The Power of Halting in Security Games

ProTeCS 2025

Igors Stepanovs

Halting in code-based game-playing proofs

Code-Based Game-Playing Proofs
and the Security of Triple Encryption

Mihir Bellare *   Phillip Rogaway †

Adversary:

abort(x)

- Adversary halts immediately
- Adversary returns x

Security game:

abort(true)
or
abort(false)

- Game halts immediately
- Game returns true or false

Example: abort(false) in multi-key reduction

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE) where adversary has access to oracle:

$\underline{\textsc{NewGroup}(g)}$

require $\mathsf{K}[g] = \perp$

$\mathsf{K}[g] \leftarrow_{\$} \{0,1\}^{\mathsf{NE.kl}}$

Weaker security notion (e.g. IND-CPA, KR, or OW) where adversary has access to oracle:

$\underline{\textsc{NewGroup}(g)}$

require $\mathsf{K}[g] = \perp$

$\mathsf{K}[g] \leftarrow_{\$} \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE) where adversary has access to oracle:

$\underline{\textsc{NewGroup}(g)}$

require $\mathsf{K}[g] = \bot$

$\mathsf{K}[g] \leftarrow\!\!\$ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\underline{\textsc{NewGroup}(g)}$

require $\mathsf{K}[g] = \bot$

$\mathsf{K}[g] \leftarrow\!\!\$ \{0,1\}^{\mathsf{NE.kl}}$

If $\mathsf{K}[g] \in S$ then

  bad $\leftarrow$ true

$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Weaker security notion (e.g. IND-CPA, KR, or OW) where adversary has access to oracle:

$\underline{\textsc{NewGroup}(g)}$

require $\mathsf{K}[g] = \bot$

$\mathsf{K}[g] \leftarrow\!\!\$ \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and
E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE)
where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
  $\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}} \setminus S$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Weaker security notion (e.g. IND-CPA, KR, or OW)
where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE) where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
    bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
    bad $\leftarrow$ true
    $\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}} \setminus S$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
    bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Weaker security notion (e.g. IND-CPA, KR, or OW) where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE) where adversary has access to oracle:

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
  $\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}} \setminus S$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Rely on unique keys for other proof steps

Birthday bound

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Weaker security notion (e.g. IND-CPA, KR, or OW) where adversary has access to oracle:

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE) where adversary has access to oracle:

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
  $\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}} \setminus S$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Rely on unique keys for other proof steps

Birthday bound

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  bad $\leftarrow$ true
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Equivalent

Weaker security notion (e.g. IND-CPA, KR, or OW) where adversary has access to oracle:

$\textsc{NewGroup}(g)$
require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{NE.kl}}$

Example: abort(false) in multi-key reduction

Symmetric Signcryption and
E2EE Group Messaging in Keybase

Joseph Jaeger[1], Akshaya Kumar[1], and Igors Stepanovs[2]

Strong security notion (e.g. AE)
where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\$ \{0,1\}^{\mathsf{NE.kl}}$

Equivalent

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\$ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Birthday bound

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\$ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$
  abort(false)
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Free transition

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\$ \{0,1\}^{\mathsf{NE.kl}}$
If $\mathsf{K}[g] \in S$ then
  $\mathsf{bad} \leftarrow \mathsf{true}$
$S \leftarrow S \cup \{\mathsf{K}[g]\}$

Equivalent

Weaker security notion (e.g. IND-CPA, KR, or OW)
where adversary has access to oracle:

$\textsc{NewGroup}(g)$

require $\mathsf{K}[g] = \bot$
$\mathsf{K}[g] \leftarrow\$ \{0,1\}^{\mathsf{NE.kl}}$

Rely on unique keys for other proof steps

Example: abort(false) in PRP-PRF switch

$\mathsf{X}(u)$ // $|u| = 128$

$\textbf{if } \mathsf{A}[u] = \bot :$

$\quad \mathsf{A}[u] \leftarrow\!\$\ \{0,1\}^{128} \setminus \mathsf{A}$

$\quad \mathsf{B}[\mathsf{A}[u]] \leftarrow u$

$\textbf{return } \mathsf{A}[u]$

Random permutation
(lazy sampling)

Analysis of the Telegram Key Exchange*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[2], Eyal Ronen[3], and Igors Stepanovs[4]

Example: abort(false) in PRP-PRF switch

$\mathsf{X}(u)$ // $|u| = 128$

$\textbf{if } \mathsf{A}[u] = \bot :$

$\quad \mathsf{A}[u] \leftarrow\!\!\$ \{0,1\}^{128} \setminus \mathsf{A}$

$\quad \mathsf{B}[\mathsf{A}[u]] \leftarrow u$

$\textbf{return } \mathsf{A}[u]$

$\mathsf{X}(u)$ // $|u| = 128$

$\textbf{if } \mathsf{A}[u] = \bot :$

$\quad \mathsf{A}[u] \leftarrow\!\!\$ \{0,1\}^{128}$

$\quad \textbf{if } \mathsf{A}[u] \in \mathcal{R} :$

$\qquad \mathtt{bad}_2 \leftarrow \mathtt{true}$

$\qquad \mathsf{A}[u] \leftarrow\!\!\$ \{0,1\}^{128} \setminus \mathcal{R}$ // G_3–G_4

$\qquad \textbf{abort}(\mathtt{false})$ // G_5

$\quad \mathsf{B}[\mathsf{A}[u]] \leftarrow u$

$\quad \mathcal{D} \leftarrow \mathcal{D} \cup \{u\} ; \mathcal{R} \leftarrow \mathcal{R} \cup \{\mathsf{A}[u]\}$

$\textbf{return } \mathsf{A}[u]$

Random permutation
(lazy sampling)

$\mathbf{G_4}$ – random permutation
$\mathbf{G_5}$ – random function

$$\Pr[G_4] - \Pr[G_5] \le \Pr[\mathrm{bad}_2^{G_5}]$$

Analysis of the Telegram Key Exchange*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[2], Eyal Ronen[3], and Igors Stepanovs[4]

Example: abort(false) in PRP-PRF switch

```
X(u)    // |u| = 128
if A[u] = ⊥ :
    A[u] ←$ {0,1}^128 \ A
    B[A[u]] ← u
return A[u]
```

Random permutation
(lazy sampling)

```
X(u)    // |u| = 128
if A[u] = ⊥ :
    A[u] ←$ {0,1}^128
    if A[u] ∈ R :
        bad_2 ← true
        A[u] ←$ {0,1}^128 \ R    // G3–G4
        abort(false)    // G5
    B[A[u]] ← u
    D ← D ∪ {u} ; R ← R ∪ {A[u]}
return A[u]
```

G_4 – random permutation
G_5 – random function

$$\Pr[G_4] - \Pr[G_5] \leq \Pr[\mathsf{bad}_2^{G_5}]$$

```
X(u)    // |u| = 128
if A[u] = ⊥ :
    A[u] ←$ {0,1}^128
    if A[u] ∈ R :
        abort(false)    // G6–G9
        A[u] ←$ {0,1}^128 \ R    // G10–G11
    B[A[u]] ← u
    D ← D ∪ {u} ; R ← R ∪ {A[u]}
return A[u]
```

G_9 – random function
G_{10} – random permutation

$$\Pr[G_9] - \Pr[G_{10}] \leq 0.$$

Analysis of the Telegram Key Exchange*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[2], Eyal Ronen[3], and Igors Stepanovs[4]

Example: abort(false) in PRP-PRF switch

$\mathsf{X}(u)$ // $|u| = 128$

- **if** $\mathsf{A}[u] = \bot$:
  - $\mathsf{A}[u] \leftarrow\$ \{0,1\}^{128} \setminus \mathsf{A}$
  - $\mathsf{B}[\mathsf{A}[u]] \leftarrow u$
- **return** $\mathsf{A}[u]$

Random permutation
(lazy sampling)

$\mathsf{X}(u)$ // $|u| = 128$

- **if** $\mathsf{A}[u] = \bot$:
  - $\mathsf{A}[u] \leftarrow\$ \{0,1\}^{128}$
  - **if** $\mathsf{A}[u] \in \mathcal{R}$:
    - $\mathtt{bad}_2 \leftarrow \mathtt{true}$
    - $\mathsf{A}[u] \leftarrow\$ \{0,1\}^{128} \setminus \mathcal{R}$ // G_3–G_4
    - **abort**(false) // G_5
  - $\mathsf{B}[\mathsf{A}[u]] \leftarrow u$
  - $\mathcal{D} \leftarrow \mathcal{D} \cup \{u\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathsf{A}[u]\}$
- **return** $\mathsf{A}[u]$

G_4 – random permutation
G_5 – random function

$$\Pr[G_4] - \Pr[G_5] \leq \Pr[\mathrm{bad}_2^{G_5}]$$

$\mathsf{X}(u)$ // $|u| = 128$

- **if** $\mathsf{A}[u] = \bot$:
  - $\mathsf{A}[u] \leftarrow\$ \{0,1\}^{128}$
  - **if** $\mathsf{A}[u] \in \mathcal{R}$:
    - **abort**(false) // G_6–G_9
    - $\mathsf{A}[u] \leftarrow\$ \{0,1\}^{128} \setminus \mathcal{R}$ // G_{10}–G_{11}
  - $\mathsf{B}[\mathsf{A}[u]] \leftarrow u$
  - $\mathcal{D} \leftarrow \mathcal{D} \cup \{u\}$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathsf{A}[u]\}$
- **return** $\mathsf{A}[u]$

G_9 – random function
G_{10} – random permutation

$$\Pr[G_9] - \Pr[G_{10}] \leq 0.$$

Rely on random function to prove IND$ security of a block cipher mode of operation (e.g. AES-CBC)

Analysis of the Telegram Key Exchange*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[2], Eyal Ronen[3], and Igors Stepanovs[4]

Example: abort(false) is useful beyond switching back and forth

Oracle $\text{ENC}(m_0, m_1)$

1 : **require** $(c^*_{\text{rsa}} = \bot) \wedge (|m_0| = |m_1|)$
2 : **require** $m_0, m_1 \in \mathcal{M}$
3 : $\text{attempt} \leftarrow 1$
4 : $p_{\text{rsa}} \leftarrow\$ \{0,1\}^{2048}$
5 : $z \leftarrow p_{\text{rsa}}$ // Parse p_{rsa} as an integer.
6 : $r \,\|\, c_{\text{ige}} \leftarrow p_{\text{rsa}}$ // s.t. $|r| = 256, |c_{\text{ige}}| = 1792$.
7 : **if** $\mathsf{T}[c_{\text{ige}}] \neq \bot$: **abort**(false)
8 : **if** $\text{ige-ciphertext-to-key-map}[c_{\text{ige}}] \neq \bot$:
9 :   **abort**(false)
10 : $K \leftarrow\$ \{0,1\}^{256}$
11 : $\text{ige-ciphertext-to-key-map}[c_{\text{ige}}] \leftarrow (r, K)$
12 : **if** $K \in S_{\text{IC}}$: **abort**(false)
13 : **if** $\text{ige-key-to-data-map}[K] \neq \bot$:
14 :   **abort**(false)
15 : $pad \leftarrow\$ \{0,1\}^{1536-|m_b|}$
16 : $m_{\text{padded}} \leftarrow m_b \,\|\, pad$
17 : **if** $c_{\text{ige}} = K \,\|\, m_{\text{padded}}$: **abort**(false)
18 : $h \leftarrow \mathrm{H}(K \,\|\, m_{\text{padded}})$
19 : $p_{\text{ige}} \leftarrow \text{reverse}(m_{\text{padded}}) \,\|\, h$
20 : $\text{ige-key-to-data-map}[K] \leftarrow (p_{\text{ige}}, c_{\text{ige}})$
21 : **if** $z \notin \mathbb{Z}_N$:
22 : $\text{attempt} \leftarrow \text{attempt} + 1$
23 :   **if** $\text{attempt} > \text{max-attempts}$:
24 :   **abort**(false)
25 :   **goto line** 4
26 : $c^*_{\text{rsa}} \leftarrow z^e \bmod N$
27 : $c^*_{\text{ige}} \leftarrow c_{\text{ige}}$; $K^* \leftarrow K$
28 : **return** c^*_{rsa}

Game 19 for the proof of Telegram's variant of OAEP+ scheme.

(e.g. SHA-256 is used in different contexts, with no domain separation.)

Analysis of the Telegram Key Exchange*

Martin R. Albrecht[1], Lenka Mareková[2], Kenneth G. Paterson[2], Eyal Ronen[3], and Igors Stepanovs[4]

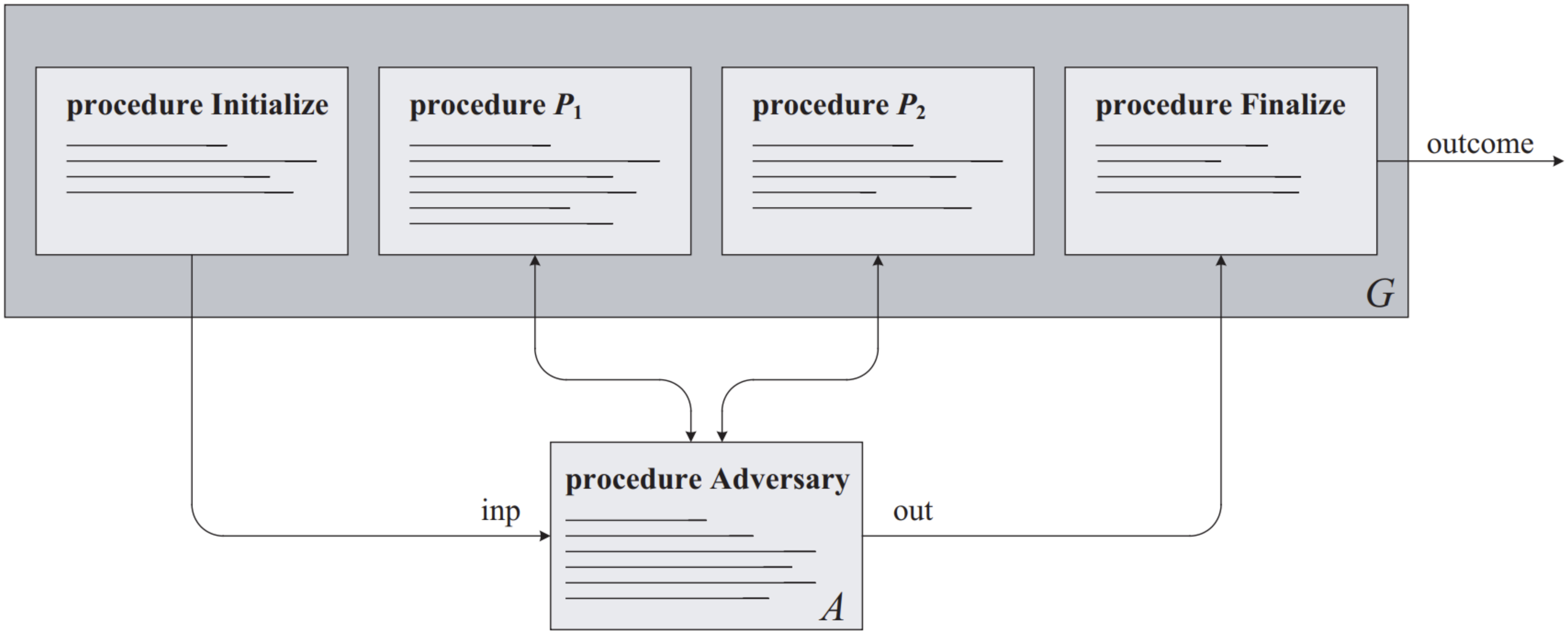
Caution: use with care

Code-Based Game-Playing Proofs and the Security of Triple Encryption

Mihir Bellare * Phillip Rogaway †

Adv := Pr[Game => true] (e.g. for search games).

When **abort(false)** is used, advantage might no longer be well-defined.



An alternative to abort(false)?

Hardening Signature Schemes via Derive-then-Derandomize: Stronger Security Proofs for EdDSA

Mihir Bellare[1] Hannah Davis[2] Zijing Di[3]

Game G_2, $\boxed{G_3}$

$\textsc{Fin}(c')$:

$\boxed{\text{if bad then return } 0}$
return c'

Game G_4

$\textsc{Priv}(X)$:

$w \leftarrow \mathbf{F}[\mathcal{S}[\mathsf{H}](\cdot, \mathcal{G}_{\mathtt{all}})](X)$
return w

$\mathcal{S}[\mathsf{H}](Y, \mathcal{G})$:

$(y, m) \leftarrow Y$
if $\exists z$ such that $(y, z, m) \in \mathcal{G}$.edges
  return z
$M \leftarrow \mathcal{G}$.FindPath(IV, y)
$M_{\mathtt{all}} \leftarrow \mathcal{G}_{\mathtt{all}}$.FindPath$(IV, y)$
if $M \neq \perp$ and $\mathsf{unpad}(M \| m) \neq \perp$ then
  if $\mathrm{T}_{\mathsf{h}}[Y, M] \neq \perp$ then $z \leftarrow \mathrm{T}_{\mathsf{h}}[Y, M]$
  else $z \leftarrow\!\!\$\ \mathsf{Out}^{-1}(\mathsf{H}(\mathsf{unpad}(M \| m)))$
    $\mathrm{T}_{\mathsf{h}}[Y, M] \leftarrow z$
else if $\mathrm{T}_{\mathsf{h}}[Y] \neq \perp$ then $z \leftarrow \mathrm{T}_{\mathsf{h}}[Y]$
else $z \leftarrow\!\!\$\ \{0,1\}^{2k}$; $\mathrm{T}_{\mathsf{h}}[Y] \leftarrow z$
if $(z \in \mathcal{G}_{\mathtt{all}}$.nodes and $(y, z, m) \notin \mathcal{G}_{\mathtt{all}}$.edges$)$
  or $M \neq M_{\mathtt{all}}$
    $\mathsf{bad} \leftarrow \mathsf{true}$
add (y, z, m) to $\mathcal{G}$.edges
add (y, z, m) to $\mathcal{G}_{\mathtt{all}}$.edges
return z

Thank you!