# 能動的攻撃者の下での XORの記号モデルと その計算論的健全性

[1)]櫻田英樹 [2,3)]川本裕輔 [3)]萩谷昌己

1) NTT コミュニケーション科学基礎研究所

2) 日本学術振興会特別研究員/ ENS Cachan

3) 東京大学大学院情報理工学系研究科

# 背景：プロトコルの安全性検証の2種類のモデル

| | 具体（計算論的）モデル | 記号モデル |
|---|---|---|
| メッセージ | ビット列 | 暗号化等を表す記号の列 |
| 参加者・攻撃者 | 多項式時間確率Turing機械 | プロセス<br>（確率的動作がない） |
| 安全性を導くための仮定 | 暗号の計算論的安全性（確率的・計算量的） | 暗号は絶対に解読されない（理想的安全性） |
| 安全性検証 | 通常は手作業 | 自動的 |

記号モデルの健全性

プロトコルが記号モデルで安全ならば、具体モデルでも安全

※プロトコルで使われる暗号プリミティブ、攻撃者の能力（受動的・能動的）、安全性の種類により様々な健全性が考えられる

# 健全性についての既存研究

Abadi, Rogaway, 2000

対称鍵暗号を用いるプロトコルについての、受動的攻撃者(eavesdropper)の下での、メッセージの識別不能性の健全性。

Micciancio, Warinschi, 2004

公開鍵暗号を用いるプロトコルについての、能動的攻撃者の下での、実行列についての安全性（攻撃の成功を表すイベントが起こらない）の健全性。

Comon-Lundh, Cortier, 2008

対称鍵暗号を用いるプロトコルについての、能動的攻撃者の下での、プロセス（参加者のプログラム）の識別不能性についての健全性。

Comon-Lundh, Hagiya, Kawamoto, Sakurada, 2009

公開鍵暗号・リング署名・ハッシュ関数を用いるプロトコルについての、能動的攻撃者の下での、プロセス（参加者のプログラム）の識別不能性。

# 本発表

（Comon-Lundh, Hagiya, Kawamoto, Sakurada）による健全性の結果を拡張し、ビット毎の排他的論理和（XOR）を用いるプロトコルを扱えるようにできるかを検討

- D. Unruhによる健全性の反例を紹介
- この反例を回避して健全性を成り立たせる方法について検討

# 具体モデル

- プロトコルの正規参加者は、次のいずれかのビット列（長さ$\eta$）を送信。一方、攻撃者は計算可能な任意のビット列を送信。
  - $\mathbf{0}=0\cdots0$
  - ランダムに生成したビット列（ノンス）
  - これらにビットごとのXOR($\oplus$)の演算をしたもの

$$\begin{array}{rcl} s & = & 00100101 \\ s' & = & 11101011 \\ \hline s \oplus s' & = & 11001110 \end{array}$$

# 記号モデル

- プロトコルの正規参加者・攻撃者はともに次の文法で定義される項（記号列）を送受信する

$$M, M' ::= s \mid \mathbf{0} \mid M \oplus M'$$

- 項の等式系として次のようなものを考える

$$\begin{aligned} M_1 \oplus M_2 &= M_2 \oplus M_1 \\ (M_1 \oplus M_2) \oplus M_3 &= M_1 \oplus (M_2 \oplus M_3) \\ M \oplus M &= \mathbf{0} \\ M \oplus \mathbf{0} &= M \end{aligned}$$

E.g.)

$$s_2 \oplus \mathbf{0} \oplus s_1 \oplus s_2 = s_1$$

# 記号モデルと具体モデルの違い

独立に生成したノンス$s_1, \ldots, s_n, s$に対し、次の等式が成り立つことがあるか？

$$s = s_{i_1} \oplus \cdots \oplus s_{i_k}$$

<u>具体モデル</u>

　ある。セキュリティパラメータ（ビット列の長さ$\eta$）に対してノンスの個数$n$が大きい（$> 3\eta$）とき、ほとんど全てのビット列が$s_{i_1} \oplus \cdots \oplus s_{i_k}$の形で表せる

<u>記号モデル</u>

　ない。等式系から導けないため。

# ノンス$s_1, s_2, s_3$からXORを用いて生成できるビット列（ $s_1, s_2, s_3$が生成する部分ベクトル空間）

| | | | |
|---|---|---|---|
| 00000 | 01000 | 10000 | 11000 |
| 00001 | 01001 | 10001 | 11001 |
| 00010 | 01010 | 10010 | 11010 |
| 00011 | 01011 | 10011 | 11011 |
| 00100 | 01100 | 10100 | 11100 |
| 00101 | 01101 | 10101 | 11101 |
| 00110 | 01110 | 10110 | 11110 |
| 00111 | 01111 | 10111 | 11111 |

1, 2, 3

- 生成できないノンスを新たに追加するとビット列は2倍
- ランダムに追加していっても十分多くとれば全体をカバー

# 記号モデルと具体モデルの違い

独立に生成したノンス$s_1, \ldots, s_n, s$に対し、次の等式が成り立つことがあるか？

$$s = s_{i_1} \oplus \cdots \oplus s_{i_k}$$

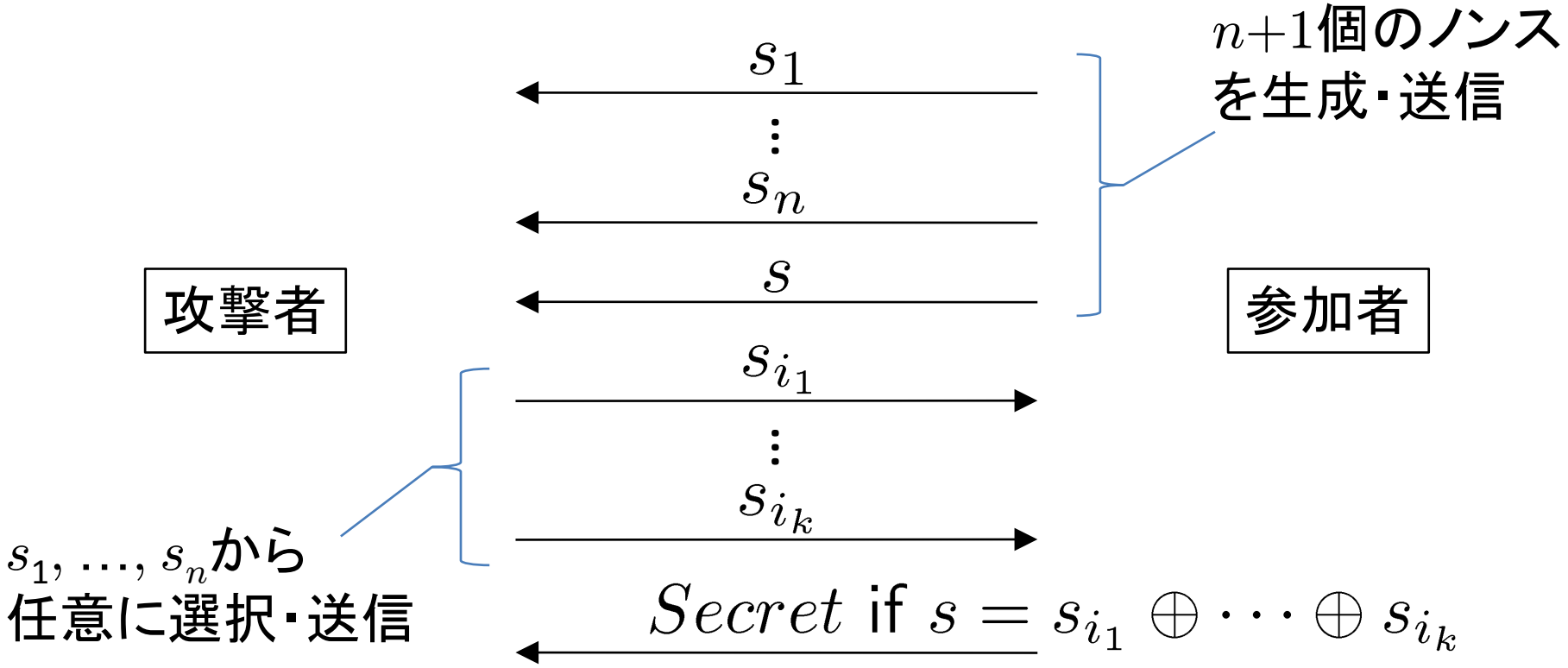
<u>具体モデル</u>

　ある。セキュリティパラメータ（ビット列の長さ$\eta$）に対してノンスの個数$n$が大きい（$> 3\eta$）とき、ほとんど全てのビット列が$s_{i_1} \oplus \cdots \oplus s_{i_k}$の形で表せる

<u>記号モデル</u>

　ない。等式系から導けないため。

# D. Unruhによる、健全性の反例

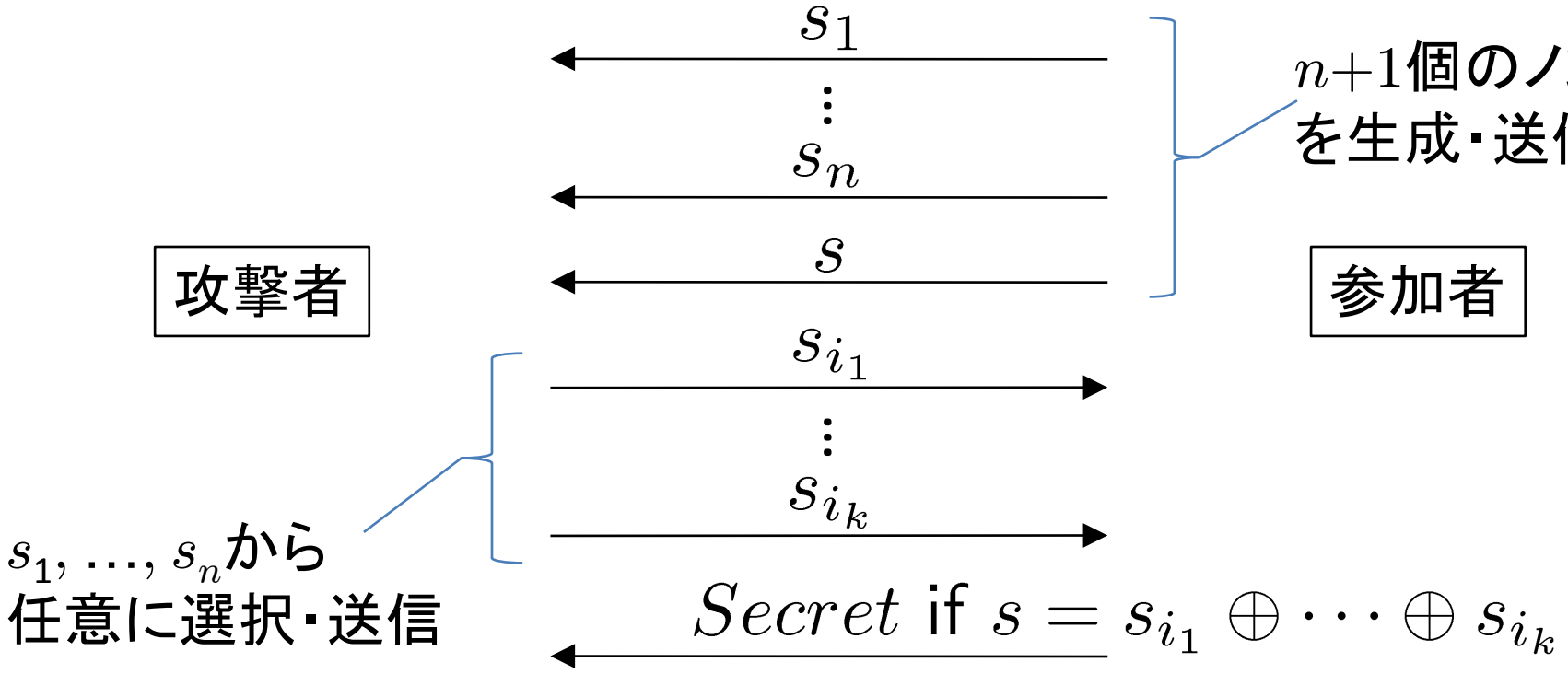



- 記号モデルでは常に $s \neq s_{i_1} \oplus \cdots \oplus s_{i_k}$ なので、秘密が漏れない
- 具体モデルでは、$n$がビット列の長さに対し大きいとき、
  攻撃者は$s_{i_1},\ldots,s_{i_k}$をうまく選び$s=s_{i_1} \oplus \cdots \oplus s_{i_k}$とすることが可能

# 健全性を成り立たせるには？

- 扱うプロトコルのクラスを制限し、D. Unruhによる反例を扱わないようにする
- 各正規参加者は、条件分岐で（プロトコルにより決まる）高々定数回しかXORを計算しないようにしたい
- このため、正規参加者はループ（while）を使わないよう制限

# D. Unruhによる、健全性の反例

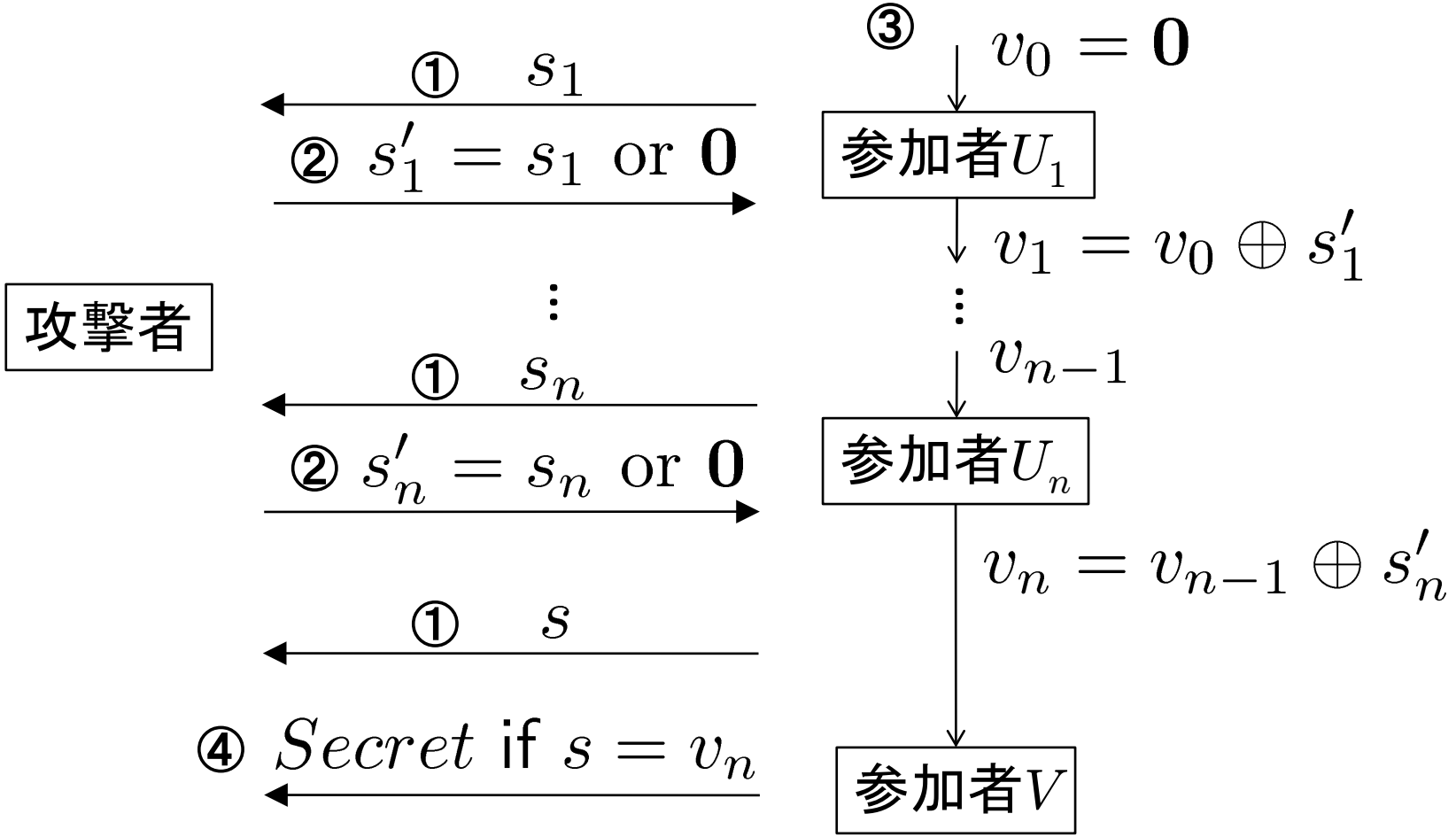



- 攻撃者がノンスの個数$n$および$k$を任意に選べる必要
- 参加者がループしてノンスを次々に送信し、受信したノンスのXORを計算することが必要
  ⇒プロトコルの制限によって回避可能

# パラレルセッションを許す場合

- （Comon-Lundh, Hagiya, Kawamoto, Sakurada）は、参加者のコピーが何個でも同時に実行することを許す
- この場合、ループに相当することが可能、先ほどの制限だけでは不十分

# D. Unruhの反例のプロトコルの変種



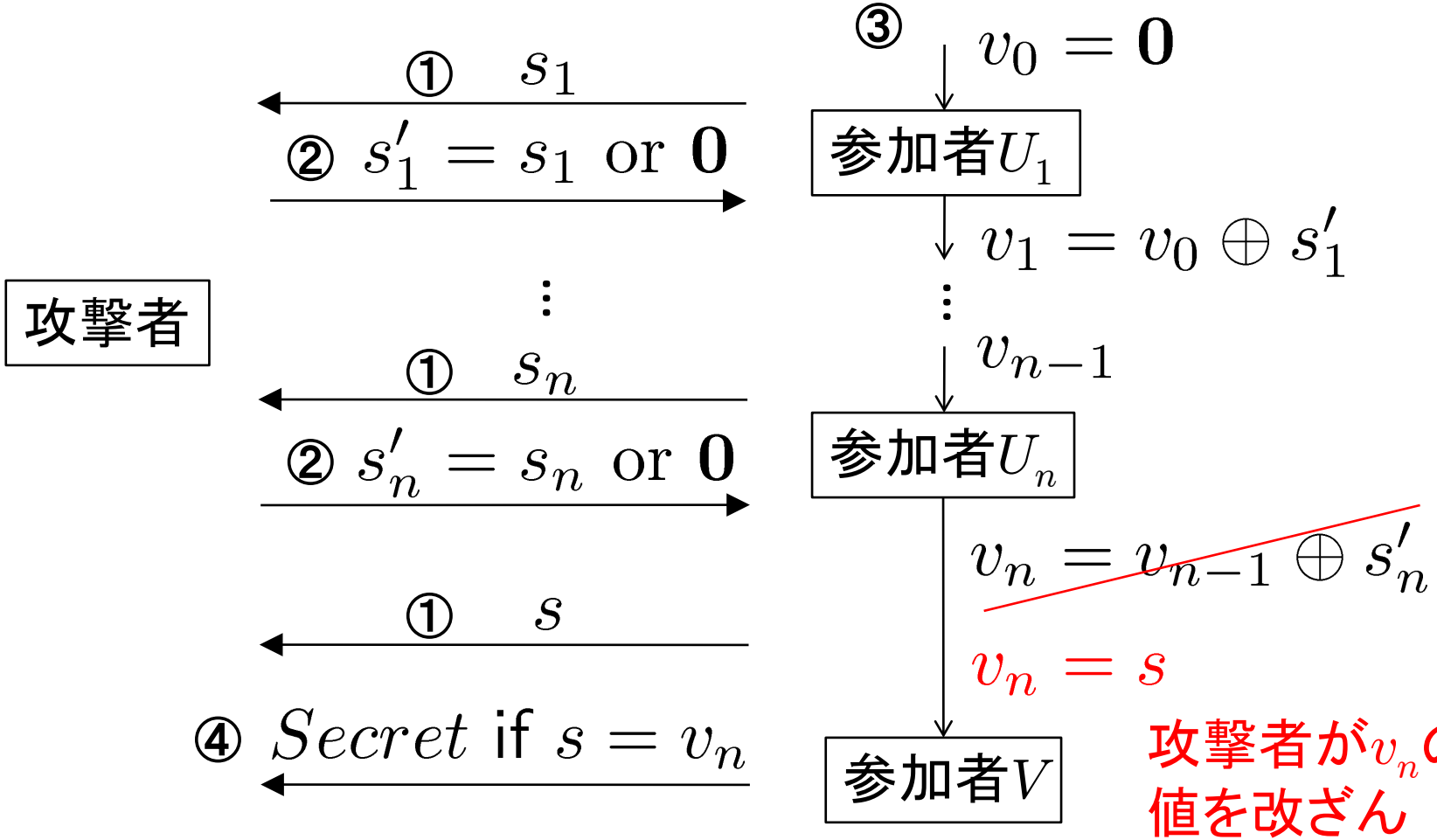

ループを直接使用しなくても、D. Unruhの反例と同様に健全性の反例になっている

# パラレルセッションを許す場合

- （Comon-Lundh, Hagiya, Kawamoto, Sakurada）は、参加者のコピーが何個でも同時に実行することを許す
- この場合、ループに相当することが可能、先ほどの制限だけでは不十分
- 新たな制限：

  正規参加者同士は、（改ざんが可能な）公開通信路を用いて通信する

# D. Unruhの反例のプロトコルの変種




制限により、記号モデルでプロトコルが安全でないため、健全性の反例とはならない

# まとめ

（Comon-Lundh, Hagiya, Kawamoto, Sakurada）による健全性の結果を拡張し、ビット毎の排他的論理和（XOR）を用いるプロトコルを扱えるようにできるかを検討

- D. Unruhによる健全性の反例を紹介
- この反例を回避して健全性を成り立たせる方法について検討