

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

形式的モデルで
自動証明

Dolev-Yao
Blanchet.

多くの研究が存在.

形式的モデルでの証明結果が,
計算量的モデルにおける安全性も
保証できる条件.

Abadi-Rogaway,
Canetti-Herzog

間接的手法

計算量的モデル
における安全性

直接的手法

暗号プリミティブ自体は形式化
せずに, 証明を行う.

Corin-Hartog,
Blanchet-Pointcheval,
Canetti et al.

計算量的モデルで
自動証明

CryptoVerif とは？

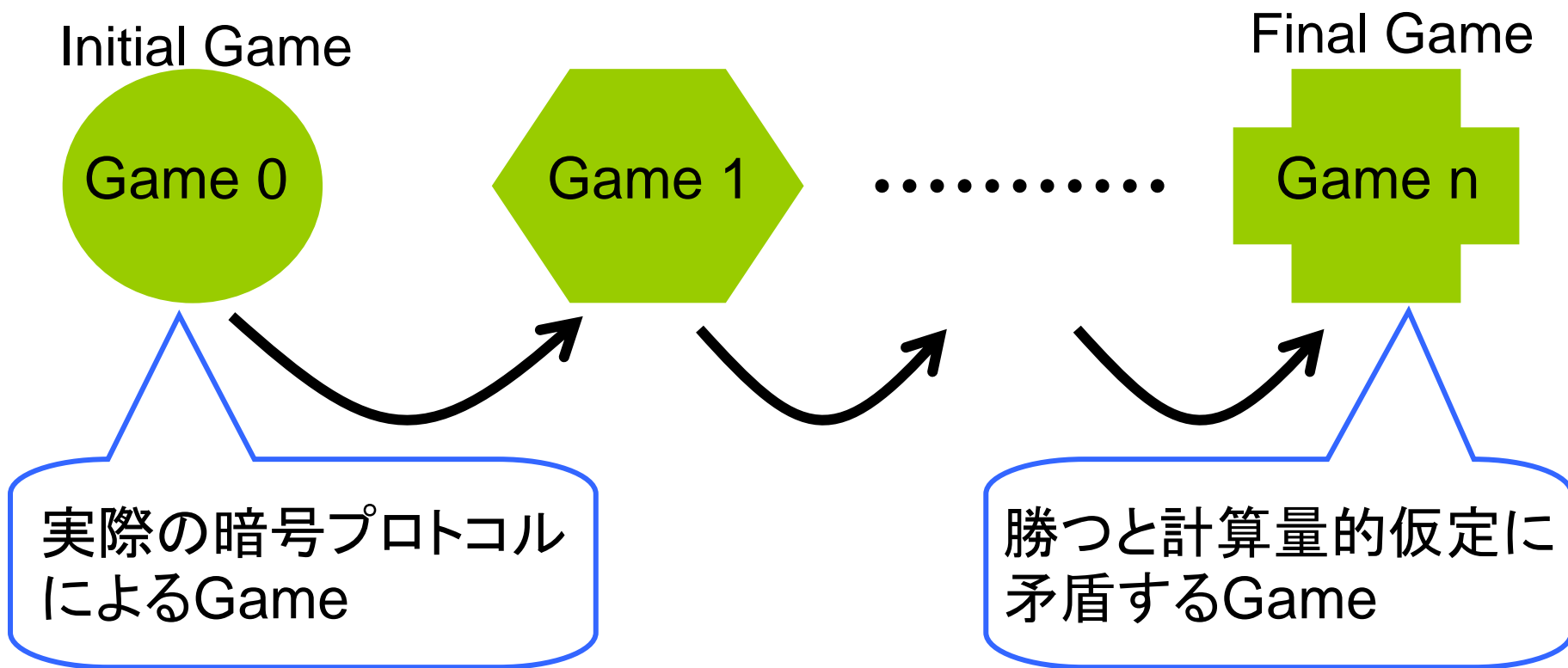
- **Game列による検証を自動で行うプログラム.**
 - Blanchetが作成. 自身のサイトで公開している.
 - FDH署名や OAEP暗号などの証明を行うためのサンプルファイルも公開
- **Game列による検証に便利なプロセス計算を提案.**
- **計算量的仮定は, **Observational Equivalence**を満たす式として記述.**
- **落とし戸付一方向性関数が存在するならば, FDH署名は **eUF-ACMA** であることを証明した.**
 - 証明結果は, Bellare, Rogaway [BR93] の結果と一致.

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。

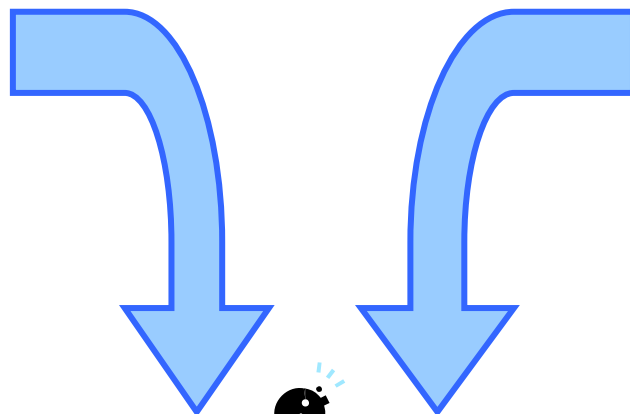


Game列による検証の自動化

Initial Game G_0



暗号プロトコルの
攻撃ゲーム



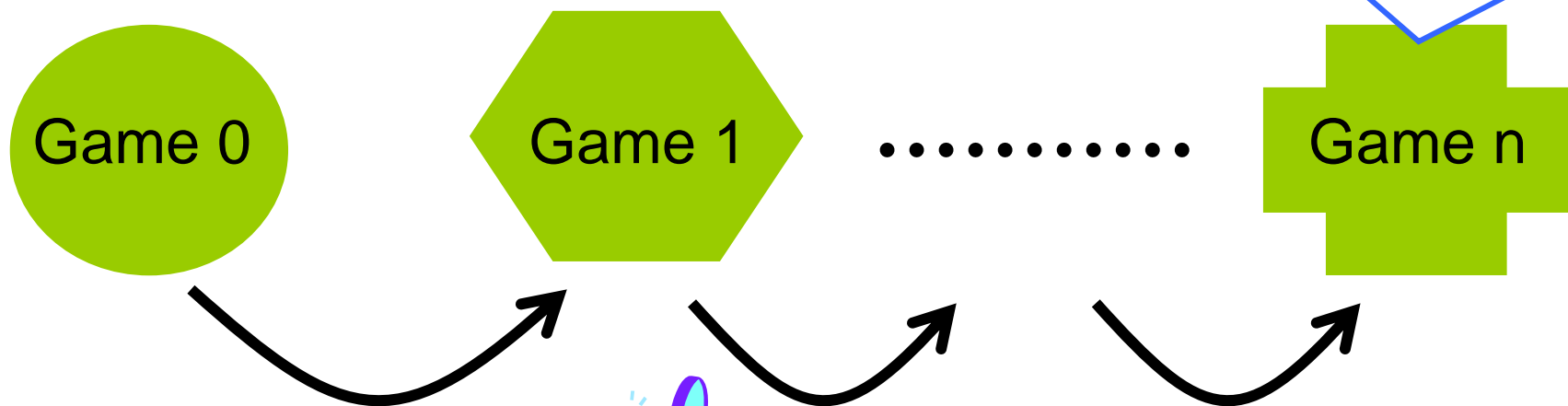
CryptoVerif

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

Game列による検証の自動化

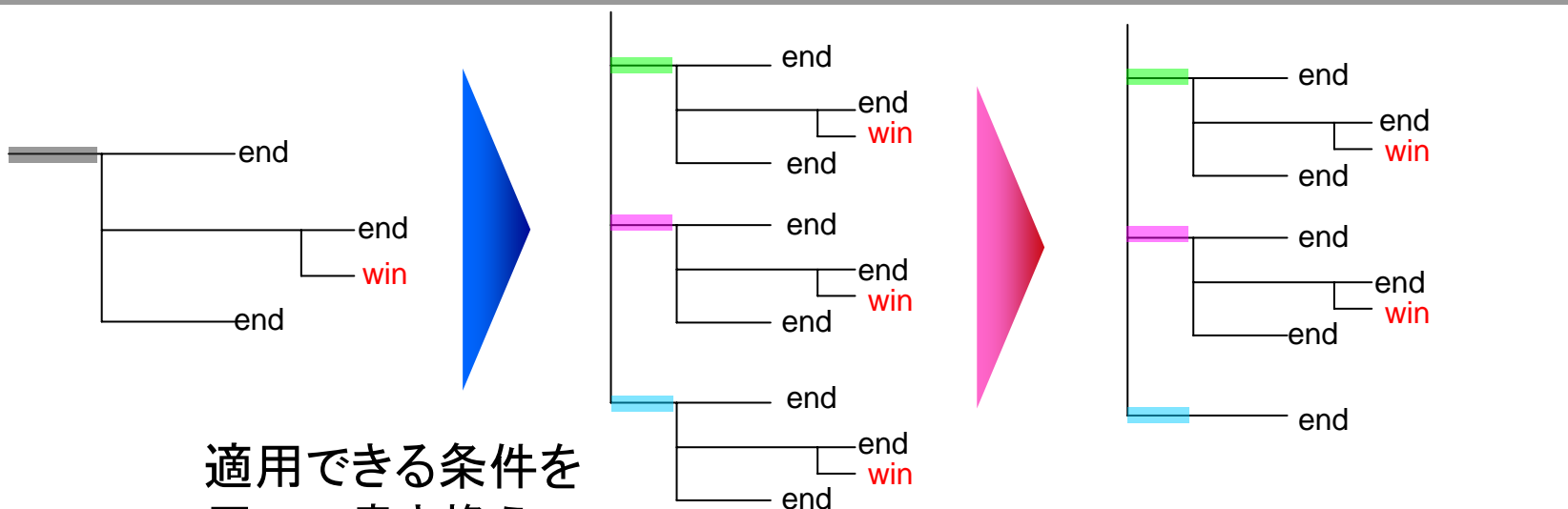
攻撃者が勝つイベントが存在しない。



CryptoVerif

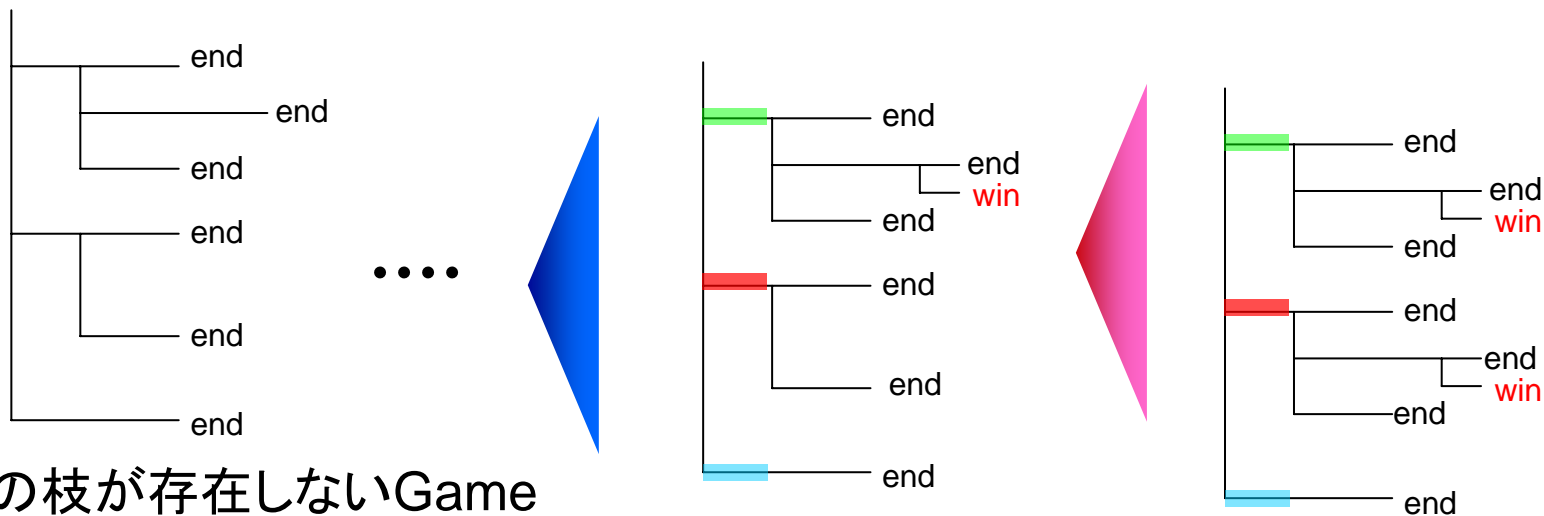
与えられた条件から
Game nに到達可能か
判定する。

CryptoVerif の動作イメージ



適用できる条件を用いて書き換え.

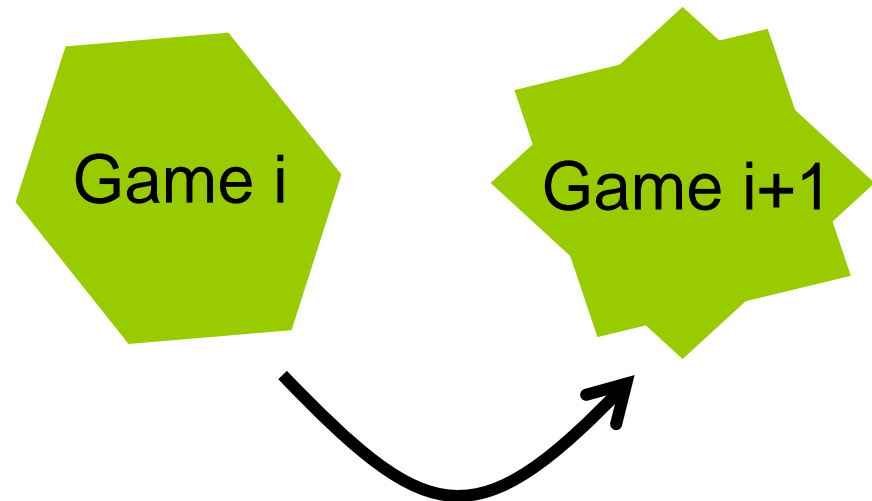
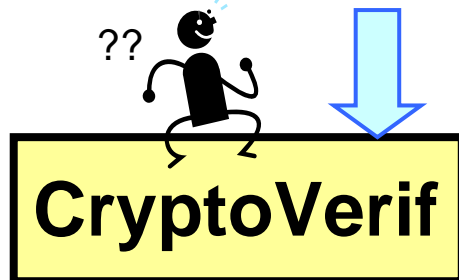
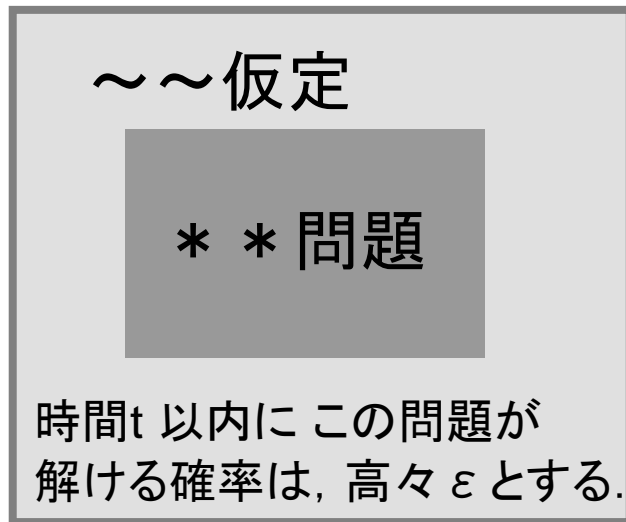
無駄な枝をCut



Win への枝が存在しないGame

計算量的仮定の取り扱い

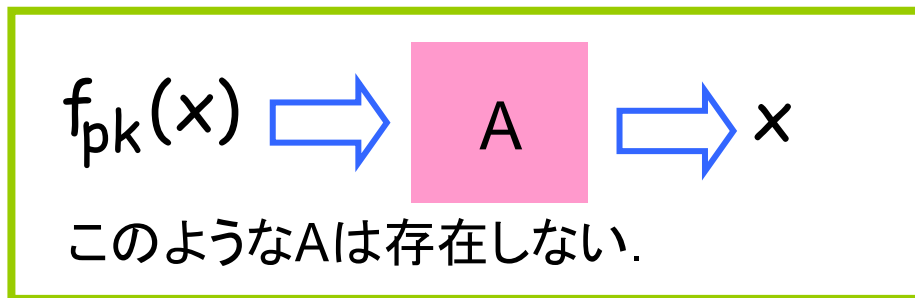
- **CryptoVerif**では、計算量的仮定をそのまま扱うことができない。
 - Blanchetのプロセス計算のルールに従って、Observational Equivalenceを満たす式として定式化しなければならない。
 - 人間が評価しなければならない部分も残されており、結構大変。



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

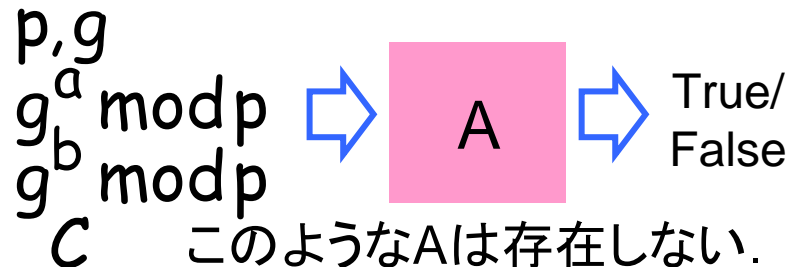
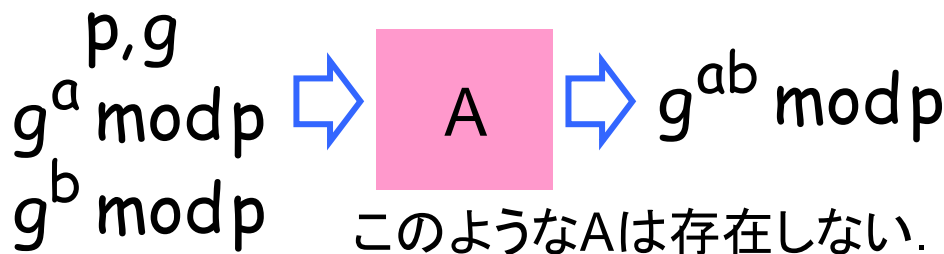
CryptoVerifで一方向性を取り扱うための記述は 既にある.



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

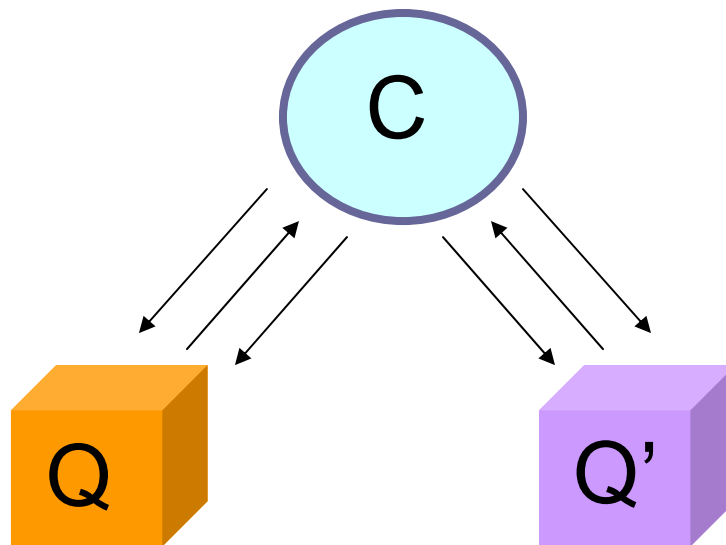
$$\forall C \forall a : (|\Pr[C[Q] \rightsquigarrow a] - \Pr[C[Q'] \rightsquigarrow a]| \leq p(t)) \\ \wedge \sum_{\mathcal{E}} |\Pr[C[Q] \rightsquigarrow \mathcal{E}] - \Pr[C[Q'] \rightsquigarrow \mathcal{E}]| \leq p(t))$$

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Q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)

$$\mathbb{G} = \{G_p\}$$

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

時間 t 以内では、いかなる \mathcal{A} であっても

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

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

CryptoVerifが取り扱える計算量的仮定

～～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



問題への能動的攻撃を行う攻撃モデルを定式化。

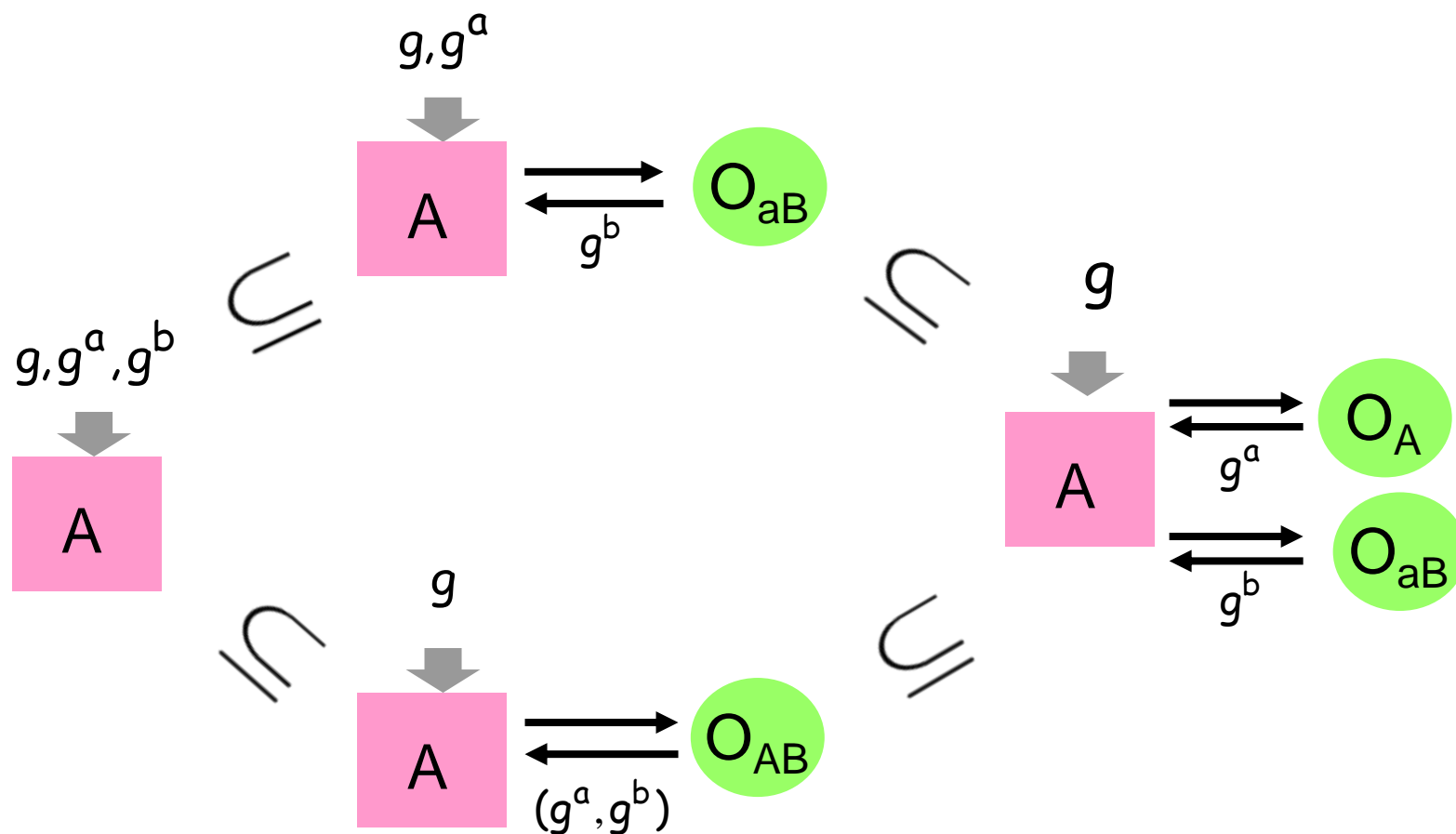
\approx_p

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.

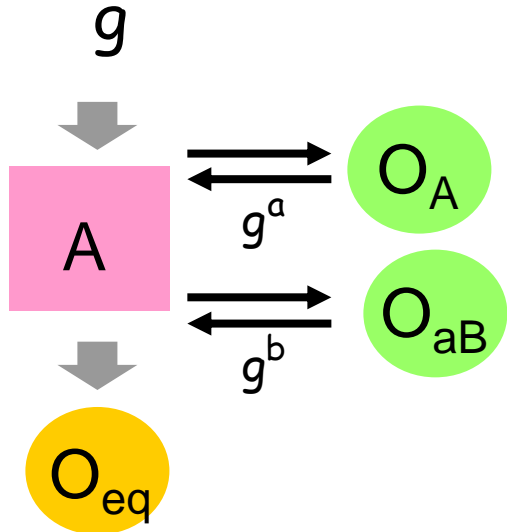


使用するオラクルによる攻撃のゴールの差

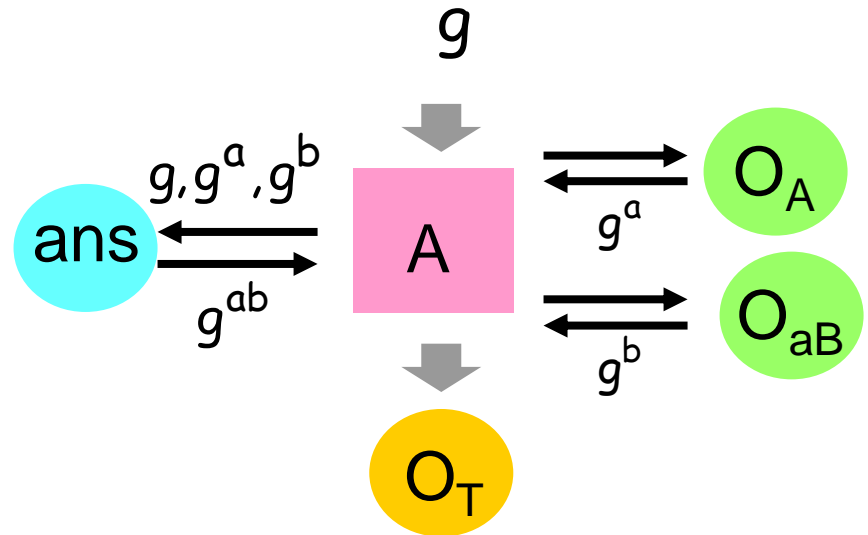
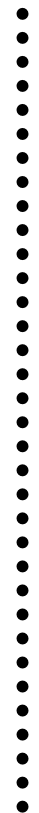
ans

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.

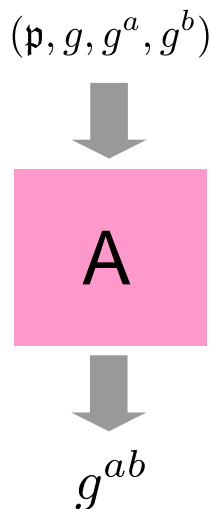


入力された組で, ans に
問い合わせしていないものの
正しい答えを返せば成功.

攻撃モデルの拡張

通常のCDH仮定

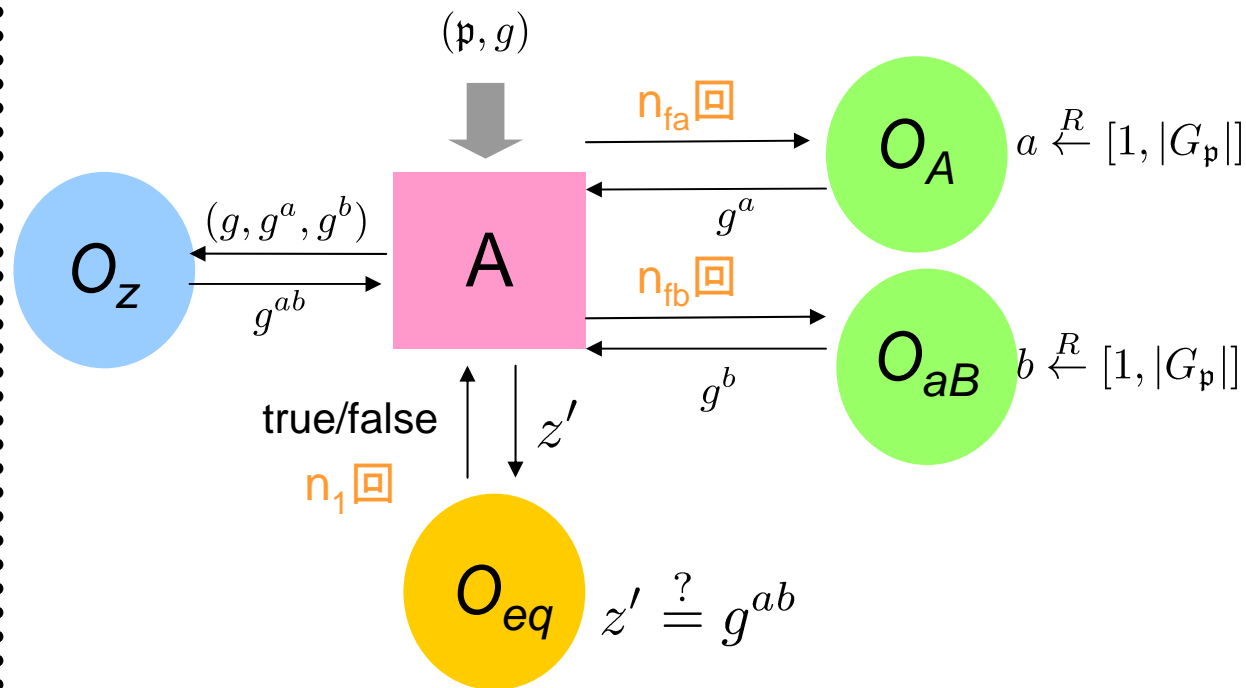
$r \xleftarrow{R} \text{seed},$
 $(p, g) \leftarrow \text{Ggen}(r),$
 $(a, b) \xleftarrow{R} [1, |G_p|]^2,$



攻撃モデルを拡張したCDH仮定

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

n_k 回

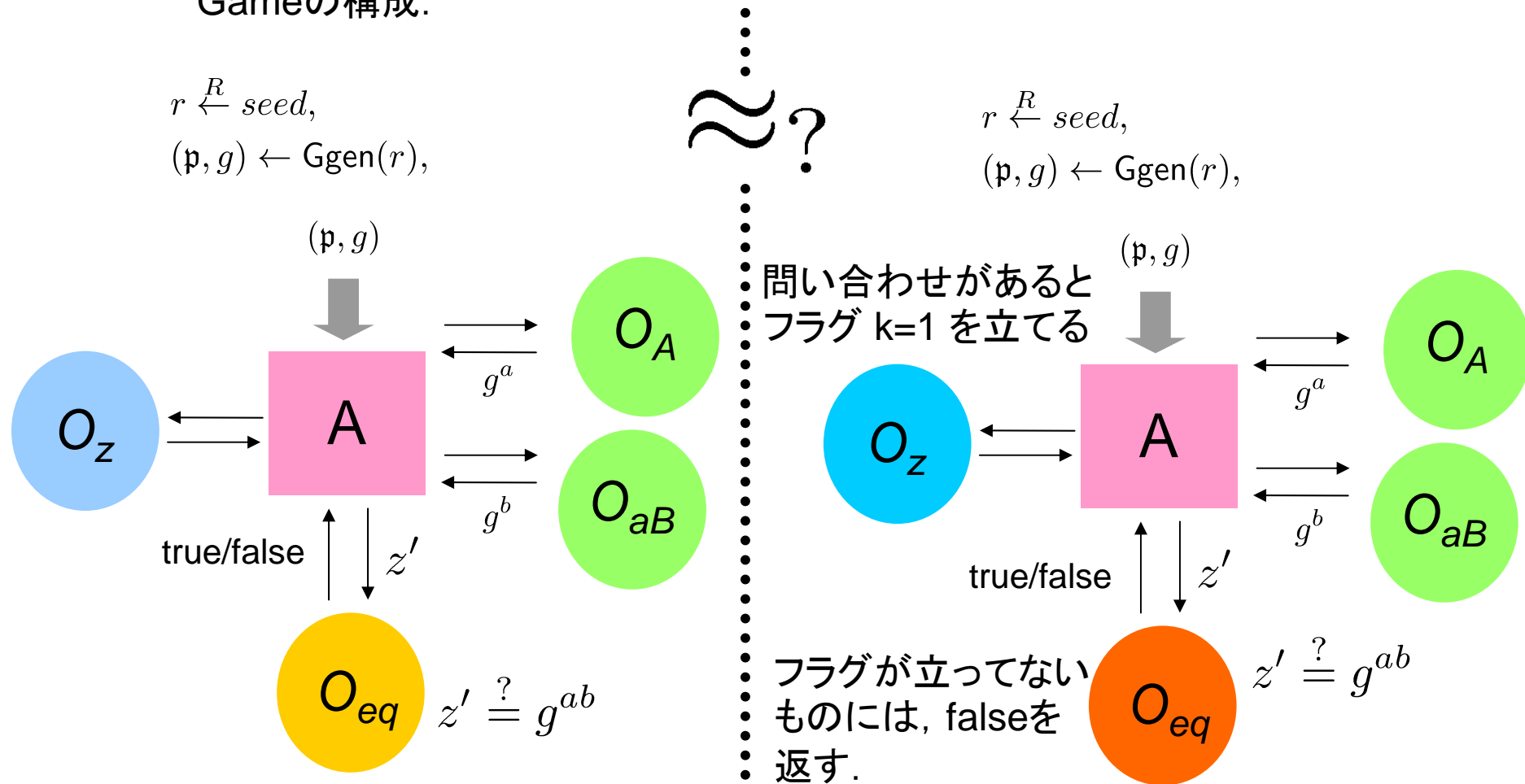


O_z に問い合わせしていない (g^a, g^b) に対して,
 $O_{eq}(z') \rightarrow \text{true}$ となったとき勝ち.

Observational Equivalent な関係で表現

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

目的: Ozに問い合わせしていない(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.



Computational Diffie-Hellman 仮定 (CDH仮定)

一般的な定式化

$$\mathbb{G} = \{G_p\}$$

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

時間 t 以内では、いかなる \mathcal{A} であっても

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

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  $O_{gr} := r \stackrel{R}{\leftarrow} \text{seed}$ ; return;
   ( $O_G := \text{return}(\text{Ggen1}(r), \text{Ggen2}(r))$ )
| foreach  $i_{fa} \leq n_{fa}$  do  $O_{ga}() := a \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;
| foreach  $i_{fb} \leq n_{fb}$  do  $O_{gb}() := b \stackrel{R}{\leftarrow} [1, |G_p|]$ ; return;
   ( $O_y() := \text{return}(f(\text{Ggen1}(r), \text{Ggen2}(r), a), f(\text{Ggen1}(r), \text{Ggen2}(r), b))$ )
| foreach  $i_1 \leq n_1$  do  $O_{eq'}(z' : G_p) := \text{return}(z' \stackrel{?}{=} f(\text{Ggen1}(r), f(\text{Ggen1}(r), \text{Ggen2}(r), a), b))$ 
|  $O_z() := \text{return}(f(\text{Ggen1}(r), f(\text{Ggen1}(r), \text{Ggen2}(r), a), b))$ )).
```

\approx_p

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

ただし、

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

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。

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

$(n_k \times LR')\{\text{return(true)}/\text{event compute}\}$
 $(n_k \times LR')\{\text{return(false)}/\text{event compute}\}$
を考える。

ShoupのDifference Lemmaより,

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

となる。また,

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

$$L \approx_p R$$

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)

$$\mathbb{G} = \{G_p\}$$

$$Adv(\mathcal{A}) = \Pr \left[\begin{array}{l} r \xleftarrow{R} \text{seed}, \\ (p, g) \leftarrow \text{Ggen}(r), \\ (a, b, c) \xleftarrow{R} [1, |G_p|]^3, \\ \text{"true"} \leftarrow \mathcal{A}(p, g, g^a, g^b, g^{ab}) \end{array} \right] - \Pr \left[\begin{array}{l} r \xleftarrow{R} \text{seed}, \\ (p, g) \leftarrow \text{Ggen}(r), \\ (a, b, c) \xleftarrow{R} [1, |G_p|]^3, \\ \text{"true"} \leftarrow \mathcal{A}(p, g, g^a, g^b, g^c) \end{array} \right]$$

時間 t 以内では、いかなる \mathcal{A} であっても

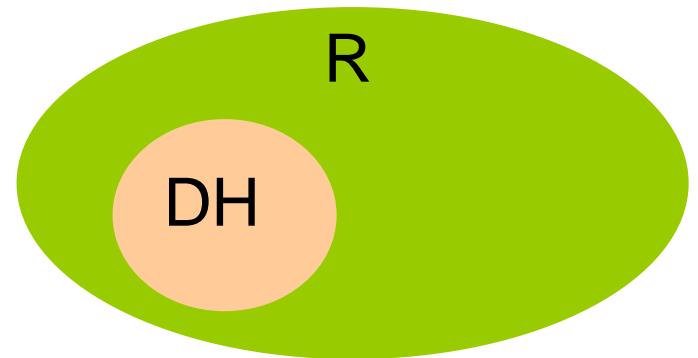
$$Adv(\mathcal{A}) \leq \epsilon$$

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

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

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

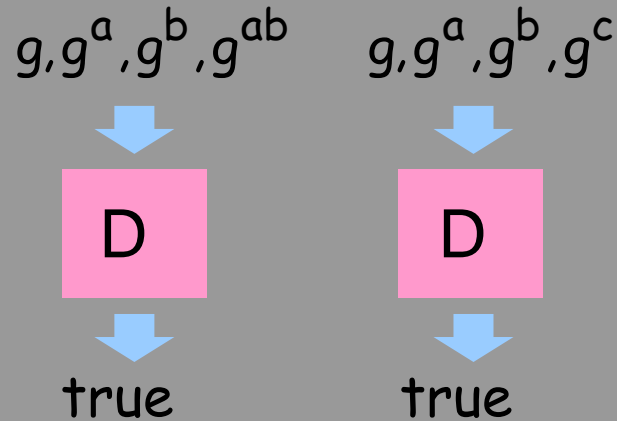
$$DH \subset R$$



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 : Ogr := r  $\stackrel{R}{\leftarrow}$  seed; return;  
      (OG := return(Ggen1(r), Ggen2(r))  
|Ogab() := (a, b)  $\stackrel{R}{\leftarrow}$  [1, |Gp|]2; return;  
      (Oy() := return(f(Ggen1(r), Ggen2(r), a), f(Ggen1(r), Ggen2(r), b),  
                    f(Ggen1(r), f(Ggen1(r), Ggen2(r), a), b))  
|OT(z' : bool) := return(z'  $\stackrel{?}{=} true$ )).
```


\approx_{ε}

```
R : Ogr := r  $\stackrel{R}{\leftarrow}$  seed; return;  
      (OG := return(Ggen1(r), Ggen2(r))  
|Ogab() := (a, b, c)  $\stackrel{R}{\leftarrow}$  [1, |Gp|]3; return;  
      (Oy() := return(f(Ggen1(r), Ggen2(r), a), f(Ggen1(r), Ggen2(r), b),  
                    f(Ggen1(r), Ggen2(r), c))  
|OT(z' : bool) := 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 >>>