

# TOSHIBA

Leading Innovation >>>

## Blanchetフレームワークにおける CDH仮定の定式化方針について

○ 花谷 嘉一 †  
    國分 雄一 ‡  
    米山 一樹 ‡  
    太田 和夫 ‡

† (株) 東芝 研究開発センター

‡ 電気通信大学

安全性証明の手間の軽減や、証明中のミスを防ぐために形式的検証技術が注目されている。

Blanchetのフレームワークはその一種。



証明に必要な書き換え規則の生成には手間がかかるし、ミスが混入する恐れがある。

しかし、フレームワークの性質上、絶対に書き換え規則は必要。

一回だけ頑張っておき、以降 それを使い回せるならなんとか我慢できそう。

使い回す機会が多いと思われる、  
計算量的仮定の定式化方針を探ろう。

# 背景



計算量的仮定の定式化は、できればやりたくない……

The design of such equivalences can be **delicate**, but this is a **one-time effort**: the same equivalence can be reused for proofs that rely on the same assumption. [BP06]

Blanchetの定式化した 一方向性置換で証明できるもの.

FDH署名, PFDH署名の**UF-CMA**  
[BR93]の公開鍵暗号の**IND-CPA, IND-CCA**

確かに、使いまわしできてみたい……

ただ、使い回しがきく理由について明確に議論されていない。  
CDH仮定も定式化さえすれば使い回せるのかな？

# 本日の内容

- Blanchetフレームワークのための計算量的仮定の定式化  
汎用性の高い計算量的仮定の定式化を行うためには、  
どんな方針で定式化すればいいのか？

## 4通りの安全性

4通りの定式化した  
CDH仮定

安全性 CDH仮定	1	2	3	4
1				
2		?		
3				
4				

CryptoVerifを用いて検証を実行  
それぞれ証明できる安全性を調べる。

結果：より強力な能動的攻撃モデルを採用して定式化すると、  
汎用性が高い。

# 目次

---

- 計算量的仮定を書き換え規則で表現する手順
- CDH仮定の定式化方針 … 4通り
- 評価する安全性 … 4通り
- CDH仮定による証明結果
- まとめ

# 計算量的仮定を書き換え規則で表現する手順

普通の仮定  
問題

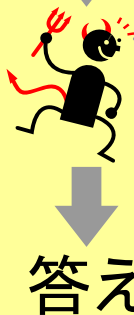


正しく答えるのは  
難しい。

計算量的仮定の書き換え規則

問題

解きやすそうな問題を選ぶ。  
解くためのヒントを得る。



計算量的仮定の下での、能動的攻撃による差



- ・ 書き換えの形を考える. 証明できるかどうかに影響
- ・ 観測等価性を評価する. 証明結果の正しさに影響

計算量的仮定の定式化には、手間がかかる！

---

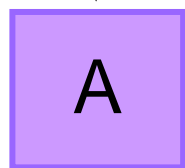
# CDH仮定の定式化

# 落とし戸付一方向性置換

普通のモデル

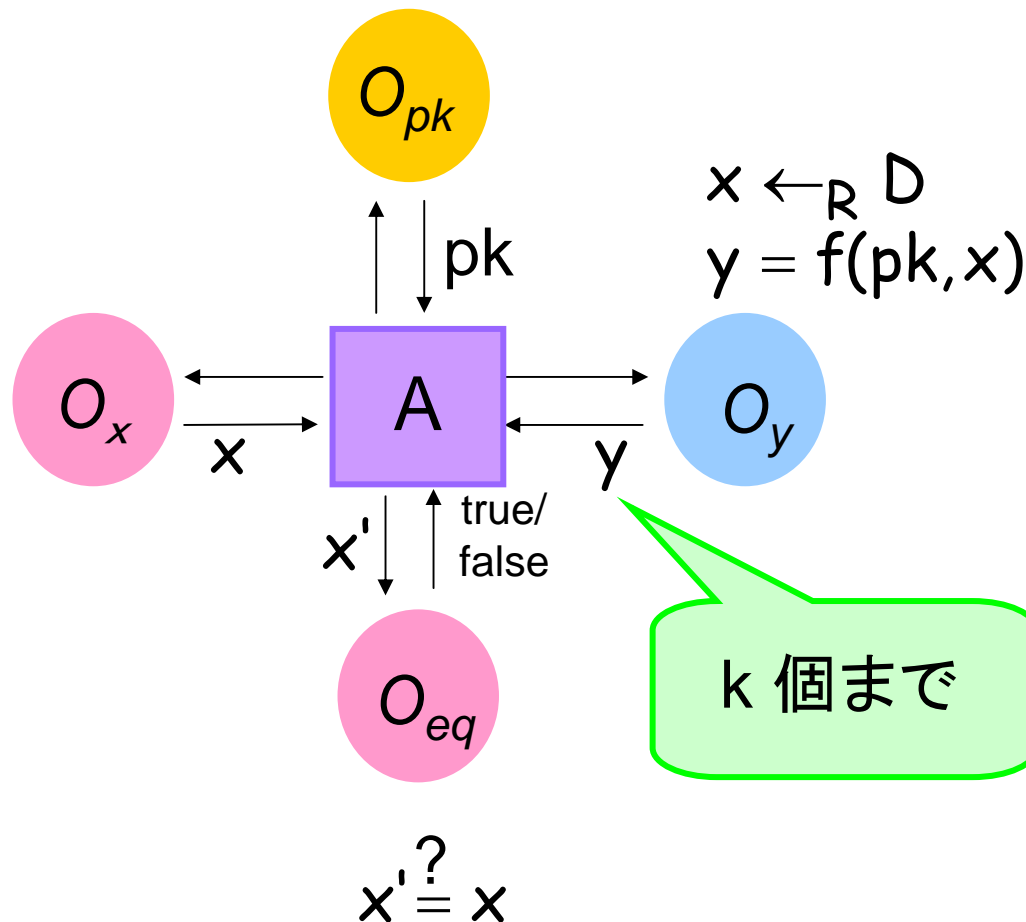
$$x \leftarrow_R D$$
$$y = f(pk, x)$$

$$pk, y = f(pk, x)$$



$x'$

定式化しているモデル



k 個まで



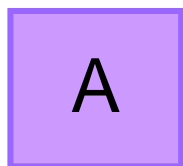
# 計算 Diffie-Hellman 仮定

## 普通のモデル

$$(p, g) \leftarrow \text{Ggen}(r)$$

$$(a, b) \stackrel{R}{\leftarrow} [1, |G_p|]^2$$

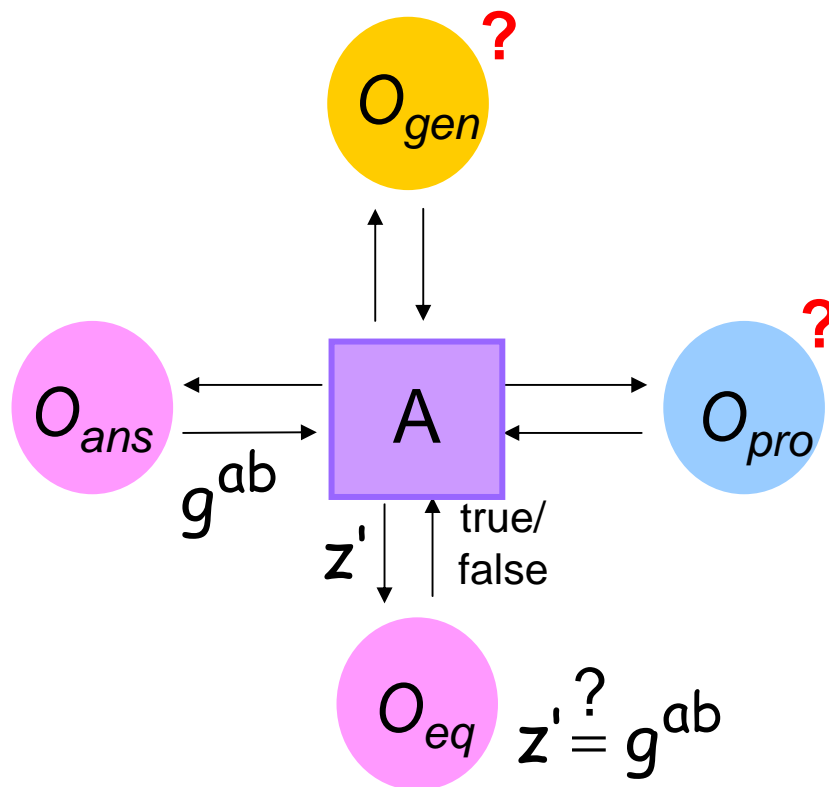
$$p, g, g^a, g^b$$



$z'$

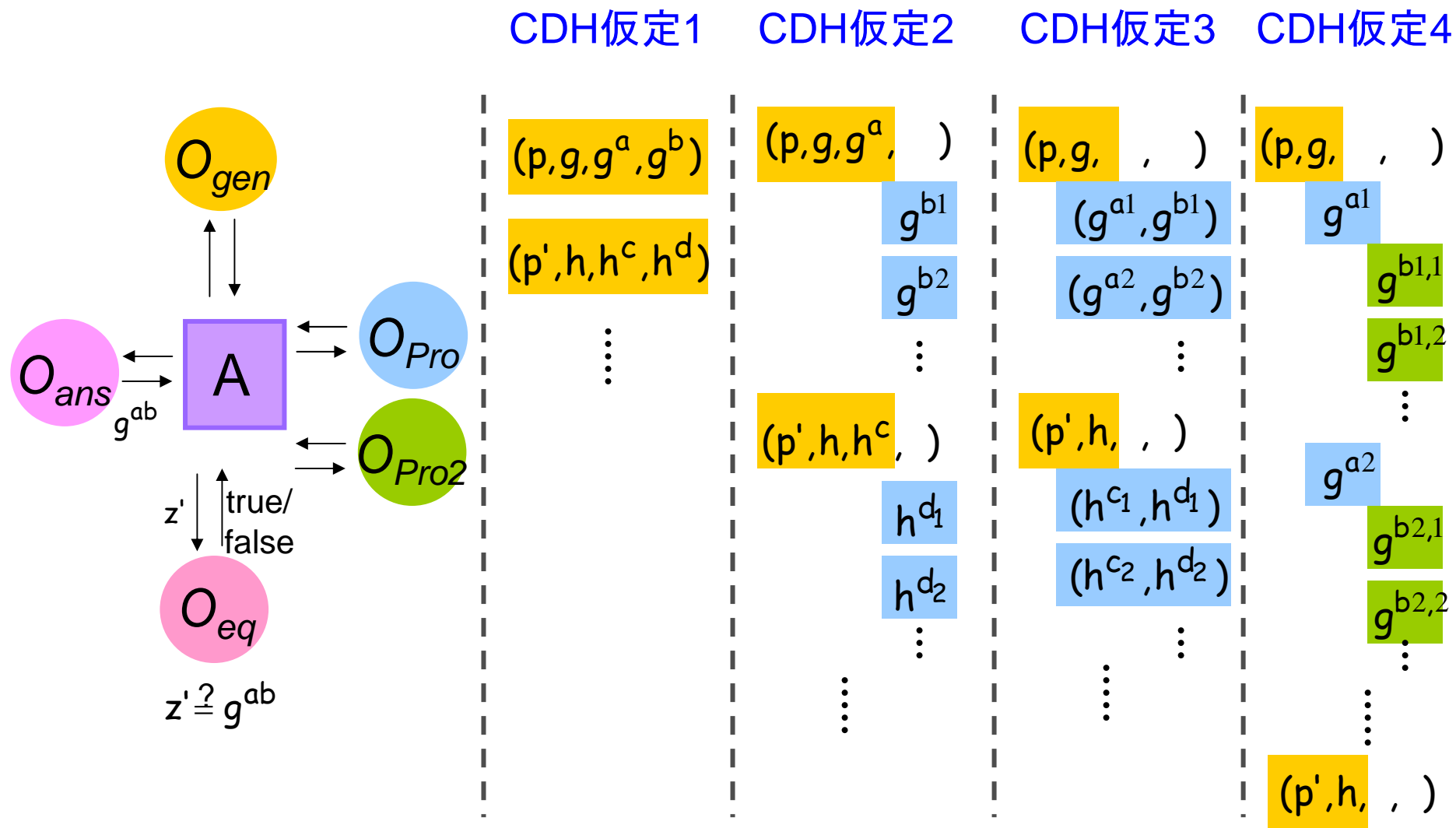
時間  $t$  以内では、いかなる  $A$  であっても  $z' = g^{ab}$  となる確率は  $\varepsilon$  以下。

## 定式化するモデル



問題の4成分をどのように与えればよいか？

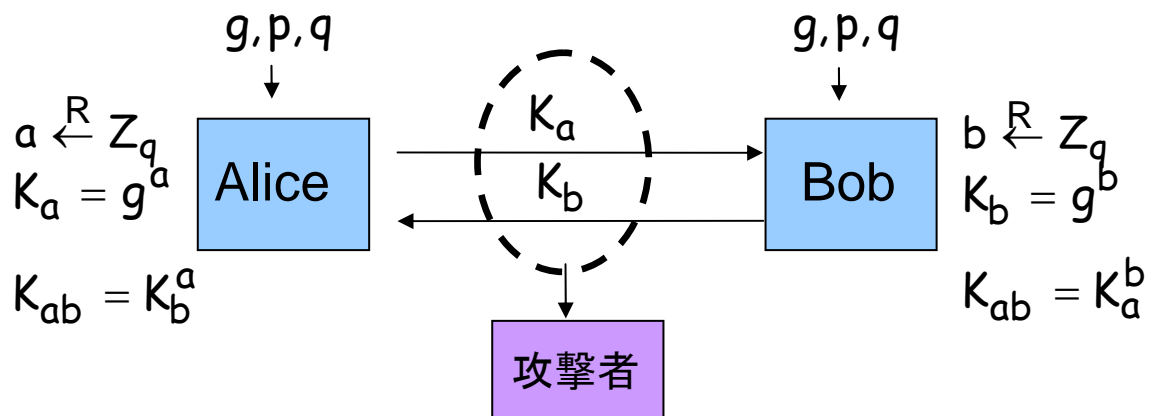
# 4つの能動的攻撃モデル



---

# 検証する安全性

# 盗聴攻撃



共有鍵  $K_{ab}$  を求める

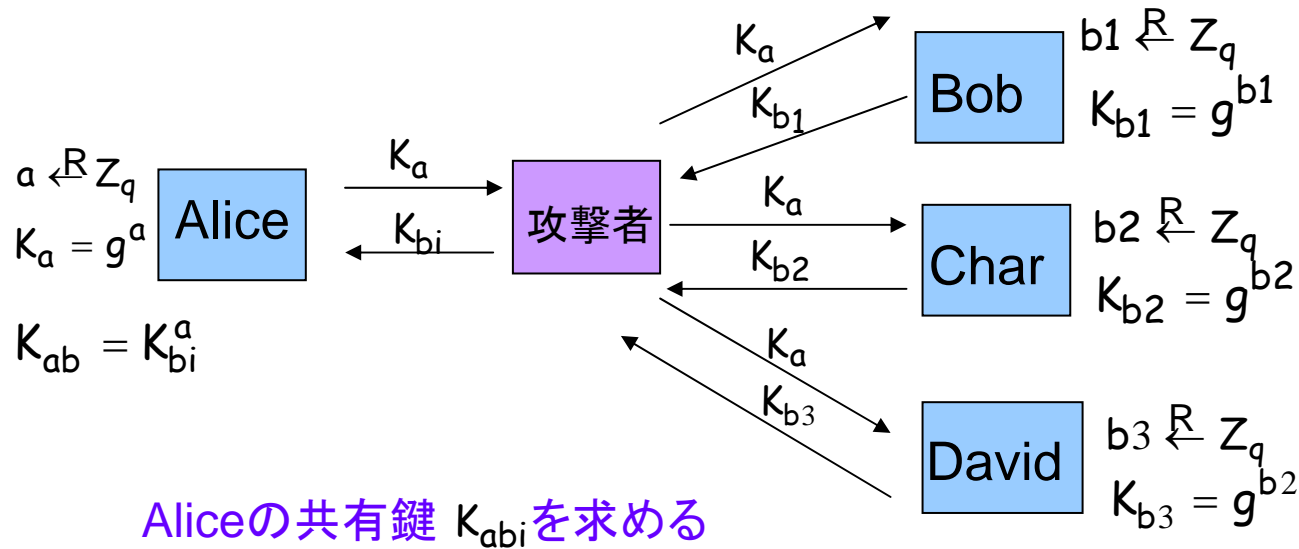
## 安全性1:

一回だけ鍵交換を実行.  
2者間の共有鍵  $K_{ab}$  を求めるのは  
難しい.

## 安全性2:

複数回 鍵交換を実行.  
2者間の共有鍵  $K_{ab}$  を1つ求める  
ことすら難しい.

# Unknown Key Share 攻撃 [CBH05]



## 安全性3:

Aliceが一回だけ鍵交換を実行.  
Aliceの共有鍵  $K_{abi}$  を求めるのは  
難しい.

## 安全性4:

Aliceが複数回 鍵交換を実行.  
Aliceの共有鍵  $K_{abi}$  を1つ求める  
ことすら難しい.

---

# CryptoVerif を用いた安全性検証

# 実験デモ

安全性 CDH仮定	1	2	3	4
1	○			
2				
3				
4	▲			

▲ : CryptoVerifで証明する際,  
“auto”では証明できなかった.  
仮定の適用箇所を指定することで  
証明可能.

# 結果

安全性 CDH仮定	1	2	3	4
1	○	×	×	×
2	○	×	×	×
3	▲	▲	×	×
4	▲	▲	▲	▲

CDH仮定4が、最も多くの証明に適用できた。

強力な能動的攻撃モデルを採用して定式化すれば、汎用性が高まることが確認できた。



# まとめ

---

Blanchetフレームワークのために、  
使いまわしのきく書き換え規則として  
計算量的仮定を定式化する方針を探った。

- 方法
- CDH仮定を4通りのモデルで定式化
  - CryptoVerifでDH鍵交換の4通りの安全性を検証

- 結果
- 定式化したCDH仮定が使用可能なことを確認
  - できるだけ強力な能動的攻撃モデルを採用して定式化すれば、使い回しが効く傾向がある。
  - ただし、強力な能動的攻撃モデルを採用した場合、証明には人間の補助が必要となった。

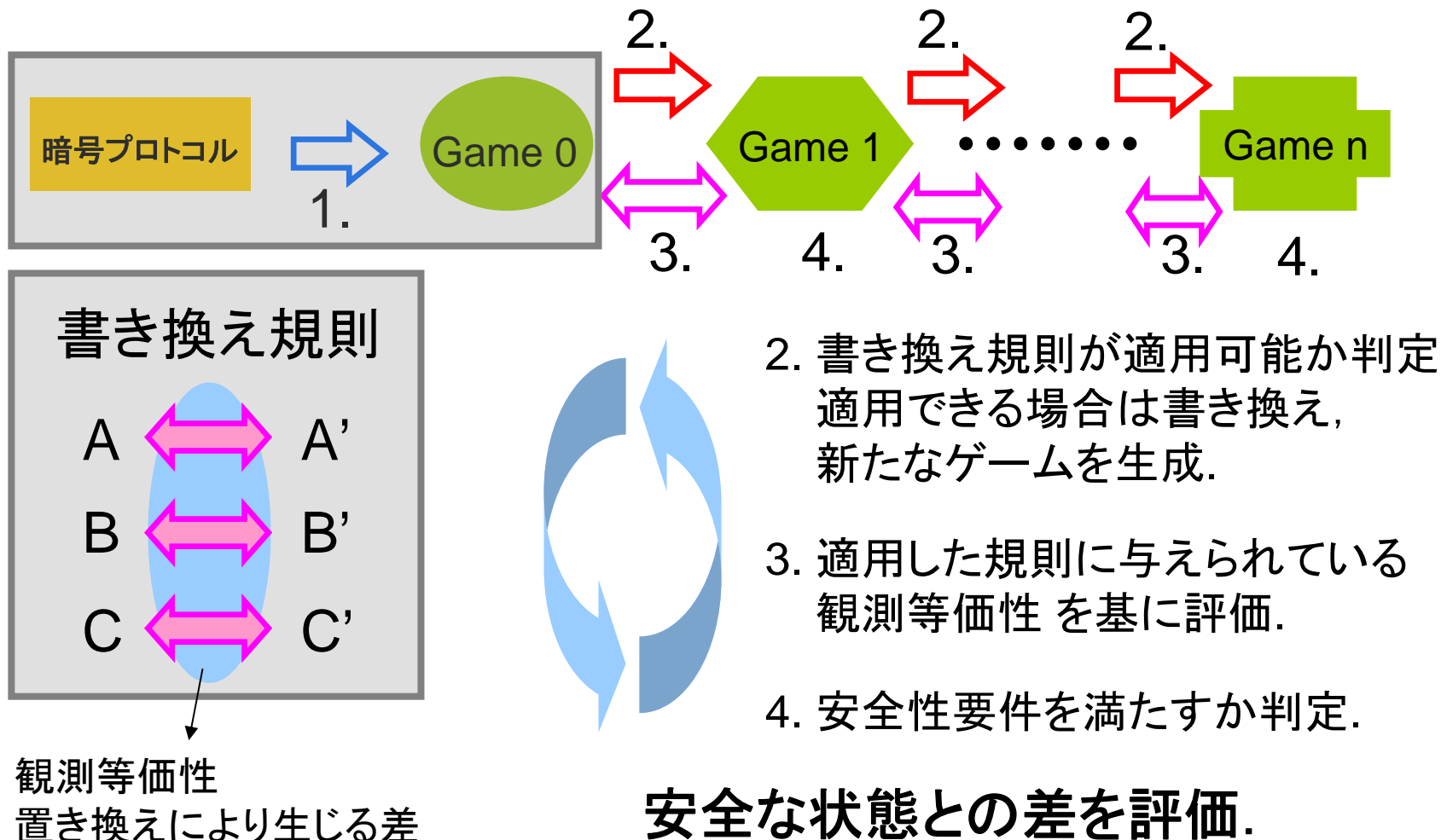
**TOSHIBA**

**Leading Innovation >>>**

---

# 参考資料

# Blanchetフレームワーク



フレームワークに基づく自動検証ツール [CryptoVerif](#) が公開されている.

# 証明結果

**安全性1** (CDH仮定1,2,3,4 の下で同じ証明結果となった.)

RESULT Proved event bad  $\implies$  false with probability

$\text{PCDH}(\text{time} + \text{time}(\text{context for game 4}))$

RESULT  $\text{time}(\text{context for game 4}) = \text{time}(\text{true})$

**安全性2** (CDH仮定3,4 の下で同じ証明結果となった.)

RESULT Proved event bad  $\implies$  false with probability

$na * \text{PCDH}(-3. * \text{time}(f) + 3. * na * \text{time}(f) + \text{time} + \text{time}(\text{context for game 4}))$

RESULT  $\text{time}(\text{context for game 4}) = \text{time}(\text{true}) * na$

**安全性3** (CDH仮定4 でのみ証明できた.)

RESULT Proved event bad  $\implies$  false with probability

$nb * \text{PCDH}(2. * nb * \text{time}(f) + -2. * \text{time}(f) + \text{time} + \text{time}(\text{context for game 4}))$

RESULT  $\text{time}(\text{context for game 4}) = \text{time}(\text{true}) * nb$

**安全性4** (CDH仮定4 でのみ証明できた.)

RESULT Proved event bad  $\implies$  false with probability

$na * nb * \text{PCDH}(2. * na * nb * \text{time}(f) + na * \text{time}(f) + -3. * \text{time}(f) + \text{time} + \text{time}(\text{context for game 4}))$

RESULT  $\text{time}(\text{context for game 4}) = \text{time}(\text{true}) * na * nb$

# CDH仮定1

```
L : foreach  $i_k \leq n_k$  do  $O_{gr} := r \stackrel{R}{\leftarrow} seed$ ;  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ;  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
  ( $O_G := \text{return}(Ggen1(r), Ggen2(r))$ )  
  |  $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$   
  |  $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$   
  | foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) := \text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$   
  |  $O_{ans}() := \text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))
```

$\approx_p^{CDH1}(t=n_k \text{Succ}_{\mathbb{G}}^{CDH}(t+(n_k-1)t_{Ggen}+3(n_k-1)t_f)$

```
R : foreach  $i_k \leq n_k$  do  $O_{gr} := r \stackrel{R}{\leftarrow} seed$ ;  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ;  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
  ( $O_G := \text{return}(Ggen1(r), Ggen2(r))$ )  
  |  $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$   
  |  $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$   
  | foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) :=$   
    if defined( $k$ ) then  $\text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$   
    else  $\text{return}(\text{false})$   
  |  $O_{ans}() := k \leftarrow \text{mark}$ ;  $\text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))
```

# CDH仮定2

```
L : foreach  $i_k \leq n_k$  do  $O_{gr} := r \stackrel{R}{\leftarrow} seed$ ;  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
  ( $O_G := \text{return}(Ggen1(r), Ggen2(r))$   
  |  $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$   
  | foreach  $i_f \leq n_{f2}$  do  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$ ;  
    | foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) := \text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$   
    |  $O_{ans}() := \text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))
```

$\approx_p^{CDH2}(t) = n_k n_{f2} \text{Succ}_{\mathbb{G}}^{CDH}(t + (n_k - 1)t_{Ggen} + (2n_k n_{f2} + n_k - 3)t_f)$

```
R : foreach  $i_k \leq n_k$  do  $O_{gr} := r \stackrel{R}{\leftarrow} seed$ ;  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
  ( $O_G := \text{return}(Ggen1(r), Ggen2(r))$   
  |  $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$   
  | foreach  $i_{f2} \leq n_f$  do  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$   
    | foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) :=$   
      if defined( $k$ ) then  $\text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$   
      else  $\text{return}(\text{false})$   
    |  $O_{ans}() := k \leftarrow \text{mark}$ ;  $\text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))
```

# CDH仮定3

```
L : foreach  $i_k \leq n_k$  do  $O_{gr}() := r \stackrel{R}{\leftarrow} seed$ ; return;  
  ( $O_G() := \text{return}(Ggen1(r), Ggen2(r))$ )  
  |foreach  $i_f \leq n_{f3}$  do  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ;  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$ )  
    | $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$ )  
    |foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) := \text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ );  
    | $O_{ans}() := \text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ )).
```

$\approx_p^{CDH3}(t) = n_k n_{f3} \text{Succ}_G^{CDH}(t + (n_k - 1)t_{Ggen} + 3(n_k n_{f3} - 1)t_f)$

```
R : foreach  $i_k \leq n_k$  do  $O_{gr}() := r \stackrel{R}{\leftarrow} seed$ ; return;  
  ( $O_G() := \text{return}(Ggen1(r), Ggen2(r))$ )  
  (foreach  $i_f \leq n_{f3}$  do  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ;  $O_{gaB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$ )  
    | $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$ )  
    |foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) :=$   
      if defined( $k$ ) then return( $z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b)$ );  
      else return(false)  
    | $O_{ans}() := k \leftarrow \text{mark}$ ; return( $f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b)$ ))))
```



# CDH仮定4

```
L : foreach  $i_k \leq n_k$  do  $O_{gr}() := r \stackrel{R}{\leftarrow} seed$ ; return;  
  ( $O_G() := \text{return}(Ggen1(r), Ggen2(r))$ );  
  |foreach  $i_{fa} \leq n_{fa}$  do  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$ )  
    |foreach  $i_{fb} \leq n_{fb}$  do  $O_{gAB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
      ( $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$ )  
      |foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) := \text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ )  
        ( $O_{ans}() := \text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))))
```

$\approx_p^{CDH4} = n_k n_{fa} n_{fb} \text{Succ}_{\mathbb{C}}^{CDH}(t + (n_k - 1)t_{Ggen} + (2n_k n_{fa} n_{fb} + n_k n_{fa} - 3)t_f)$

```
R : foreach  $i_k \leq n_k$  do  $O_{gr}() := r \stackrel{R}{\leftarrow} seed$ ; return;  
  ( $O_G() := \text{return}(Ggen1(r), Ggen2(r))$ )  
  |foreach  $i_{fa} \leq n_{fa}$  do  $O_{gA}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
    ( $O_A() := \text{return}(f(Ggen1(r), Ggen2(r), a))$ )  
    |foreach  $i_{fb} \leq n_{fb}$  do  $O_{gAB}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;  
      ( $O_{aB}() := \text{return}(f(Ggen1(r), Ggen2(r), b))$ )  
      |foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) :=$   
        if defined( $k$ ) then  $\text{return}(z' \stackrel{?}{=} f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$   
        else  $\text{return}(\text{false})$   
        ( $O_{ans}() := k \leftarrow \text{mark}$ ;  $\text{return}(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))$ ))))
```