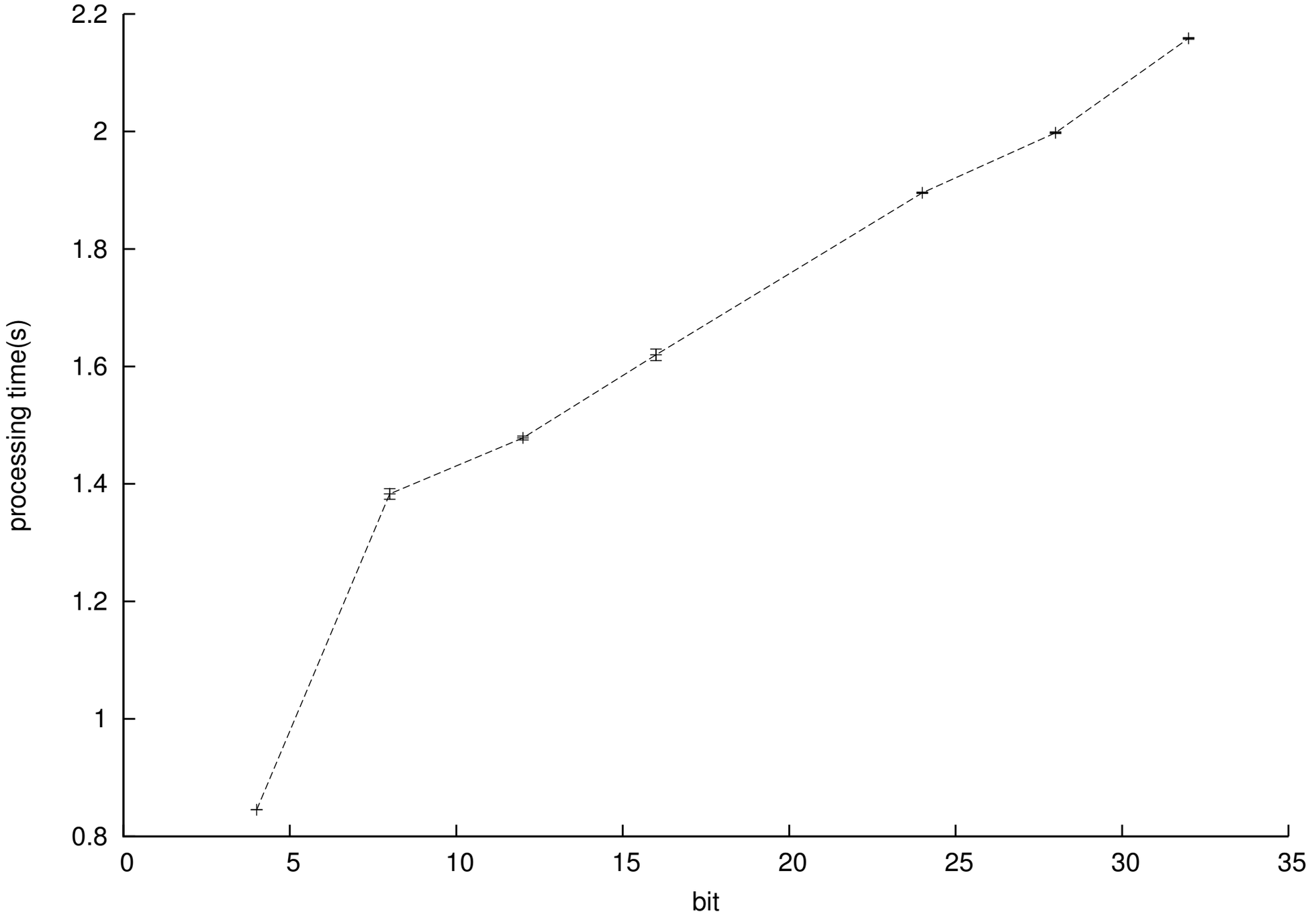
n = 140240, $\mu = 2^{30}$の時,

1/65000の確率で危険な領域に

実装したプログラムの処理時間



Fairplayによる評価の処理時間



足し算から引き算への変更

従来[3]	本提案
$s_A = a - s_B$	$s_A = a + s_B$
$a = 20, s_B = 100$	$a = 20, s_B = 100$
$sA = 20 - 100 \bmod 256$ $= 48 \bmod 256$	$sA = 20 + 100 \bmod 256$ $= 120 \bmod 256$
$a = 48 + 100 \bmod 256$	$a = 120 - 100 \bmod 256$
Fairplayで正しくmodをとる必要性	