TOSHIBA
Leading Innovation

CryptoVerif 適用のための Diffie-Hellman仮定の定式化

○花谷 嘉一†
村谷 博文†

† 東芝 研究開発センター

目次

- **背景**
  - 形式的モデルと計算量的モデル
  - Game列による安全性証明
  - CryptoVerif **[BP06]**
- **本発表の目的**
- **CDH仮定の定式化**
  - Observational Equivalence
  - Blanchetのプロセス計算による定式化
  - 証明の概要
- **DDH仮定の定式化の方針**
- **まとめ**

安全性証明のモデル

形式的モデル

- Dolev-Yaoモデル

特徴

暗号プリミティブは理想的な安全性を持っていると仮定して，暗号プロトコルの評価を行うモデル.

長所

自動検証可能.

短所

暗号プリミティブ自体の脆弱性は無視.

計算量的モデル

- 証明可能安全性

特徴

ある計算量的な問題は困難と仮定すると，暗号プロトコルは安全であることを保証する手法.

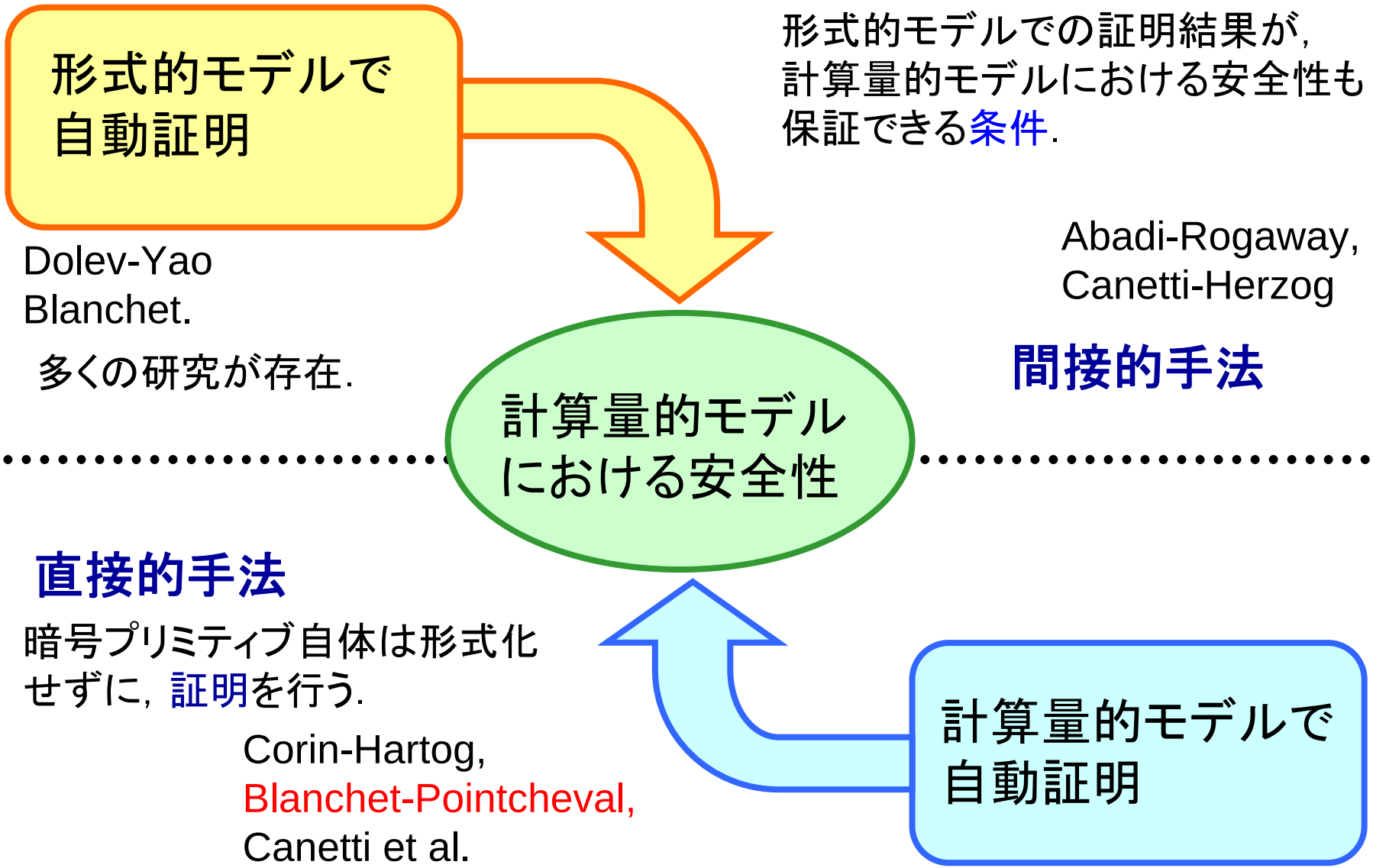
長所

暗号プリミティブ自体の脆弱性も考慮.

短所

証明が複雑で誤りやすい.

計算量的モデルでの安全性の保証



CryptoVerif とは？

- **Game列による検証を自動で行うプログラム.**
  - Blanchetが作成. 自身のサイトで公開している.
  - FDH署名や OAEP暗号などの証明を行うためのサンプルファイルも公開

- **Game列による検証に便利なプロセス計算を提案.**
- **計算量的仮定は, Observational Equivalenceを満たす式として記述.**

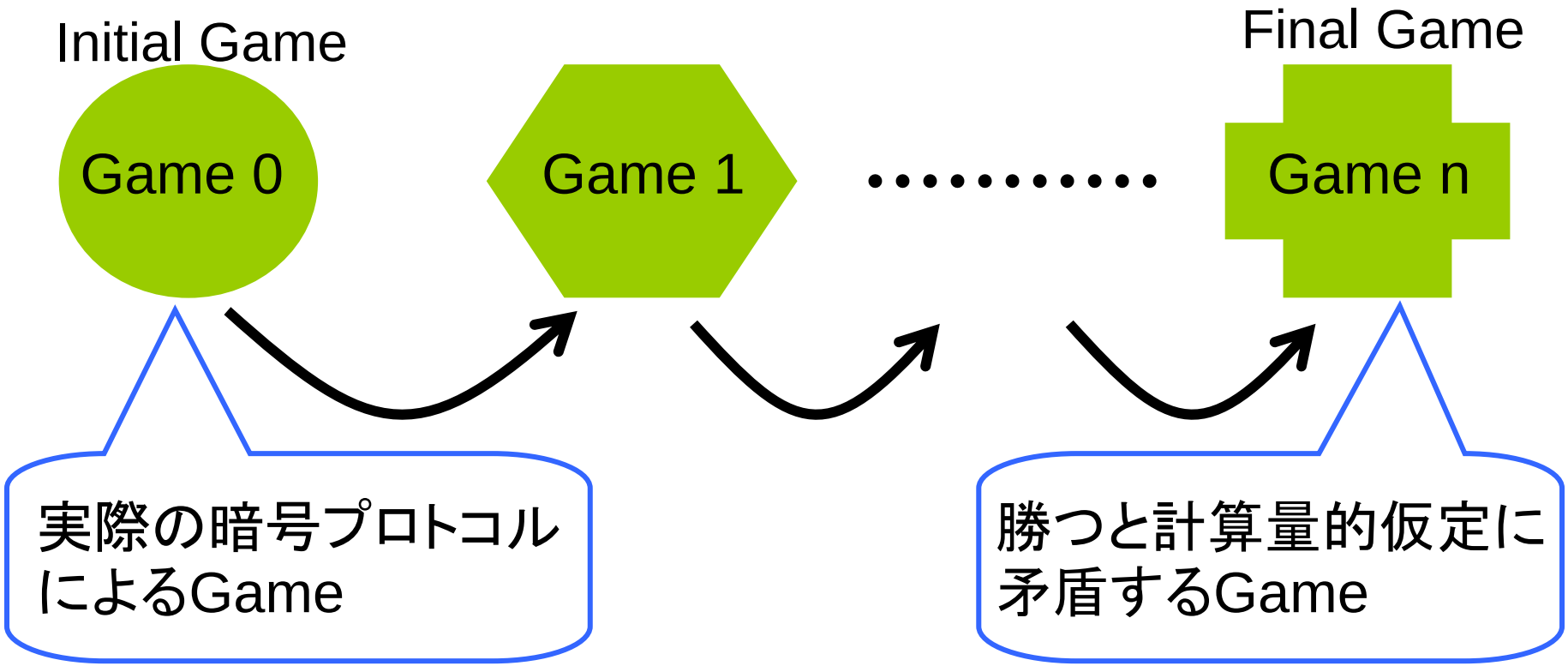
- **落とし戸付一方向性関数が存在するならば, FDH署名は eUF-ACMA であることを証明した.**
  - 証明結果は, Bellare, Rogaway [BR93] の結果と一致.

Game列による検証

- **安全性証明技法の一つ**

  各Game間の攻撃成功率の差を評価することで,
  Game 0 と Game n の攻撃成功率の差を評価する.

計算量的問題を攻撃問題に帰着する場合.



Game列による検証の自動化

Initial Game G_0

Game 0

暗号プロトコルの
攻撃ゲーム

条件

変数の定義域
関数の性質
オラクルの性質
計算量的な仮定
仮定による変換法
などなど

CryptoVerif

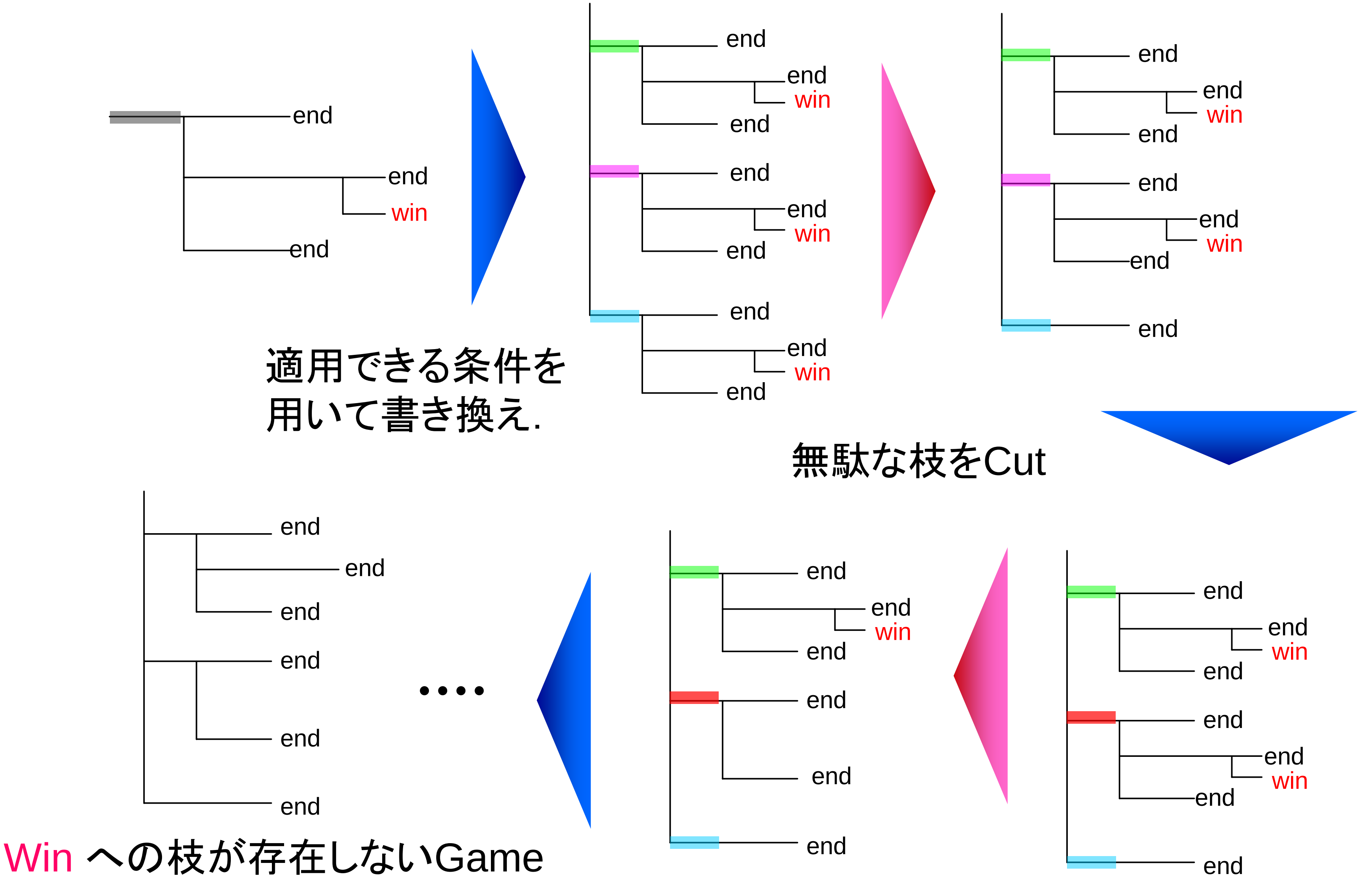
Game列による検証の自動化

攻撃者が勝つイベントが存在しない.

CryptoVerif

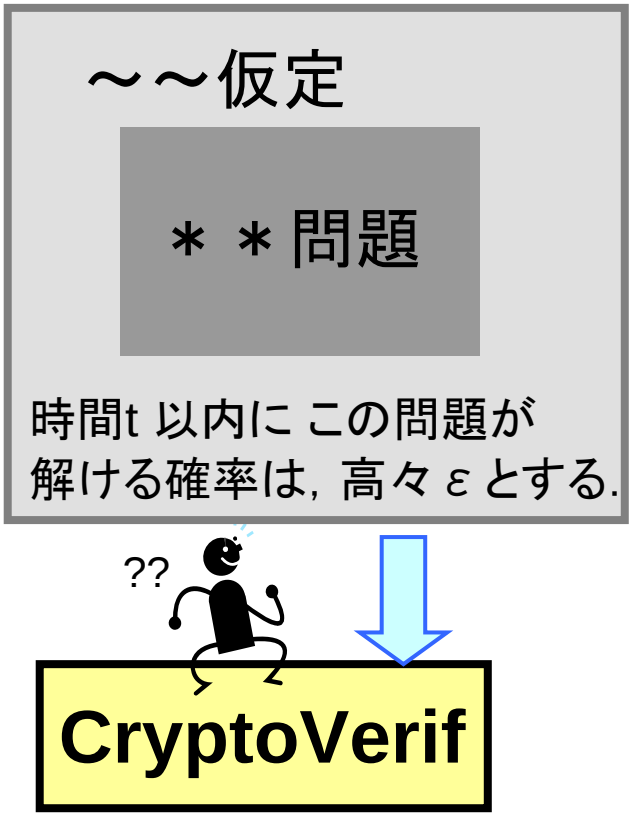
与えられた条件からGame nに到達可能か判定する.

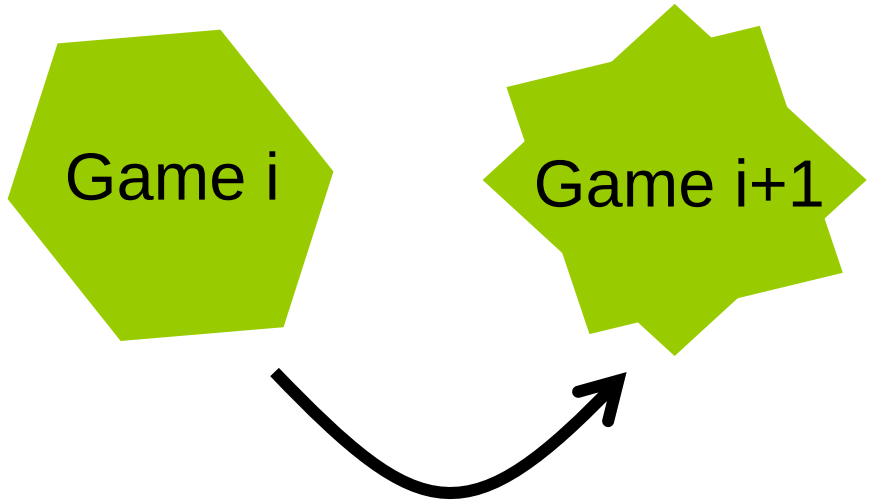
CryptoVerif の動作イメージ



計算量的仮定の取り扱い

- **CryptoVerifでは，計算量的仮定をそのまま扱うことができない．**
  - Blanchetのプロセス計算のルールに従って，Observational Equivalenceを満たす式として定式化しなければならない．
  - 人間が評価しなければならない部分も残されており，結構大変．

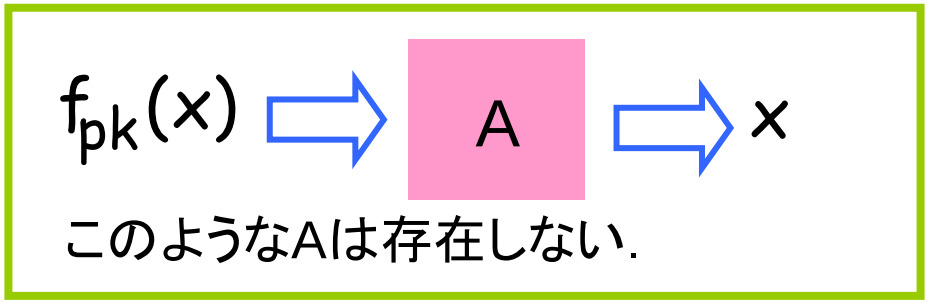




計算量的な仮定に基づいた変換とは，
何を何に置き換える変換で，
それによってどのような差が生じるか？

本発表の目的

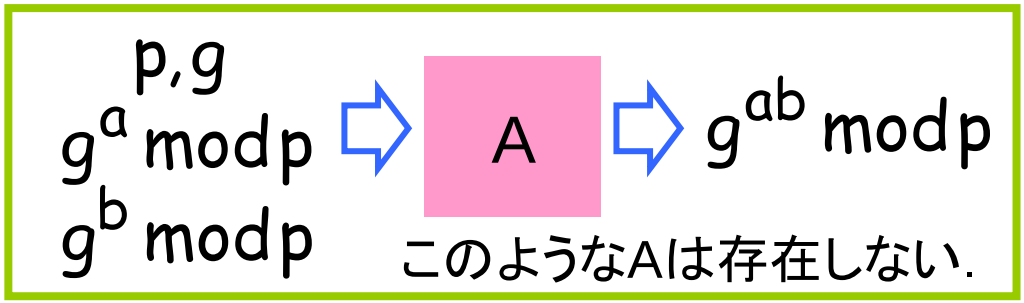
CryptoVerifで一方向性を取り扱うための記述は 既にある.

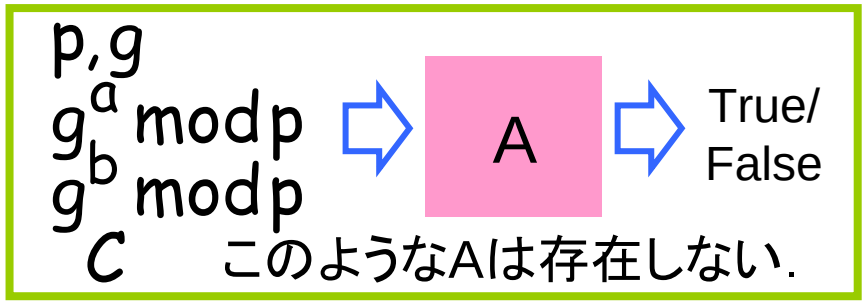


一方向性以外の計算量的仮定も記述可能なのか？

RSA逆関数問題，素因数分解問題，CDH問題，DDH問題など
様々な計算量的仮定にも対応できる？？

CDH仮定とDDH仮定をCryptoVerifで取り扱うための定式化を行う.





目次

- 背景
  - Game列による安全性証明
  - Blanchetらの方法 **[BP06]**
- 本発表の目的
- **CDH仮定の定式化**
  - Observational Equivalence
  - Blanchetのプロセス計算による定式化
  - 証明
- **DDH仮定の定式化の方針**
- **まとめ**

Observational Equivalence

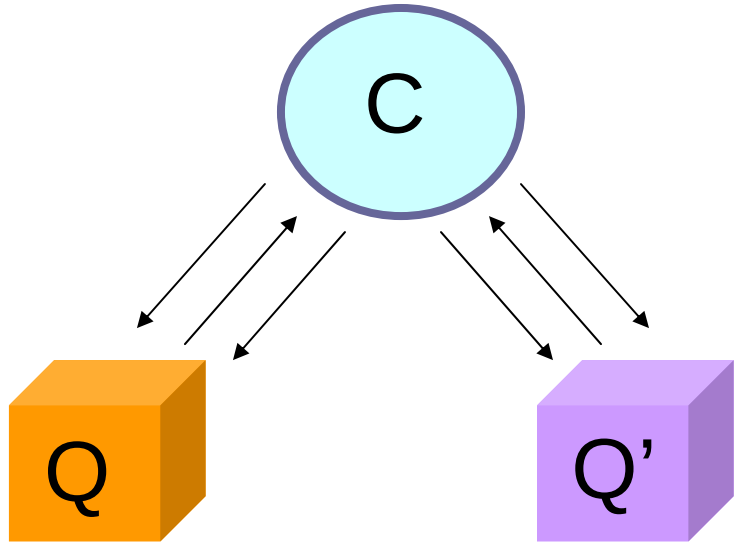
$$\forall C \forall a : (|\Pr[C[Q] \rightsquigarrow a] - \Pr[C[Q'] \rightsquigarrow a]| \leq p(t)$$
$$\wedge \sum_{\mathcal{E}} |\Pr[C[Q] \rightsquigarrow \mathcal{E}] - \Pr[C[Q'] \rightsquigarrow \mathcal{E}]| \leq p(t))$$

QとQ‘は 確率pまでObservational Equivalence といい,
$Q \approx_p Q'$ と書く.

CはQとQ‘を acceptable で実行時間が高々t の context, aは bitstring, $\mathcal{E}$ はEvent の列 である.

直感的な理解

どんなContext Cをであっても,
Cがプロセスを使っているか,
Q’を使っているか, 入出力を観察して
見分けられる確率は高々p .



Computational Diffie-Hellman **仮定** (CDH**仮定**)

$$\mathbb{G} = \{G_{\mathfrak{p}}\}$$

$$\mathsf{Succ}(\mathcal{A}) = \Pr\left[\begin{array}{l} r \xleftarrow{R} seed, \\ (\mathfrak{p}, g) \leftarrow \mathsf{Ggen}(r), \\ (a, b) \xleftarrow{R} [1, |G_{\mathfrak{p}}|]^2, \\ z' \leftarrow \mathcal{A}(\mathfrak{p}, g, g^a, g^b) : \\ z' = g^{ab} \end{array}\right]$$

時間t 以内では，いかなるAであっても

$$\mathsf{Succ}(\mathcal{A}) \leq \epsilon$$

となるとき，$\mathbb{G}$ は(t, ε)-CDH仮定を満たすという.

CryptoVerif で扱うためには，Blanchetのプロセス計算で定式化しなければならない.

計算量的仮定を取り扱うには？

通常の計算量的仮定

～～仮定

＊＊問題

時間t 以内に この問題が
解ける確率は，高々 ε とする.

計算問題の仮定

CryptoVerifが取り扱える計算量的仮定

問題への能動的攻撃を行う攻撃モデル
を定式化.

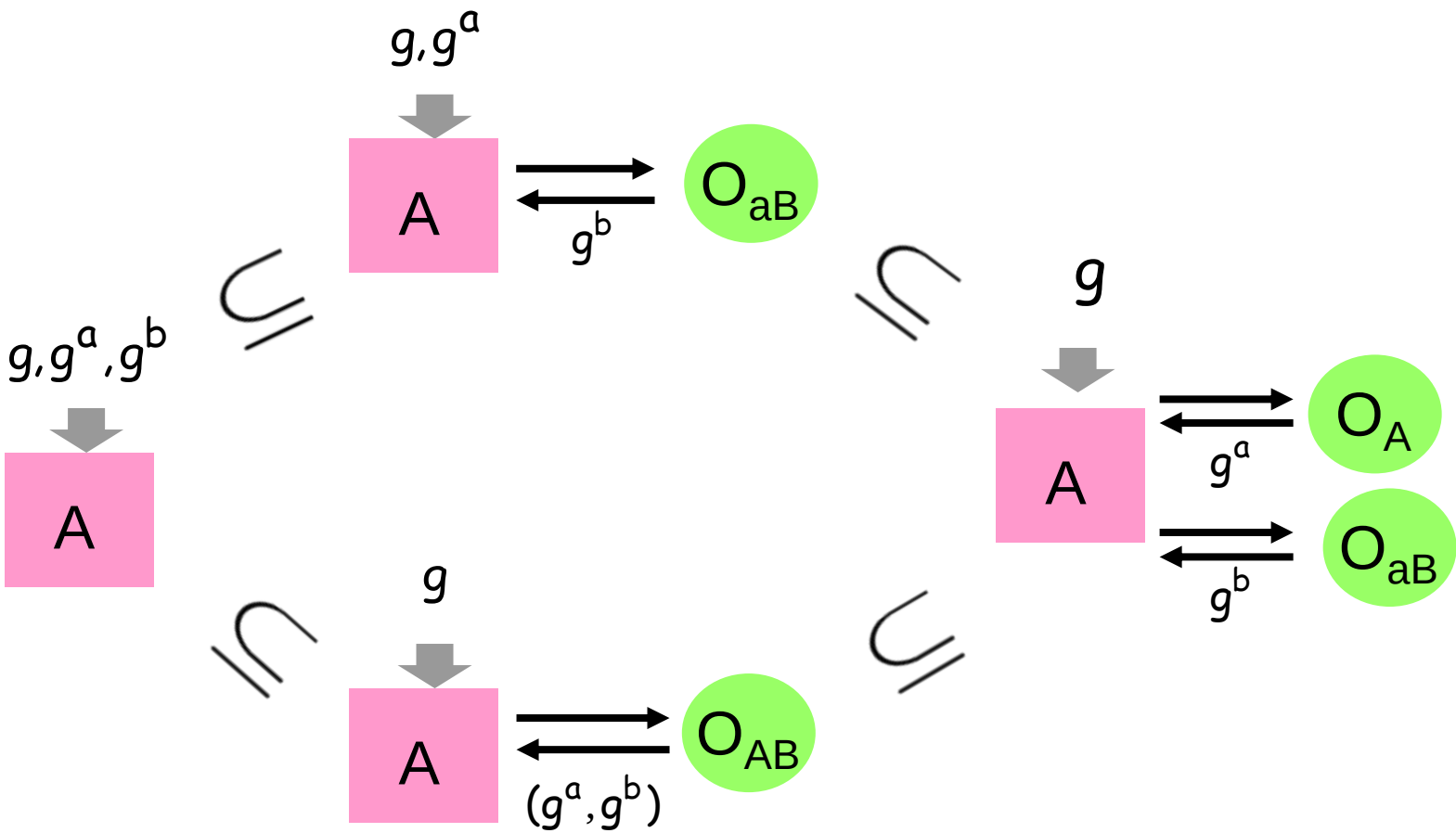
$\approx_p$

能動的攻撃の攻撃モデルだが，
絶対に攻撃成功の条件を満たさない
理想的な状況を定式化.

CDH仮定への能動的攻撃モデル

入力パターンのバリエーション.

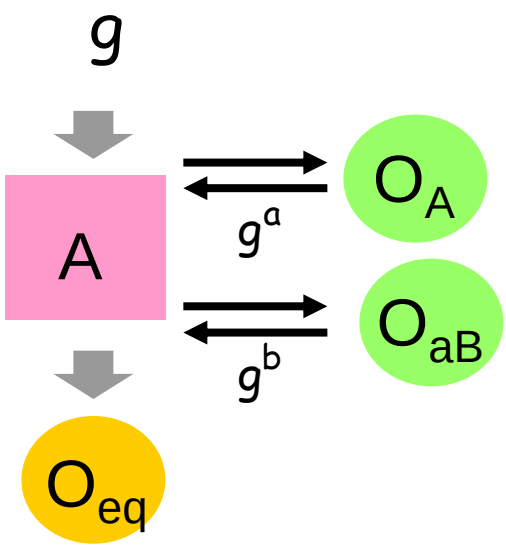
入力される問題を能動的に選べるように.



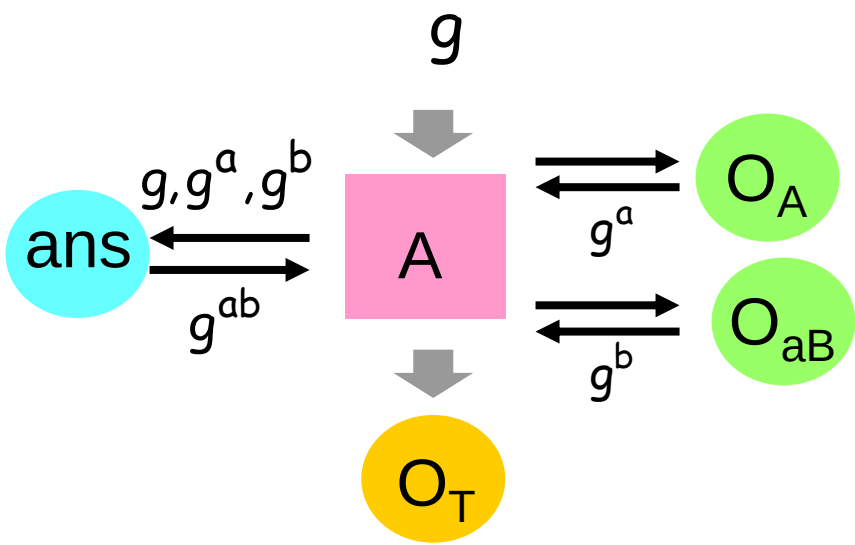
使用するオラクルによる攻撃のゴールの差

ans ― 呼び出しを認める.
　　└ 呼び出しを認めない.

入力された問題の組に対応する,
答えを返すオラクル.



入力された組のうち
正しい答えを1つでも
返せば成功.



入力された組で, ans に
問い合わせていないものの
正しい答えを返せば成功.

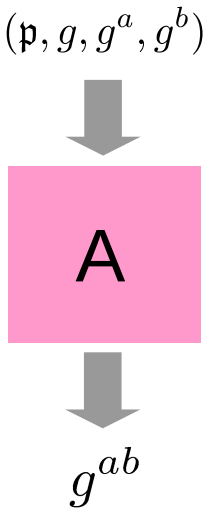
攻撃モデルの拡張

通常のCDH仮定

$r \xleftarrow{R} seed,$

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

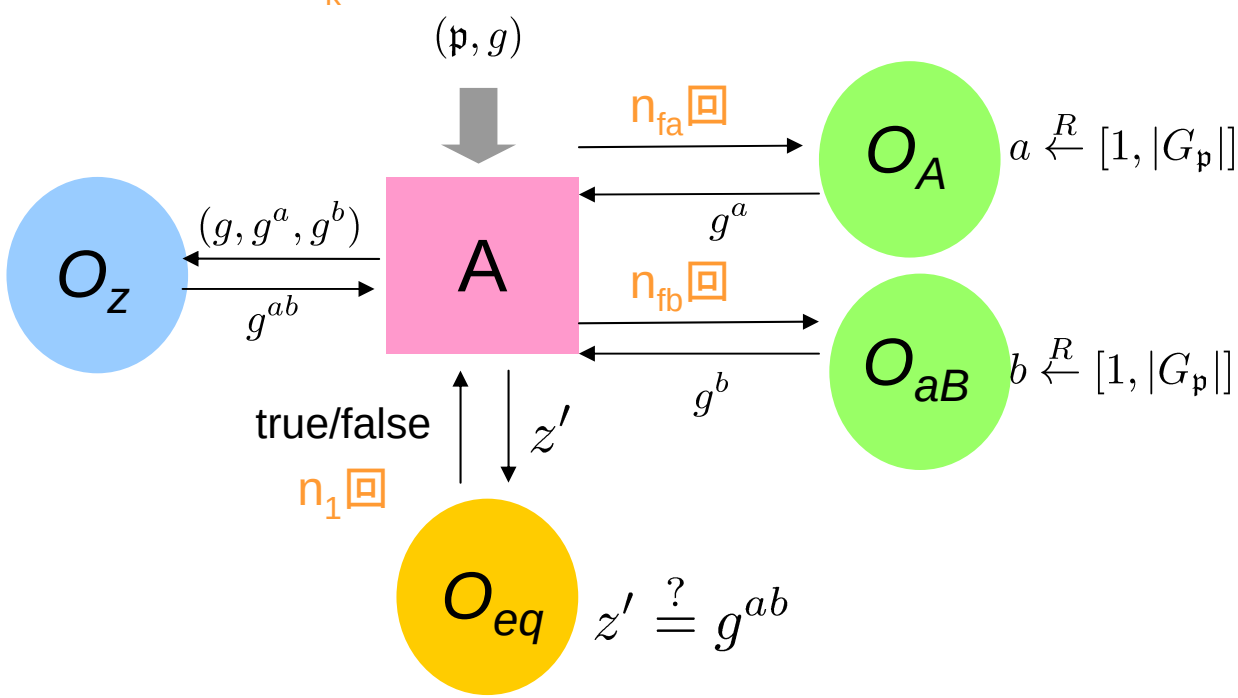
$(a, b) \xleftarrow{R} [1, |G_{\mathfrak{p}}|]^2,$



攻撃モデルを拡張したCDH仮定

$r \xleftarrow{R} seed,$

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



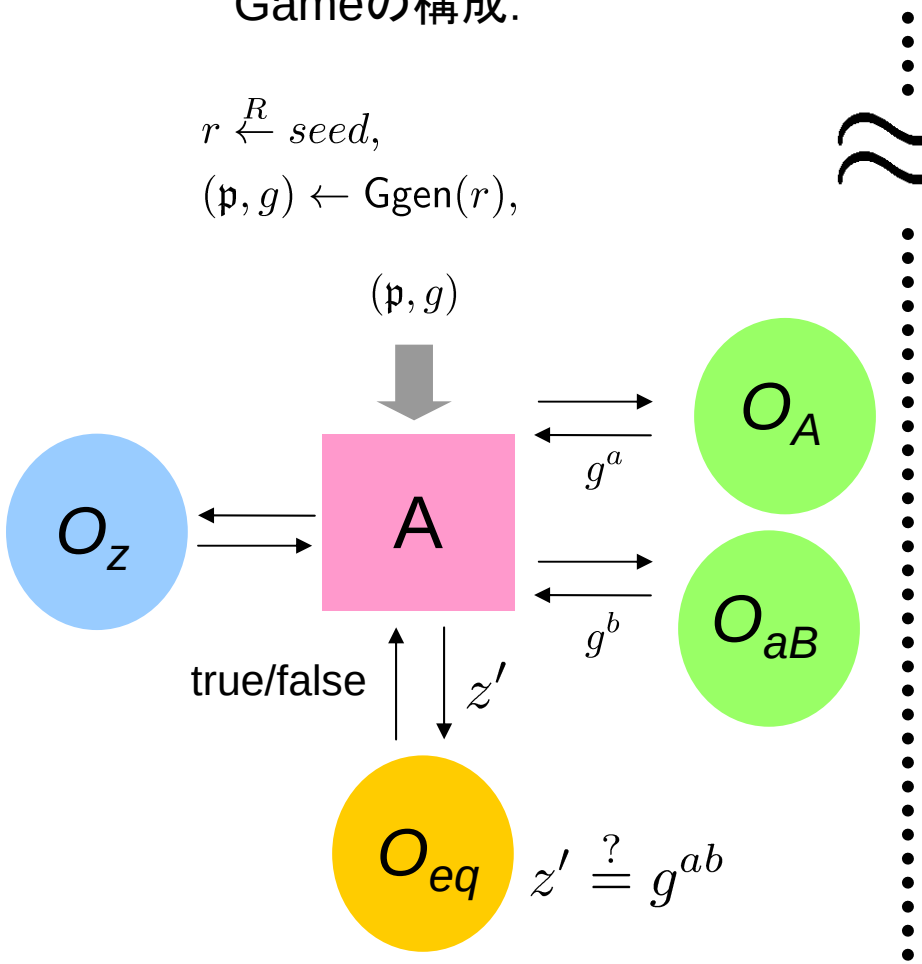
Ozに問い合わせていない(g^a, g^b)に対して，

$O_{eq}(z')$→true となったとき勝ち．

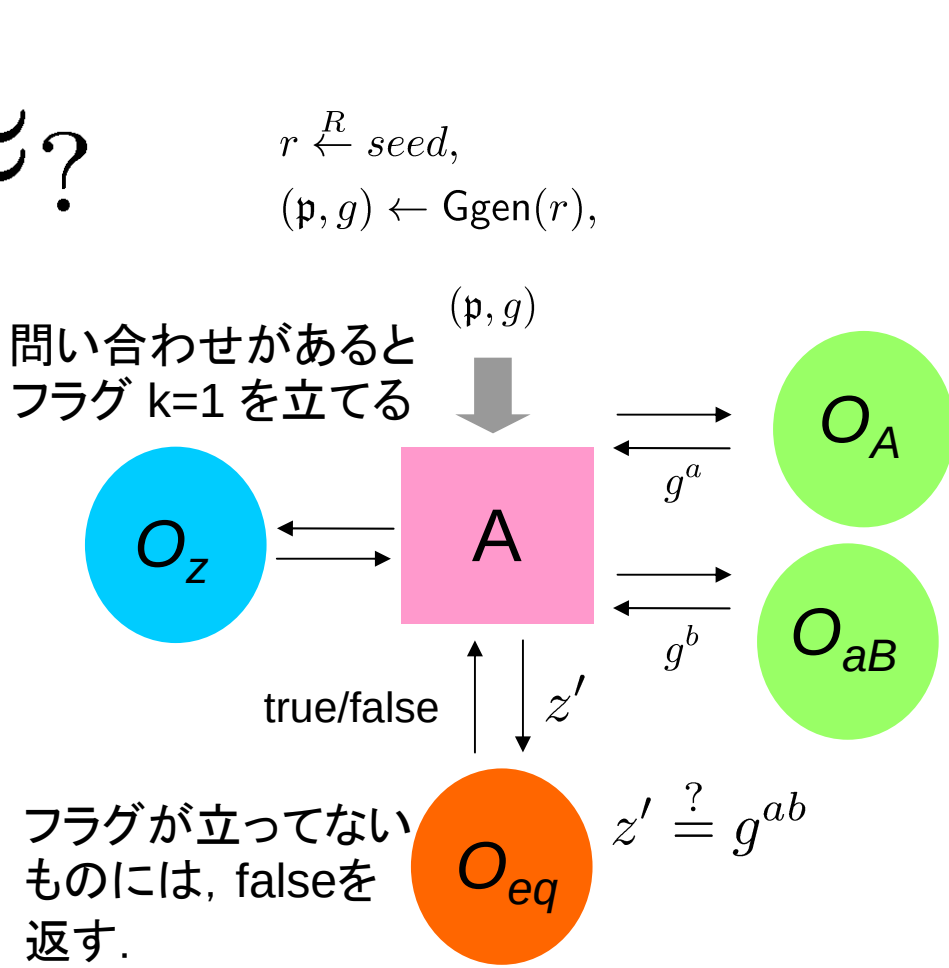
Observational Equivalent な関係で表現

勝利条件：Ozに問い合わせていない(g^a, g^b)に対して，$O_{eq}(z')\rightarrow$true となったとき勝ち．

目的：Ozに問い合わせていない(g^a, g^b)に対して，**絶対に**$O_{eq}(z')\rightarrow$true とならない Gameの構成．



$\approx ?$



Computational Diffie-Hellman **仮定** (CDH**仮定**)

一般的な定式化

$$\mathbb{G} = \{G_{\mathfrak{p}}\}$$

$$\mathrm{Succ}(\mathcal{A}) = \Pr\left[\begin{array}{l} r \xleftarrow{R} seed, \\ (\mathfrak{p}, g) \leftarrow \mathsf{Ggen}(r), \\ (a, b) \xleftarrow{R} [1, |G_{\mathfrak{p}}|]^2, \\ z' \leftarrow \mathcal{A}(\mathfrak{p}, g, g^a, g^b) : \\ z' = g^{ab} \end{array}\right]$$

時間t 以内では，いかなるAであっても

$$\mathrm{Succ}(\mathcal{A}) \leq \epsilon$$

となるとき，$\mathbb{G}$ は(t, ε)-CDH仮定を満たす．

CryptoVerif で取り扱うための定式化

L : **foreach** $i_k \leq n_k$ **do** $O_{gr} := r \xleftarrow{R} seed$; **return**;
($O_G :=$ **return**(Ggen1(r), Ggen2(r))
| **foreach** $i_{fa} \leq n_{fa}$ **do** $O_{ga}() := a \xleftarrow{R} [1, |G_{\mathfrak{p}}|]$; **return**;
| **foreach** $i_{fb} \leq n_{fb}$ **do** $O_{gb}() := b \xleftarrow{R} [1, |G_{\mathfrak{p}}|]$; **return**;
($O_y() :=$ **return**(f(Ggen1(r), Ggen2(r), a), f(Ggen1(r), Ggen2(r), b))
| **foreach** $i_1 \leq n_1$ **do** $O_{eq'}(z' : G_{\mathfrak{p}}) :=$ **return**($z' \stackrel{?}{=} f$(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))
| $O_z() :=$ **return**(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b)))).

$$\approx_p$$

R : **foreach** $i_k \leq n_k$ **do** $O_{gr} := r \xleftarrow{R} seed$; **return**;
($O_G :=$ **return**(Ggen1(r), Ggen2(r))
| **foreach** $i_{fa} \leq n_{fa}$ **do** $O_{ga}() := a \xleftarrow{R} [1, |G_{\mathfrak{p}}|]$; **return**;
| **foreach** $i_{fb} \leq n_{fb}$ **do** $O_{gb}() := b \xleftarrow{R} [1, |G_{\mathfrak{p}}|]$; **return**;
($O_y() :=$ **return**($z' = f$(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))
| **foreach** $i_1 \leq n_1$ **do** $O_{eq'}(z' : G_{\mathfrak{p}}) :=$
if defined(k) **then return**($z' \stackrel{?}{=} f$(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))
else return(false)
| $O_z() := k \leftarrow$ mark; **return**(f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b)))).

ただし，

$$p(t) = n_k n_{fa} n_{fb} \times \mathrm{Succ}(t + (n_k n_{fa} n_{fb} + n_k n_{fa} - 2)t_f + (n_k - 1)t_{\mathsf{Ggen}})$$

証明の概要

従来のCDH仮定のLRと
攻撃モデルを能動的攻撃に拡張したLR'を考える.

LR'で $(i_{fa}, i_{fb}) \in [1, n_{fa}] \times [1, n_{fb}]$ で初めて攻撃が成功する場合を考える.
C_i[LR]が $(i_{fa}, i_{fb}) \in [1, n_{fa}] \times [1, n_{fb}]$ まで LR'をシミュレートできるC_iを構成する.

LR'で **event compute** の起こる確率pを評価する.

$$p(t) = n_k n_{fa} n_{fb} \times \mathsf{Succ}(t + (n_k n_{fa} n_{fb} + n_k n_{fa} - 2)t_f + (n_k - 1)t_{\mathsf{Ggen}})$$

$(n_k \times LR')\{\mathbf{return}(\mathrm{true})/\mathbf{event}\ \mathrm{compute}\}$
$(n_k \times LR')\{\mathbf{return}(\mathrm{false})/\mathbf{event}\ \mathrm{compute}\}$
を考える.

ShoupのDifference Lemmaより,

$$(n_k \times LR')\{\mathbf{return}(\mathrm{true})/\mathbf{event}\ \mathrm{compute}\} \approx_p (n_k \times LR')\{\mathbf{return}(\mathrm{false})/\mathbf{event}\ \mathrm{compute}\}$$

となる. また,

$$(n_k \times LR')\{\mathbf{return}(\mathrm{true})/\mathbf{event}\ \mathrm{compute}\} \approx_0 L$$
$$(n_k \times LR')\{\mathbf{return}(\mathrm{false})/\mathbf{event}\ \mathrm{compute}\} \approx_0 R$$

$$L \approx_p R$$

目次

- 背景
  - Game列による安全性証明
  - Blanchetらの方法 **[BP06]**
- 本発表の目的
- CDH仮定の定式化
  - Observational Equivalence
  - Blanchetのプロセス計算による定式化
  - 証明
- **DDH仮定の定式化の方針**
- **まとめ**

Decisional Diffie-Hellman 仮定 (DDH仮定)

$$\mathbb{G} = \{G_{\mathfrak{p}}\}$$

$$Adv(\mathcal{A}) = \Pr\begin{bmatrix} r \xleftarrow{R} seed, \\ (\mathfrak{p}, g) \leftarrow \mathsf{Ggen}(r), \\ (a, b, c) \xleftarrow{R} [1, |G_{\mathfrak{p}}|]^3, \\ "true" \leftarrow \mathcal{A}(\mathfrak{p}, g, g^a, g^b, g^{ab}) \end{bmatrix} - \Pr\begin{bmatrix} r \xleftarrow{R} seed, \\ (\mathfrak{p}, g) \leftarrow \mathsf{Ggen}(r), \\ (a, b, c) \xleftarrow{R} [1, |G_{\mathfrak{p}}|]^3, \\ "true" \leftarrow \mathcal{A}(\mathfrak{p}, g, g^a, g^b, g^c) \end{bmatrix}$$

時間t 以内では，いかなるAであっても

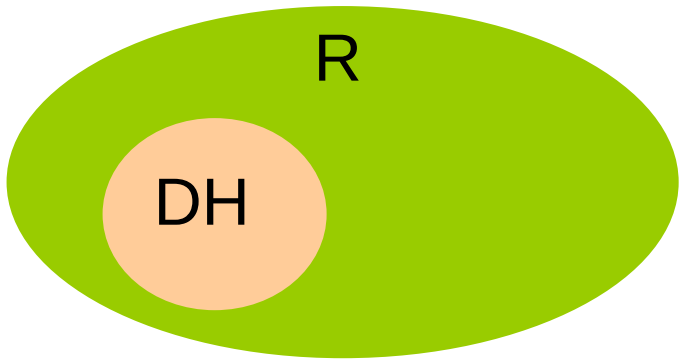
$$Adv(\mathcal{A}) \leq \epsilon$$

となるとき，$\mathbb{G}$ は(t, ε)-DDH仮定を満たすという．

$$DH = \{(g, g^a, g^b, g^{ab}) \mid a, b \in |\langle g \rangle|\}$$

$$R = \{(g, g^a, g^b, g^c) \mid a, b, c \in |\langle g \rangle|\}$$

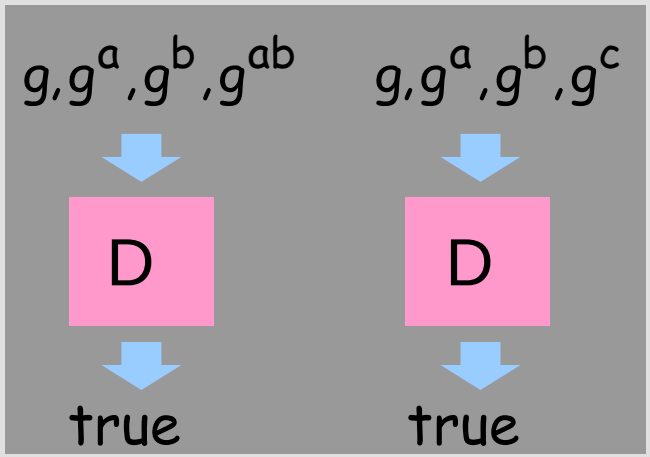
$$DH \subset R$$



識別問題を取り扱うには？

識別問題の仮定

DDH仮定



時間 t 以内で，この問題に対するアドバンテージは，高々 ε である．

$$L : O_{gr} := r \xleftarrow{R} seed; \mathbf{return};$$
$$(O_G := \mathbf{return}(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r))$$
$$|O_{gab}() := (a,b) \xleftarrow{R} [1, |G_{\mathrm{p}}|]^2; \mathbf{return};$$
$$(O_y() := \mathbf{return}(f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), a), f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), b),$$
$$f(\mathsf{Ggen1}(r), f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), a), b))$$
$$|O_T(z' : bool) := \mathbf{return}(z' \stackrel{?}{=} true))).$$

$$\approx_{\varepsilon}$$

$$R : O_{gr} := r \xleftarrow{R} seed; \mathbf{return};$$
$$(O_G := \mathbf{return}(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r))$$
$$|O_{gab}() := (a,b,c) \xleftarrow{R} [1, |G_{\mathrm{p}}|]^3; \mathbf{return};$$
$$(O_y() := \mathbf{return}(f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), a), f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), b),$$
$$f(\mathsf{Ggen1}(r), \mathsf{Ggen2}(r), c))$$
$$|O_T(z' : bool) := \mathbf{return}(z' \stackrel{?}{=} true))).$$

ほぼ自明に定式化可能．

攻撃モデル拡張の必要性は？

- **CDH仮定の定式化の際には攻撃モデルの拡張を行った.**

何故か？　ある署名方式への能動的攻撃が，CDH問題への能動的攻撃にあたるものがあるから.

- **では，DDH仮定は？**

KW署名では，拡張の必要はない.

もし，方式への攻撃が DDH仮定への能動的攻撃にあたるものがあるならば，攻撃モデルの拡張が必要.

まとめ

- **Blanchetのプロセス計算で**
  - 能動的攻撃モデルでCDH仮定が定式化できることを示した.
  - 受動的攻撃モデルならば, DDH仮定は ほぼ自明に定式化できることを示した.
    ただし, Observational Equivalenceの定義を満たすためには, アドバンテージ ε がある条件を満たす必要有.

- **Gap Diffie-Hellman仮定(DDH問題は簡単に解けるけど, CDH問題は難しい)の定式化について, 近く報告予定.**

- **今後の課題.**
  - 今回, 定式化したものを使った自動証明例の作成.


TOSHIBA
Leading Innovation