Task-PIOA: 電子署名に対する能動的攻撃者の扱いについて

2008/3/8

米山 一樹

The University of Electro-Communications

本発表の概要

- Task-PIOAフレームワークにおける**能動的**攻撃者モデルの定式化事例の提案
  - 対象プリミティブ：
    電子署名（FDH署名）
  - 攻撃者モデル：
    適応的選択文書攻撃を行う存在的偽造者

電子署名における能動的攻撃

Task-PIOAフレームワークにおける
初？の能動的攻撃者モデルの定式化

目次

- 背景，研究の目的
- 能動的攻撃者モデルの定式化
  - FDH署名における偽造者（Real system）
  - 電子署名の理想機能（Ideal system）
- 安全性証明の概要
  - 落とし戸付き一方向性関数の再定式化
- まとめ

Task-PIOAフレームワーク

- **シミュレーションパラダイム**に基づく安全性検証手法
  - 直接的に**計算量的困難性**を扱うことが可能
  - **形式的**検証可能（自動検証ツールは未実装）

攻撃者モデル

- 2つの分類軸
  - 攻撃者の振る舞い：能動的 or 受動的
  - 参加者支配のタイミング：適応的 or 非適応的

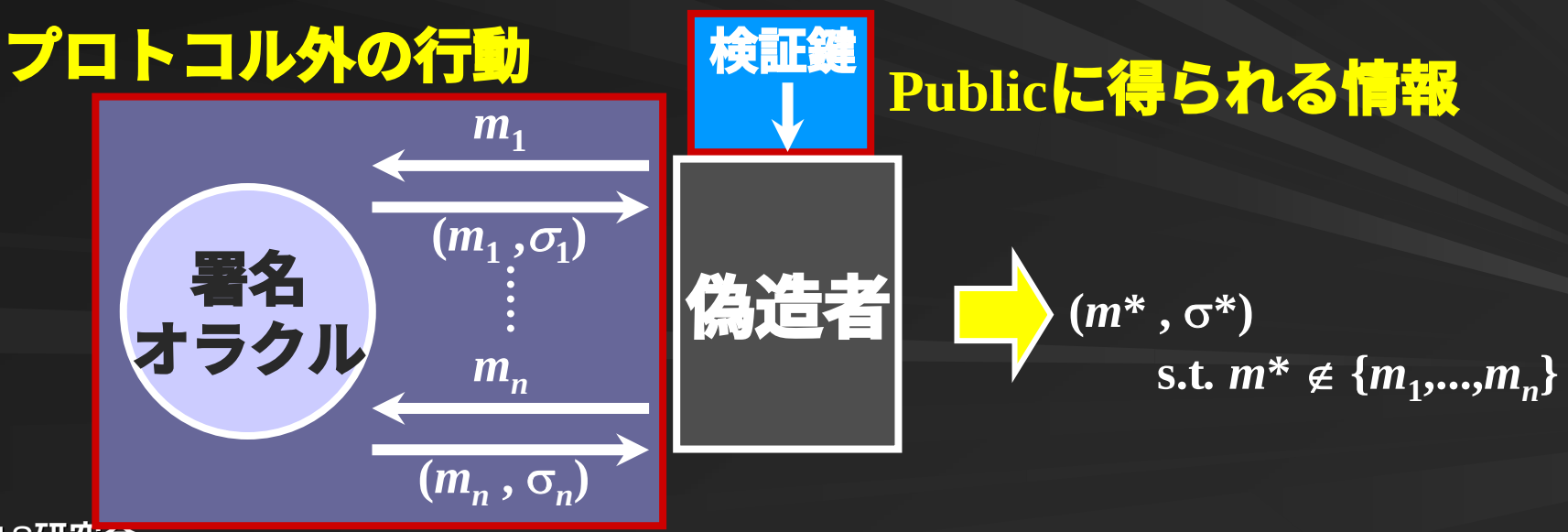
	非適応的	適応的
受動的	紛失通信 [CCK+07] 鍵交換 [KYOK07] ゼロ知識証明 [CMP07]	鍵交換 [YKO07]
能動的	電子署名 [本発表の目的]	?

能動的攻撃者モデルをどのように定式化するか？

電子署名における能動的攻撃者

- 能動的攻撃者：
  - Publicに得られる情報 + **プロトコル外の行動で得られた情報**を用いて攻撃を行う．
  - 参加者の支配とは意味が異なることに注意．

- 適応的選択文書攻撃を行う存在的偽造者



能動的攻撃者モデルの定式化

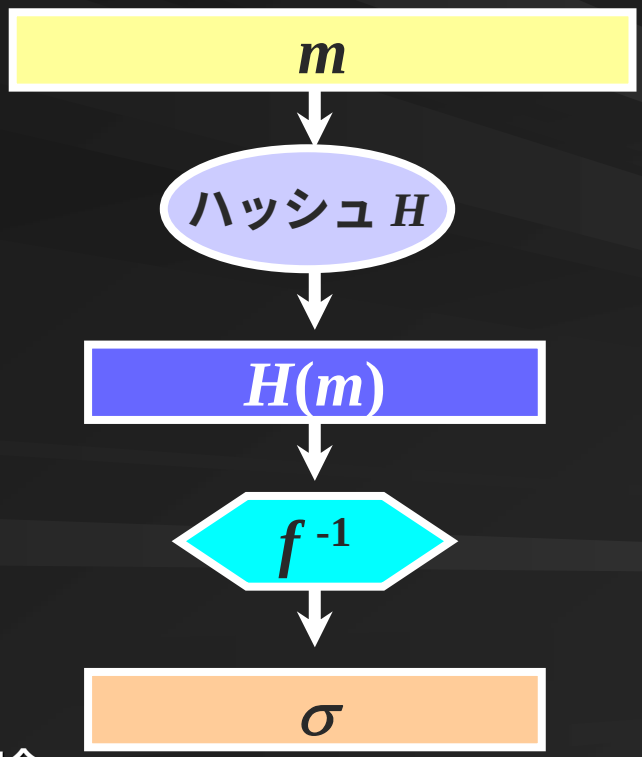
定式化のポイント

- 登場人物
  - Real system (FDH署名) : 署名者，検証者，偽造者，環境
  - Ideal system : 理想機能，シミュレータ，環境
- 適応的選択文書攻撃
  - 偽造者（Real）とシミュレータ（Ideal）の
    インターフェイスに反映．
  - これは簡単．
- 存在的偽造不可能性
  - 理想機能でうまく反映する必要あり．

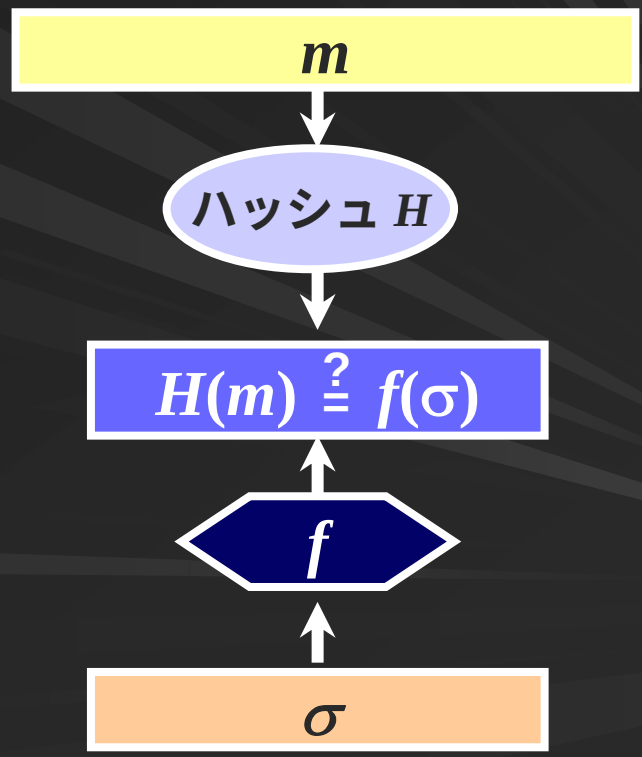
FDH署名

- 署名鍵 $sk : f^{-1}$，検証鍵 $vk : f$
  （ただし，f は落とし戸付き置換）

- 署名生成

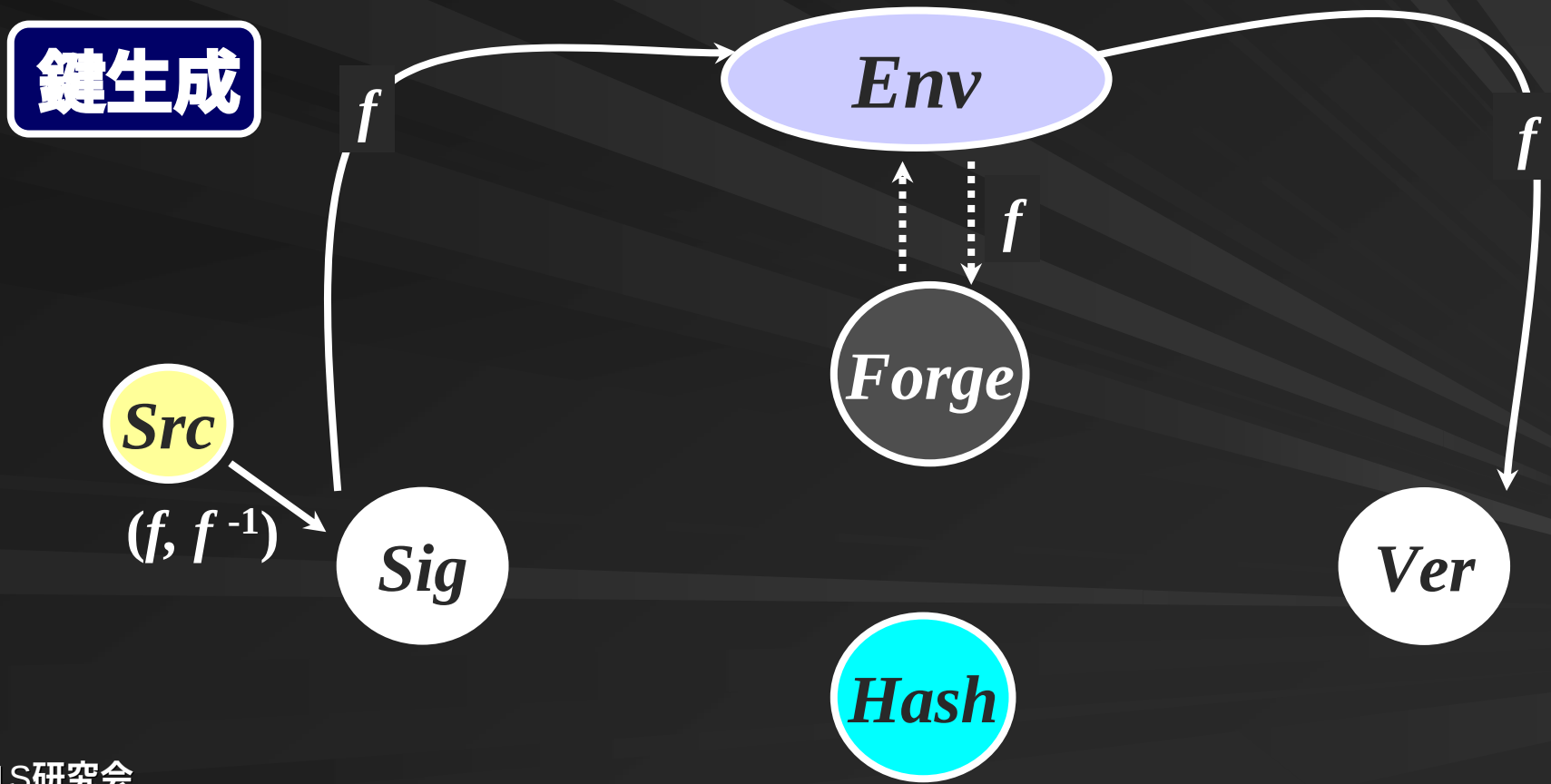


- 署名検証



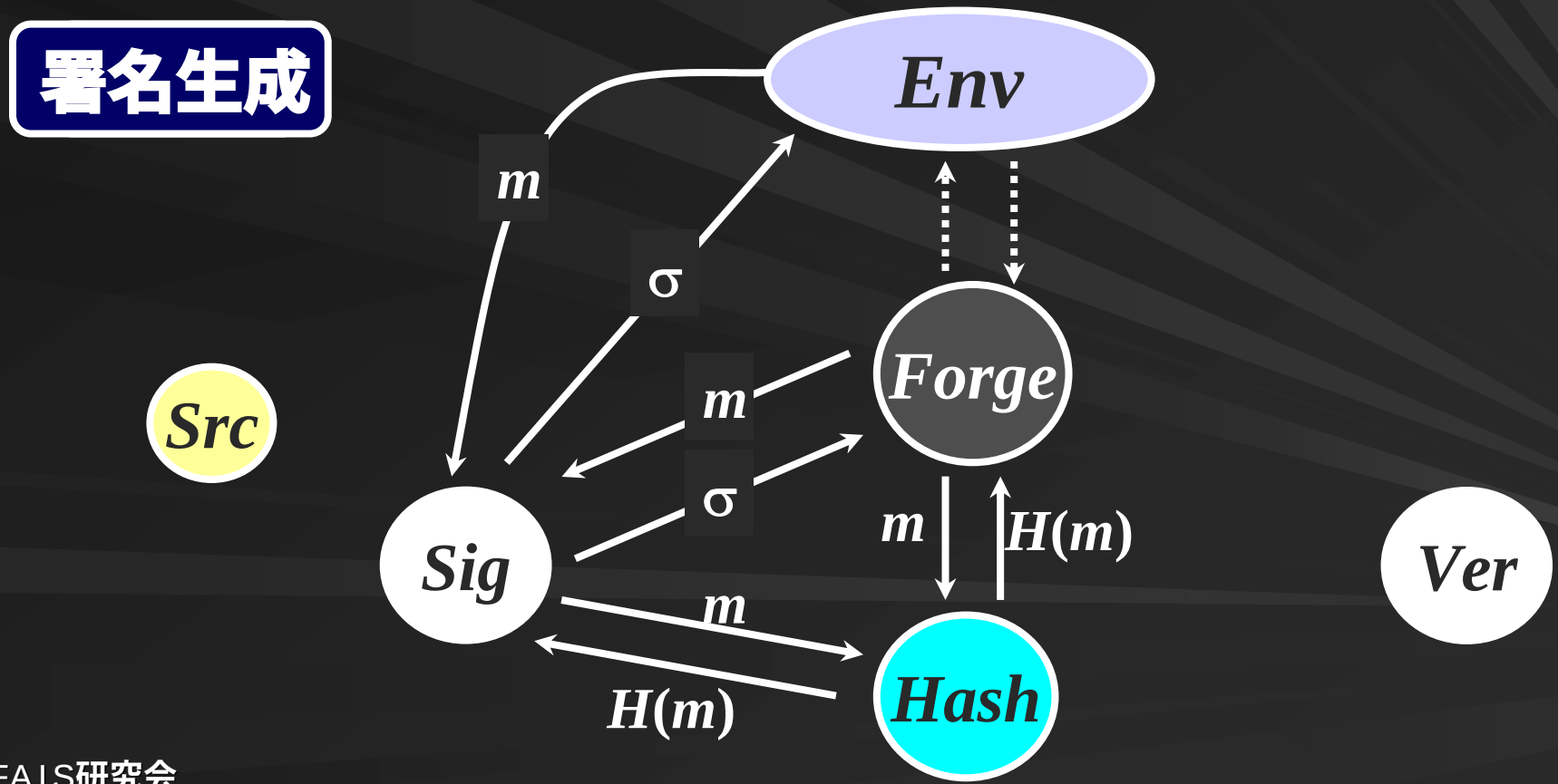
Real system（RS）の構成

- *RS* は以下のtask-PIOA の結合で構成される．
  - *RS = Sig* || *Ver* || *Forge* || *Env* || *Src*(*Tdpp*) || *Hash*
    (*Src*(*Tdpp*)：落とし戸付き置換ペアを選ぶオートマトン)



Real system（RS）の構成

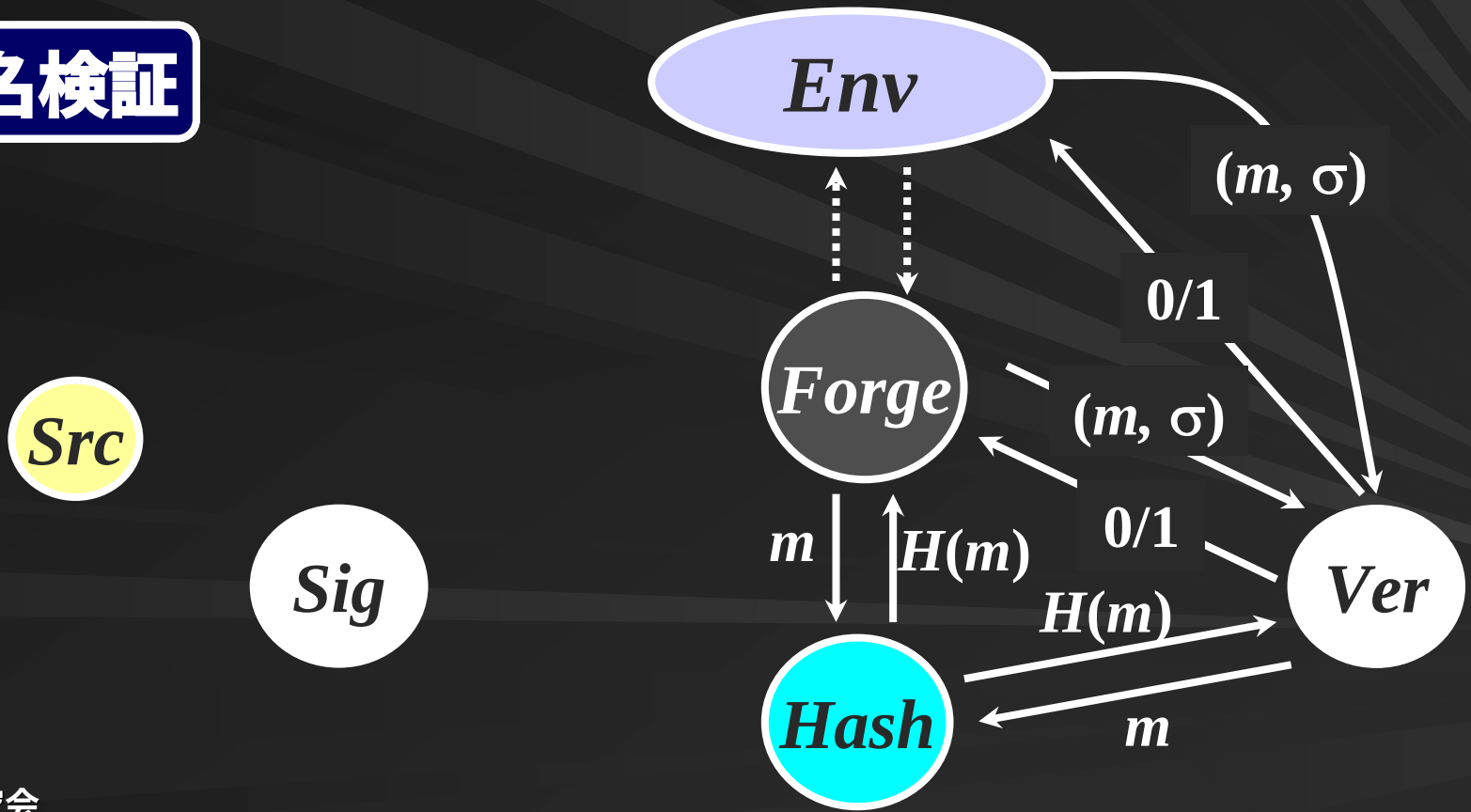
- *RS* は以下のtask-PIOA の結合で構成される．
  - *RS = Sig || Ver || Forge || Env || Src(Tdpp) || Hash*
    (*Src(Tdpp)*：落とし戸付き置換ペアを選ぶオートマトン)



Real system（RS）の構成

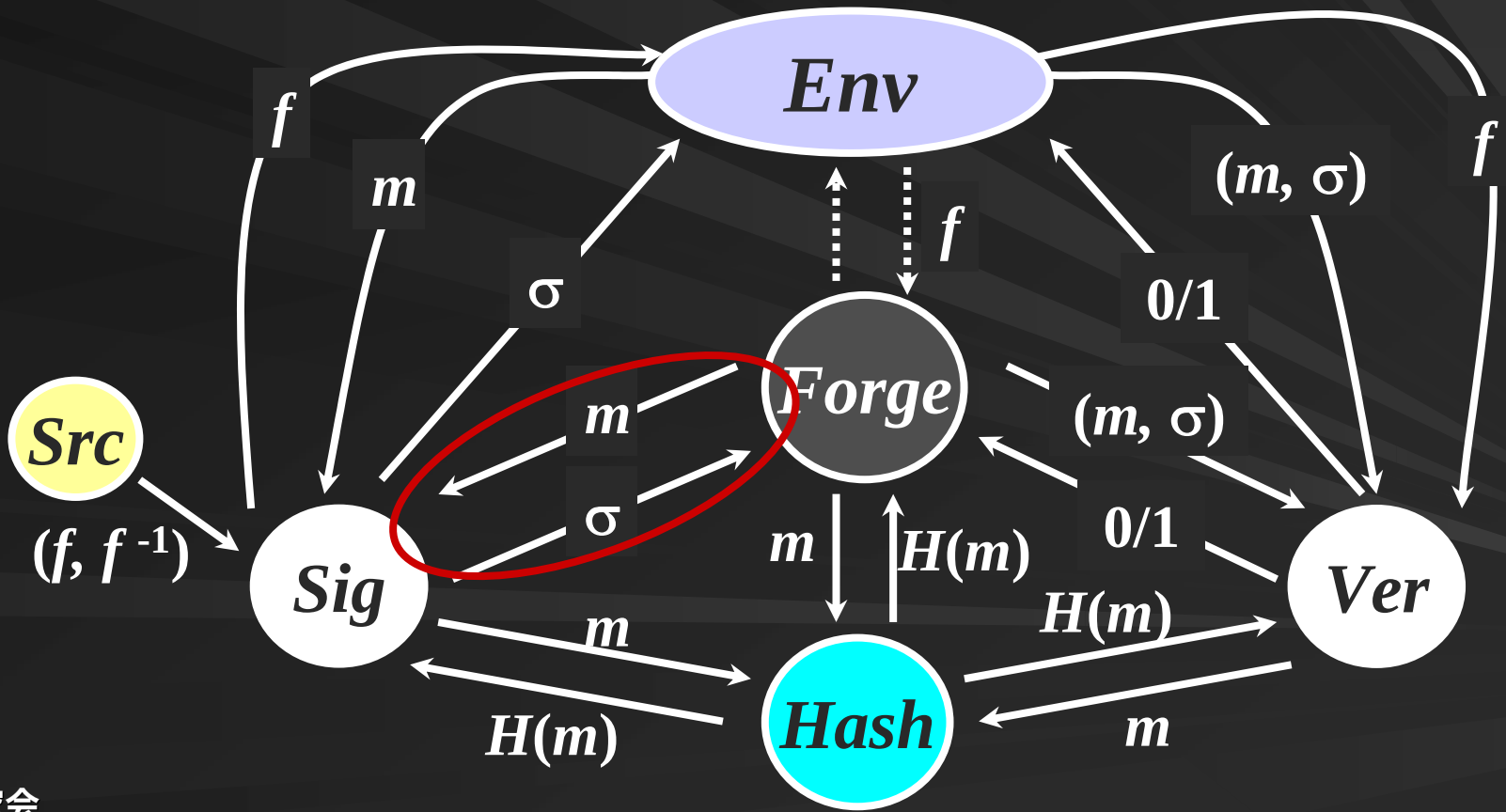
- *RS* は以下のtask-PIOA の結合で構成される．
  - *RS = Sig || Ver || Forge || Env || Src(Tdpp) || Hash*
    (*Src(Tdpp)*：落とし戸付き置換ペアを選ぶオートマトン)

署名検証



Real system（RS）の構成

- *RS* は以下のtask-PIOA の結合で構成される．
  - *RS = Sig || Ver || Forge || Env || Src(Tdpp) || Hash*
    (*Src(Tdpp)*：落とし戸付き置換ペアを選ぶオートマトン)



偽造者のcode

$Forge(MID, \mathcal{M}, \mathcal{D}, Tdp, Tdpp, \psi)$:

Signature:

Input:
- $out(f)_{fval}$, $f \in Tdp$
- $hashed(h)_{hval}$, $h \in MID \times \mathcal{D}$
- $out'(\sigma)_{\sigma val}$, $\sigma \in MID \times \mathcal{D}$
- $out'(\phi)_{\phi val}$, $\phi \in MID \times \{0,1\}$
- if $Sig \in \psi$ then
  - $rand(p)_{pval}$, $p \in Tdpp$
  - $out'(f)_{fval}$, $f \in Tdp$
  - $in(m)_{mval}$, $m \in MID \times \mathcal{M}$
- if $Ver \in \psi$ then
  - $in(V)_{Vval}$, $V \in MID \times \mathcal{M} \times \mathcal{D} \times Tdp$
- Arbitrary other new input actions

Output:
- $in(m)_{mval}$, $m \in MID \times \mathcal{M}$
- $in(V)_{Vval}$, $V \in MID \times \mathcal{M} \times \mathcal{D} \times Tdp$
- $hash(m)_{mval}$, $m \in MID \times \mathcal{M}$
- if $Sig \in \psi$ then
  - $out(f)_{fval}$, $f \in Tdp$
  - $out(\sigma)_{\sigma val}$, $\sigma \in MID \times \mathcal{D}$
  - $reply(p)_{pval}$, $p \in Tdpp$
- if $Ver \in \psi$ then $out(\phi)_{\phi val}$, $\phi \in MID \times \{0,1\}$
- Arbitrary other new output actions

Internal:
- Arbitrary new internal actions

State:
- $fval \in Tdp \cup \perp$, initially $\perp$
- $mval \in (MID \to \mathcal{M} \cup \perp)$, initially $\perp$
- $\sigma val \in (MID \to \mathcal{D} \cup \perp)$, initially $\perp$
- $Vval \in (MID \to (\mathcal{M} \cup \perp) \times (\mathcal{D} \cup \perp) \times (Tdp \cup \perp))$, initially $\perp$
- $\phi val \in (MID \to [0,1, \perp])$, initially $\perp$
- $hval \in (MID \to \mathcal{D} \cup \perp)$, initially $\perp$
- if $Sig \in \psi$ then $pval \in Tdpp \cup \perp$, initially $\perp$
- Arbitrary other new variables

Transitions:

$out(f)_{fval}$
- Effect:
  - if $fval = \perp$ then $fval := f$

$hashed(h)_{hval}$
- Effect:
  - if $hval(h.id) = \perp$ then $hval(h.id) := h.message$

$out'(\sigma)_{\sigma val}$
- Effect:
  - if $\sigma val(\sigma.id) = \perp$ then $\sigma val(\sigma.id) := \sigma.signature$

$out'(\phi)_{\phi val}$
- Effect:
  - if $\phi val(\phi.id) = \perp$ then $\phi val(\phi.id) := \phi.result$

$in(m)_{mval}$
- Precondition:
  - $mval(m.id) = m.message$
- Effect:
  - none

$in(V)_{Vval}$
- Precondition:
  - $Vval(V.id).message = V.message$,
  - $Vval(V.id).signature = V.signature$,
  - $Vval(V.id).verkey = V.verkey$
- Effect:
  - none

$hash(m)_{mval}$
- Precondition:
  - $mval(m.id) = m.message$
- Effect:
  - none

$out(f)_{fval}$
- Precondition:
  - $fval \neq \perp$
- Effect:
  - none

$out(\sigma)_{\sigma val}$
- Precondition:
  - $\sigma val(\sigma.id) = \sigma.signature$
- Effect:
  - none

$out(\phi)_{\phi val}$
- Precondition:
  - $\phi val(\phi.id) = \phi.result$
- Effect:
  - none

$rand(p)_{pval}$, $out'(f)_{fval}$, $in(m)_{mval}$ or $in(V)_{Vval}$
- Effect:
  - Arbitrary changes to new state variables

New input action
- Effect:
  - Arbitrary changes to new state variables

New output or internal action
- Precondition:
  - Arbitrary
- Effect:
  - Arbitrary changes to new state variables

Tasks:
- $\{in(*)_{mval}\}$, $\{in(*)_{Vval}\}$, $\{hash(*)_{mval}\}$.
- If $Sig \in \psi$ then $\{out(*)_{fval}\}$, $\{out(*)_{\sigma val}\}$.
- If $Ver \in \psi$ then $\{out(*)_{\phi val}\}$.

適応的選択文書攻撃

電子署名の理想機能

- 本質的には…
  - 「正規の署名生成を経た署名だけが検証に合格」

正規署名

- 例外：
  - 署名者が支配されている場合
  - 正しい検証鍵で検証していない場合
  - 偽造者が正規署名を知っているメッセージに対して（正規署名とは異なる）偽造が行われた場合

検証のイベント分け

署名者が支配されていない場合

input: (m, σ, vk')

output: $\phi \in \{0, 1\}$

- $vk' \neq vk$ → $\phi = \theta$ (Simが決定)
- $vk' = vk$
  - $\exists$ legitimate $(m_j, \sigma_j) = (m, \sigma)$ → $\phi = 1$
  - all legitimate $(m_j, \sigma_j) \neq (m, \sigma)$
    - $\exists$ legitimate $m_j = m$ → $\phi = \theta$ (Simが決定)
    - all legitimate $m_j \neq m$ → $\phi = 0$

存在的偽造不可能性

理想機能のcode

$Funct(MID, \mathcal{M}, \mathcal{D}, Tdp, \psi)$:

Signature:

Input:
- $reply(f)_{fval}$, $f \in Tdp$
- $in(m)_{mval}$, $m \in MID \times \mathcal{M}$
- $reply(\sigma)_{\sigma val}$, $\sigma \in MID \times \mathcal{D}$
- $in(V)_{Vval}$, $V \in MID \times \mathcal{M} \times \mathcal{D} \times Tdp$
- $reply(\theta)_{\theta val}$, $\theta \in MID \times \{0, 1\}$

Output:
- $out(f)_{fval}$, $f \in Tdp$
- $ask(m)_{mval}$, $m \in MID \times \mathcal{M}$
- $out'(\sigma)_{\sigma val}$, $\sigma \in MID \times \mathcal{D}$
- $out'(\phi)_{\phi val}$, $\phi \in MID \times \{0, 1\}$
- if $Sig \notin \psi$ then $out(\sigma)_{\sigma val}$, $\sigma \in MID \times \mathcal{D}$
- if $Ver \notin \psi$ then $out(\phi)_{\phi val}$, $\phi \in MID \times \{0, 1\}$

Internal:
- if $Sig \notin \psi$ then $fix_i - \phi val$ $(1 \le i \le |MID|)$
- if $Sig \in \psi$ then $fix'_i - \phi val$ $(1 \le i \le |MID|)$

State:
- $fval \in Tdp \cup \perp$, initially $\perp$
- $mval \in (MID \to \mathcal{M} \cup \perp)$, initially $\perp$
- $\sigma val \in (MID \to \mathcal{D} \cup \perp)$, initially $\perp$
- $Vval \in (MID \to (\mathcal{M} \cup \perp) \times (\mathcal{D} \cup \perp) \times (Tdp \cup \perp))$, initially $\perp$
- $\phi val \in (MID \to [0, 1, \perp])$, initially $\perp$
- $\theta val \in (MID \to [0, 1, \perp])$, initially $\perp$

Transitions:

$reply(f)_{fval}$
Effect:
if $fval = \perp$ then $fval := f$

$in(m)_{mval}$
Effect:
if $mval(m.id) = \perp$ then $mval(m.id) := m.message$

$reply(\sigma)_{\sigma val}$
Effect:
if $\sigma val(\sigma.id) = \perp$ then $\sigma val(\sigma.id) := \sigma.signature$

$in(V)_{Vval}$
Effect:
if $Vval(V.id) = \perp$ then $Vval(V.id).message := V.message$,
$Vval(V.id).signature := V.signature$,
$Vval(V.id).verkey := V.verkey$

$reply(\theta)_{\theta val}$
Effect:
if $\theta val(\theta.id) = \perp$ then $\theta val(\theta.id) := \theta.result$

$out(f)_{fval}$
Precondition:
$fval \neq \perp$
Effect:
none

$ask(m)_{mval}$
Precondition:
$mval(m.id) = m.message$
Effect:
none

$out(\sigma)_{\sigma val}$ or $out'(\sigma)_{\sigma val}$
Precondition:
$\sigma val(\sigma.id) = \sigma.signature$
Effect:
none

$ask(V)_{Vval}$
Precondition:
$Vval(V.id).message = V.message$,
$Vval(V.id).signature = V.signature$,
$Vval(V.id).verkey = V.verkey$
Effect:
none

$out(\phi)_{\phi val}$ or $out'(\phi)_{\phi val}$
Precondition:
$\phi val(\phi.id) = \phi.result$
Effect:
none

$fix_i - \phi val$ $(1 \le i \le |MID|)$
Precondition:
$fval \neq \perp, Vval(i) \neq \perp, \theta val(i) \neq \perp, \phi val(i) = \perp$
Effect:
if $(\bigvee_{j \in MID} ((mval(j) = Vval(i).message) \wedge (\sigma val(j) = Vval(i).signature))) \wedge (fval = Vval(i).verkey)$
then $\phi val(i) = 0$
else, if $(\bigwedge_{j \in MID} (mval(j) \neq Vval(i).message)) \wedge (fval = Vval(i).verkey)$
then $\phi val(i) = 0$
else, $\phi val(i) = \theta val(i)$

$fix'_i - \phi val$ $(1 \le i \le |MID|)$
Precondition:
$fval \neq \perp, Vval(i) \neq \perp, \theta val(i) \neq \perp, \phi val(i) = \perp$
Effect:
if $(\bigvee_{j \in MID} ((mval(j) = Vval(i).message) \wedge (\sigma val(j) = Vval(i).signature))) \wedge (fval = Vval(i).verkey)$
then $\phi val(i) = 0$
else, $\phi val(i) = \theta val(i)$

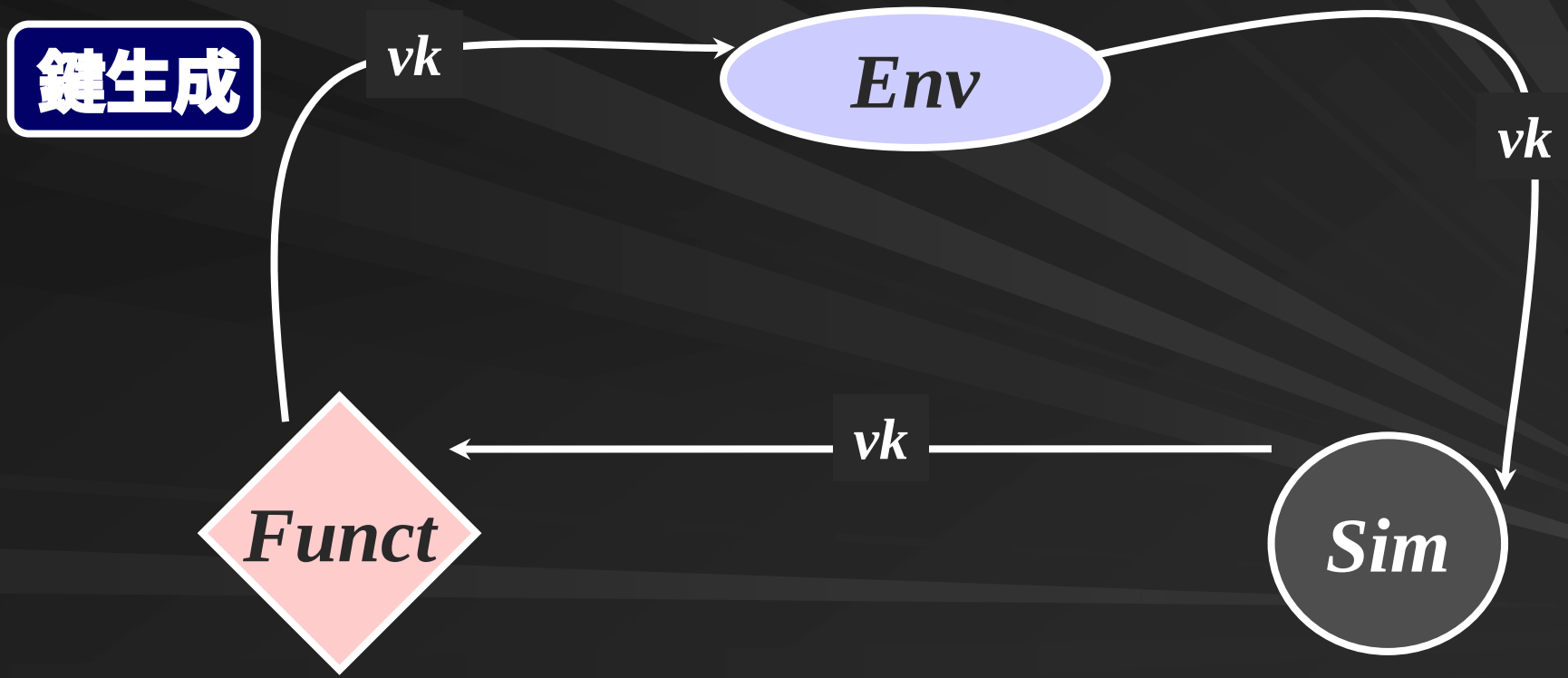
Tasks:
$\{out(*)_{fval}\}$, $\{ask(*)_{mval}\}$, $\{out'(*)_{\sigma val}\}$, $\{ask(*)_{Vval}\}$, $\{out'(*)_{\phi val}\}$.
If $Sig \notin \psi$ then $fix_i - \phi val$ $(1 \le i \le |MID|)$, $\{out(*)_{\sigma val}\}$.
If $Sig \in \psi$ then $fix'_i - \phi val$ $(1 \le i \le |MID|)$.
If $Ver \notin \psi$ then $\{out(*)_{\phi val}\}$.

適応的選択
文書攻撃

存在的偽造
不可能

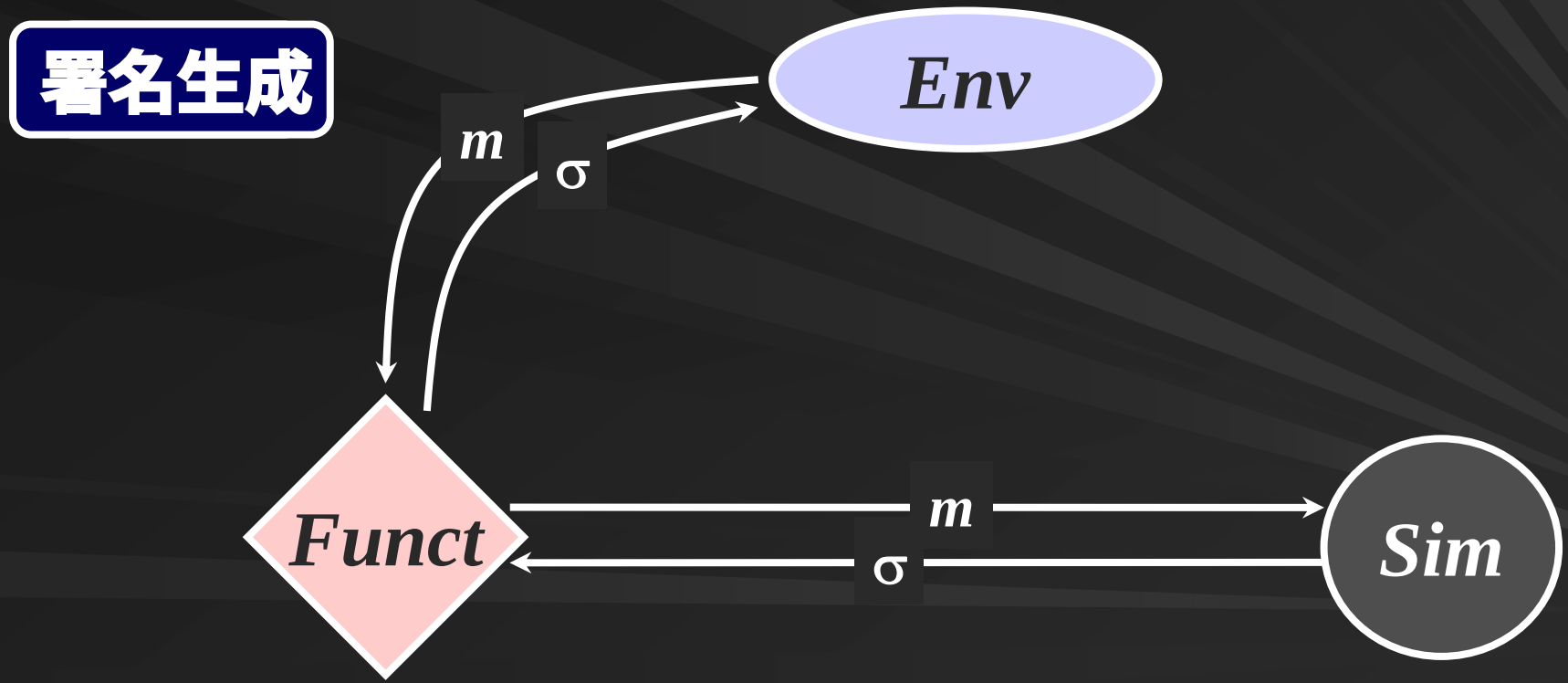
Ideal system（IS）の構成

- *IS*は以下のtask-PIOA の結合で構成される．
  - *IS = Funct || Sim || Env*



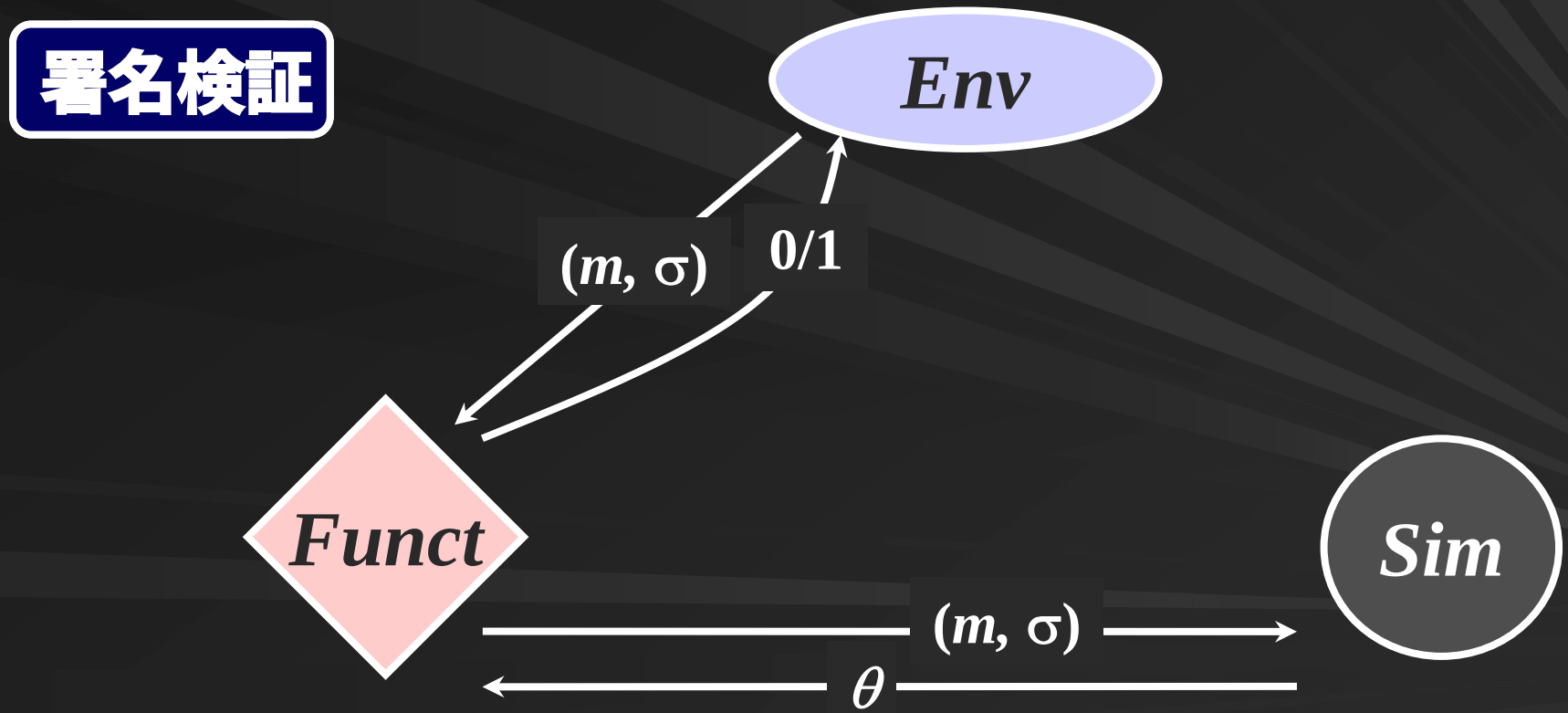
Ideal system（IS）の構成

- *IS*は以下のtask-PIOA の結合で構成される．
  - *IS = Funct || Sim || Env*



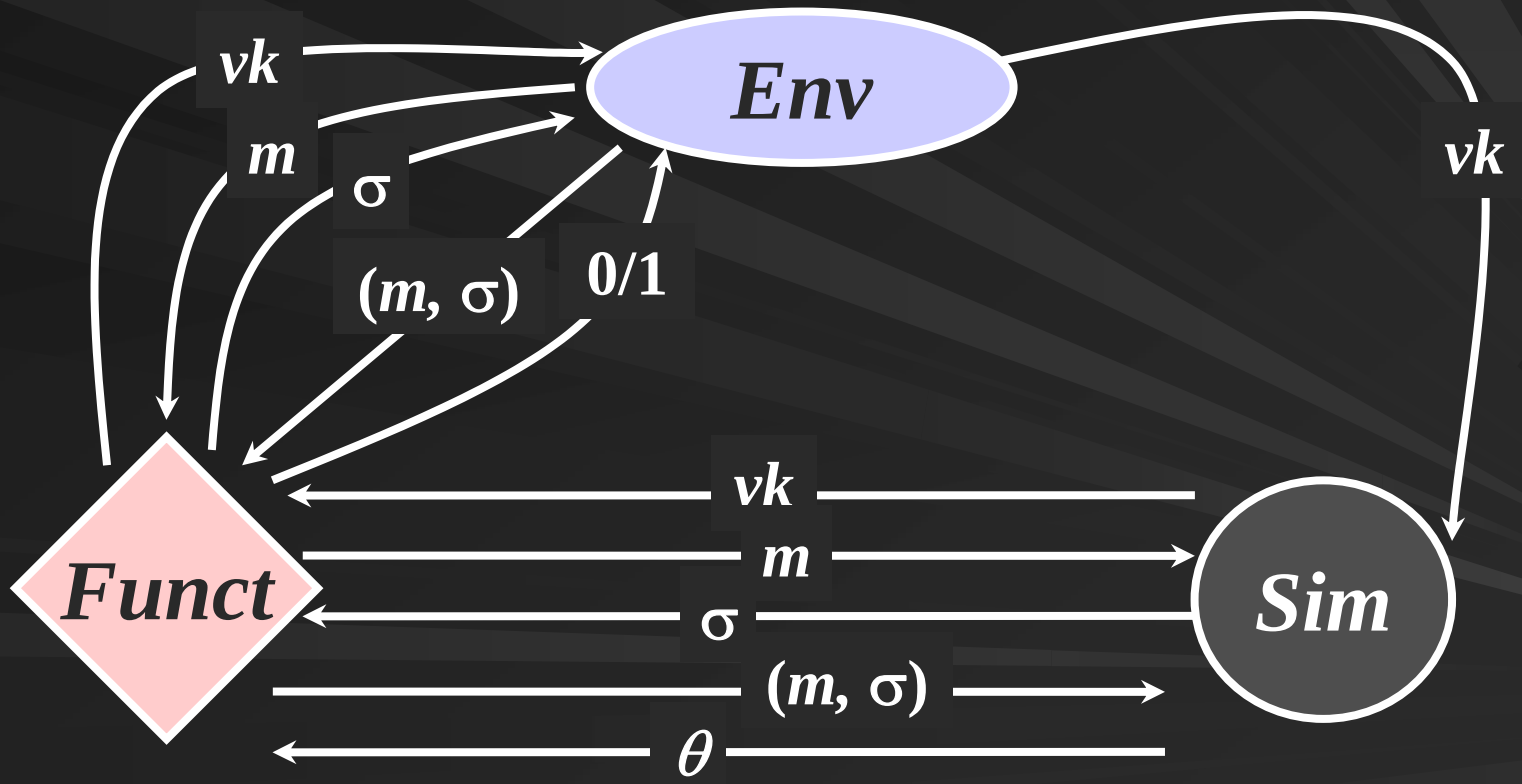
Ideal system（IS）の構成

- *IS*は以下のtask-PIOA の結合で構成される．
  - *IS = Funct || Sim || Env*



Ideal system（IS）の構成

- *IS*は以下のtask-PIOA の結合で構成される．
  - *IS = Funct || Sim || Env*



安全性証明の概要

$RS \leq_{neg,pt} IS$ 証明の分割

- ≤ の推移性を用いて複数の部分に分割．
- 落とし戸付き置換の一方向性による分割
  - $SOW' \leq_{neg,pt} SOW$
- 中間システム (*Int*) による分割
  - 完全インプリメンテーション（等価変換）

落とし戸付き関数の一方向性

計算量理論的モデルでの定式化

$\forall$ PPT adversary A

$\Pr[f(x) = y \mid A(y) = x] \leq \mathrm{negl}.$

Task-PIOAフレームワークで再定式化

Task-PIOAでの定式化

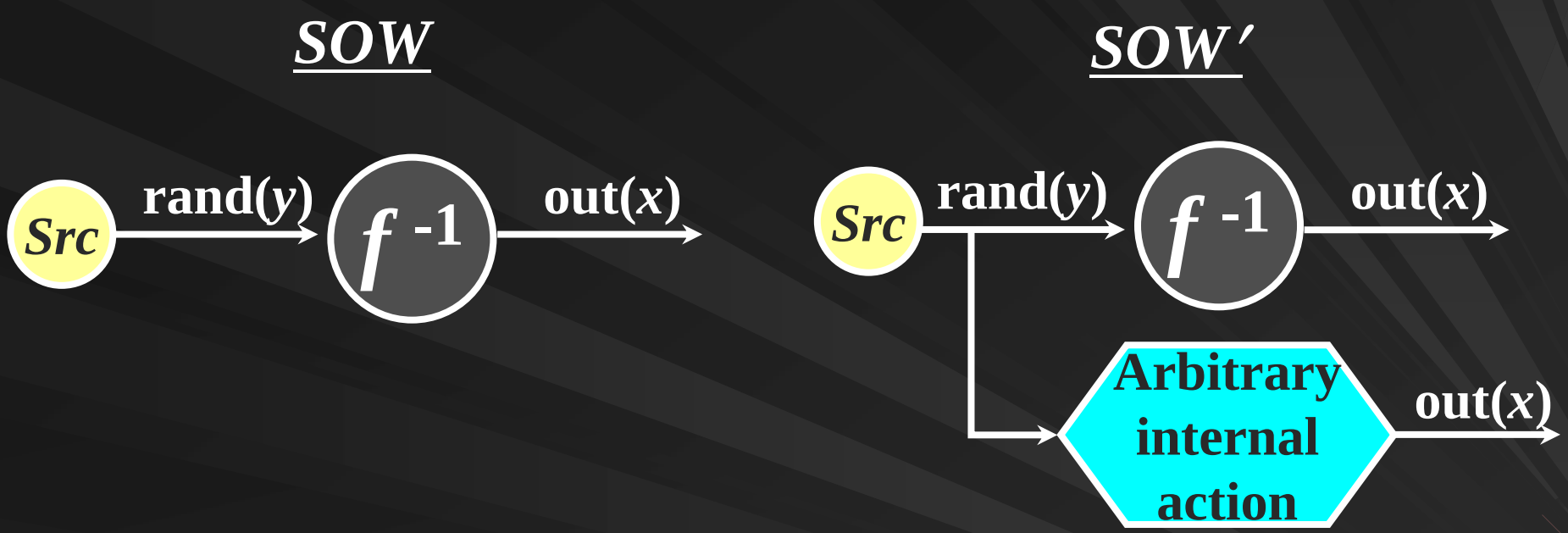
$SOW' \leq_{neg,\, pt} SOW$

ただし,

SOW': 何らかの手段で y から x を出力する.

SOW : f^{-1}を用いて y から x を出力する.

SOWPとSOWP′



- $SOW' \leq_{neg,pt} SOW$

⇒ f^{-1}を用いずに x を出力できる確率はnegl.

まとめ

- Task-PIOAフレームワークは電子署名における能動的攻撃を扱うことが可能
  - FDH署名の適応的選択文書攻撃に対する存在的偽造不可能性を（hand proof的に）証明
  - ただし，今回の定式化では署名回数オーダー分のStateとTaskが必要

- 今後の課題
  - 能動的かつ適応的攻撃者に関する考察
  - 証明不可能性の考察