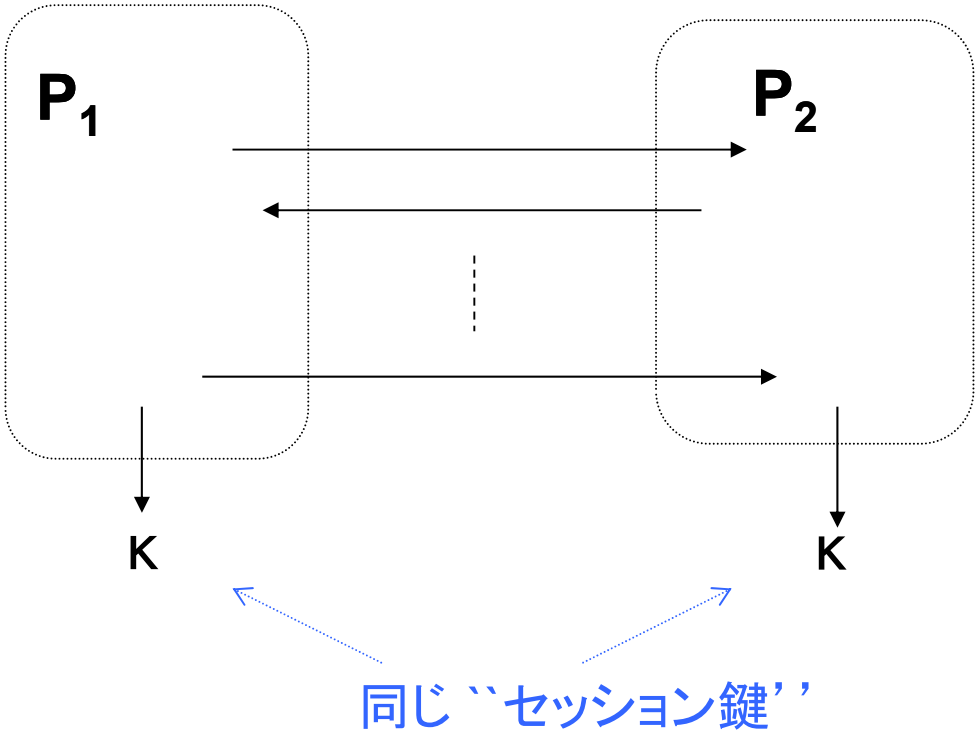
鍵共有プロトコル

IISEC

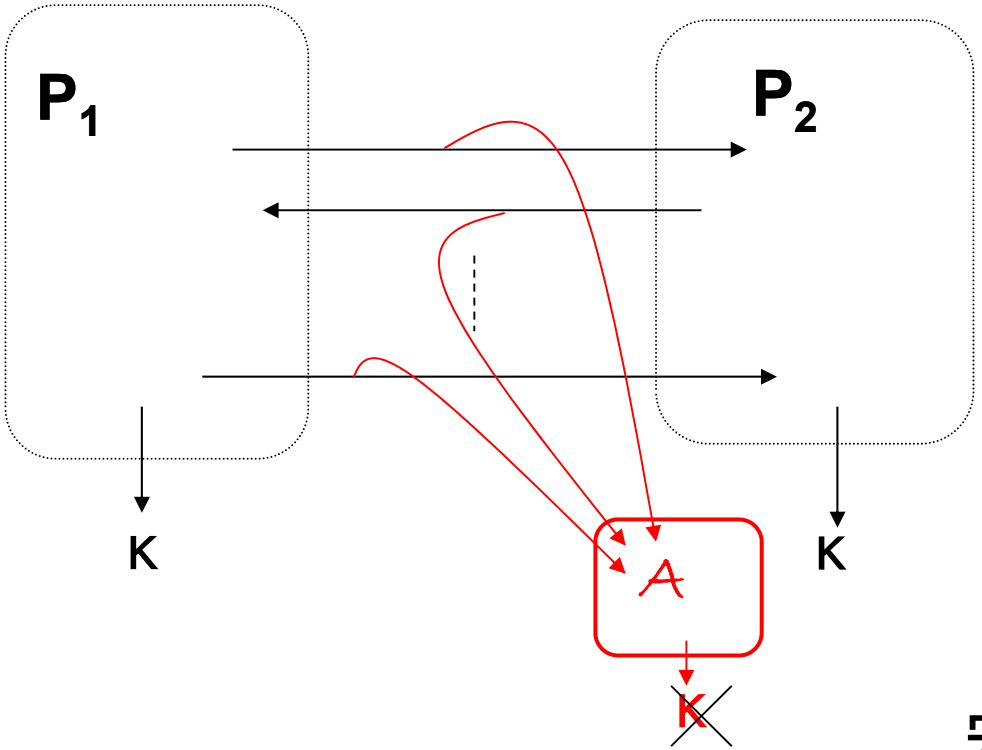
有田正剛

鍵共有プロトコルとは？



予め秘密を共有しておくことなしに、
いくつかのメッセージをやり取りすることで、
P_1とP_2のみが知ることになる秘密情報（セッション鍵）K
を共有するためのプロトコル

鍵共有プロトコルの安全性



安全性条件SK

P_1とP_2がやり取りする全てのメッセージを
敵Aが覗き見ても、
敵Aにはκのどのような部分情報も分からない。

そんなことが
可能なのか？

素数を法としたべき乗数

p: 素数
q: p-1 を割る素数
g : mod p で位数がqの整数

$<p, q, g>$ $\xrightarrow{\text{指定}}$ $\{1, g \bmod p, g^2 \bmod p, \ldots, g^{q-1} \bmod p\}$

全て異なる巨大な数の（算法つき）集合
$(g^q \bmod p = 1)$

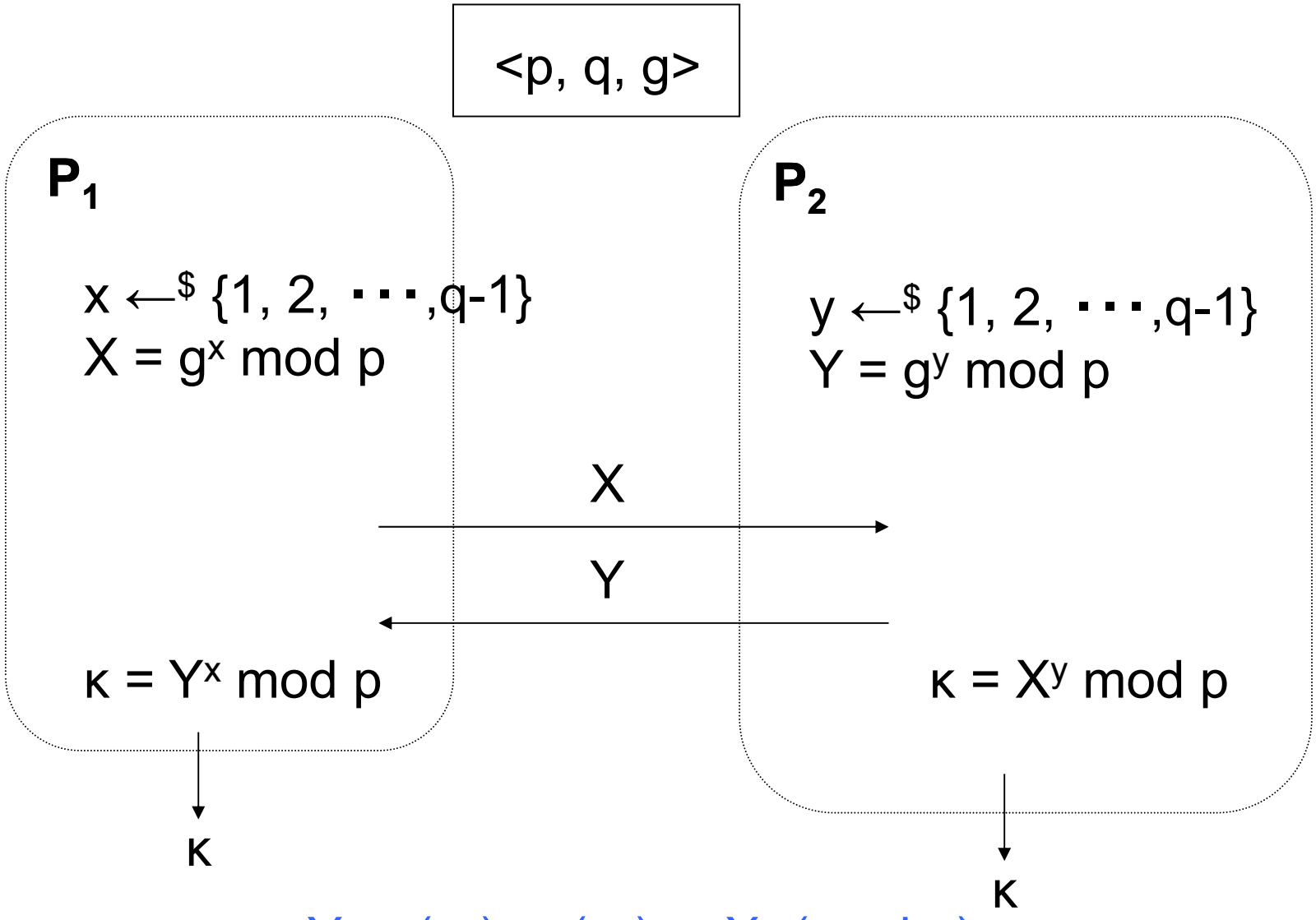
例

$<p=43, q=7, g=4>$

$g^0 \bmod p = 1$, $g \bmod p = 4$
$g^2 \bmod p = 16$, $g^3 \bmod p = 21$
$g^4 \bmod p = 41$, $g^5 \bmod p = 35$
$g^6 \bmod p = 11$

$<p=43, q=7, g=4>$ $\longrightarrow$ $\{1,4,16,21,41,35,11\}$

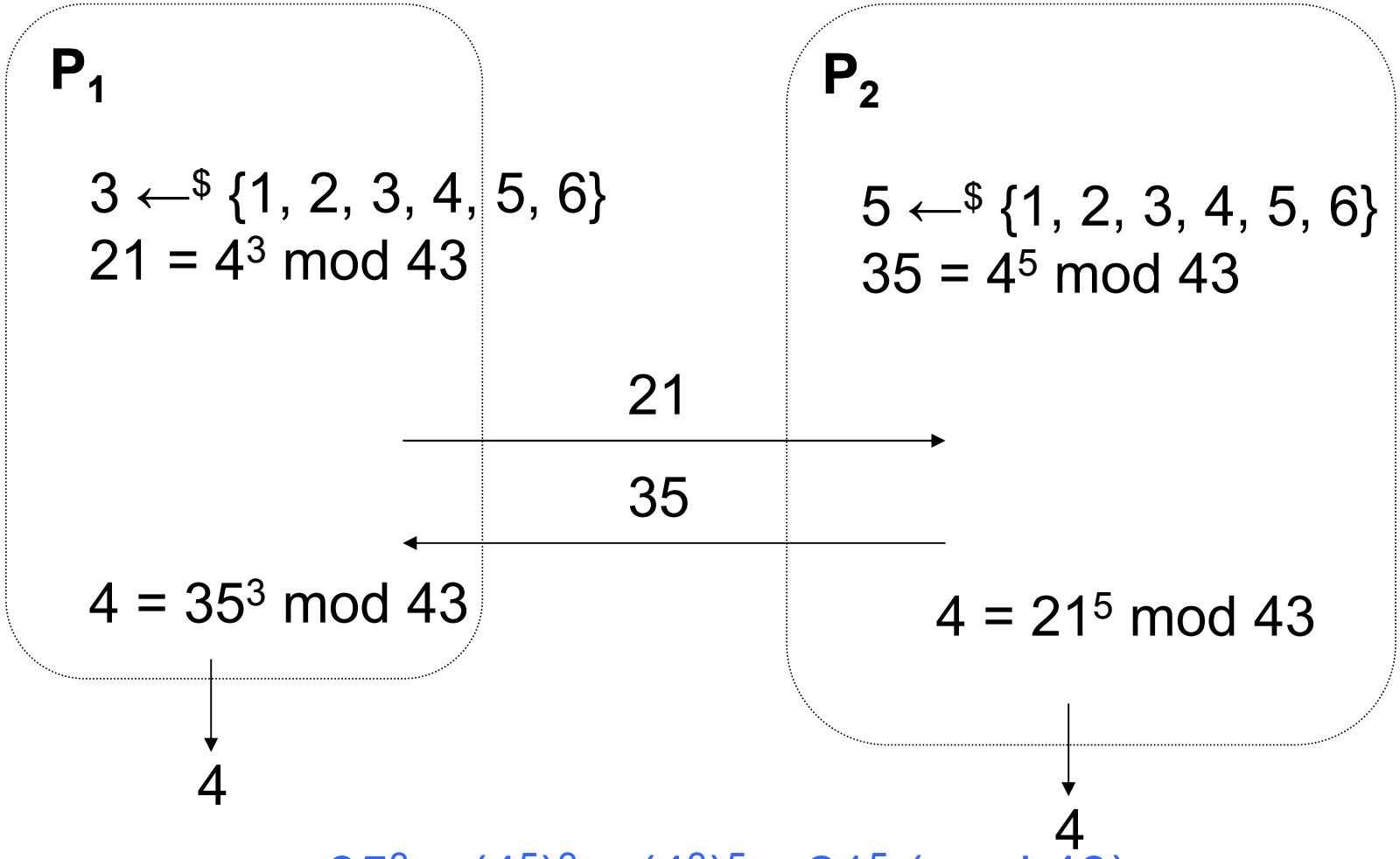
鍵共有プロトコルDH



$Y^x = (g^y)^x = (g^x)^y = X^y \pmod p$

鍵共有プロトコルDHの実行例

<p=43, q=7, g=4>



$35^3 = (4^5)^3 = (4^3)^5 = 21^5 \pmod{43}$

CDH仮定

p: 素数
q: p-1 を割る素数
g : mod p で位数がq

<p, q, g>に対して

CDH仮定

$a, b \xleftarrow{\$} \{1,2,\cdots,q\}$
どんな効率的なアルゴリズムも、$g^a \bmod p$ と $g^b \bmod p$ を与えられて、$g^{ab} \bmod p$ を求めることはできない。

- <p=43, q=7, g=4>
  $g^3 \bmod p = 21$, $g^5 \bmod p = 35$ $\xrightarrow{?}$ $g^{15} \bmod p = 4$

DDH仮定

<p, q, g>に対して

DDH仮定

$a, b, c \leftarrow^{\$} \{1,2,\cdots,q\}$
どんな効率的なアルゴリズムも、$g^a \bmod p$ と $g^b \bmod p$ を与えられても、$g^{ab} \bmod p$ と $g^c \bmod p$ を区別できない。

すなわち、

$g^a \bmod p$ と $g^b \bmod p$ を教わっても $g^{ab} \bmod p$ はまったく分からない！
（情けない話ではある）

- <p=43, q=7, g=4>
  (21, 35, 4) OR (21, 35, 11) どちらが正しい?

鍵共有プロトコルDHの安全性

<p, q, g>がDDH仮定を満たす。

→

鍵共有プロトコルDHは安全性条件SKを満たす。

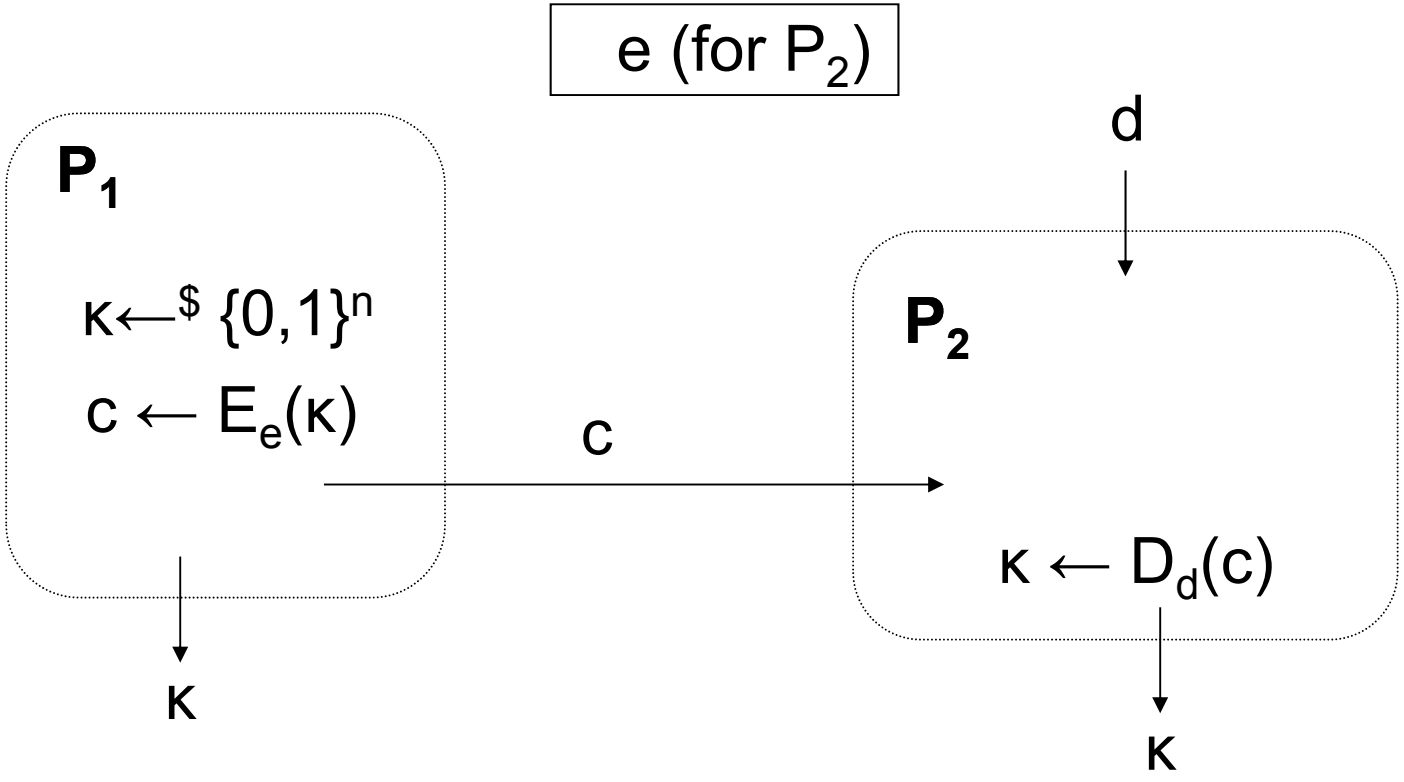
∵

DDH仮定より、
敵Aが$X(=g^x)$と$Y(=g^y)$を見ても$\kappa(=g^{xy})$について
何も分からない。

鍵共有プロトコルEB

（G, E, D): 公開鍵暗号、 n: セキュリティパラメータ

(e, d) ← G(n) /* e: 公開鍵, d: 私有鍵 */



鍵共有プロトコルEBの安全性

公開鍵暗号（G, E, D)がIND-CPA（暗号文からメッセージのどのような部分情報も漏れない）

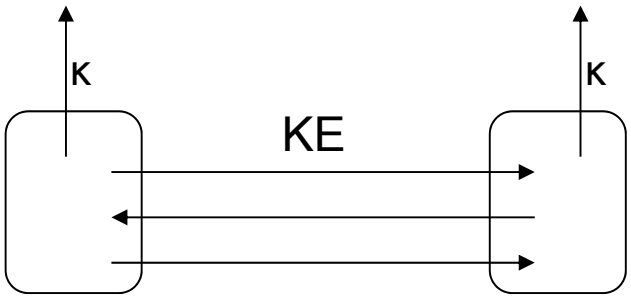
→

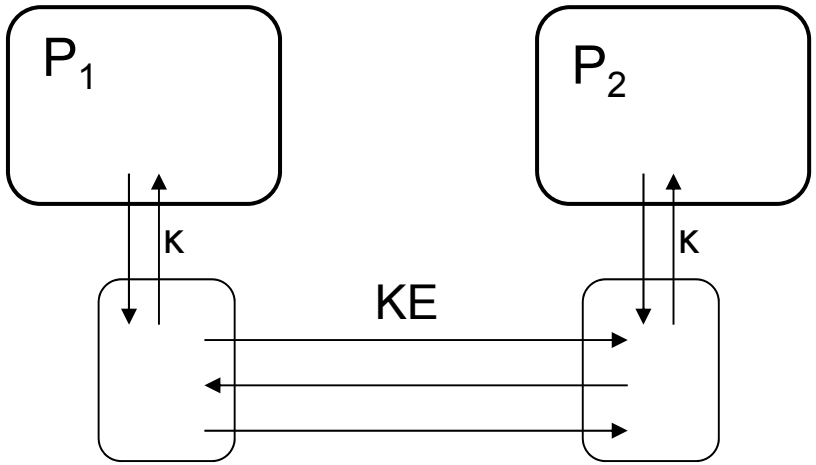
鍵共有プロトコルEBは安全性条件SKを満たす。

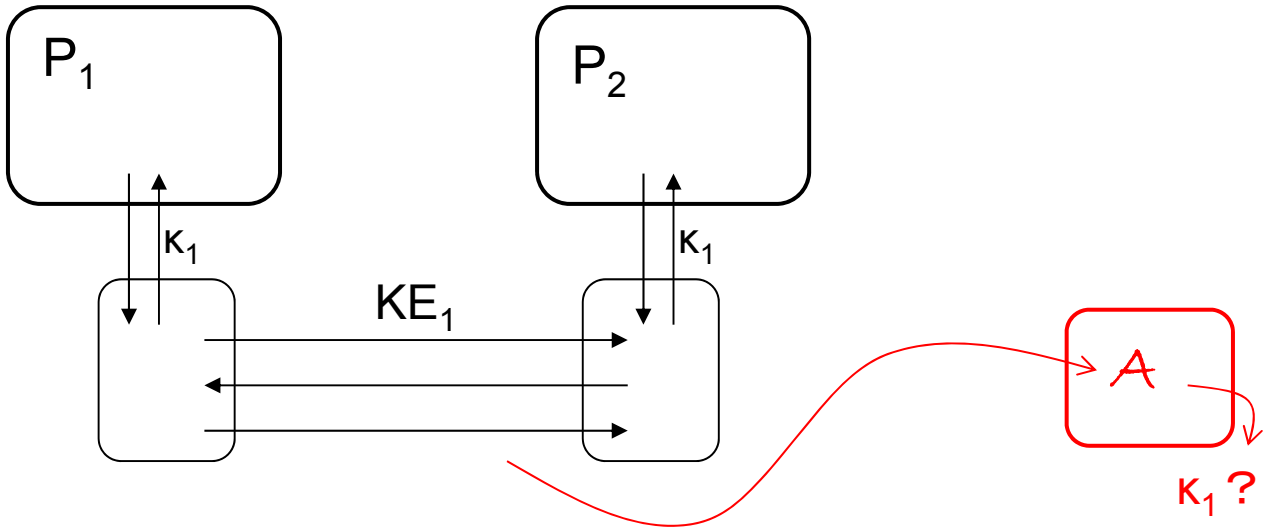
∵

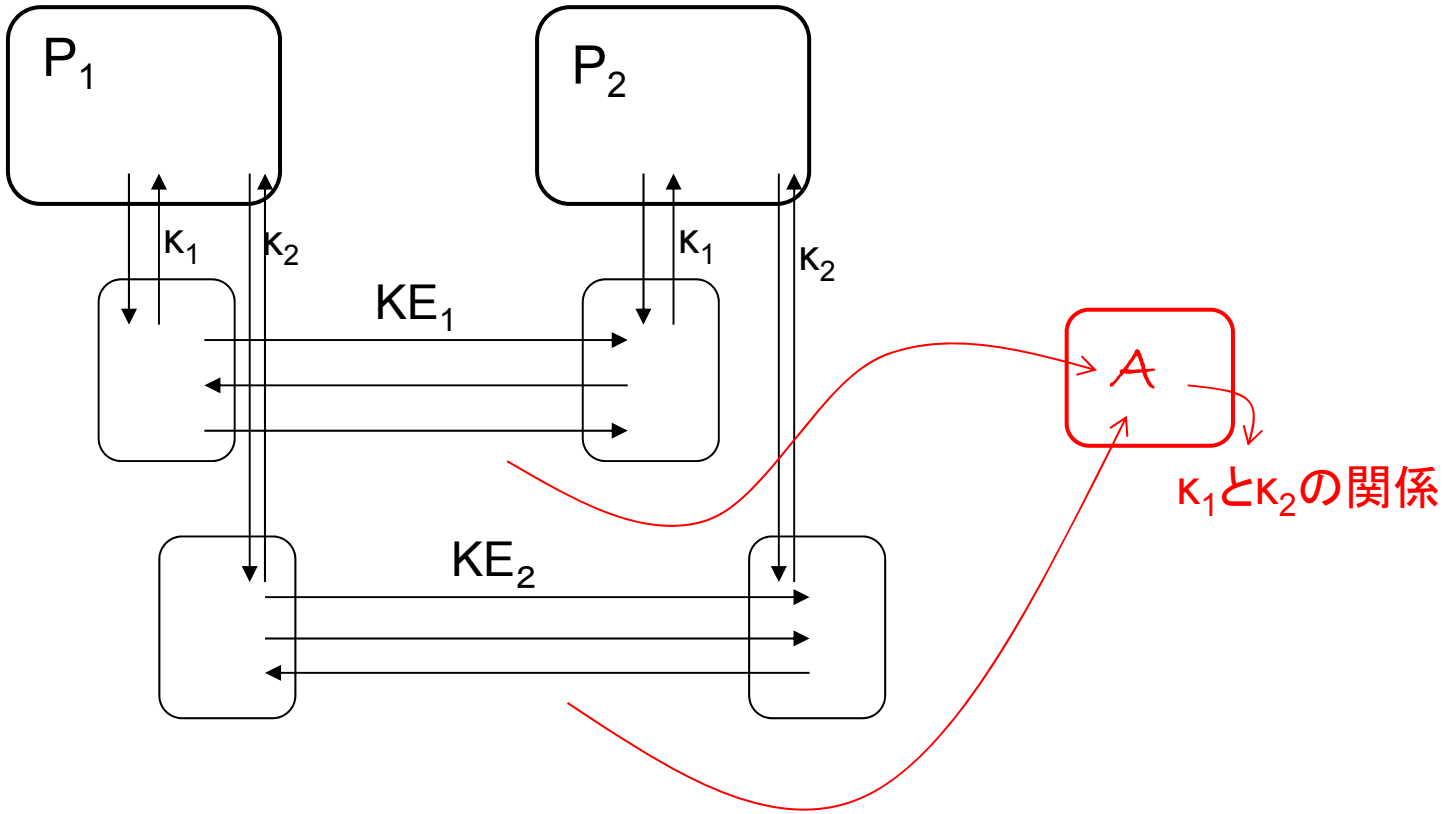
（G,E,D）のIND-CPAより、
敵がc (←$E_e(\kappa)$)を見てもκについて何も分からない。

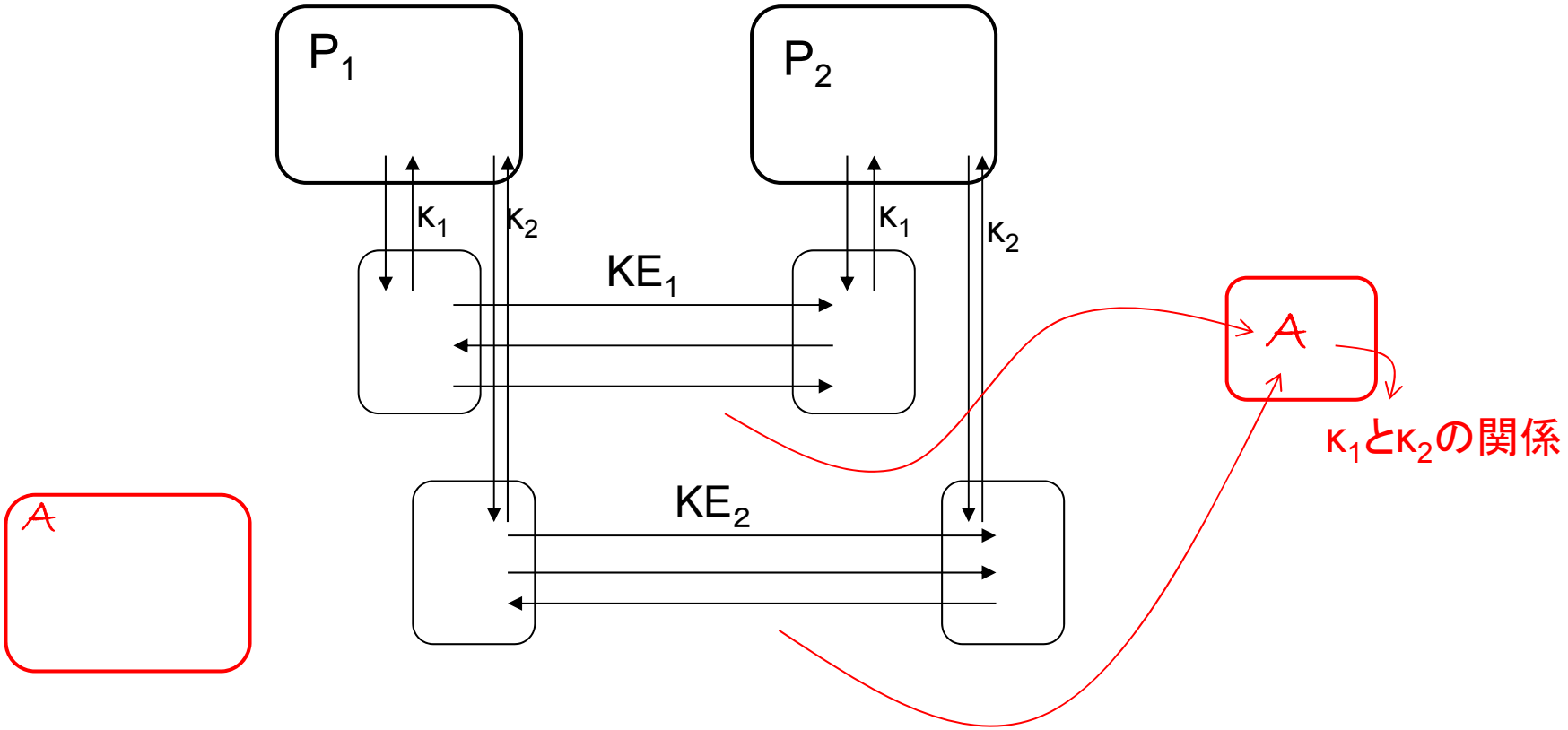
これでOKか？

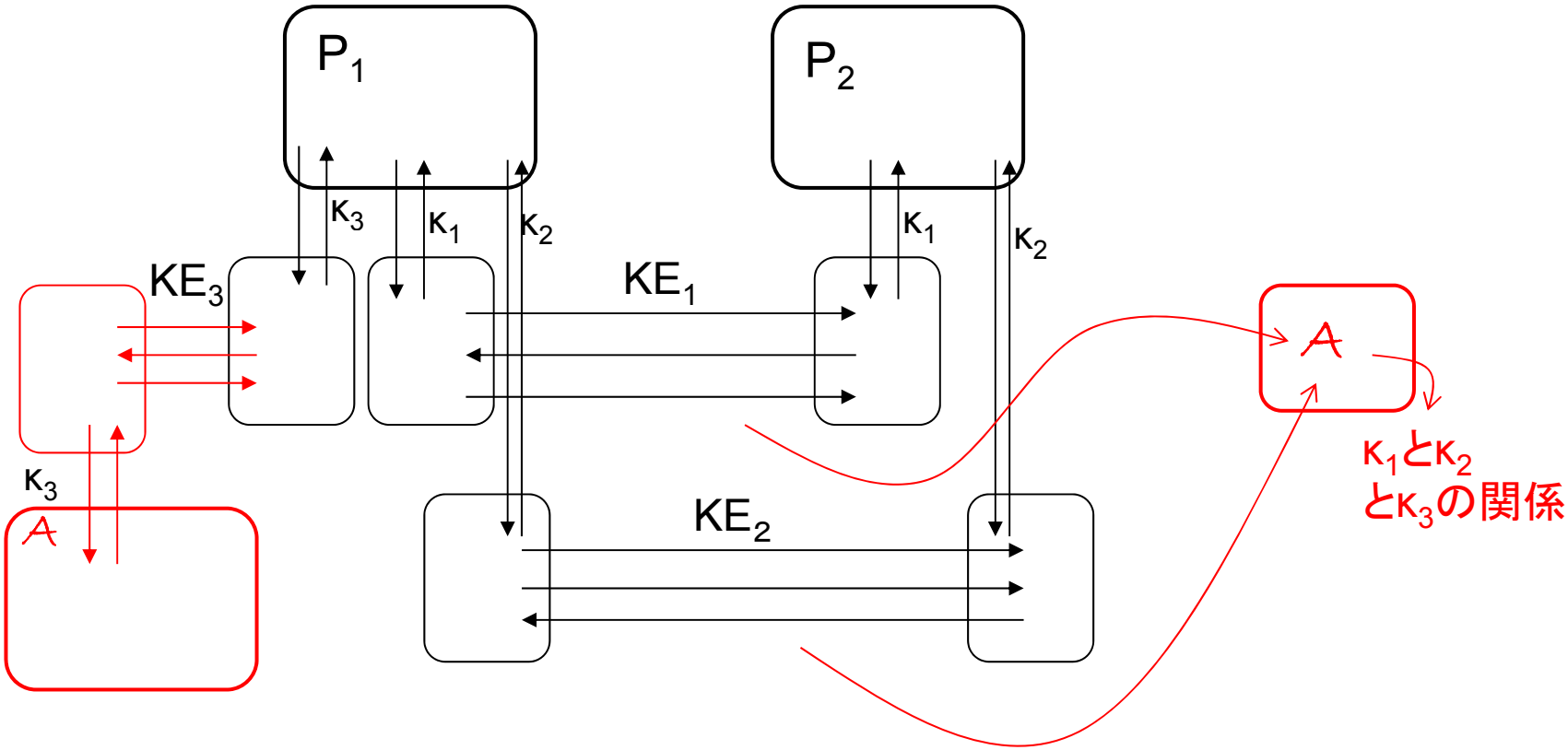

K
KE
K


P1
P2
K
K
KE


P1
P2
K1
K1
KE1
A
K1 ?

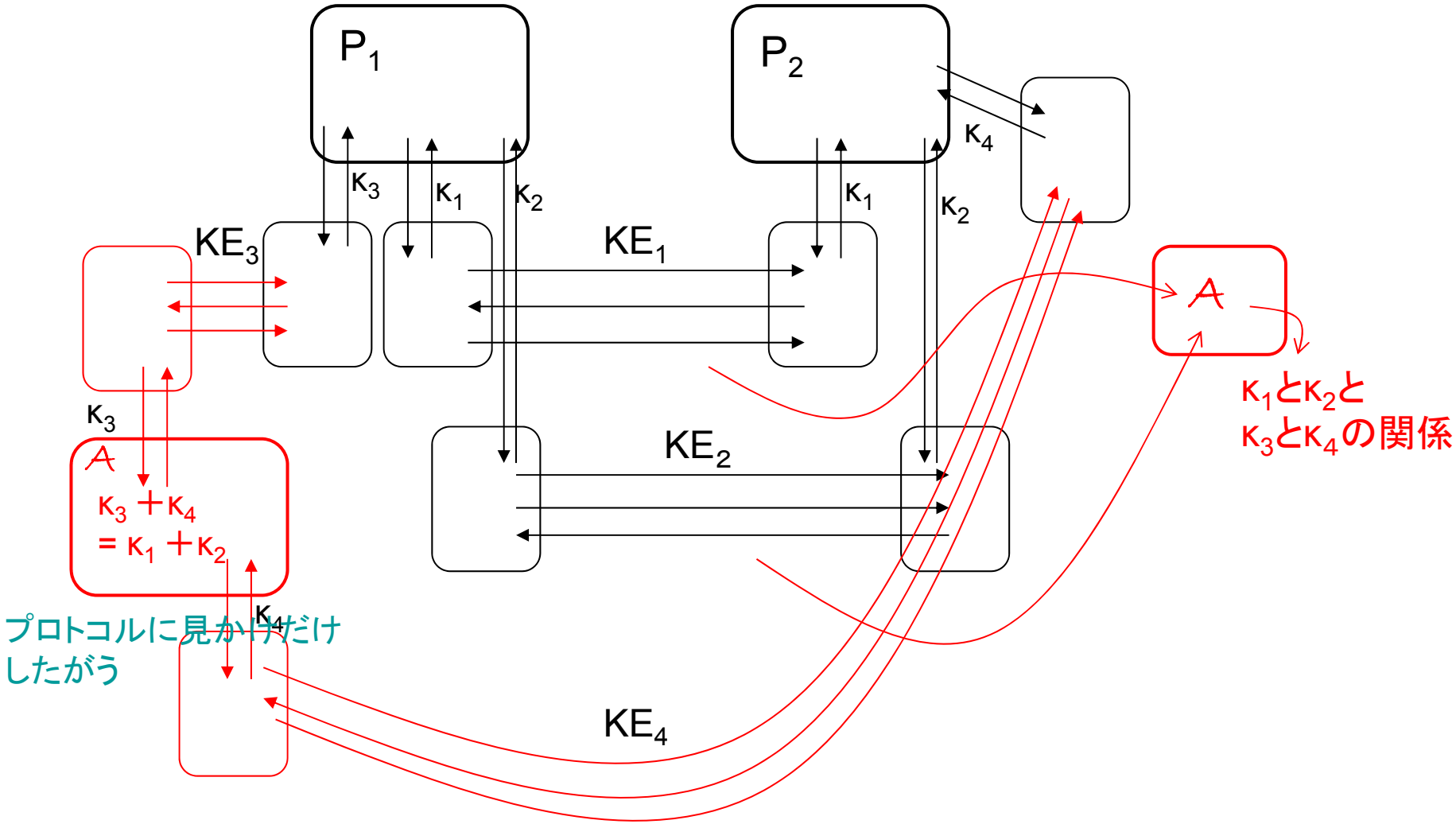

P1
P2
κ1
κ2
κ1
κ2
KE1
KE2
A
κ1とκ2の関係


P1
P2
K1
K2
K1
K2
KE1
KE2
A
A
K1とK2の関係


P1
P2
κ3
κ1
κ2
κ1
κ2
KE3
KE1
KE2
κ3
A
A
κ1とκ2
とκ3の関係

コンカレント環境における鍵共有

$g^a \bmod p$ と $g^b \bmod p$ を教わっても
$g^{ab} \bmod p$ と $g^c \bmod p$ との区別がつかない



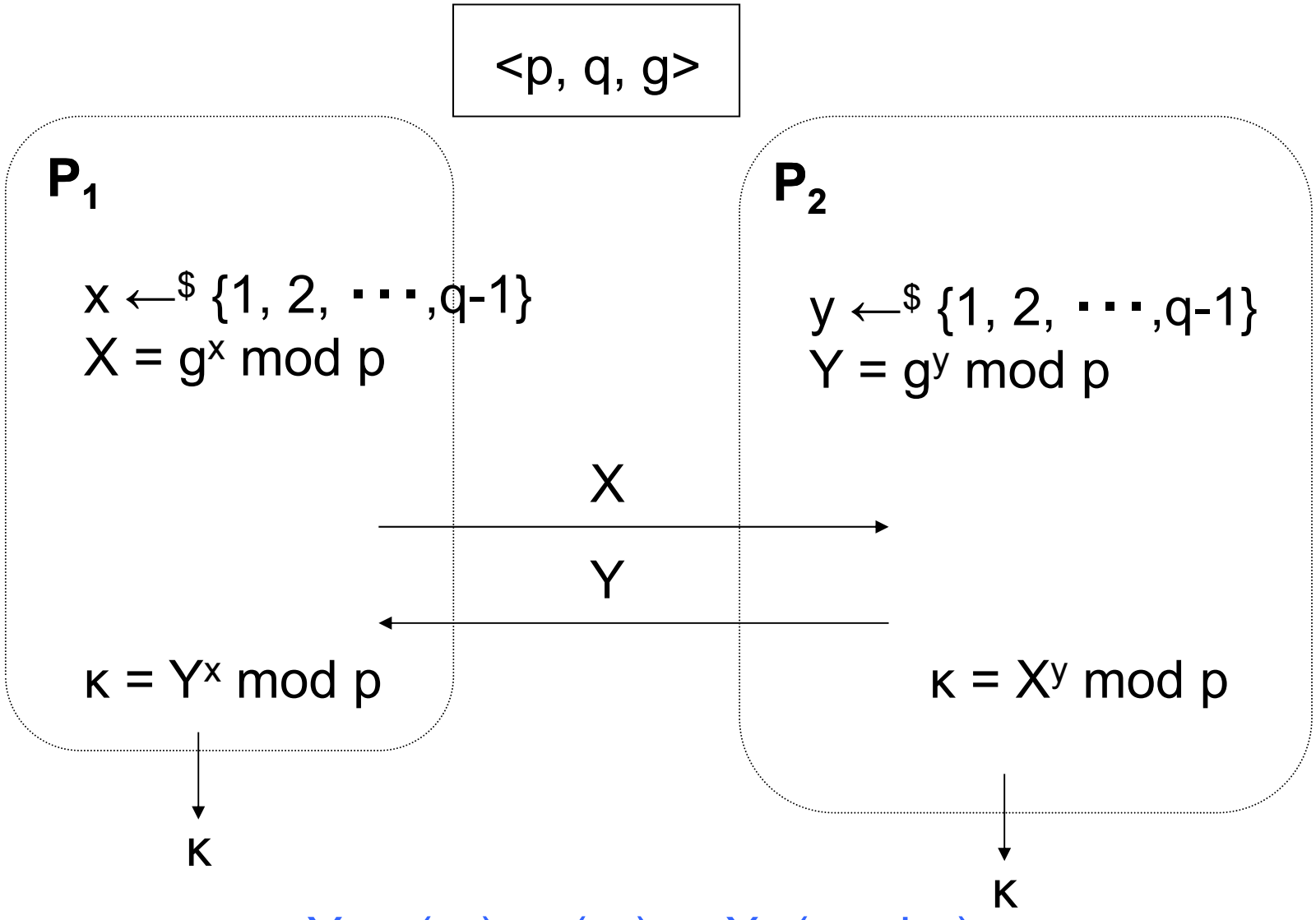
安全性条件CSK

（設定１） 鍵共有プロトコルは**同時に複数のコピー**が実行される。
（設定２） **敵A自身**も鍵共有プロトコルに**参加**できる。
（正当な参加者または、なりすまし）
（設定３） 敵Aは**終了した**任意の鍵共有プロトコルを選び、その
セッション鍵κを入手できる。

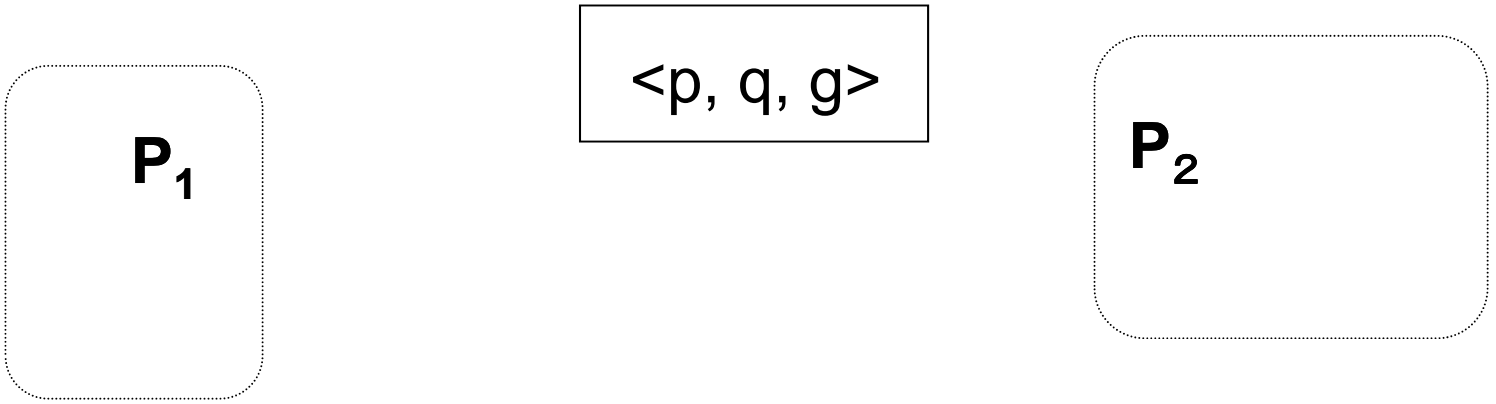
という環境において

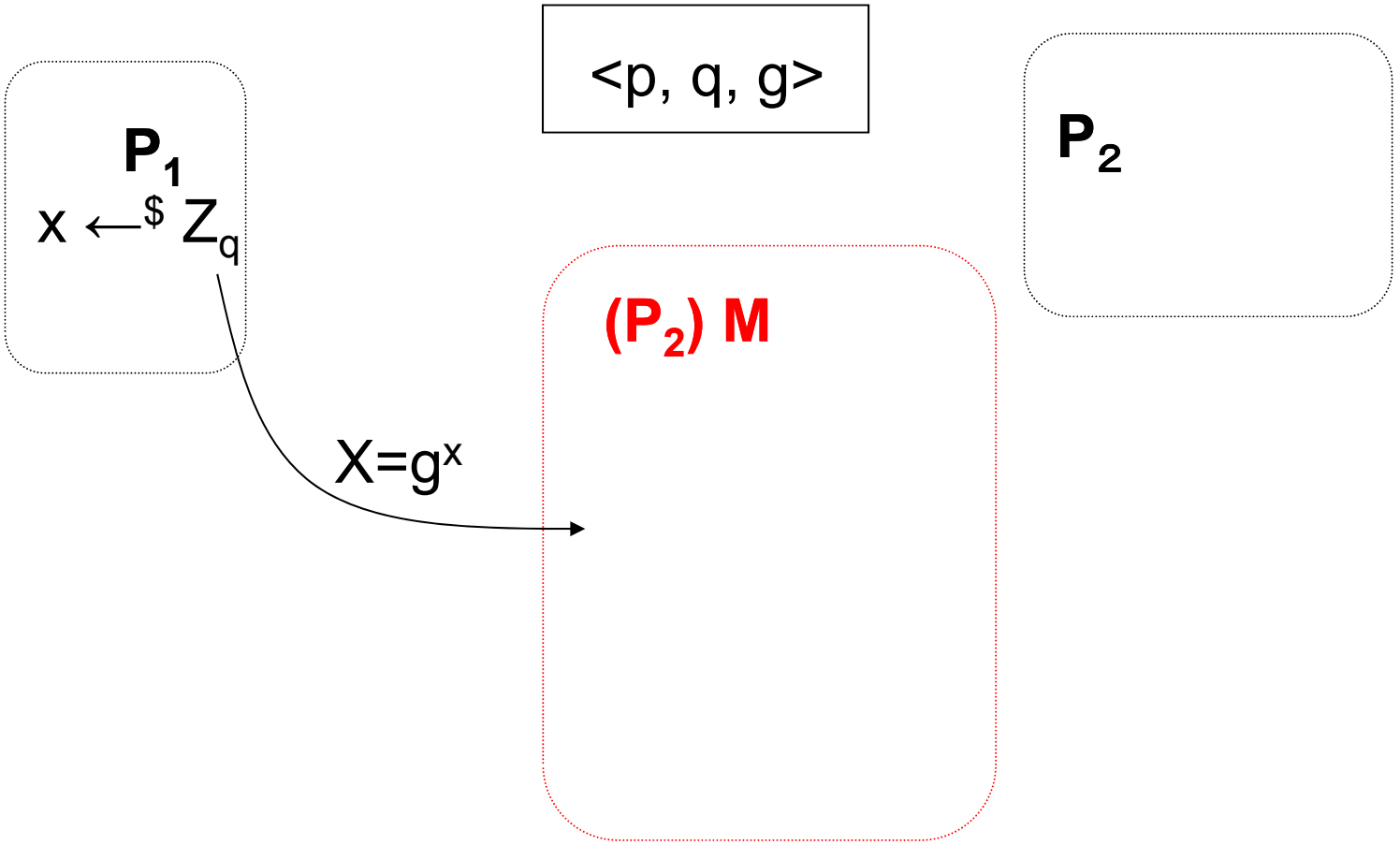
（条件１） **P_1とP_2さえ**正しく振舞えば、正しくセッション鍵κが
共有される。
（条件２） 全てのメッセージを敵Aが覗き見ても
（設定3の手段を用いない限り）セッション鍵κの
どのような部分情報も分からない。

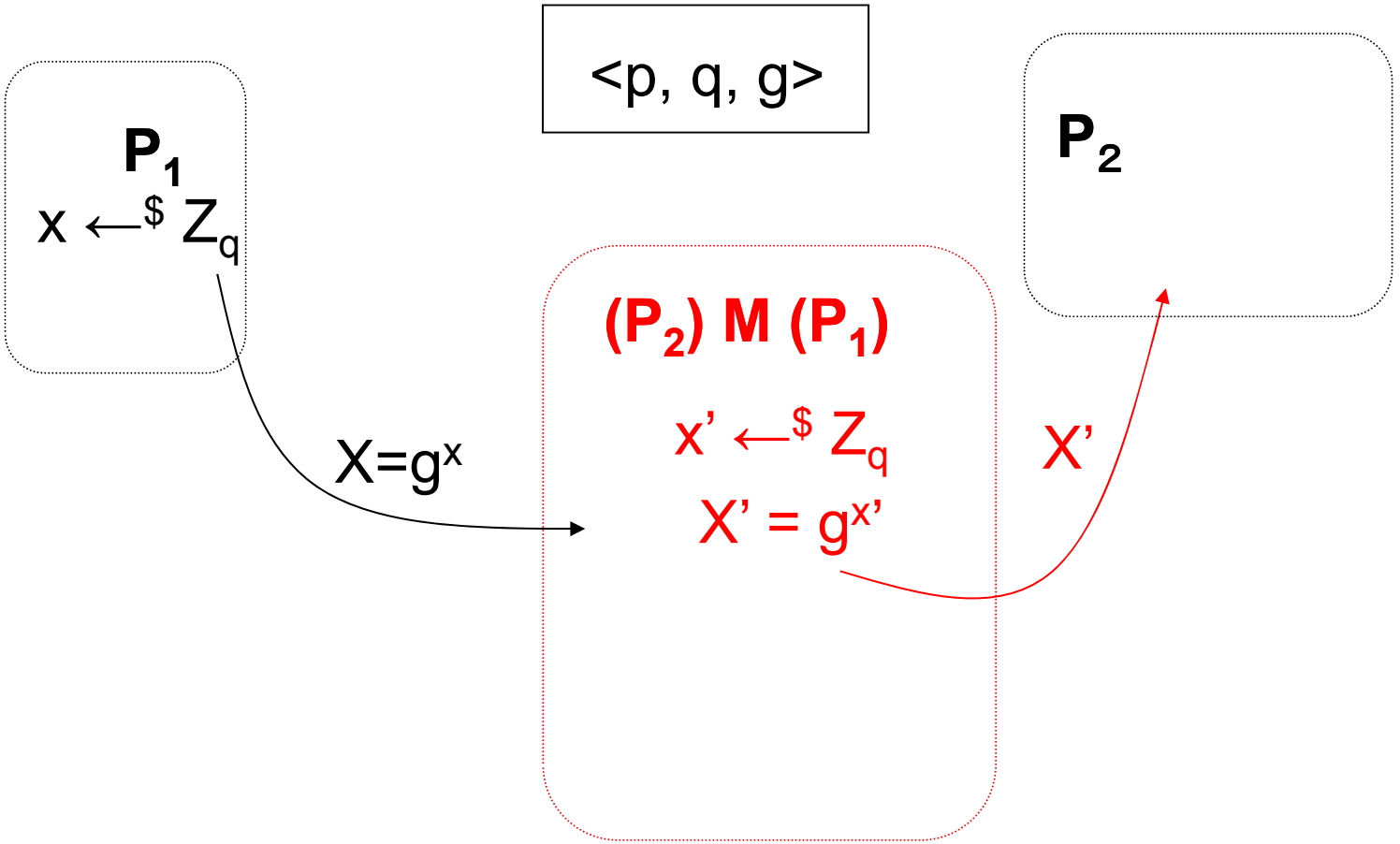
鍵共有プロトコルDH（再）

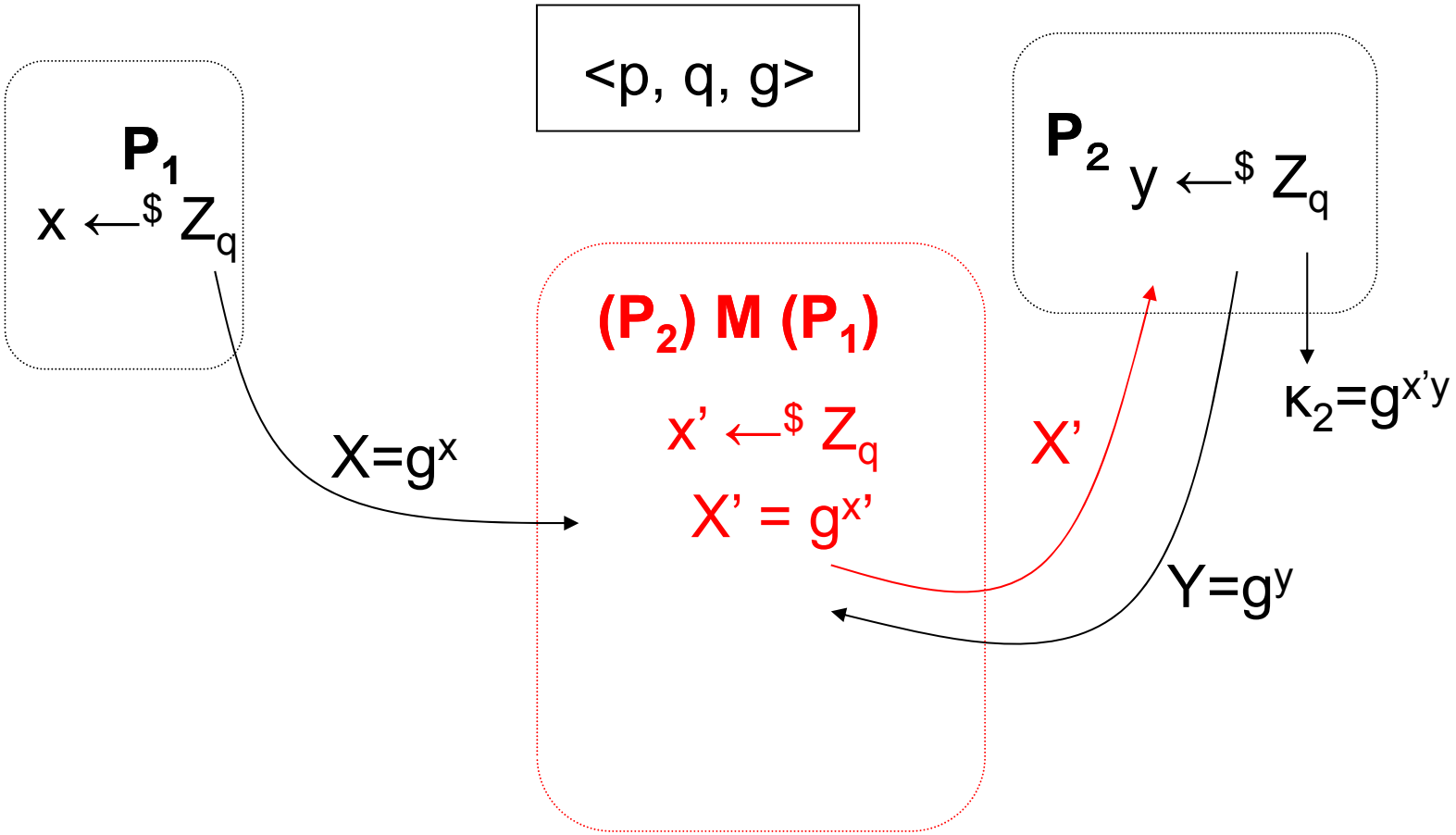


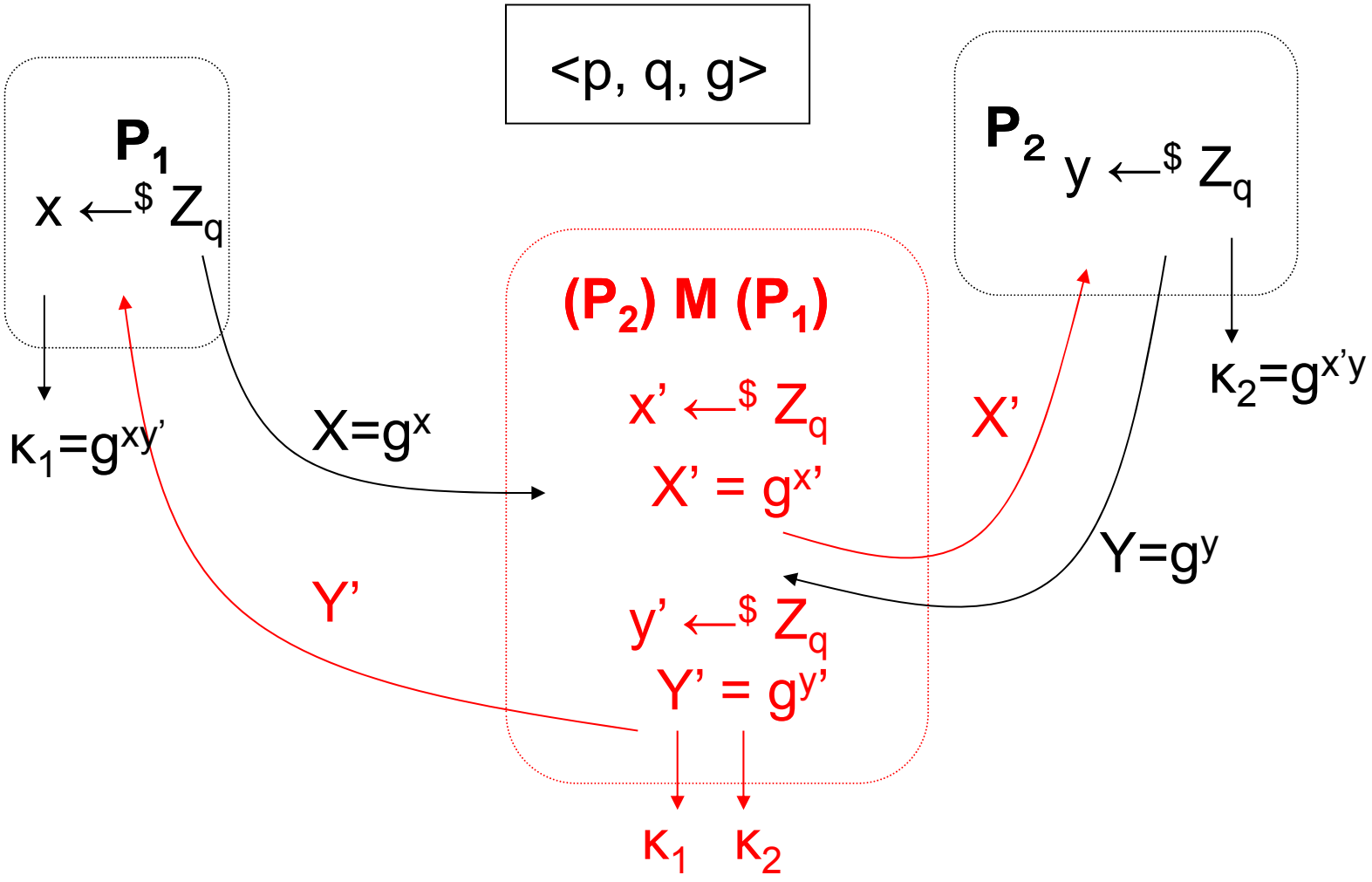
$Y^x = (g^y)^x = (g^x)^y = X^y \pmod p$


P_1
<p, q, g>
P_2

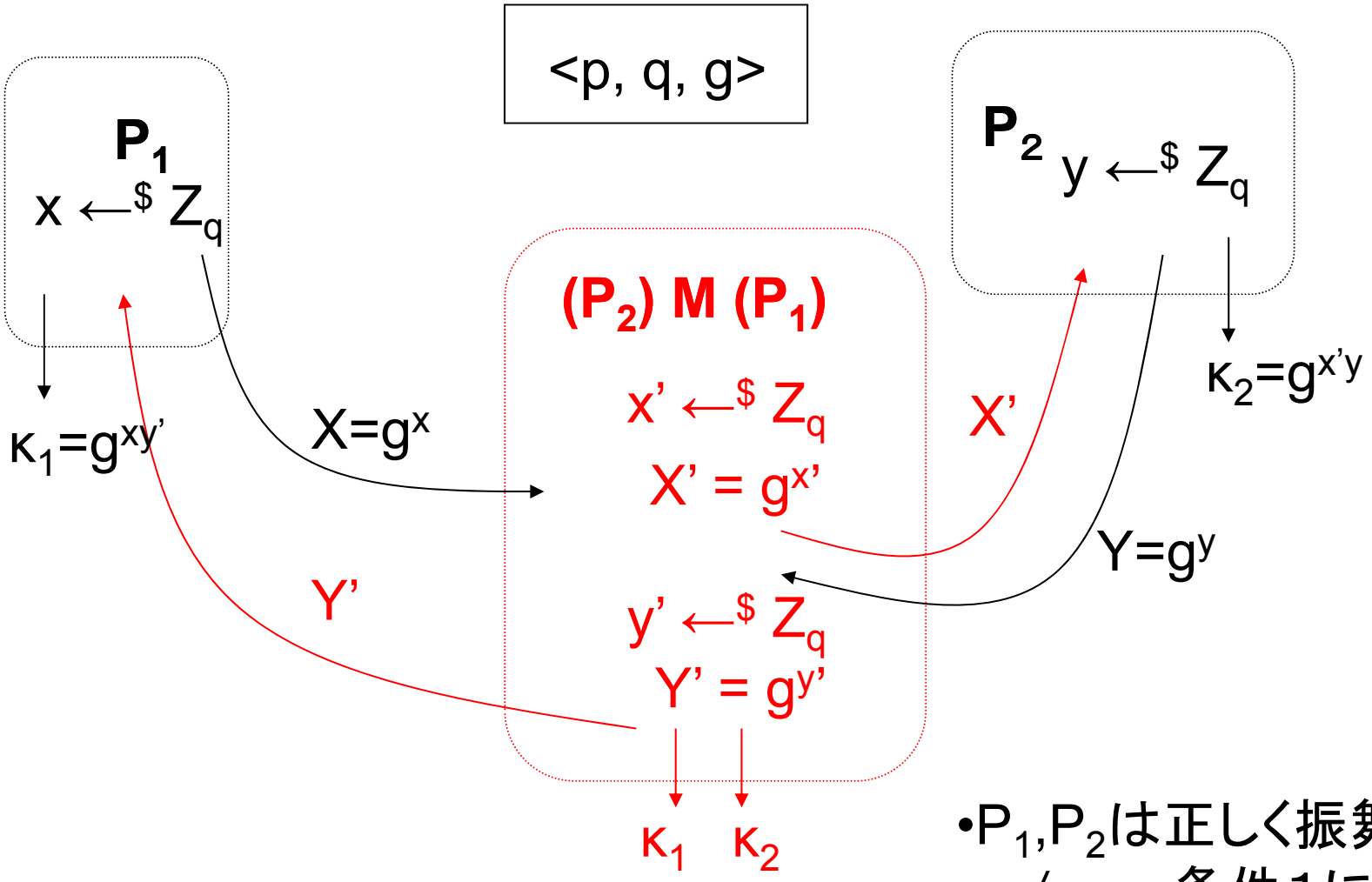

<p, q, g>
P1
x ←$ Zq
X=gx
(P2) M
P2


<p, q, g>
P1
x ←$ Zq
P2
(P2) M (P1)
x' ←$ Zq
X' = gx'
X=gx
X'


<p, q, g>
P1
x ←$ Zq
X=gx
(P2) M (P1)
x' ←$ Zq
X' = gx'
X'
P2 y ←$ Zq
κ2=gx'y
Y=gy


<p, q, g>
P1
x ←$ Zq
P2 y ←$ Zq
(P2) M (P1)
x' ←$ Zq
X' = gx'
y' ←$ Zq
Y' = gy'
X=gx
X'
Y=gy
Y'
κ1=gxy'
κ2=gx'y
κ1 κ2

DHに対する中間者攻撃

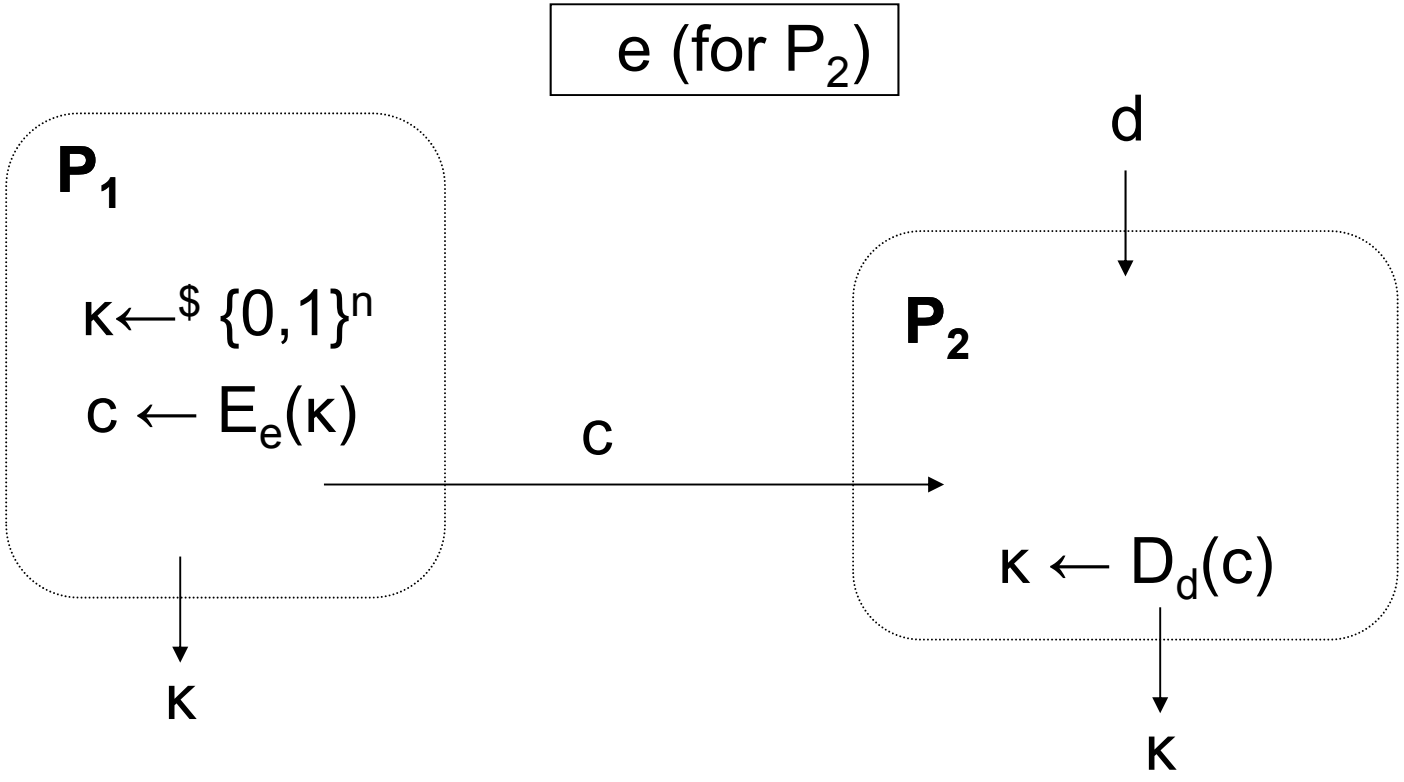


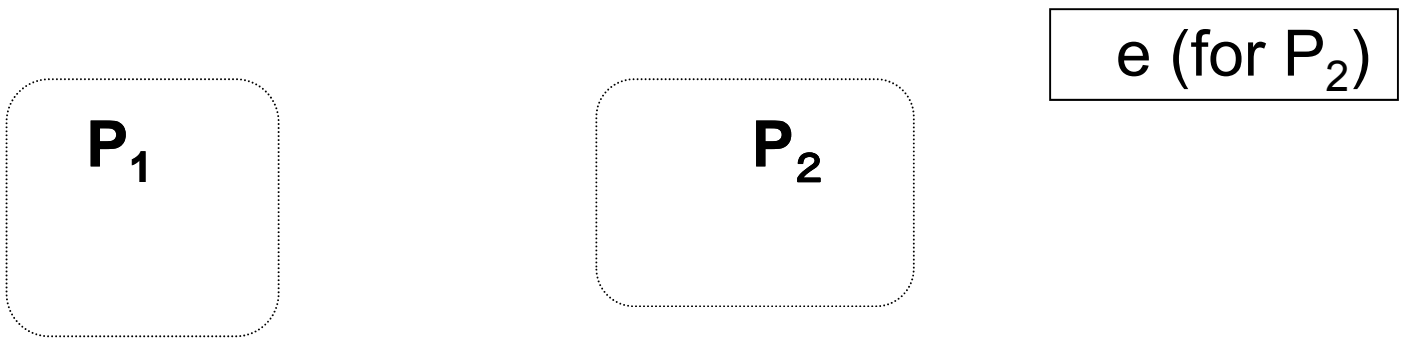
- P_1, P_2は正しく振舞っている。
- $\kappa_1 \neq \kappa_2$： 条件1に違反
- P_1, P_2にとって全く正常に見える。

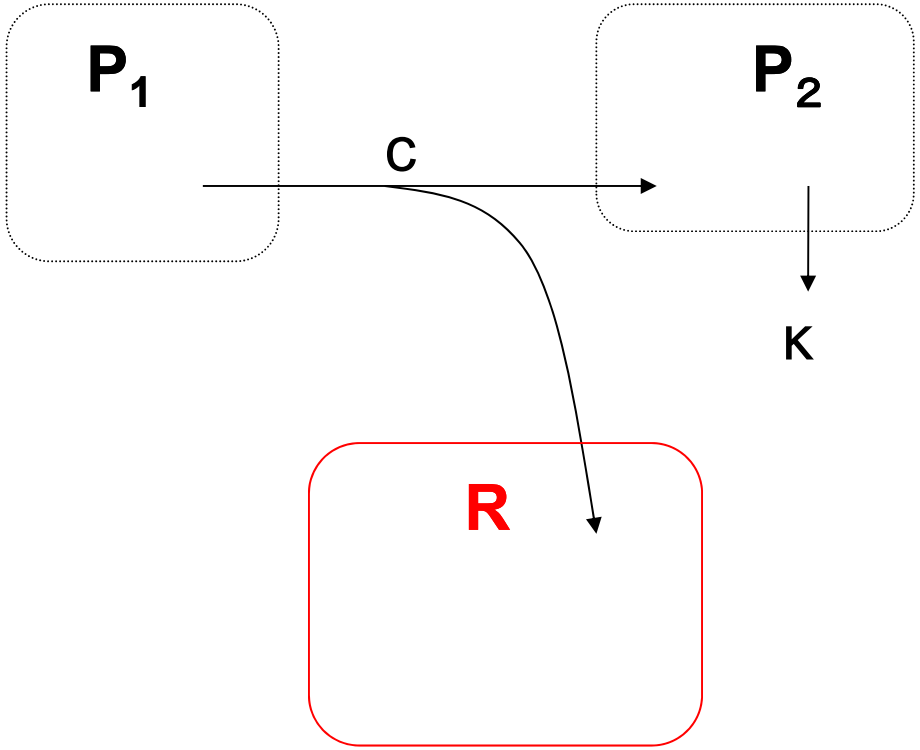
鍵共有プロトコルEB（再）

（G, E, D): 公開鍵暗号、　n: セキュリティパラメータ

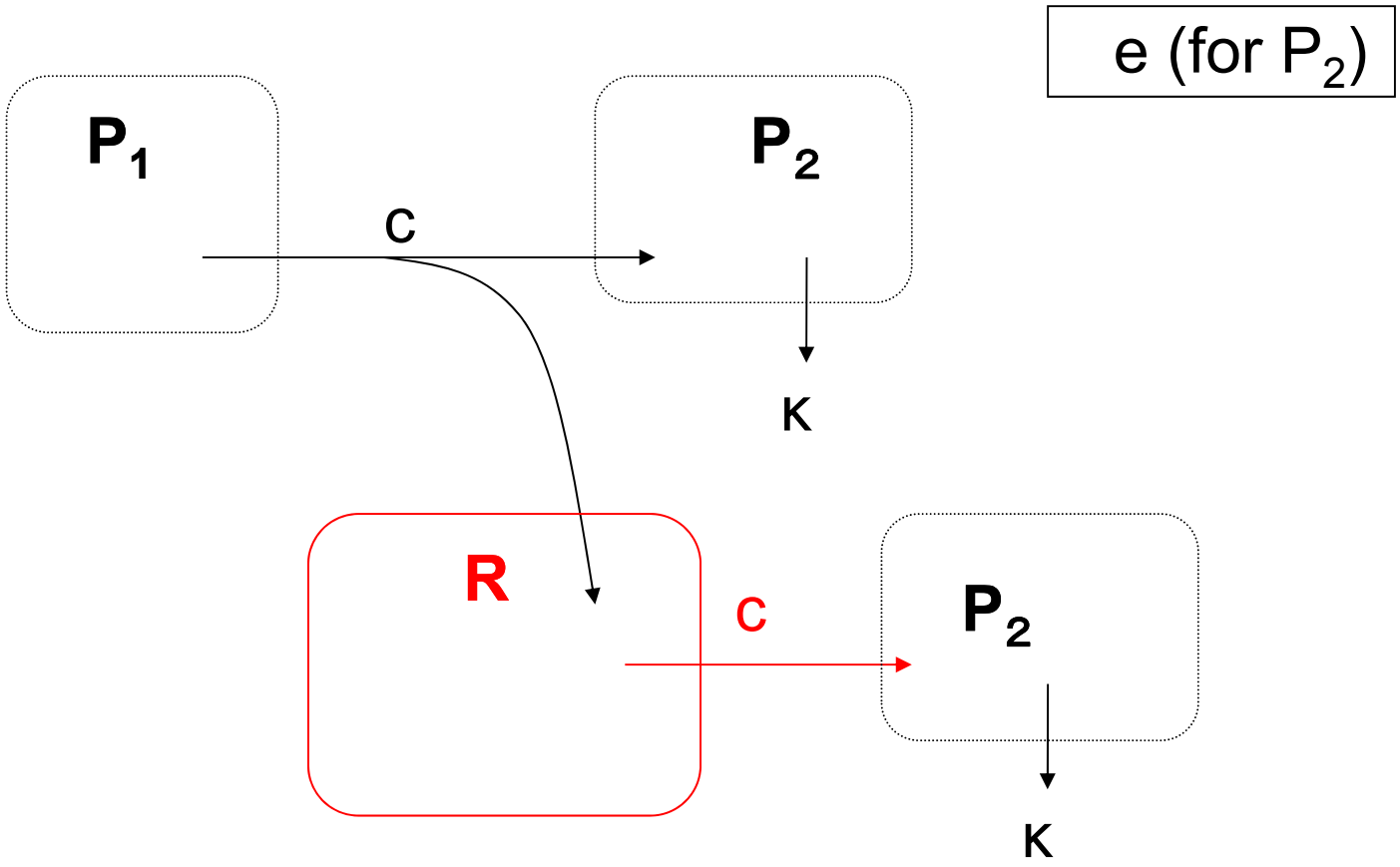
(e, d) ← G(n)　/* e: 公開鍵, d: 私有鍵 */

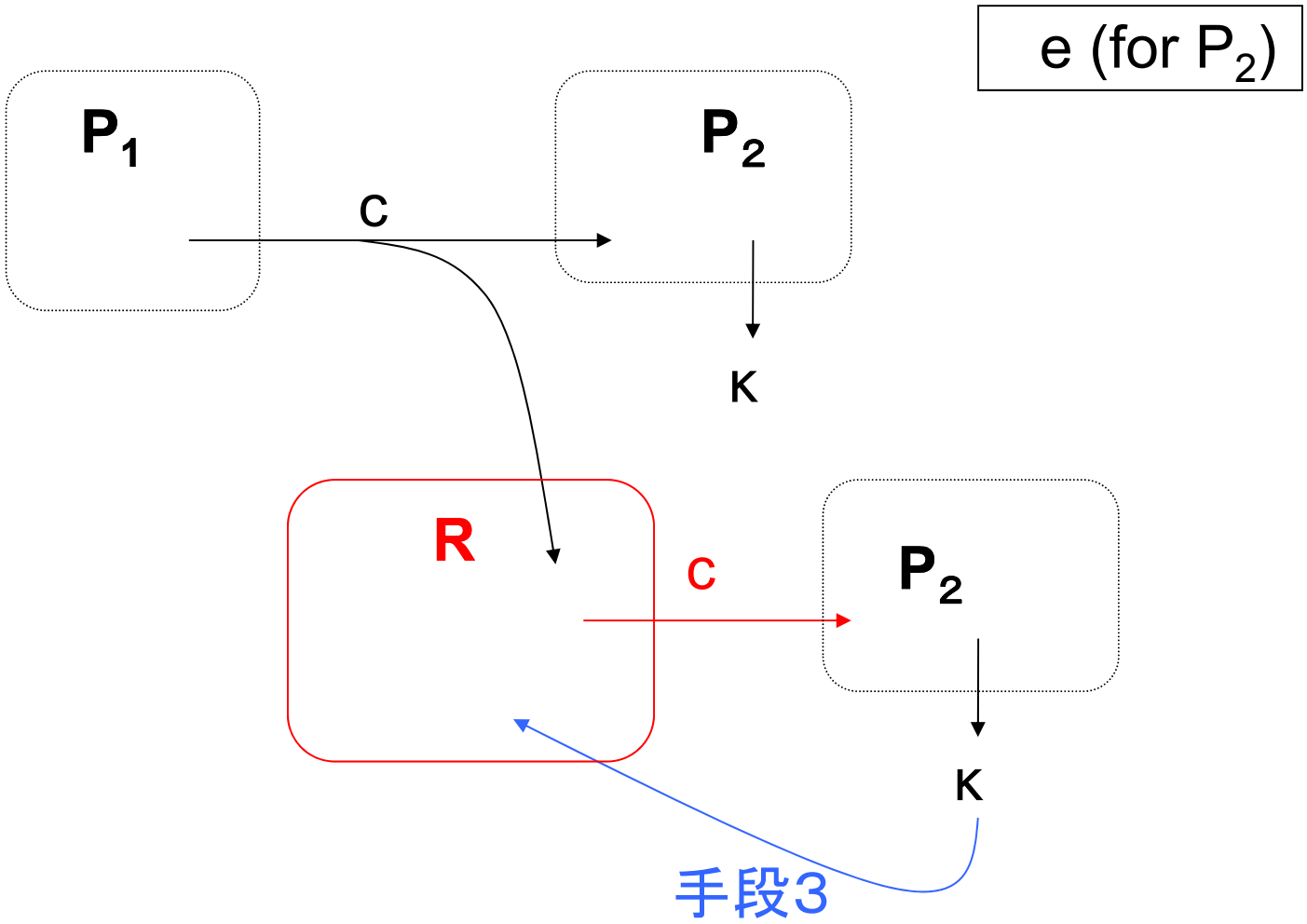


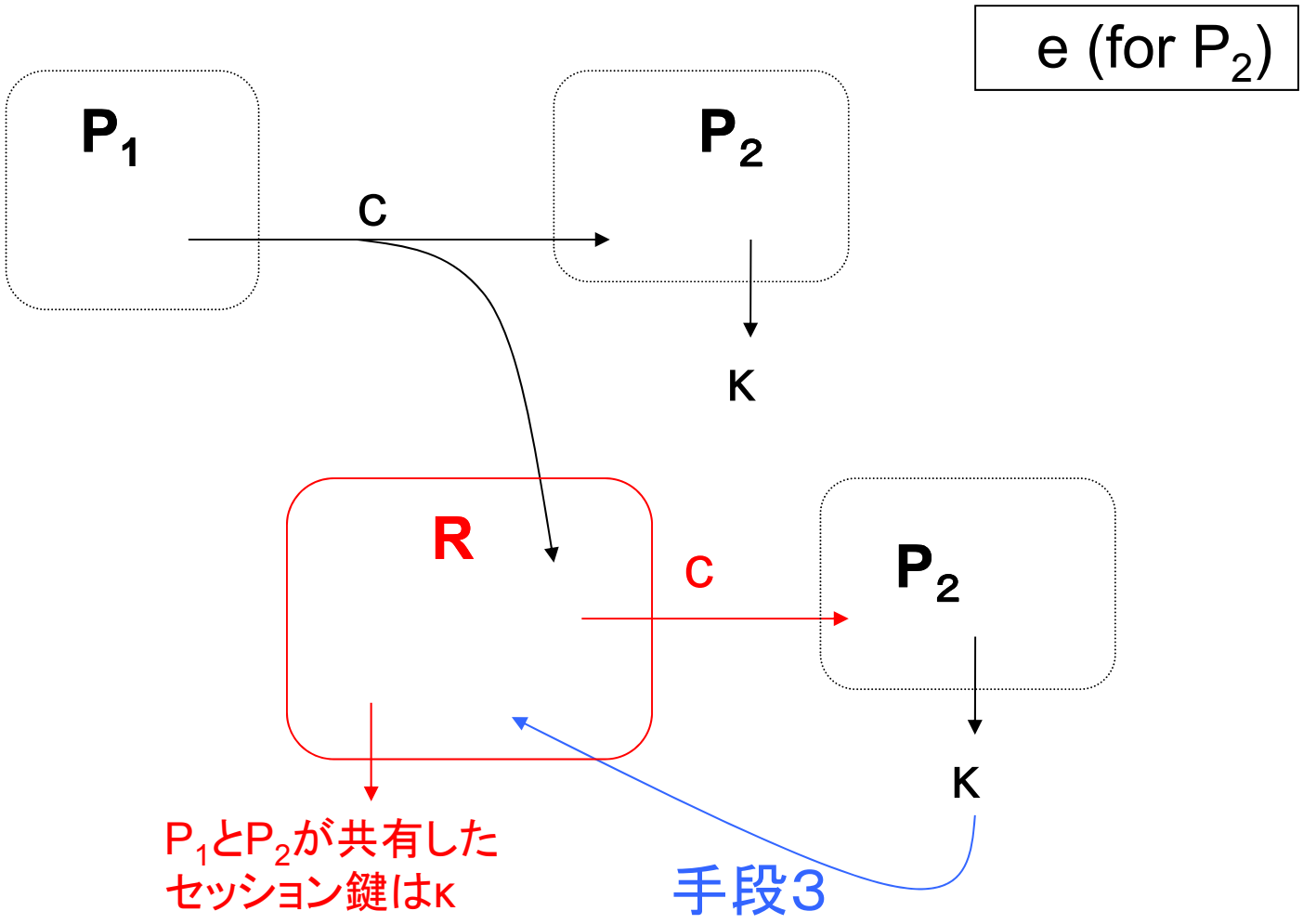

e (for P_2)
P_1
P_2



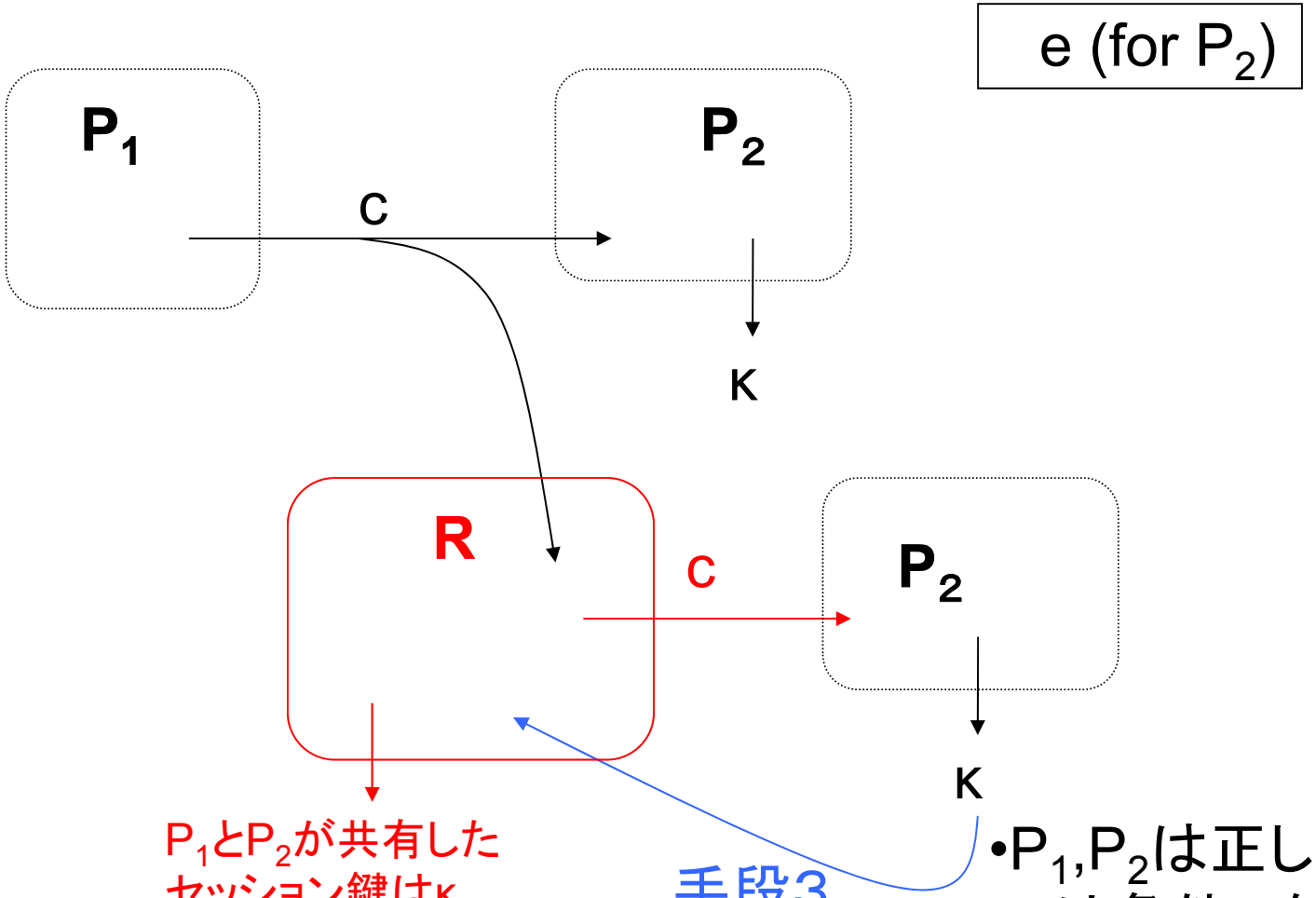
e (for P_2)


e (for P_2)
P_1
P_2
c
K
R
c
P_2
K


e (for P2)
P1
P2
C
K
R
C
P2
K
手段3


e (for P_2)
P_1
P_2
C
K
R
C
P_2
K
P_1とP_2が共有した
セッション鍵はK
手段3

EBに対する再送攻撃



- P_1,P_2は正しく振舞っている。
- Rは条件2を破っている。
- P_1,P_2にとって全く正常に見える。

鍵共有プロトコルの強化

– コンカレント環境に耐えられるように

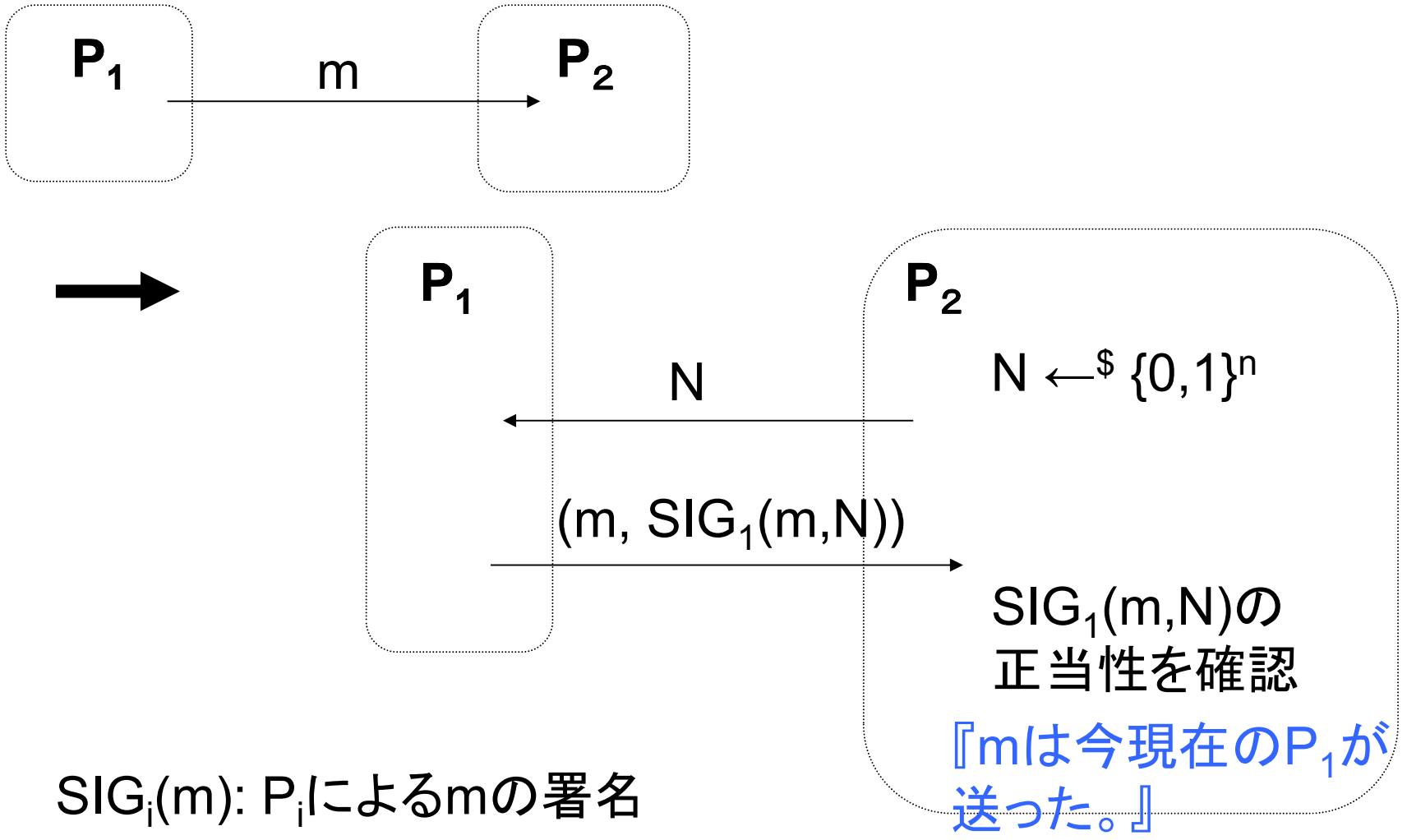
方針

- **各メッセージにセッション識別子をつける。**
  異なるセッションのメッセージが混じらないように。

- **各メッセージに送信者識別子をつける。**
  攻撃者にメッセージを流用されないように。

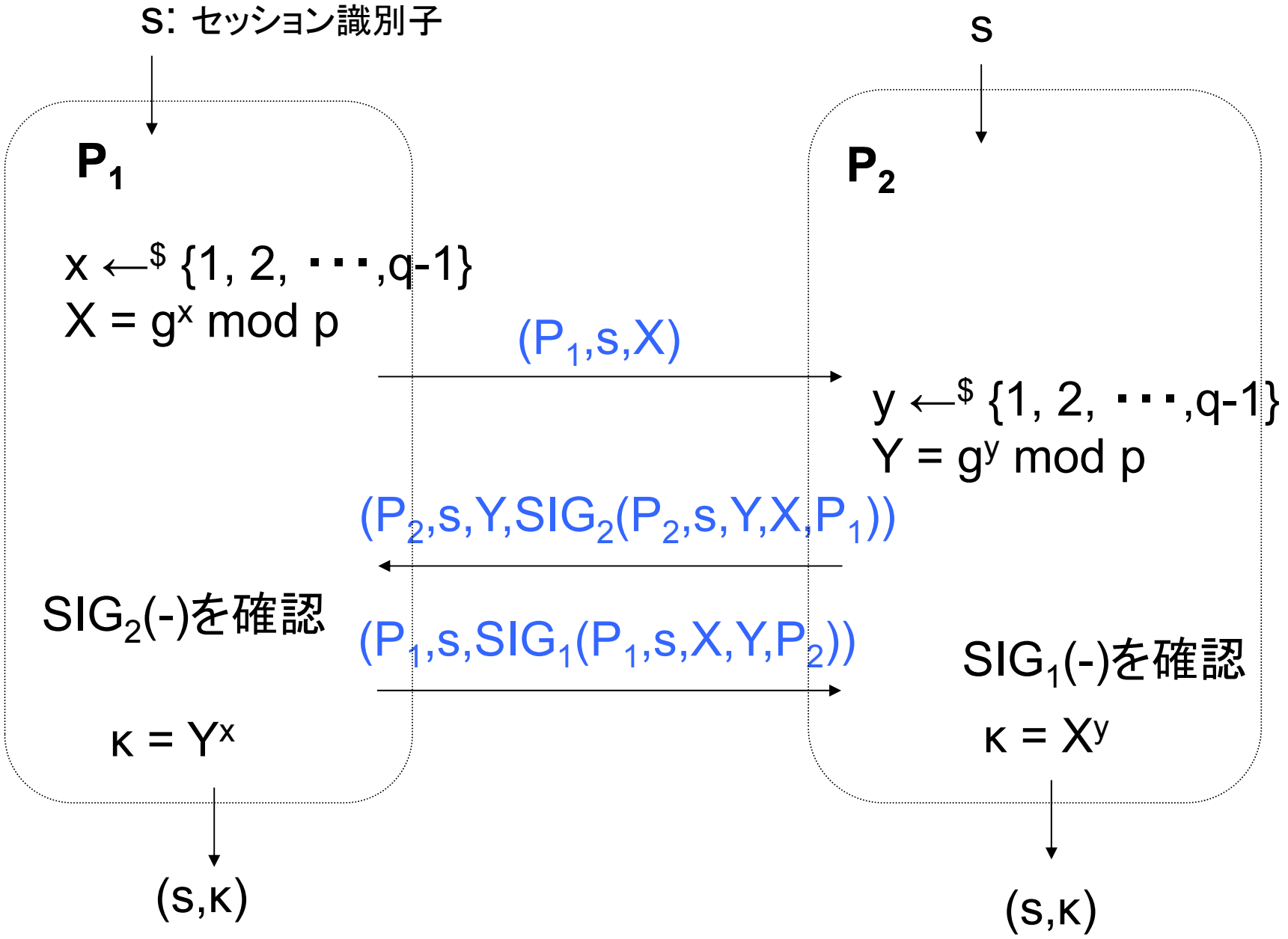
- **各メッセージの送信に認証メカニズムを組み込む。**
  通信相手P_1は確かに今現在のP_1だ。

各メッセージの送信に認証メカニズムを組み込む:

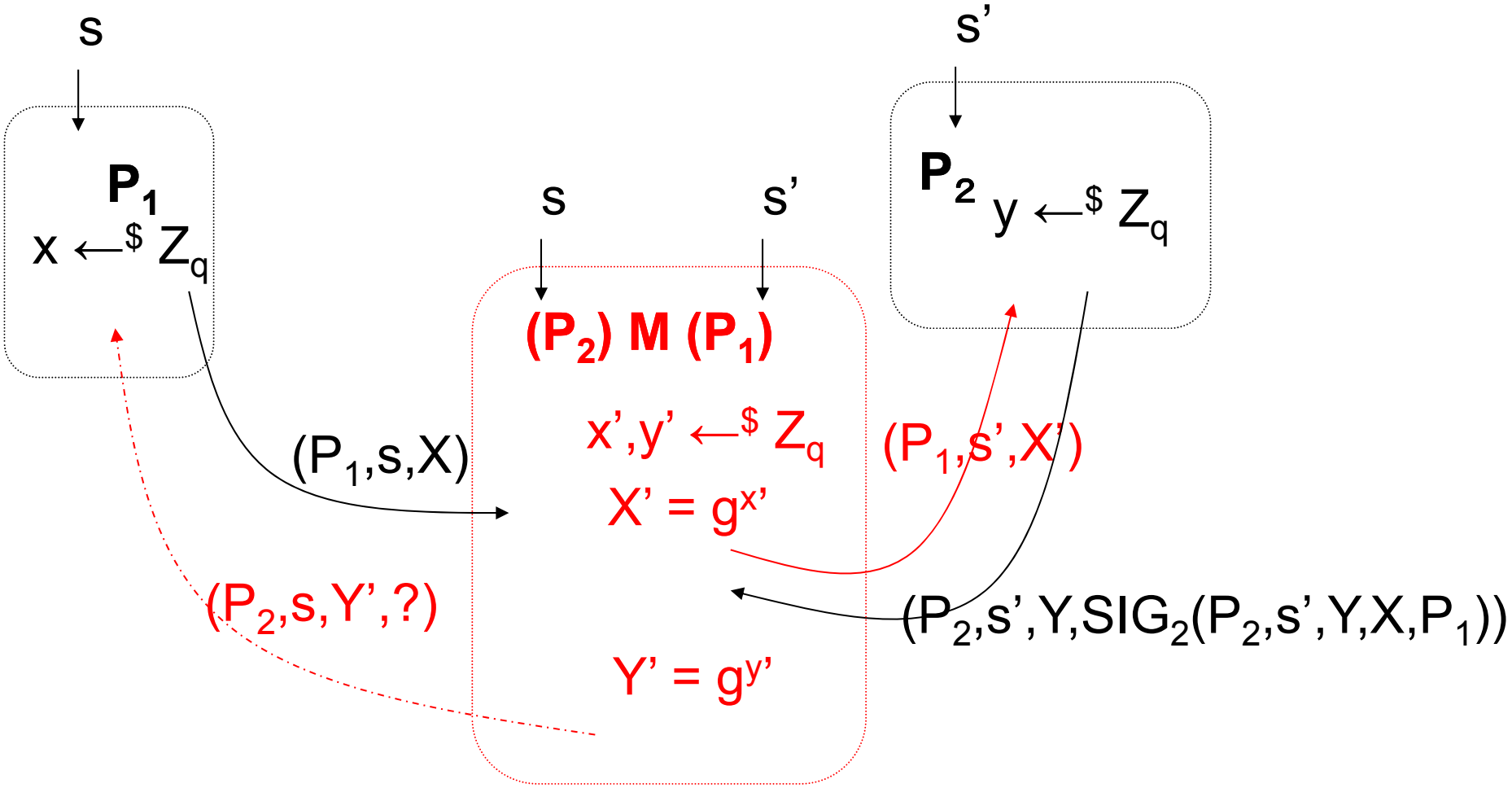


$SIG_i(m)$: P_iによるmの署名

鍵共有プロトコルDH-SIG

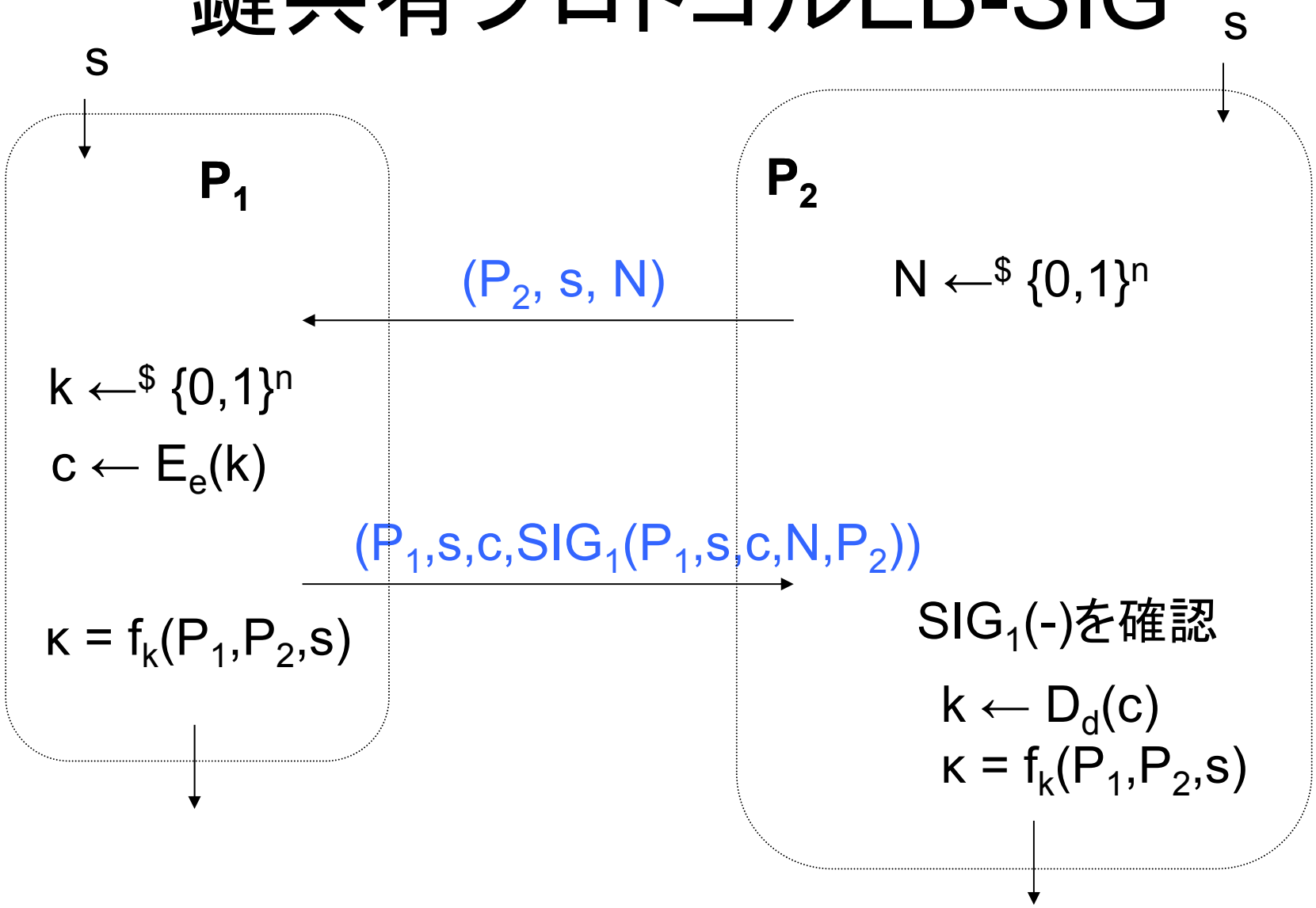


DH-SIGに対する中間者攻撃



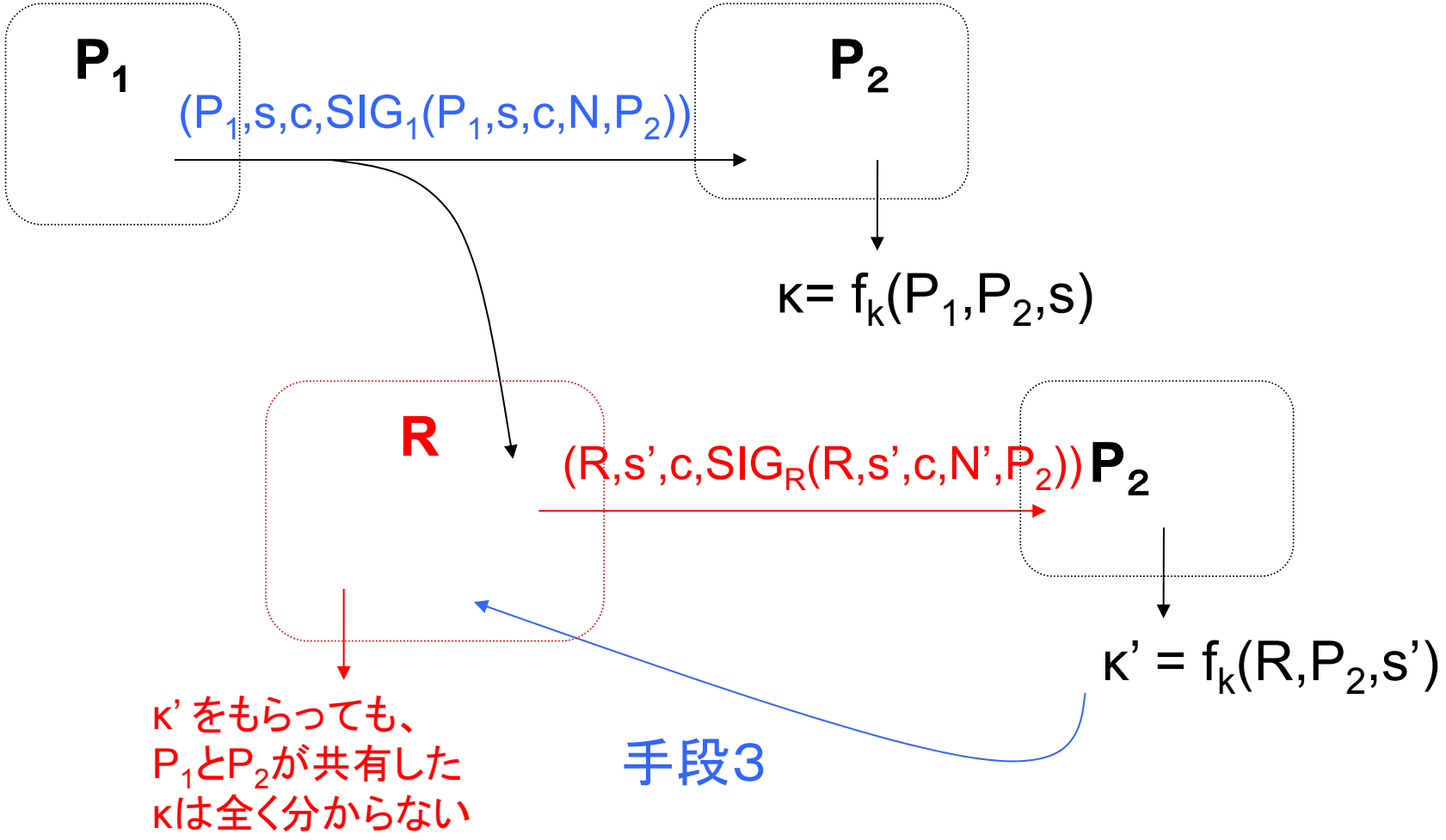
MはP1やP2に成りすますことができない。
DH-SIGにMは通用しない。

鍵共有プロトコルEB-SIG



$f_k(\cdot)$: 擬似ランダム関数
（kを知らなければランダム関数と区別がつかない）

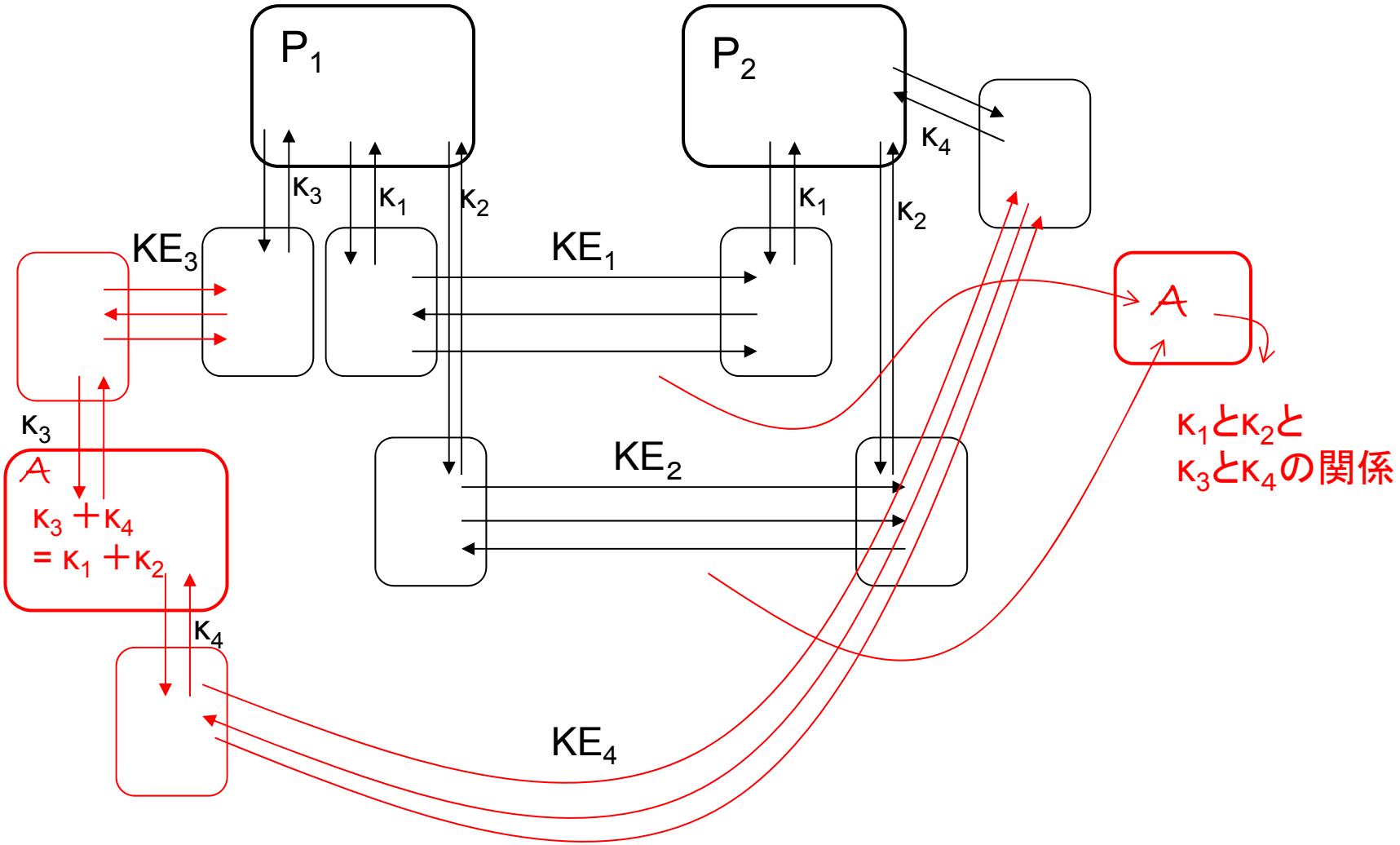
EB-SIGに対する再送攻撃



RはEB-SIGには通用しない。

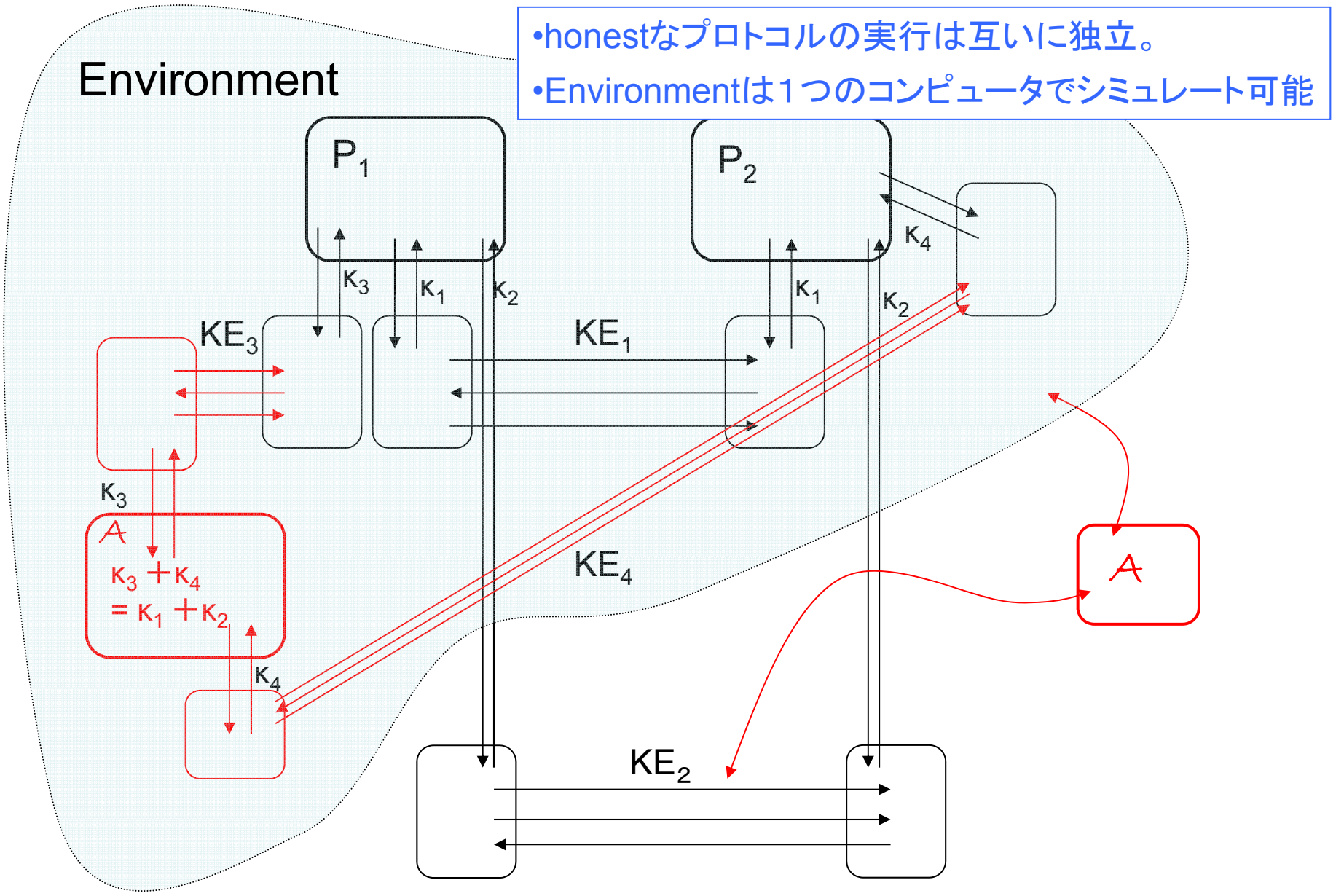
だからといって
安全と言えるか？

コンカレント環境における鍵共有(再)

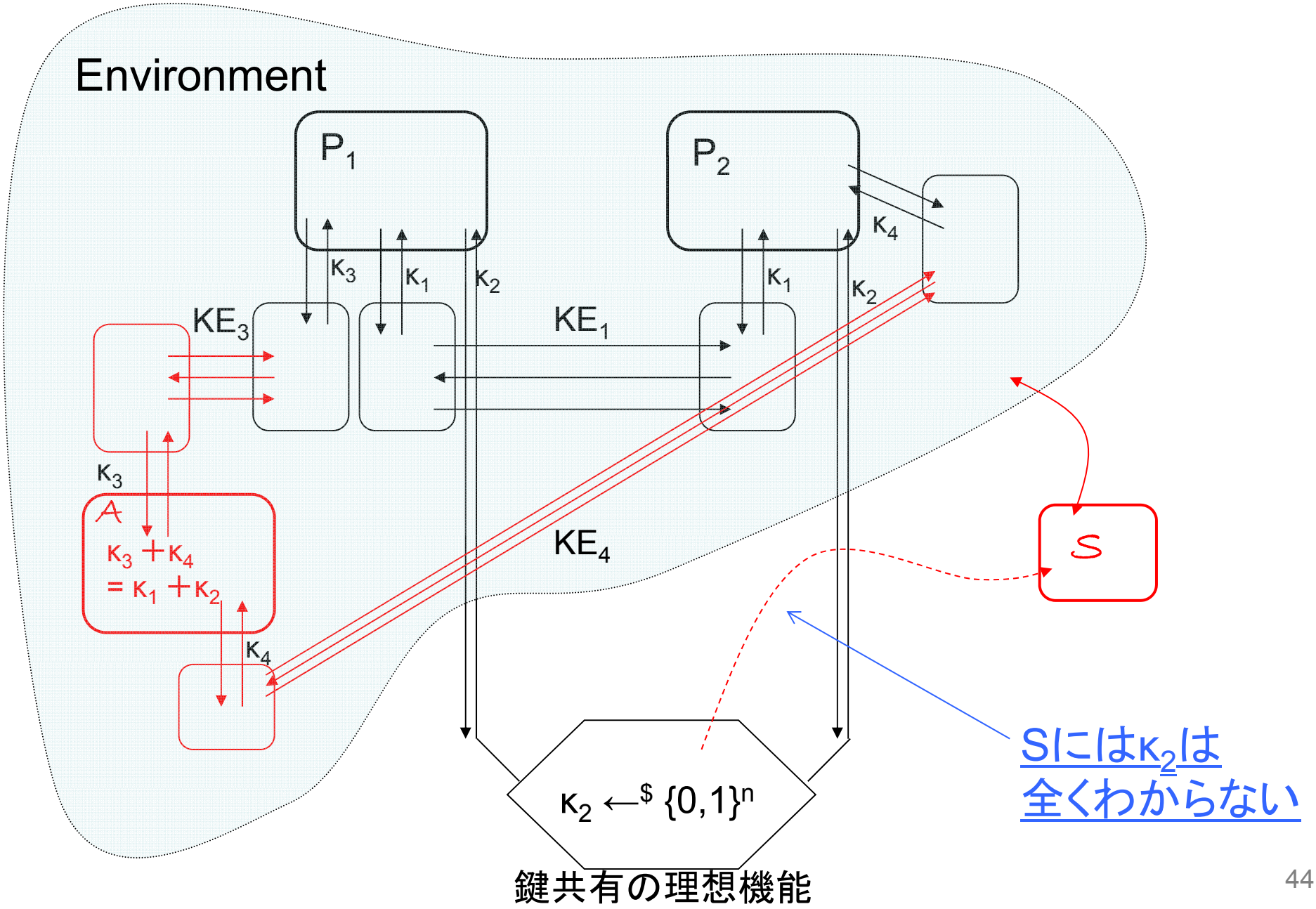


Real Game

- honestなプロトコルの実行は互いに独立。
- Environmentは1つのコンピュータでシミュレート可能



Ideal Game



鍵共有の理想機能

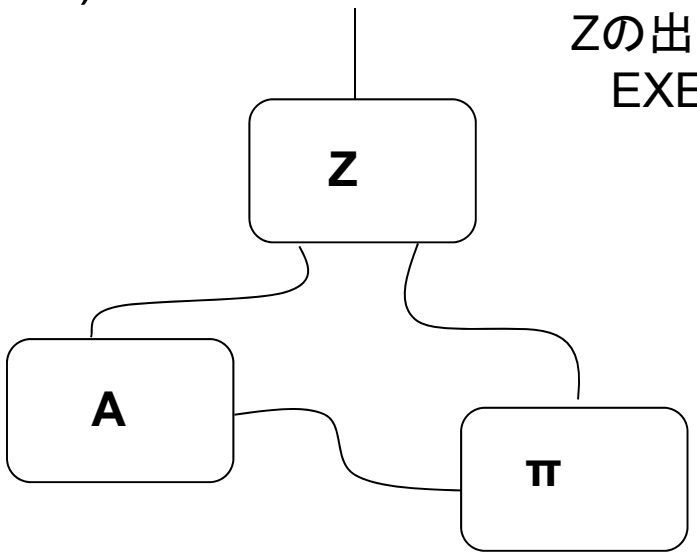
鍵共有プロトコルの安全性定義の考え方

どのような Environment にとっても
うまくSをつくられると、自分のいる世界が
Real Game (相手が (KE, A)) なのか
Ideal Game (相手が(F_{KE}, S)) なのか
区別が付かない。

このとき、KEは安全性条件CSKを満たす。

プロトコル実行モデル(UC)

Z: 環境（Environment)
A: アドバーサリー
π：プロトコル



Zの出力：
$EXEC_{\pi,A,Z}$

- Zはπへの入力をセット。πの出力を読める。
- πのメッセージはAが配達。Aはπを実行するパーティをコラプトできる。
- ZとAは同時並行に動作。

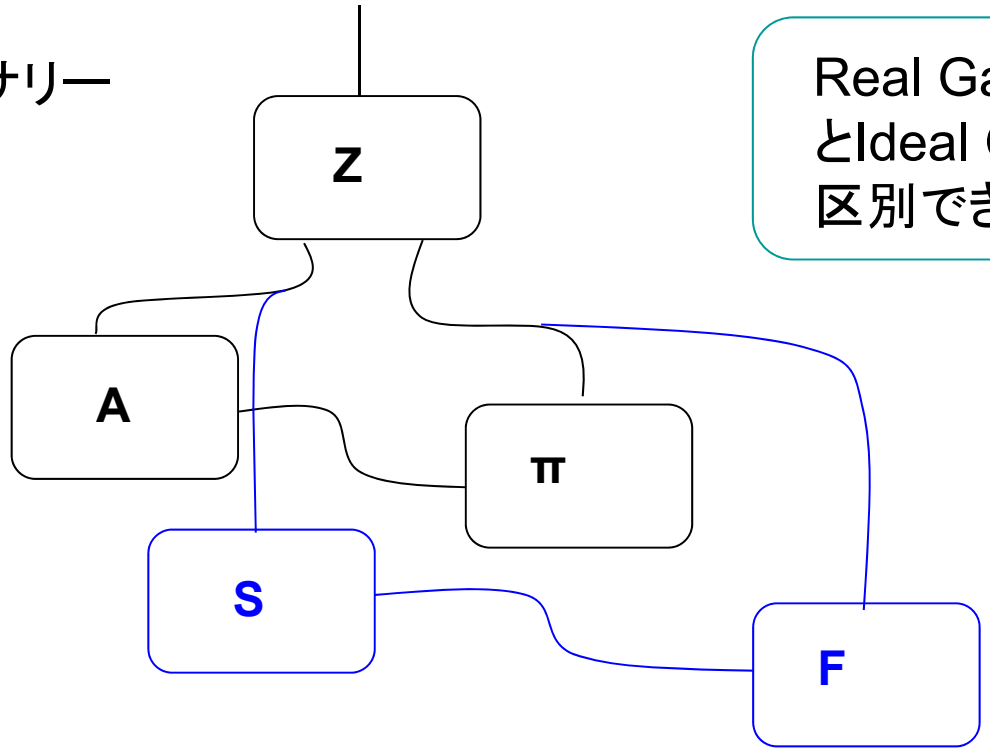
安全性定義(UC)

πが機能Fを 安全に実現

⇔

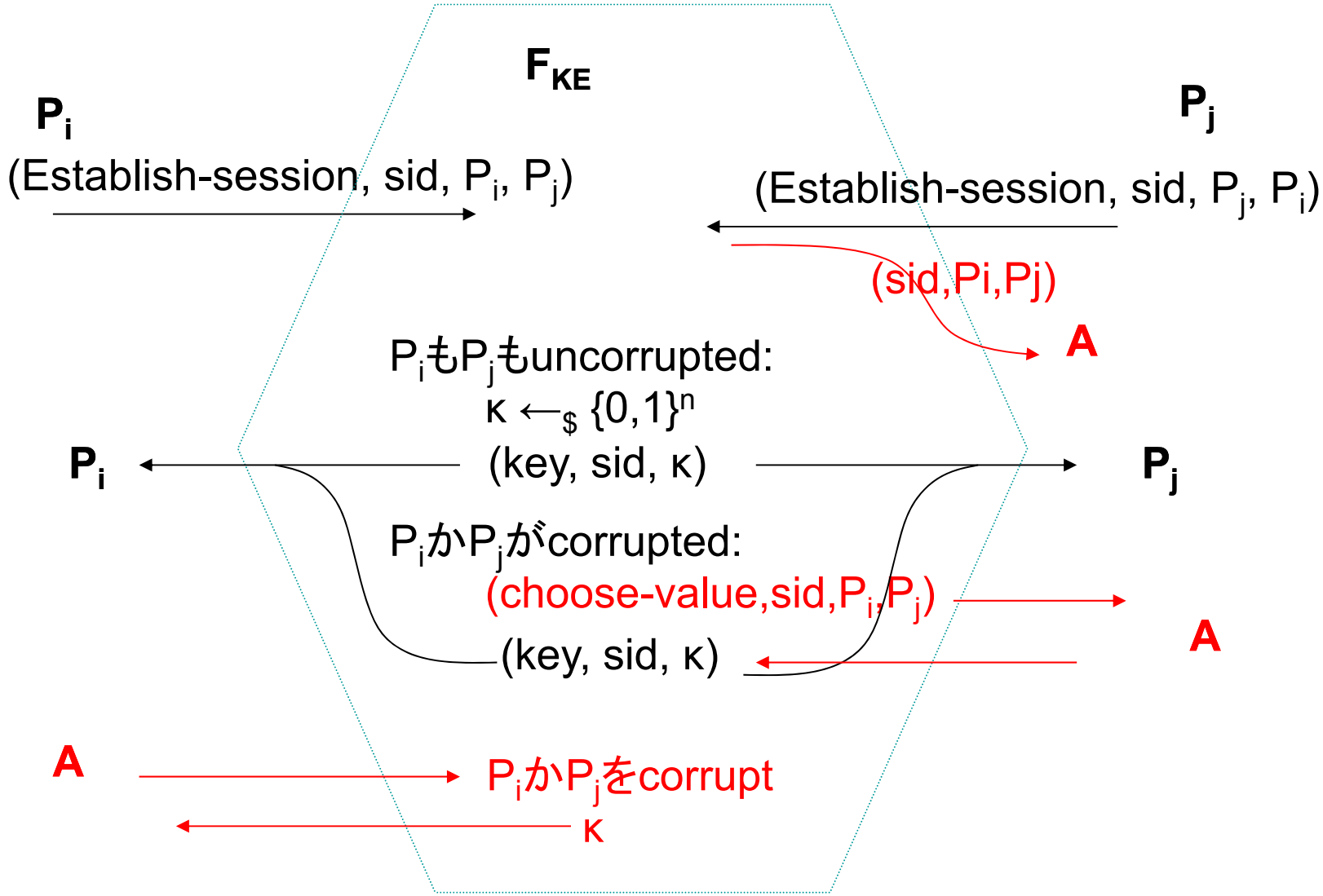
$\forall A\ \exists S\ \forall Z \quad EXEC_{\pi,A,Z} \equiv_c EXEC_{F,S,Z}$

Z: 環境
A: アドバーサリー
π: プロトコル



Real Game
とIdeal Gameが
区別できない

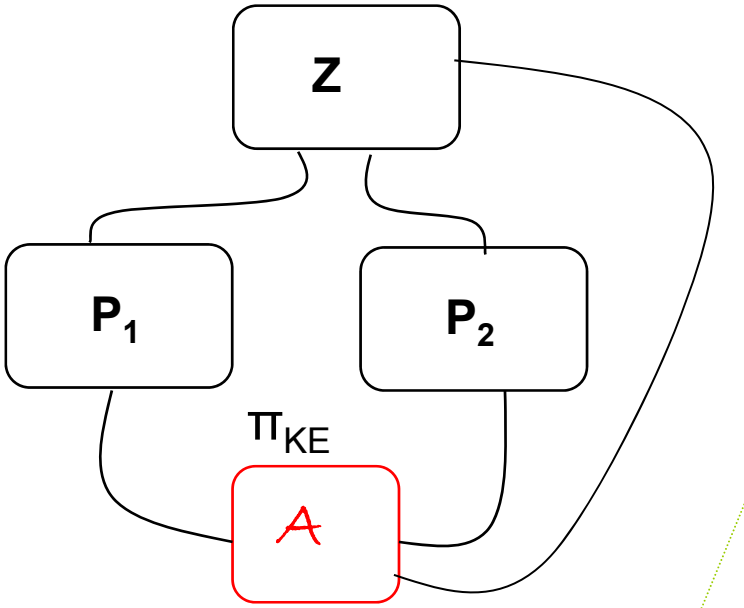
鍵共有機能F_{KE}

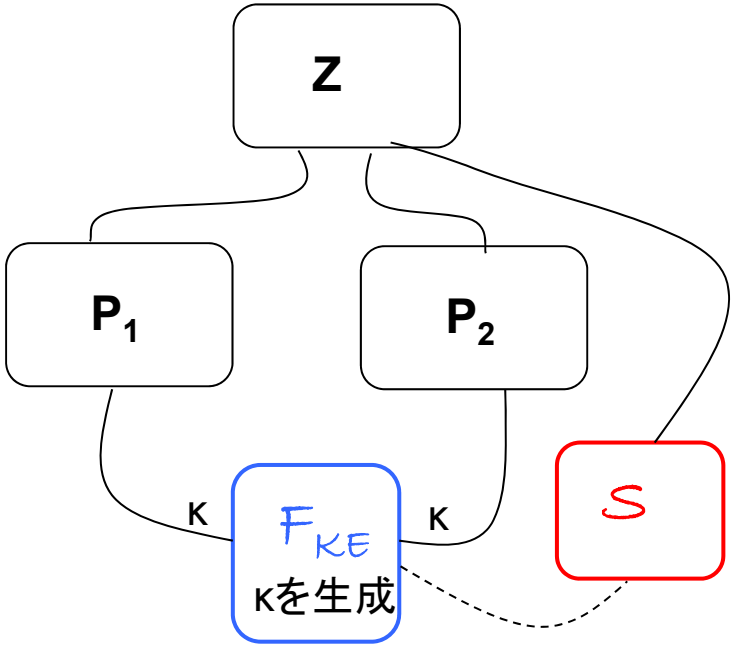

F_{KE}
P_i
P_j
(Establish-session, sid, P_i, P_j)
(Establish-session, sid, P_j, P_i)
(sid,Pi,Pj)
A
P_iもPjもuncorrupted:
$\kappa \leftarrow_{\$} \{0,1\}^n$
P_i
(key, sid, κ)
P_j
P_iかP_jがcorrupted:
(choose-value,sid,P_i,P_j)
A
(key, sid, κ)
A
P_iかP_jをcorrupt
κ

鍵共有プロトコルπ_{KE}が安全（機能F_{KE}を安全に実現）

環境Zには自分がどちらにいるのかわからない、ことを言う。
すなわち、どのような複雑な文脈のもとでもπ_{KE}はF_{KE}と同等。

（∀A∃S∀Z）





定理

1. <p, q, g> に対してDDH仮定が成立
   SIGが選択メッセージ攻撃に対して偽造不可

   → DH-SIG は鍵共有機能F_{KE}を安全に実現

2. (G,E,D)がIND-CCA
   SIGが選択メッセージ攻撃に対して偽造不可
   $\{f_k\}_k$が擬似ランダム関数

   → EB-SIG は鍵共有機能F_{KE}を安全に実現

ただし、適応的なコラプトはないとする。

これでOKか？

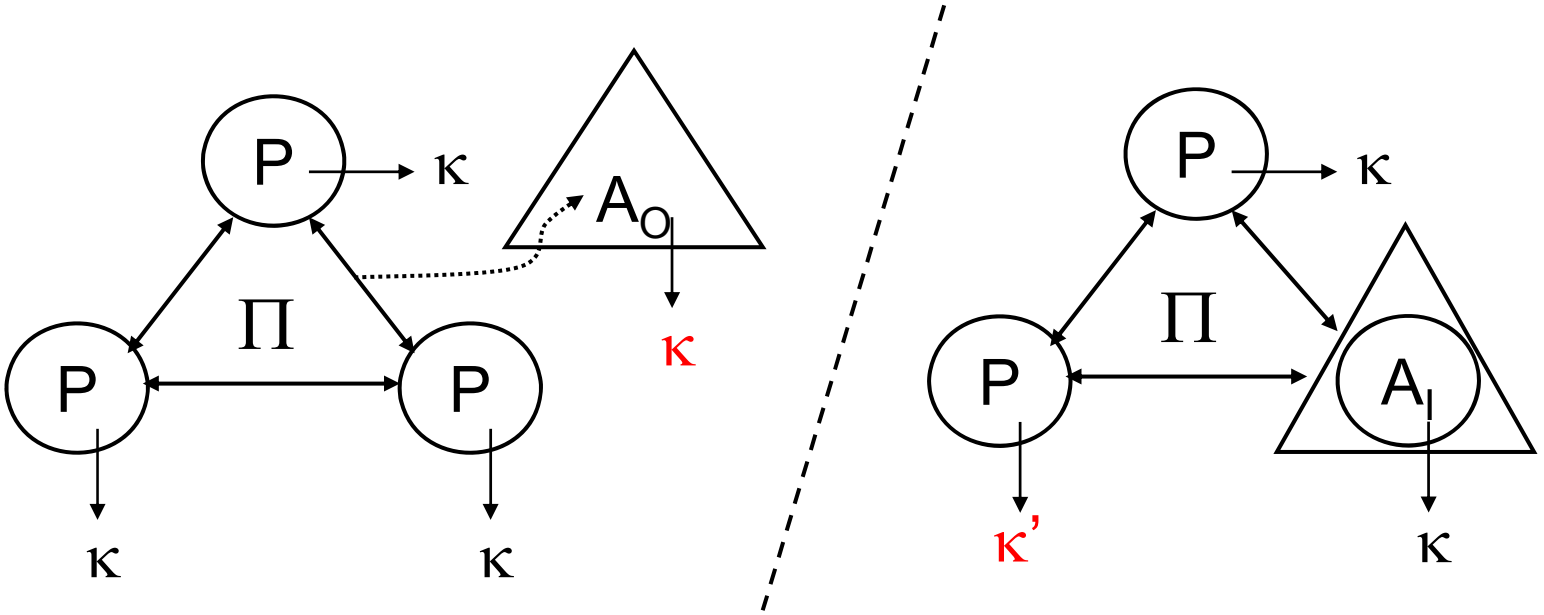
NO

- 適応的な攻撃者
- フォーワード・シークレシィ
- セッション識別子やパーティ識別子をどうするか。
- パスワード認証との併用。
- グループでの鍵共有

参考：http://lab.iisec.ac.jp/~arita/pdf/lecture123.pdf

Attacks to GKAP.

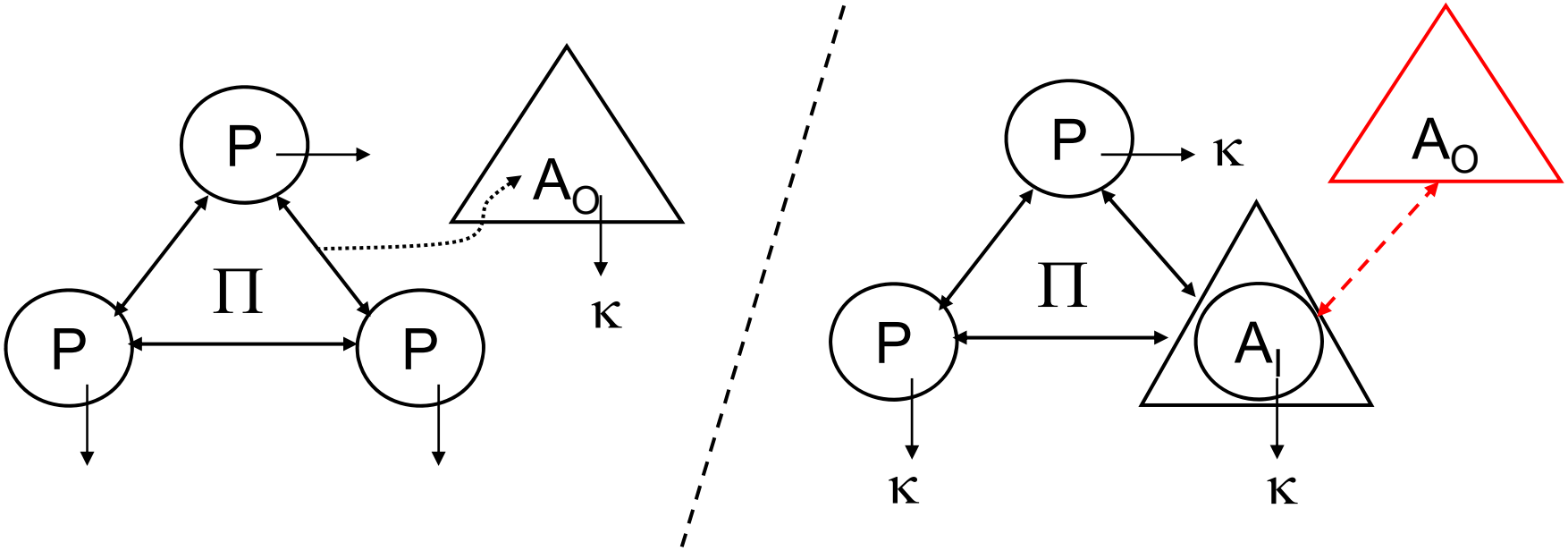
- We must consider a few kind of attacks.
  - An outside eavesdropper.
  - Malicious insiders.
  - An eavesdropper collaborates with malicious insiders.



P: Party, A_I: Adv. (Insider), A_O: Adv. (Outsider) , Π : GKAP,
κ: Session-key

Attacks to GKAP.

- We must consider a few kind of attacks.
  - An outside eavesdropper.
  - Malicious insiders.
  - An eavesdropper collaborates with malicious insiders.



P: Party, A_I: Adv. (Insider), A_O: Adv. (Outsider) , Π : GKAP,
κ: Session-key