Adaptively-Secure Secret Sharing

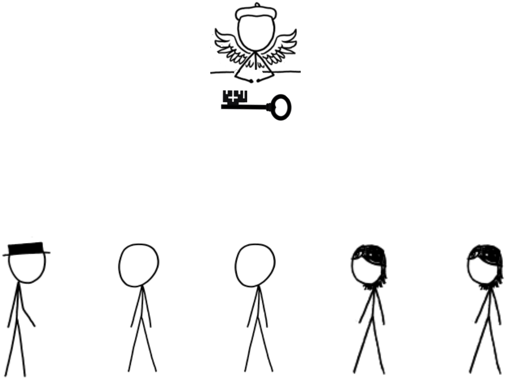
Chethan Kamath
(Joint work with Zahra Jafargholi, Karen Klein, Ilan Komargodski, Krzysztof Pietrzak and Daniel Wichs)

IST AUSTRIA

October 20, 2017, Czech Academy of Sciences, Prague

Overview

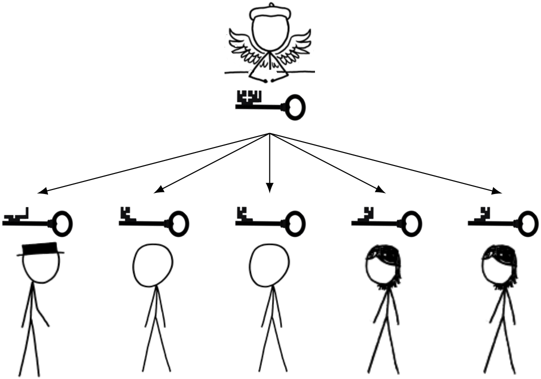
Secret Sharing



xkcd.com

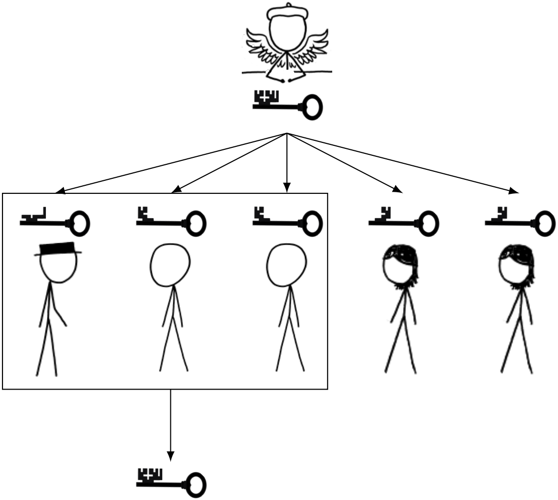
Secret Sharing

1.) Share



xkcd.com

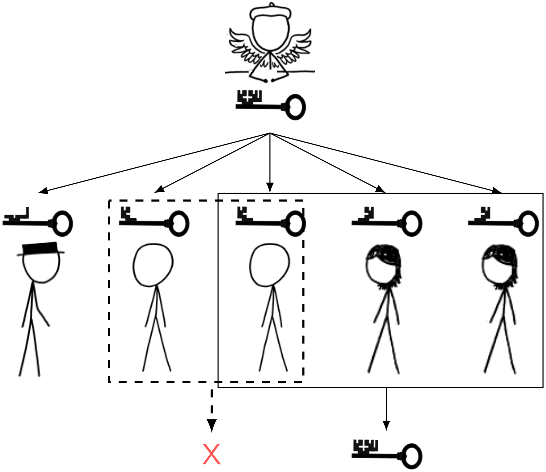
1.) Share 2.) Reconstruct



xkcd.com

Secret Sharing

1.) Share 2.) Reconstruct 3.) Access structure



xkcd.com

Definitons

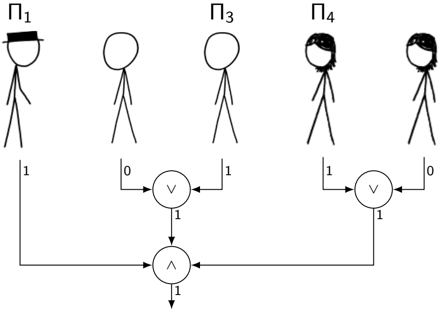
- $\mathsf{Share}(S) \to \Pi_1, \ldots, \Pi_n / \mathsf{Reconstruct}(\Pi_{\mathcal{X}}) \to S$ for $\mathcal{X} \subseteq [n]$

Definitons

- Share(S) $\to \Pi_1, \ldots, \Pi_n$/Reconstruct($\Pi_{\mathcal{X}}$) $\to S$ for $\mathcal{X} \subseteq [n]$
- Access structure: monotone Boolean circuit
  - input: $\mathbf{1}_{\mathcal{X}} \in \{0,1\}^n$
  - output: 1 if $\mathcal{X}$ is qualified, 0 otherwise

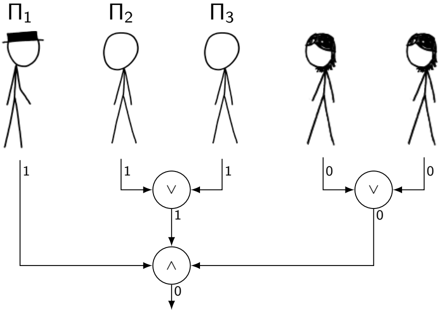
Definitons

- $\text{Share}(S) \rightarrow \Pi_1, \ldots, \Pi_n / \text{Reconstruct}(\Pi_{\mathcal{X}}) \rightarrow S$ for $\mathcal{X} \subseteq [n]$
- Access structure: monotone Boolean circuit
  - input: $\mathbf{1}_{\mathcal{X}} \in \{0,1\}^n$
  - output: 1 if $\mathcal{X}$ is qualified, 0 otherwise



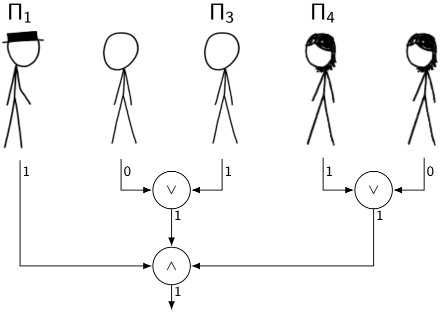
Definitons

- Share(S) $\rightarrow \Pi_1, \ldots, \Pi_n$/Reconstruct($\Pi_{\mathcal{X}}$) $\rightarrow S$ for $\mathcal{X} \subseteq [n]$
- Access structure: monotone Boolean circuit
  - input: $\mathbf{1}_{\mathcal{X}} \in \{0,1\}^n$
  - output: 1 if $\mathcal{X}$ is qualified, 0 otherwise



Definitons

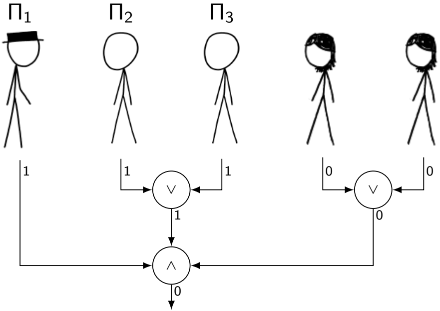
- Share(S) $\to \Pi_1, \ldots, \Pi_n$/Reconstruct($\Pi_{\mathcal{X}}$) $\to S$ for $\mathcal{X} \subseteq [n]$
- Access structure: monotone Boolean circuit
  - input: $\mathbf{1}_{\mathcal{X}} \in \{0,1\}^n$
  - output: 1 if $\mathcal{X}$ is qualified, 0 otherwise



- Completeness: qualified $\mathcal{X}$ can reconstruct

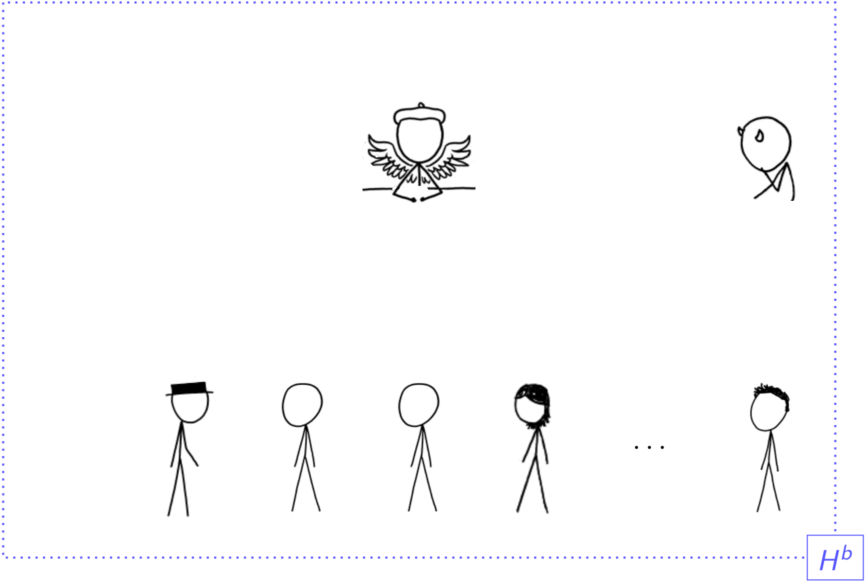
Definitons

- $\text{Share}(S) \to \Pi_1, \ldots, \Pi_n / \text{Reconstruct}(\Pi_{\mathcal{X}}) \to S$ for $\mathcal{X} \subseteq [n]$
- Access structure: monotone Boolean circuit
  - input: $\mathbf{1}_{\mathcal{X}} \in \{0,1\}^n$
  - output: 1 if $\mathcal{X}$ is qualified, 0 otherwise

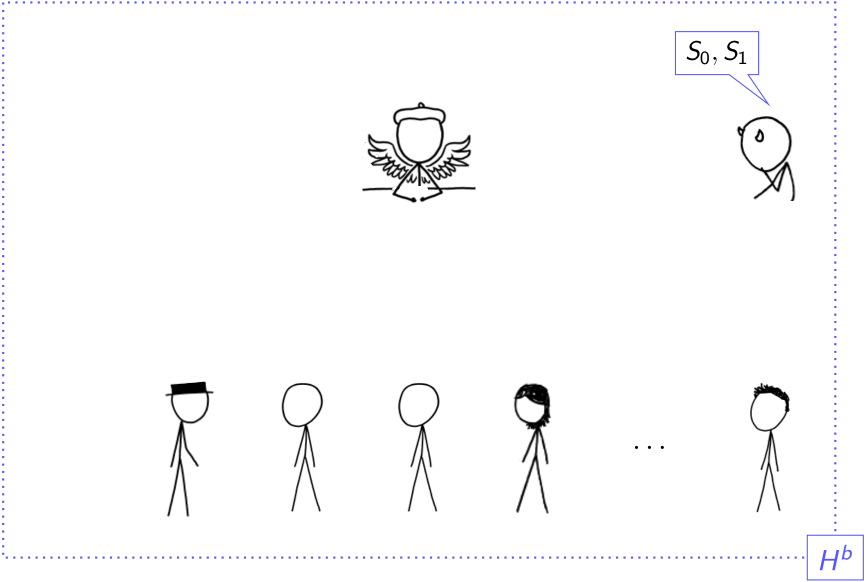


- Completeness: qualified $\mathcal{X}$ can reconstruct
- Security: unqualified $\mathcal{X}$ learns *nothing* about S

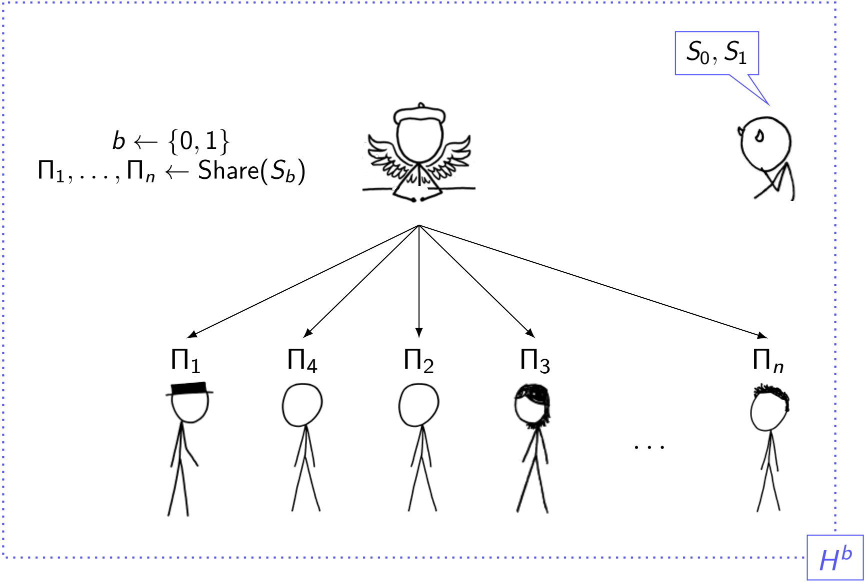
Security: Selective Game



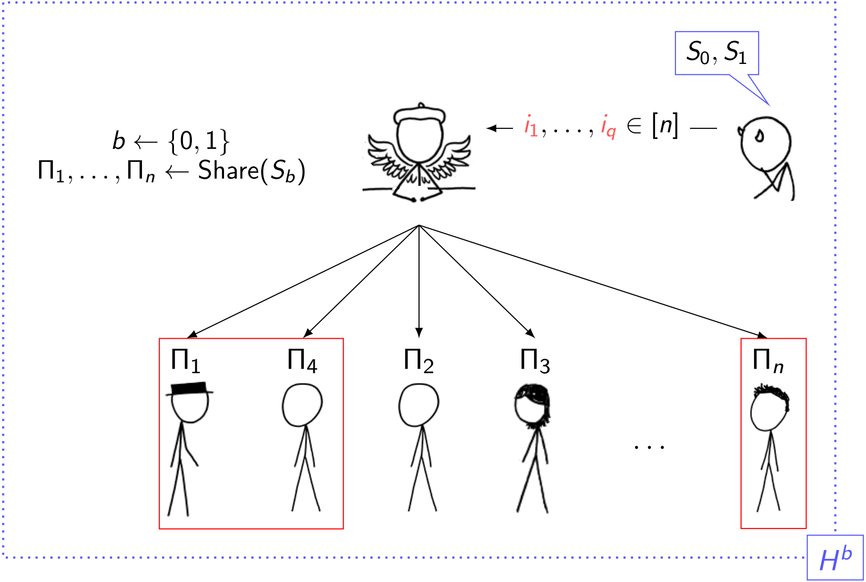
Security: Selective Game



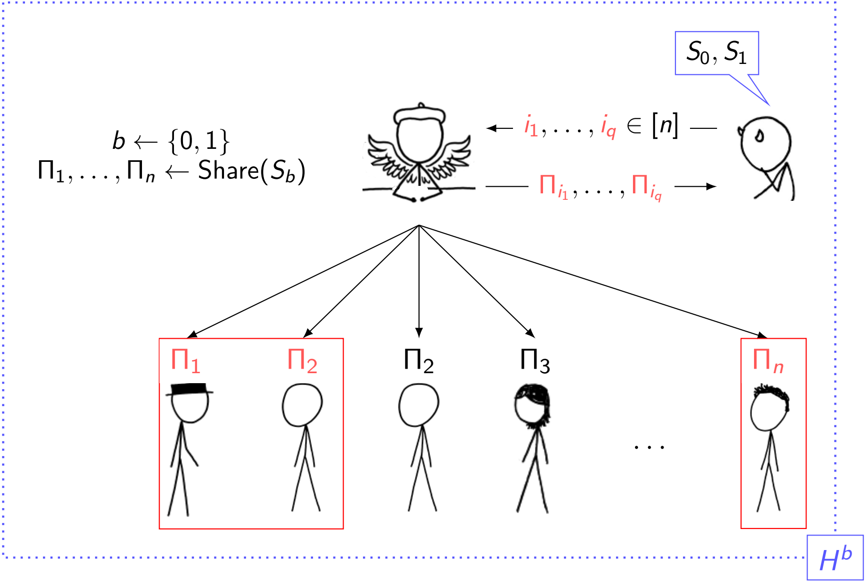
Security: Selective Game



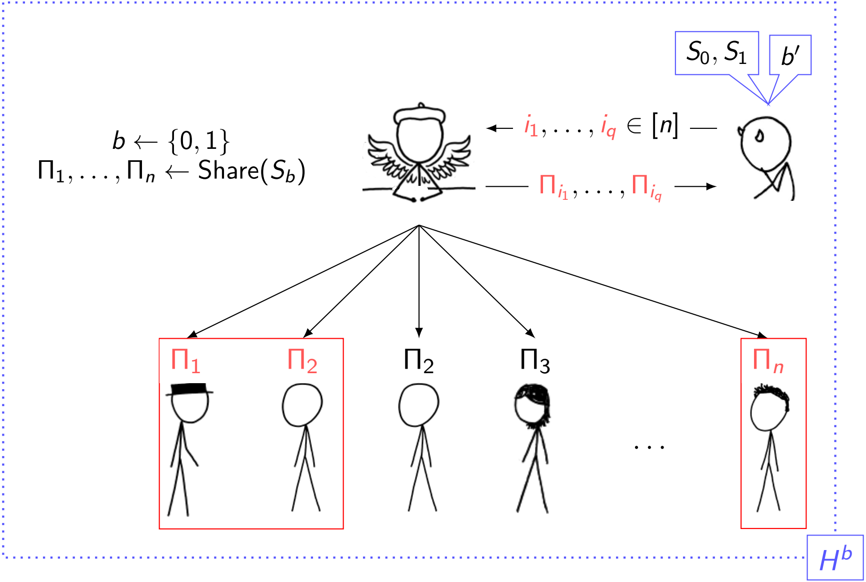
Security: Selective Game



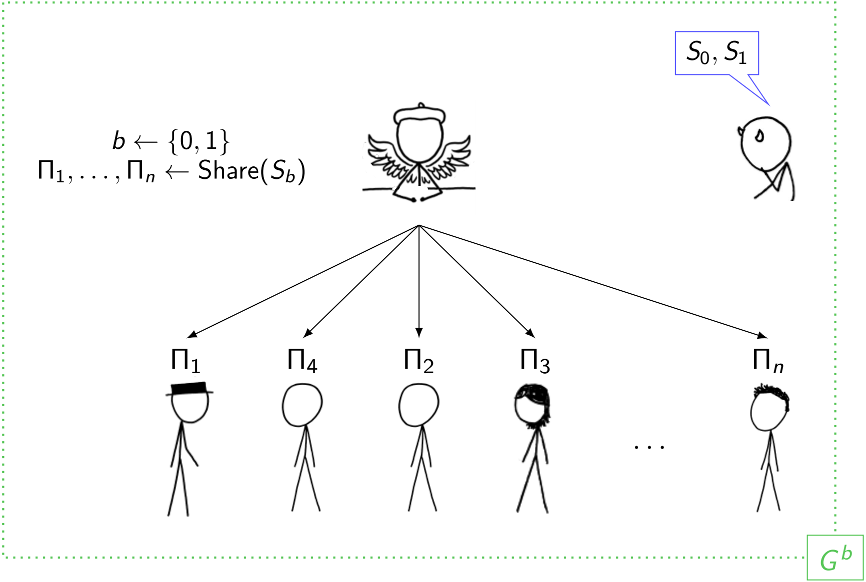
Security: Selective Game



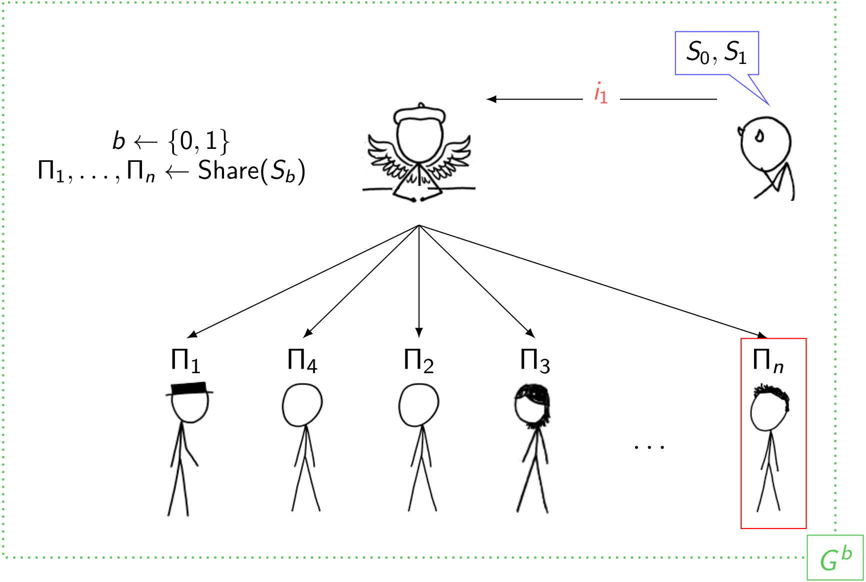
Security: Selective Game



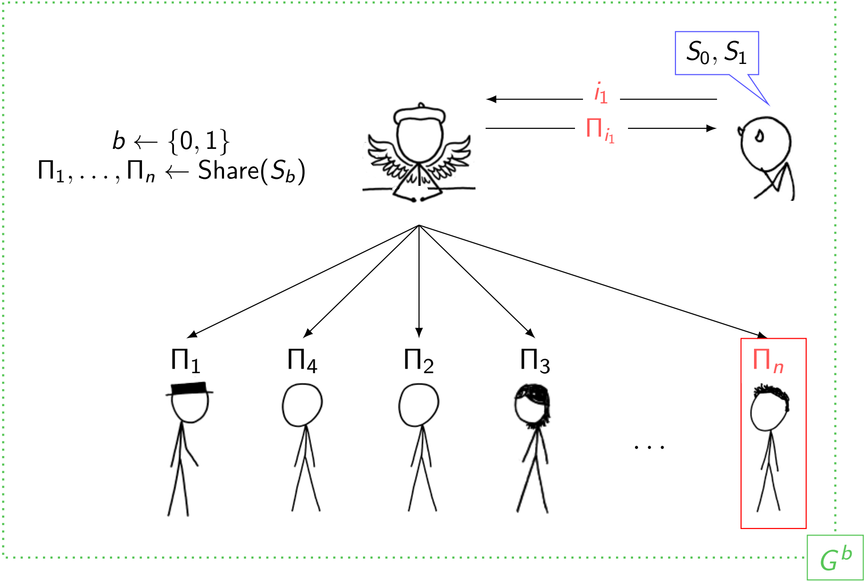
Security: Adaptive Game



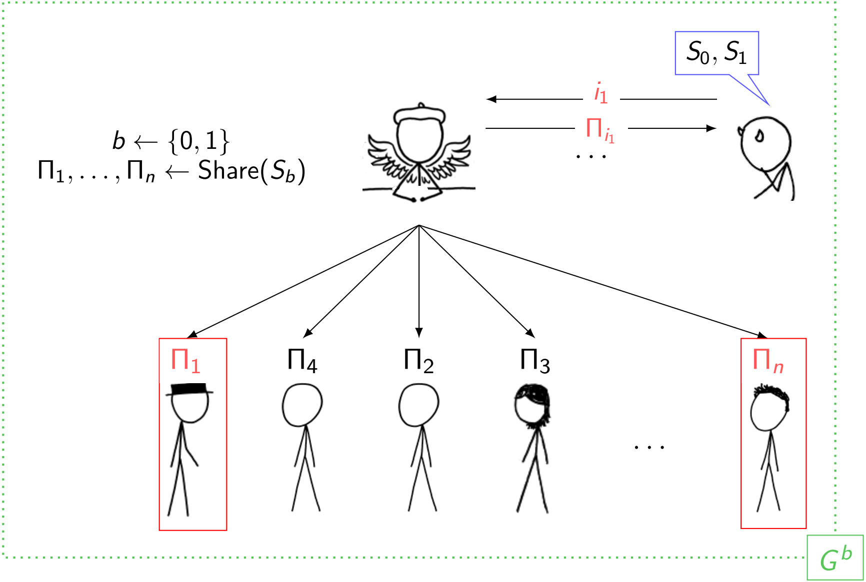
Security: Adaptive Game



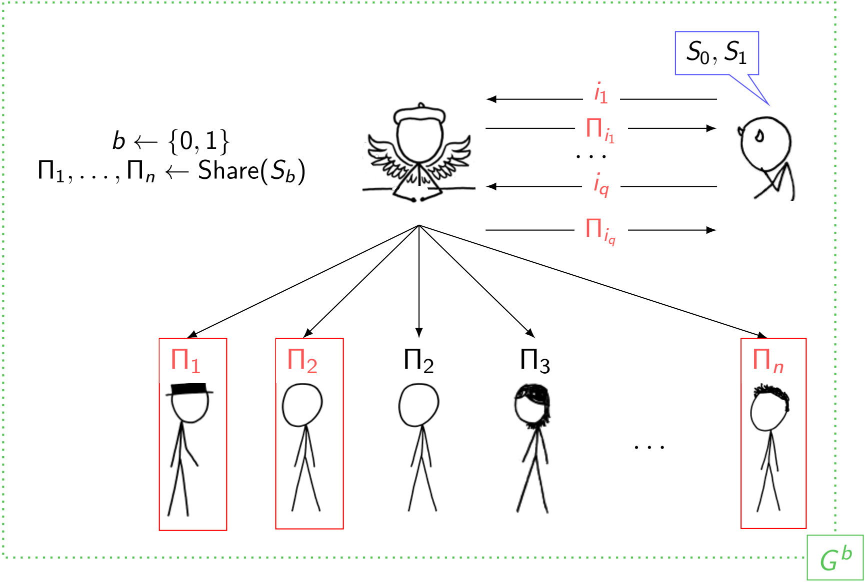
Security: Adaptive Game



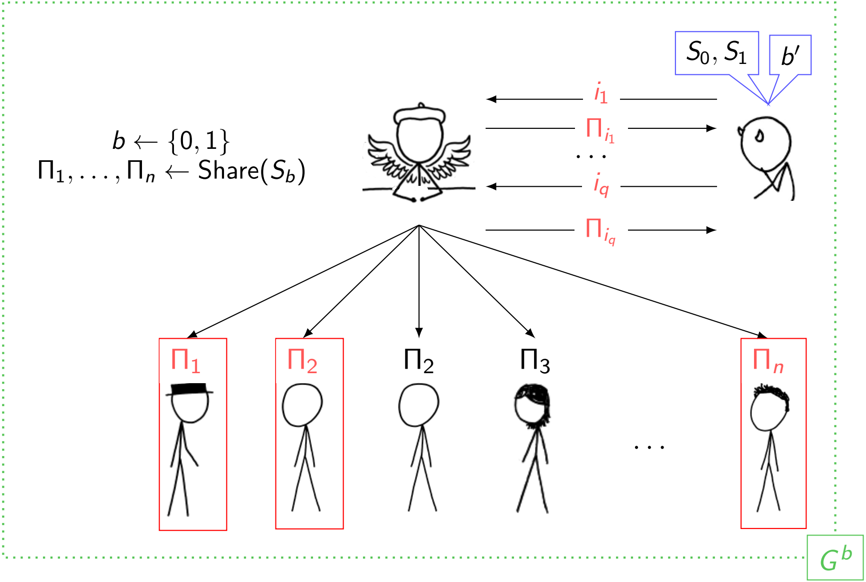
Security: Adaptive Game



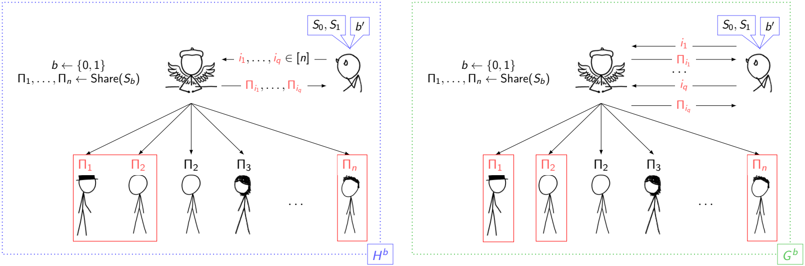
Security: Adaptive Game



Security: Adaptive Game

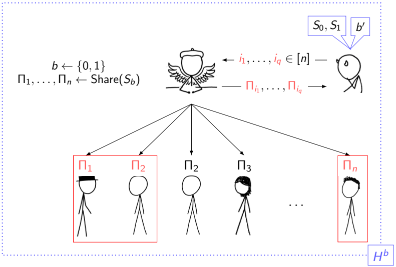


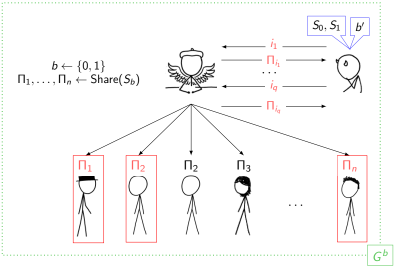
Security...



- Adversary wins if $b' = b$
- The secret sharing scheme is ϵ-(selectively/adaptively)-secure if $P[b' = b] < 1/2 + \epsilon$

Security...





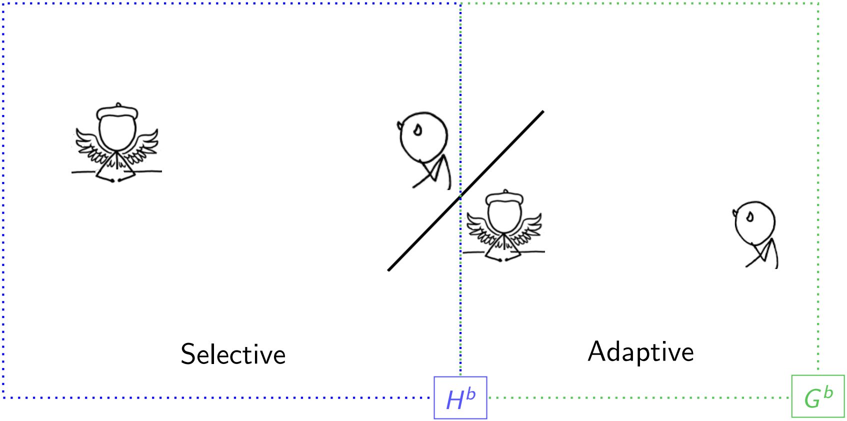
- Adversary wins if $b' = b$
- The secret sharing scheme is ϵ-(selectively/adaptively)-secure if $P[b' = b] < 1/2 + \epsilon$
- Adversary: computational or unbounded
  - Computationally-secure: ϵ is negligible for all adversaries
  - Negligible function: grows slower than any inverse polynomial
  - Equivalently: G^0 and G^1/H^0 and H^1 are indistinguishable ($\leftrightarrow$)

Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}

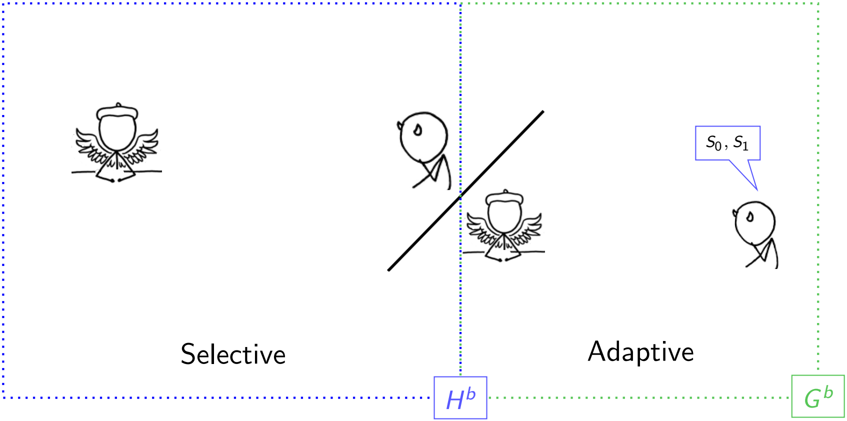
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



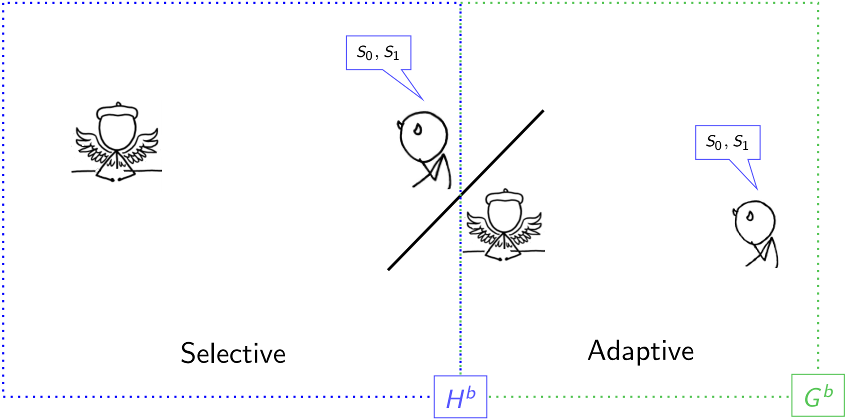
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



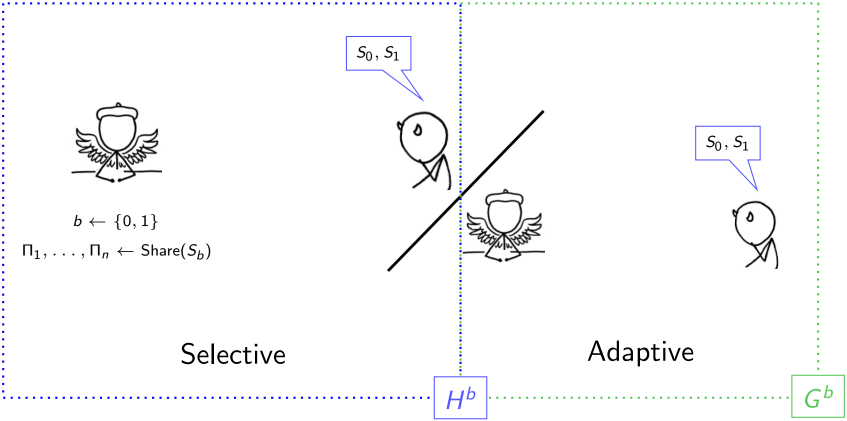
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



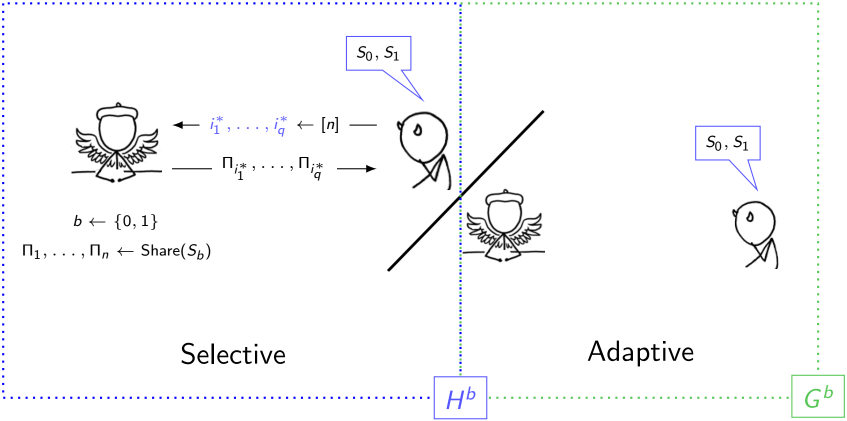
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



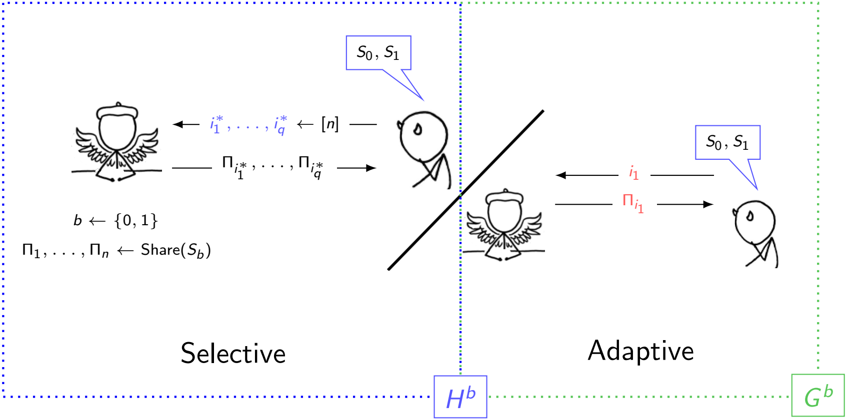
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



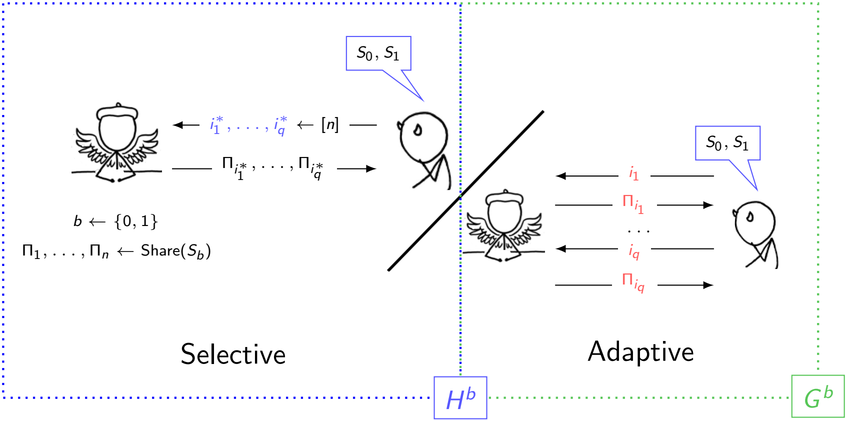
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



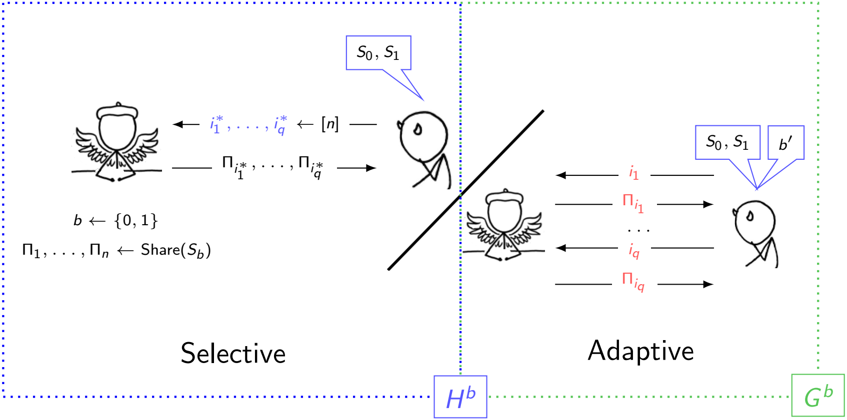
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



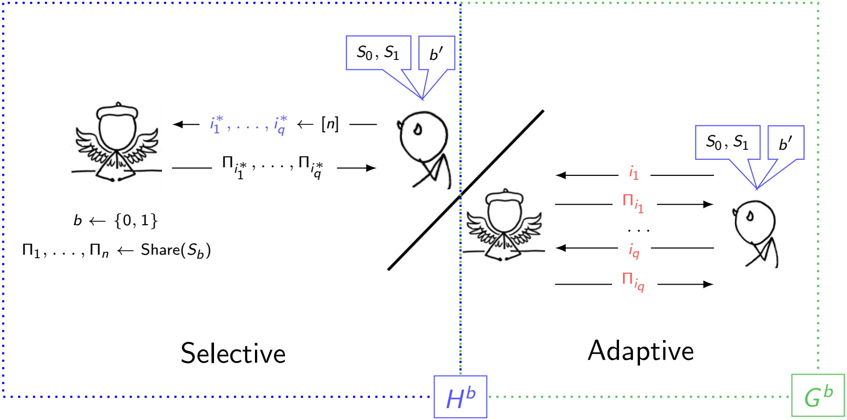
Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



Selective to Adaptive: Random Guessing

- Lemma 1: ϵ-selective security $\implies$ $\epsilon \cdot 2^n$-adaptive security:
  - Guess the participants that the *adaptive* adversary corrupts
  - Abort if guess wrong at any point during the game
  - Pr. that the guess correct is 2^{-n}



What is Known?

- Against *unbounded* adversaries:
  - Threshold [S] } selective
  - Monotone formulas [BL] } selective

  - Selective security $\implies$ adaptive security

What is Known?

- Against *unbounded* adversaries:
  - Threshold [S]
  - Monotone formulas [BL]

  } selective

  - Selective security $\implies$ adaptive security
- Against computational adversaries:
  - Monotone circuits assuming symmetric encryption [Y]
  - Every monotone access structure assuming "witness encryption" for NP [HNY]

  } selective

What is Known?

- Against *unbounded* adversaries:
  - Threshold [S]
  - Monotone formulas [BL]

  } selective

  - Selective security $\implies$ adaptive security
- Against computational adversaries:
  - Monotone circuits assuming symmetric encryption [Y]
  - Every monotone access structure assuming "witness encryption" for NP [HNY]

  } selective

  - Adaptive security *harder* to achieve:
    - Only known through random guessing

What We Show

- The exponential loss *can be* avoided for Yao's scheme

What We Show

- The exponential loss *can be* avoided for Yao's scheme

- Theorem 1: If the encryption is ϵ-secure, then for any access structure described by a Boolean circuit of size s, depth d and fan-in/fan-out δ, Yao's scheme is $\approx \epsilon \cdot (2\delta)^d \cdot s^{\delta \cdot d}$ adaptively-secure

What We Show

- The exponential loss *can be* avoided for Yao's scheme
- Theorem 1: If the encryption is ϵ-secure, then for any access structure described by a Boolean circuit of size s, depth d and fan-in/fan-out δ, Yao's scheme is $\approx \epsilon \cdot (2\delta)^d \cdot s^{\delta \cdot d}$ adaptively-secure
- Corollary 1: For *log-depth* circuits of *constant* fan-in/fan-out, quasi-polynomially-secure symmetric encryption implies adaptively-secure secret sharing

Yao's Secret Sharing

Yao's Scheme

- Symmetric encryption scheme (E, D)
  - Encrypt $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and decrypt $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

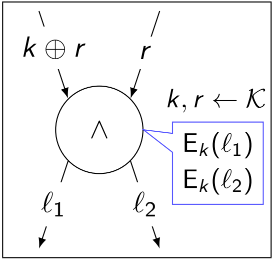
Yao's Scheme

- Symmetric encryption scheme (E, D)
  - Encrypt $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and decrypt $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

- Share: A gate associated with a key; a wire with a label
  - Label o/p wire with the secret S
  - Labels of the i/p wires given as shares to the resp. participants
  - Label other wires recursively from root

Yao's Scheme

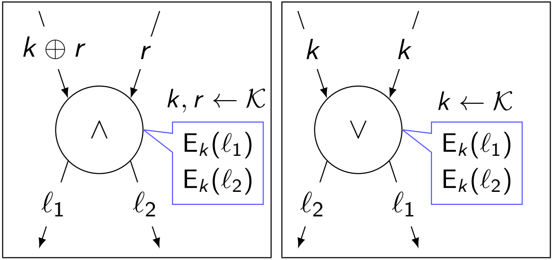
- Symmetric encryption scheme (E, D)
  - Encrypt $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and decrypt $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$
- Share: A gate associated with a key; a wire with a label
  - Label o/p wire with the secret S
  - Labels of the i/p wires given as shares to the resp. participants
  - Label other wires recursively from root



Yao's Scheme

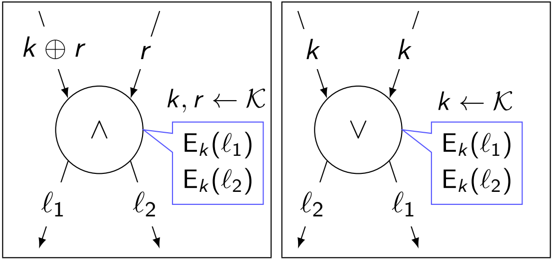
- Symmetric encryption scheme (E, D)
  - Encrypt $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and decrypt $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

- Share: A gate associated with a key; a wire with a label
  - Label o/p wire with the secret S
  - Labels of the i/p wires given as shares to the resp. participants
  - Label other wires recursively from root



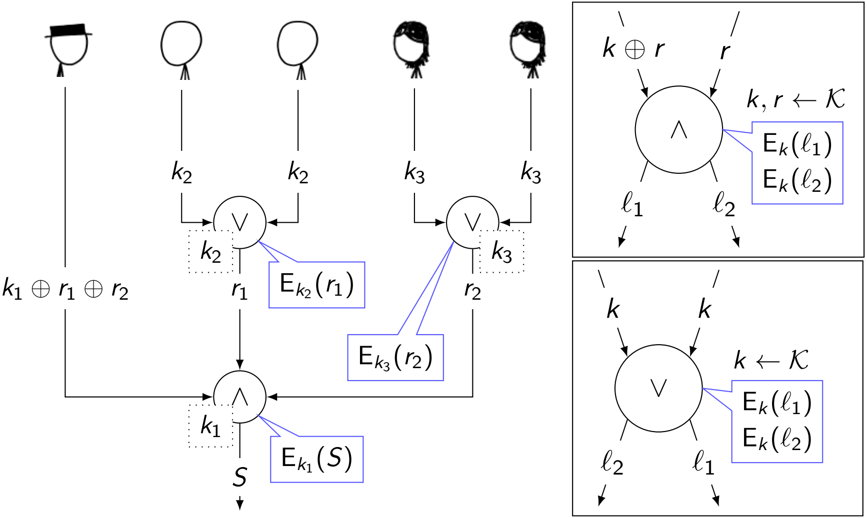
Yao's Scheme

- Symmetric encryption scheme (E, D)
  - Encrypt $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ and decrypt $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$
- Share: A gate associated with a key; a wire with a label
  - Label o/p wire with the secret S
  - Labels of the i/p wires given as shares to the resp. participants
  - Label other wires recursively from root



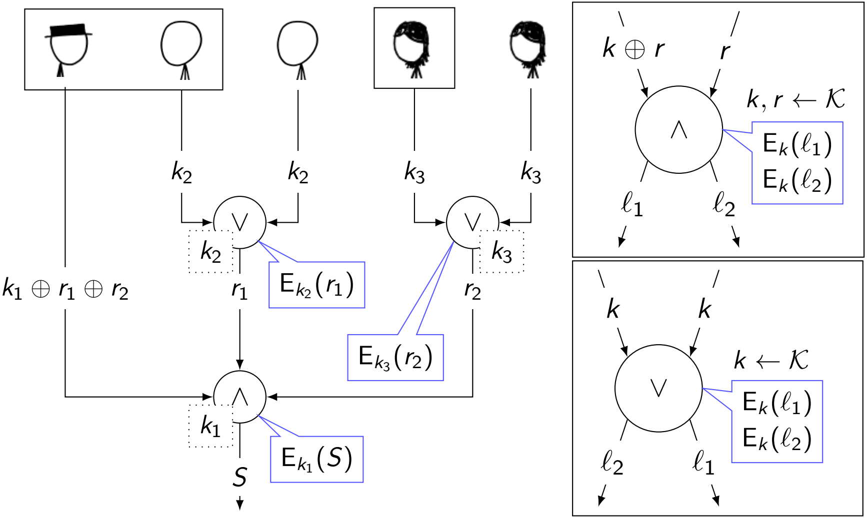
- Reconstruct does the reverse of Share

Yao's Scheme...

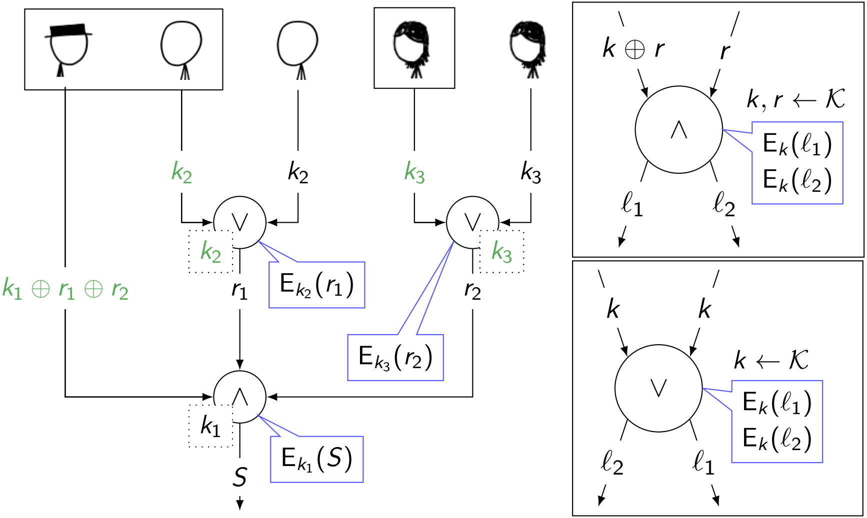


k_2
k_2
k_3
k_3
k_2
k_3
E_{k_2}(r_1)
k_1 ⊕ r_1 ⊕ r_2
r_1
r_2
E_{k_3}(r_2)
k_1
E_{k_1}(S)
S
k ⊕ r
r
k, r ← 𝒦
E_k(ℓ_1)
E_k(ℓ_2)
ℓ_1
ℓ_2
k
k
k ← 𝒦
E_k(ℓ_1)
E_k(ℓ_2)
ℓ_2
ℓ_1

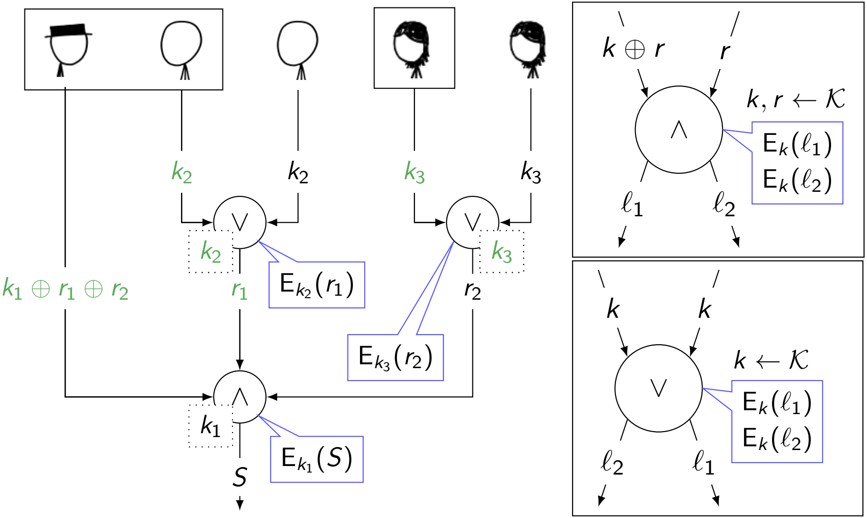
Yao's Scheme...


k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...

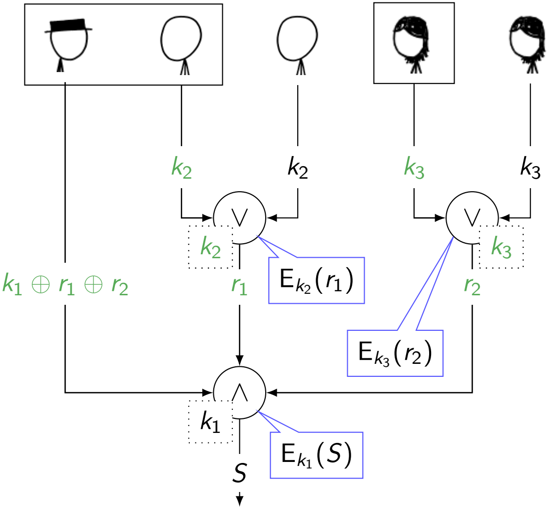

$k_1 \oplus r_1 \oplus r_2$
k_2
k_2
k_2
k_3
k_3
k_3
$E_{k_2}(r_1)$
r_1
r_2
$E_{k_3}(r_2)$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...

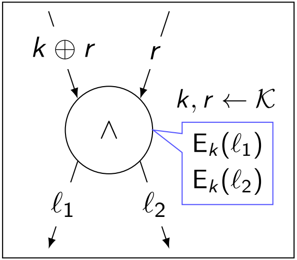


k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

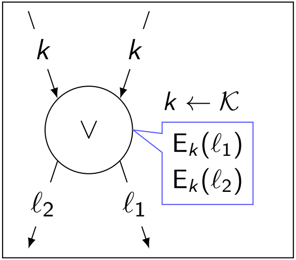
Yao's Scheme...



k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S

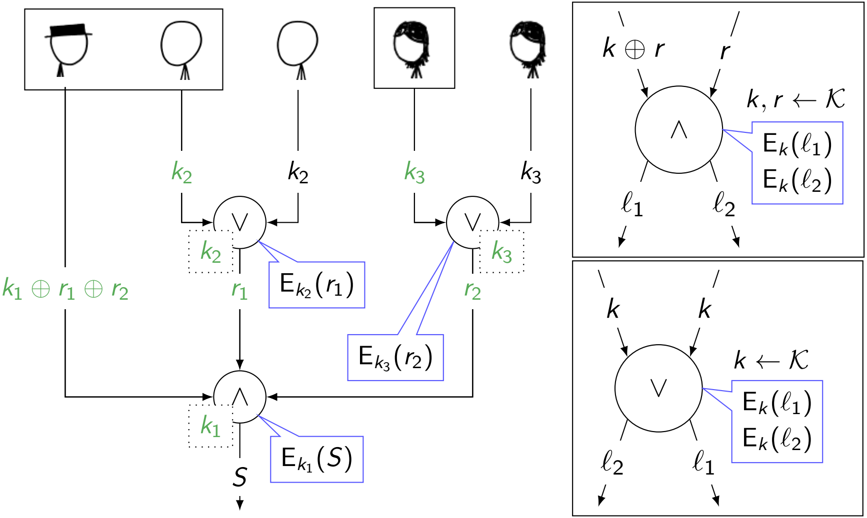


$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2



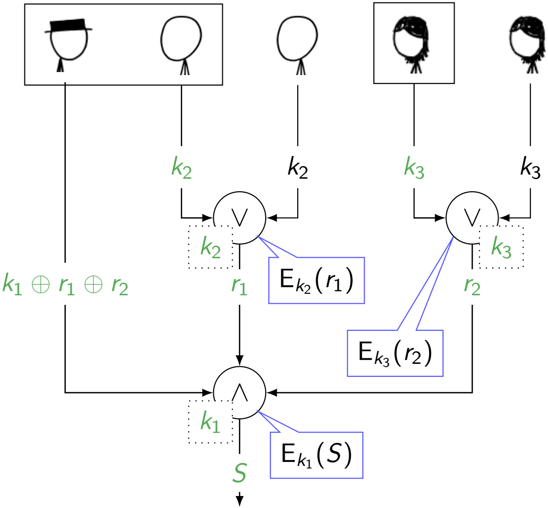
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...

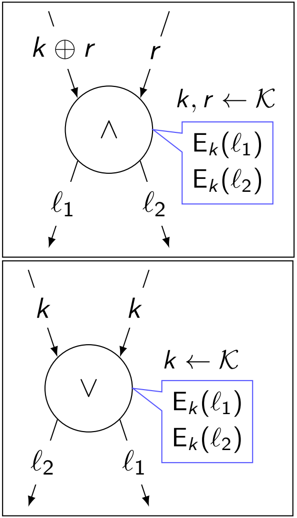


k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$E_{k_2}(r_1)$
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...

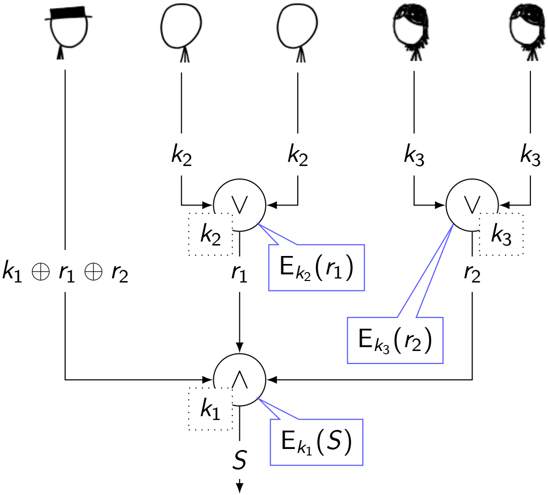


k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S

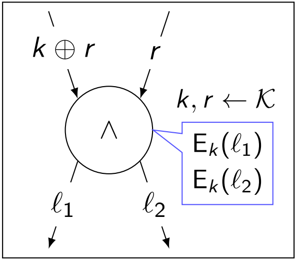


$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

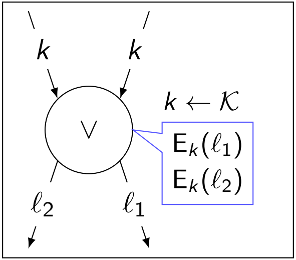
Yao's Scheme...



k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S

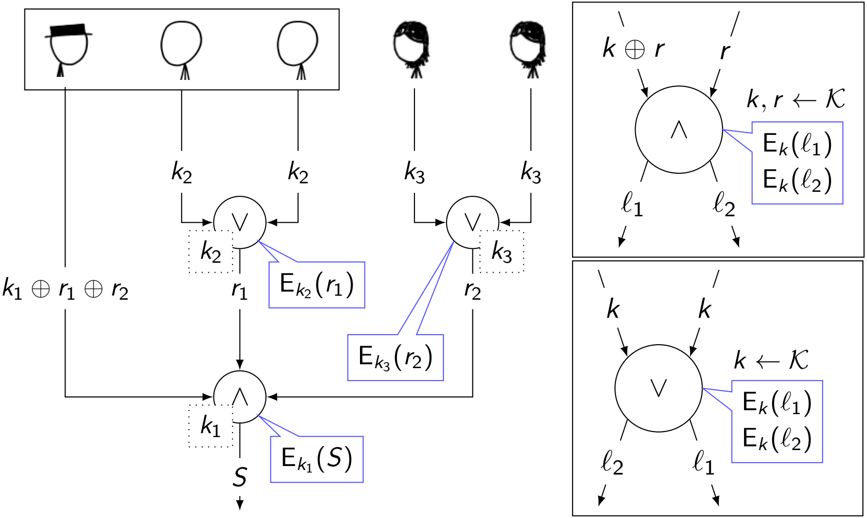


$k \oplus r$
r
$\wedge$
$k, r \leftarrow \mathcal{K}$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2

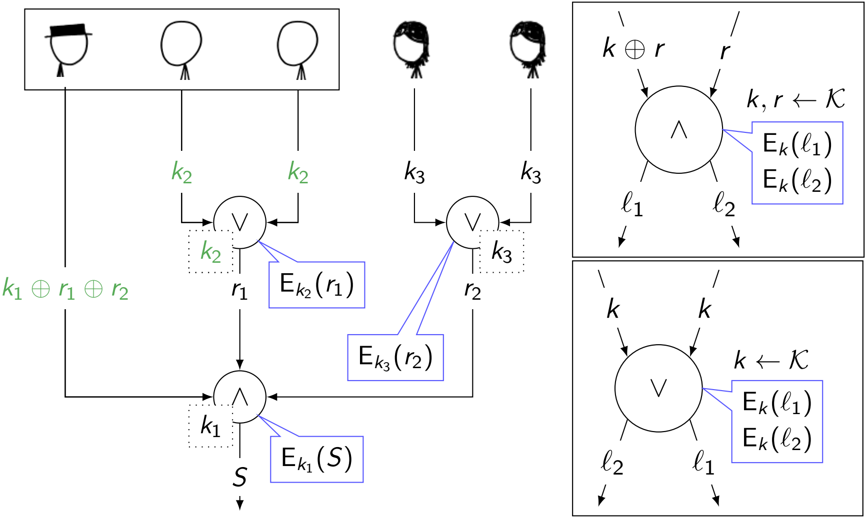


k
k
$\vee$
$k \leftarrow \mathcal{K}$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...

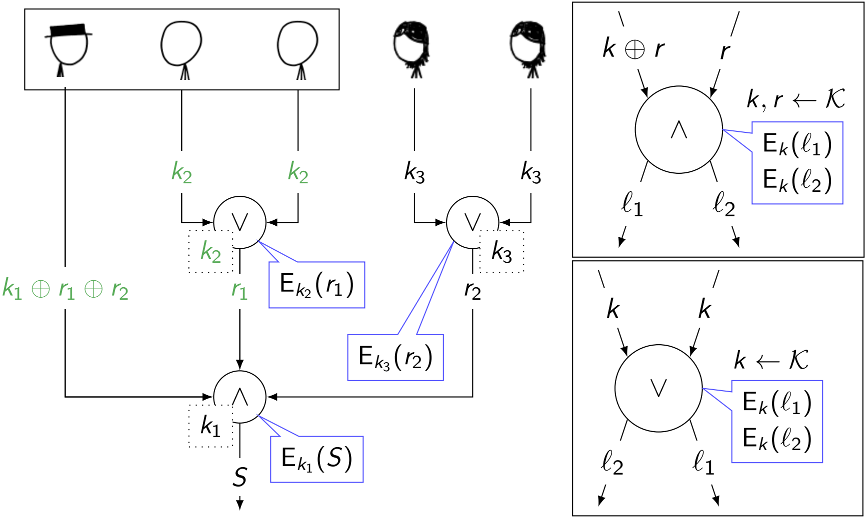

k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
$\mathsf{E}_{k_2}(r_1)$
r_2
$\mathsf{E}_{k_3}(r_2)$
$\wedge$
k_1
$\mathsf{E}_{k_1}(S)$
S
$k \oplus r$
r
$\wedge$
$k, r \leftarrow \mathcal{K}$
$\mathsf{E}_k(\ell_1)$
$\mathsf{E}_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$\vee$
$k \leftarrow \mathcal{K}$
$\mathsf{E}_k(\ell_1)$
$\mathsf{E}_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...



k_2
k_2
k_3
k_3
$\vee$
k_2
$E_{k_2}(r_1)$
$\vee$
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme...



k_2
k_2
k_3
k_3
$\vee$
$\vee$
k_2
k_3
$k_1 \oplus r_1 \oplus r_2$
r_1
r_2
$E_{k_2}(r_1)$
$E_{k_3}(r_2)$
$\wedge$
k_1
$E_{k_1}(S)$
S
$k \oplus r$
r
$k, r \leftarrow \mathcal{K}$
$\wedge$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_1
ℓ_2
k
k
$k \leftarrow \mathcal{K}$
$\vee$
$E_k(\ell_1)$
$E_k(\ell_2)$
ℓ_2
ℓ_1

Yao's Scheme: Selective Security

- Reduce to security of encryption

Yao's Scheme: Selective Security

- Reduce to security of encryption

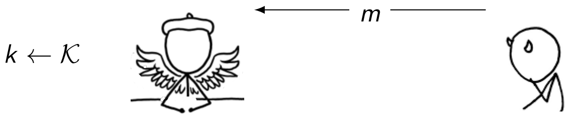
Yao's Scheme: Selective Security

- Reduce to security of encryption

$k \leftarrow \mathcal{K}$

Yao's Scheme: Selective Security

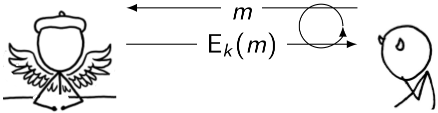
- Reduce to security of encryption



Yao's Scheme: Selective Security

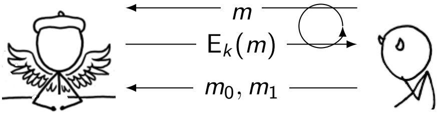
- Reduce to security of encryption

$k \leftarrow \mathcal{K}$



- Reduce to security of encryption

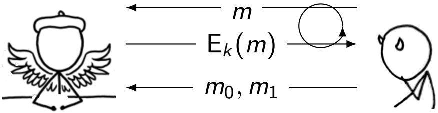
$k \leftarrow \mathcal{K}$



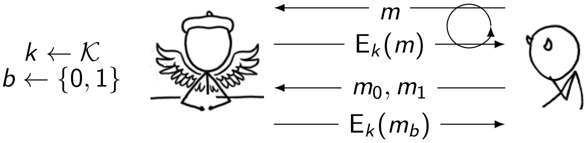
Yao's Scheme: Selective Security

- Reduce to security of encryption

$k \leftarrow \mathcal{K}$
$b \leftarrow \{0, 1\}$

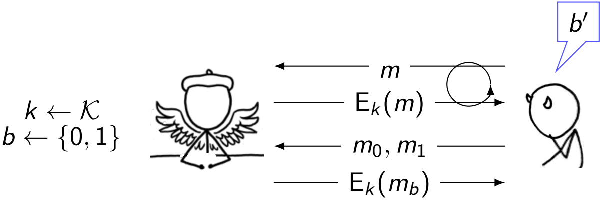


- Reduce to security of encryption



Yao's Scheme: Selective Security

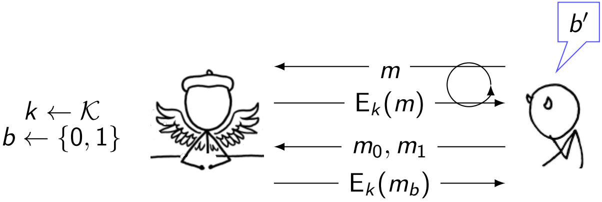
- Reduce to security of encryption



- Encryption scheme is ϵ-secure if no PPT adversary can win with pr. greater than $1/2 + \epsilon$: $E_k(m_0) \leftrightarrow E_k(m_1)$

Yao's Scheme: Selective Security

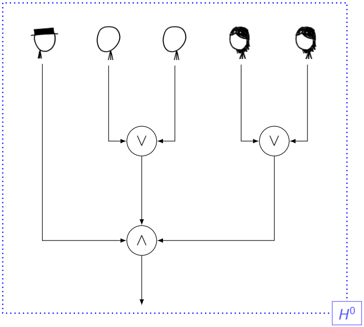
- Reduce to security of encryption



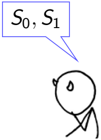
- Encryption scheme is ϵ-secure if no PPT adversary can win with pr. greater than $1/2 + \epsilon$: $E_k(m_0) \leftrightarrow E_k(m_1)$

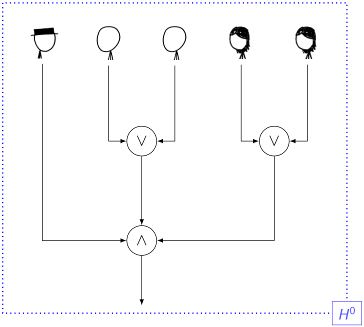
- Theorem 2 [VNS+]: If the encryption is ϵ-secure then for any access structure described by a Boolean circuit of size s the scheme is $\approx \epsilon \cdot s$-*selectively*-secure

Yao's Scheme: Selective Security...



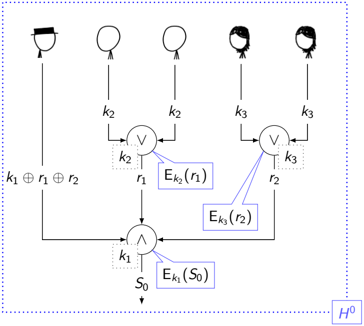
Yao's Scheme: Selective Security...





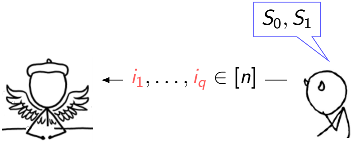
Yao's Scheme: Selective Security...

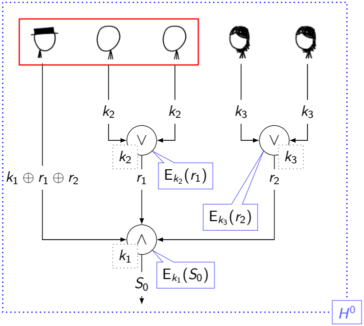
$b \leftarrow \{0,1\}$
$\Pi_1, \ldots, \Pi_n \leftarrow \mathsf{Share}(S_b)$



Yao's Scheme: Selective Security...

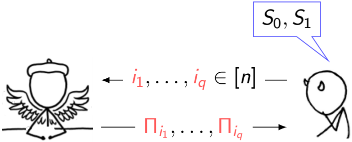
$$b \leftarrow \{0,1\}$$
$$\Pi_1, \dots, \Pi_n \leftarrow \mathsf{Share}(S_b)$$

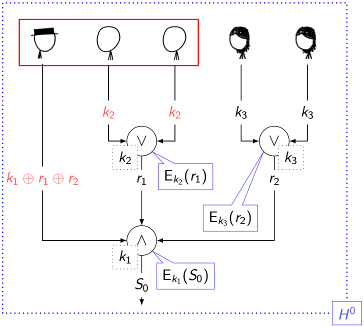




Yao's Scheme: Selective Security...

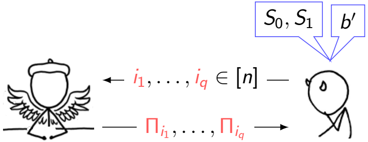
$$b \leftarrow \{0,1\}$$
$$\Pi_1, \ldots, \Pi_n \leftarrow \mathsf{Share}(S_b)$$

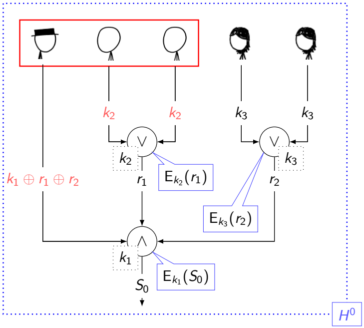




Yao's Scheme: Selective Security...

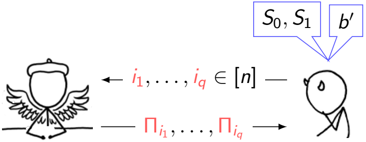
$$b \leftarrow \{0,1\}$$
$$\Pi_1, \dots, \Pi_n \leftarrow \mathsf{Share}(S_b)$$

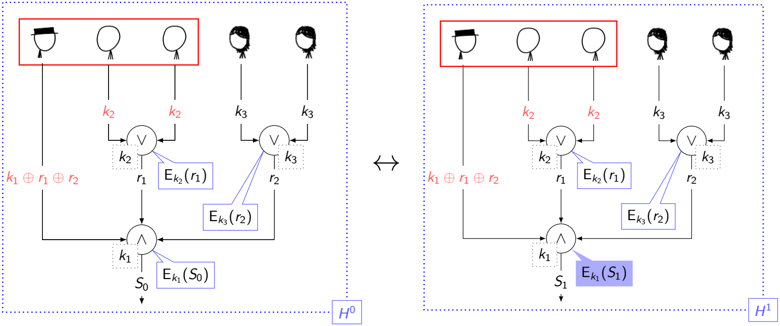




Yao's Scheme: Selective Security...

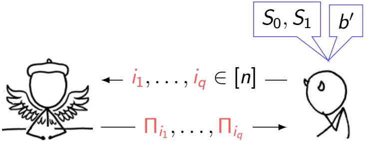
$$b \leftarrow \{0,1\}$$
$$\Pi_1, \ldots, \Pi_n \leftarrow \mathsf{Share}(S_b)$$

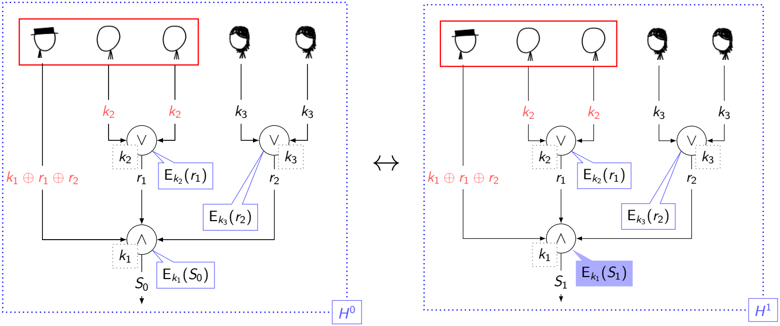




Yao's Scheme: Selective Security...

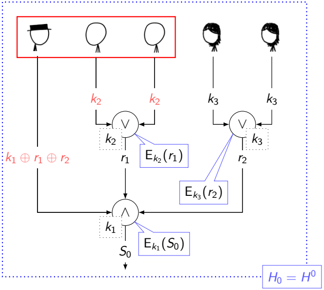
$$b \leftarrow \{0,1\}$$
$$\Pi_1, \ldots, \Pi_n \leftarrow \mathsf{Share}(S_b)$$

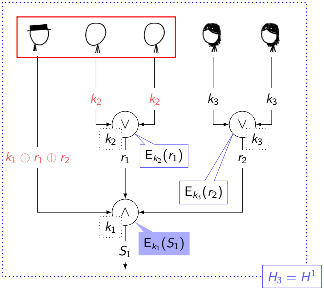




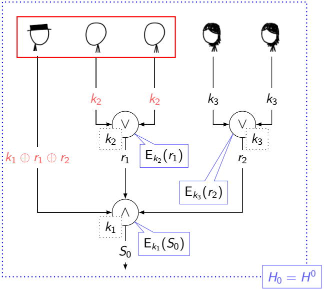
- Aim: Show that secure encryption $\implies$ $H^0 \leftrightarrow H^1$
  - Contapositive: $H^0 \not\leftrightarrow H^1$ $\implies$ encryption not secure

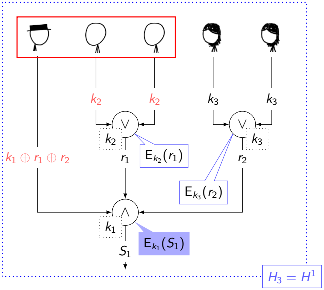
Yao's Scheme: Selective Security...

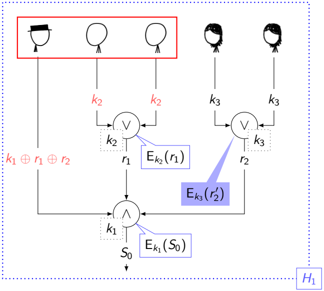




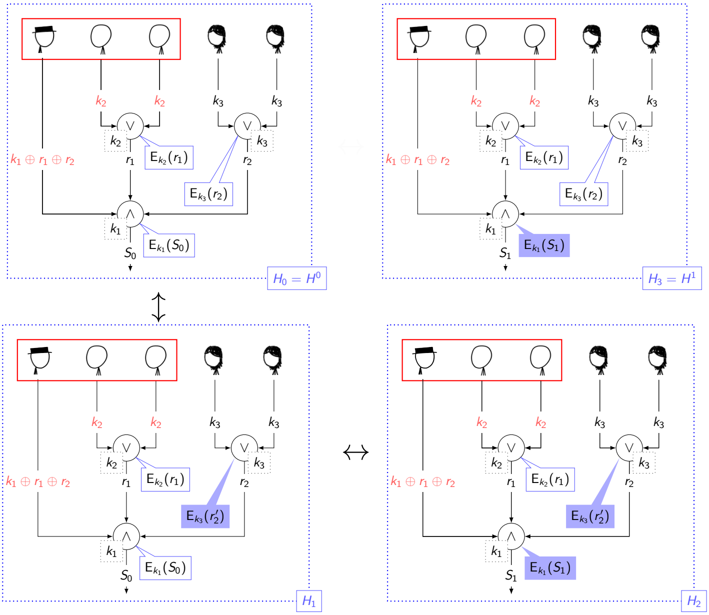
Yao's Scheme: Selective Security...



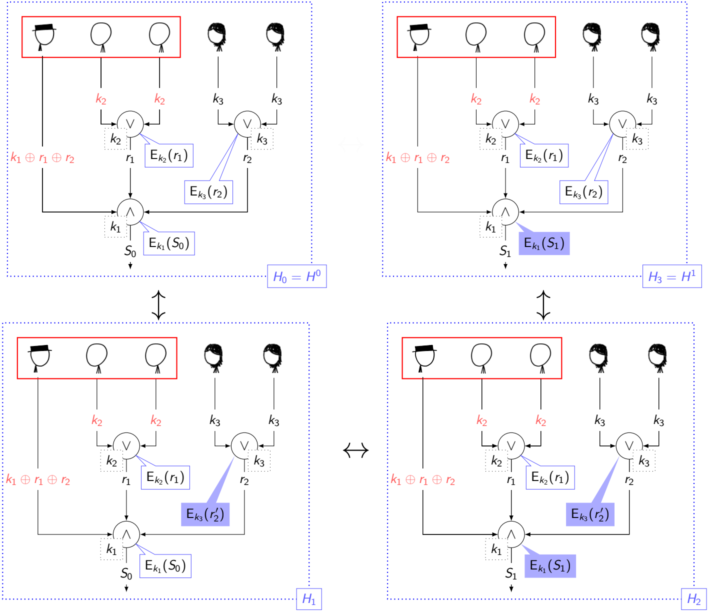




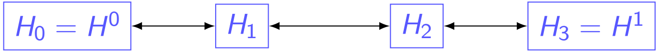
Yao's Scheme: Selective Security...



Yao's Scheme: Selective Security...



Yao's Scheme: Selective Security...



- Replace ciphertexts that the corrupt participants cannot decrypt with a bogus one
- Results in a sequence of hybrid games: the extreme games coincide with the original security game
- Show that consecutive hybrids are ϵ-indistinguishable assuming encryption is ϵ-secure: $H_i \leftrightarrow H_{i+1}$

Hybrids and Pebbling

Hybrids can be modelled using a pebbling game on the circuit

- Pebble $\implies$ bogus ciphertext/no pebble $\implies$ real ciphertext

Hybrids and Pebbling

Hybrids can be modelled using a pebbling game on the circuit

- Pebble $\Longrightarrow$ bogus ciphertext/no pebble $\Longrightarrow$ real ciphertext
- Pebbling rules: can place/remove a pebble on a gate if

Hybrids and Pebbling

Hybrids can be modelled using a pebbling game on the circuit

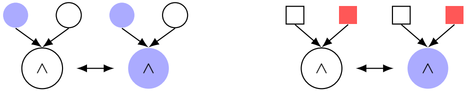
- Pebble $\Longrightarrow$ bogus ciphertext/no pebble $\Longrightarrow$ real ciphertext
- Pebbling rules: can place/remove a pebble on a gate if
  1. gate=$\vee$: i) all parent gates are pebbled and ii) all input nodes are not corrupted

Hybrids and Pebbling

Hybrids can be modelled using a pebbling game on the circuit

- Pebble $\Longrightarrow$ bogus ciphertext/no pebble $\Longrightarrow$ real ciphertext
- Pebbling rules: can place/remove a pebble on a gate if
  1. gate=$\vee$: i) all parent gates are pebbled and ii) all input nodes are not corrupted

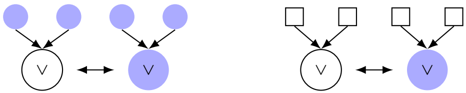
  2. gate=$\wedge$: i) one of the parents is pebbled; or ii) one of the input nodes is not corrupted



Hybrids and Pebbling

Hybrids can be modelled using a pebbling game on the circuit

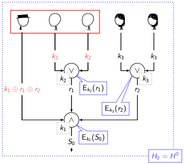
- Pebble $\Longrightarrow$ bogus ciphertext/no pebble $\Longrightarrow$ real ciphertext
- Pebbling rules: can place/remove a pebble on a gate if
  1. gate=$\vee$: i) all parent gates are pebbled and ii) all input nodes are not corrupted

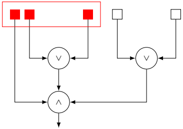


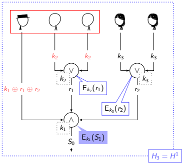
  2. gate=$\wedge$: i) one of the parents is pebbled; or ii) one of the input nodes is not corrupted

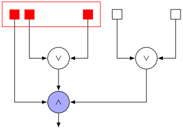
- Goal: Pebble the sink gate starting from an unpebbled state
- Pebbling sequence: $P_0, \ldots, P_\ell$, $P_i \subseteq [s]$

Hybrids and Pebbling...

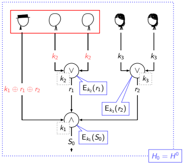


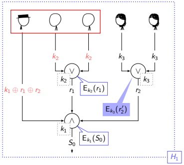


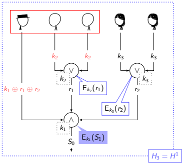


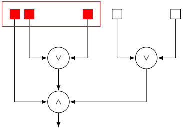


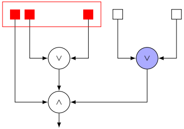
Hybrids and Pebbling...

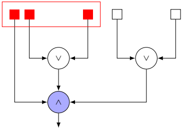




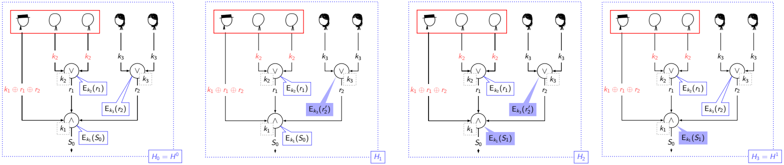


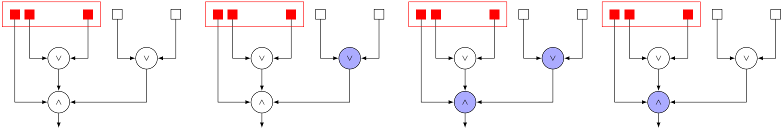




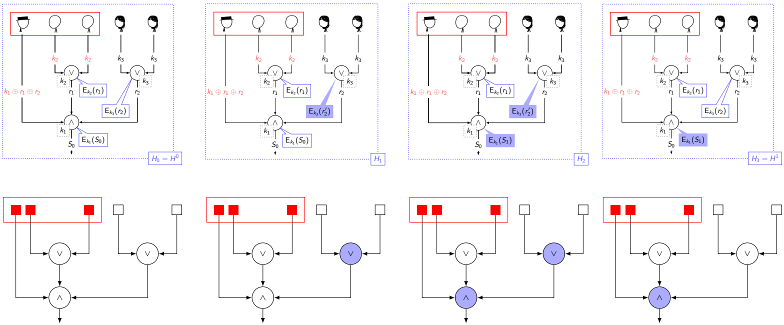


Hybrids and Pebbling...



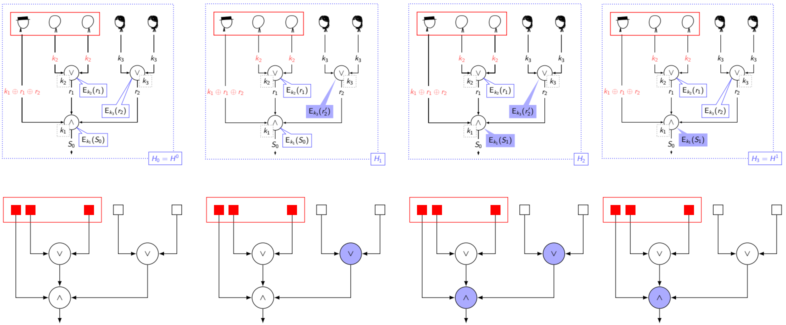


Hybrids and Pebbling...



- Any valid pebbling sequence implies a sequence of hybrids!
  - $P_0, \ldots, P_\ell \Leftrightarrow H^0 = H_0, \ldots, H_\ell = H^1$
  - Can play a hybrid game if the pebbled gates in the corresponding configuration are known
  - Neighbouring hybrids are indistinguishable

Hybrids and Pebbling...



- Any valid pebbling sequence implies a sequence of hybrids!
  - $P_0, \ldots, P_\ell \Leftrightarrow H^0 = H_0, \ldots, H_\ell = H^1$
  - Can play a hybrid game if the pebbled gates in the corresponding configuration are known
  - Neighbouring hybrids are indistinguishable
- Corollary: if encryption scheme is ϵ-secure then Yao's scheme is $\epsilon \cdot \ell$-selectively-secure

Back to Selective Security

- Theorem 2 [VNS+]: If the encryption is ϵ-secure then for any access structure described by a Boolean circuit of size s the scheme is $\approx \epsilon \cdot s$-*selectively*-secure

Back to Selective Security

- Theorem 2 [VNS+]: If the encryption is ϵ-secure then for any access structure described by a Boolean circuit of size s the scheme is $\approx \epsilon \cdot s$-*selectively*-secure

- Follows from the following pebbling strategy:
  1. Pebble level-by-level starting from the input level until o/p gate pebbled (never removing a pebble)
  2. Remove pebbles level-by-level in the reverse order
- #moves$\approx 2s$, #pebbles$= s$

Back to Selective Security

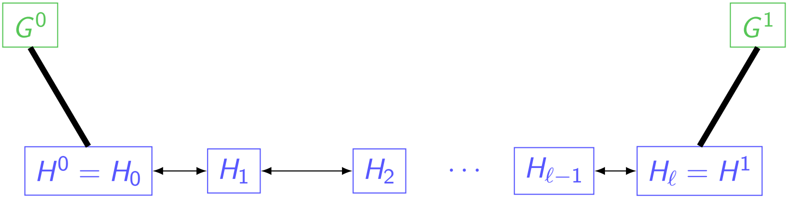
- Theorem 2 [VNS+]: If the encryption is ϵ-secure then for any access structure described by a Boolean circuit of size s the scheme is $\approx \epsilon \cdot s$-*selectively*-secure

- Follows from the following pebbling strategy:
  1. Pebble level-by-level starting from the input level until o/p gate pebbled (never removing a pebble)
  2. Remove pebbles level-by-level in the reverse order

- #moves$\approx 2s$, #pebbles$= s$

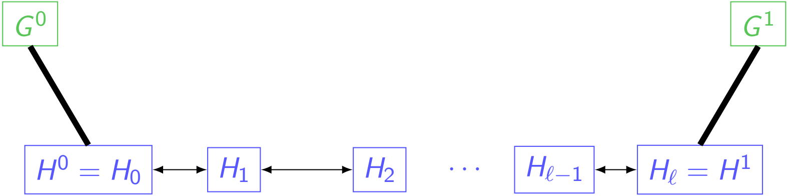
- Note: *must* know the corrupt participants

Recap



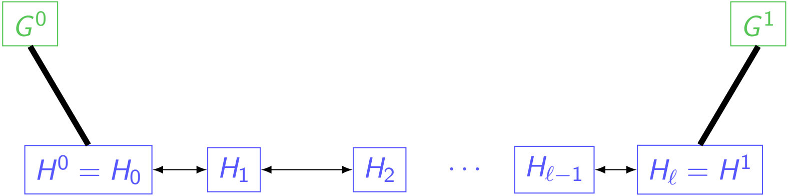
- Theorem 2 (\$): Yao's scheme is $\epsilon \cdot s$-selective-secure
- Lemma 1 (\$\$\$): ϵ-selective-secure $\implies$ $\epsilon \cdot 2^n$-adaptive-secure
- Corollary 2 (\$\$\$): Yao's scheme is $\epsilon \cdot s \cdot 2^n$-adaptive-secure

Adaptive Security: Avoiding Exponential Loss



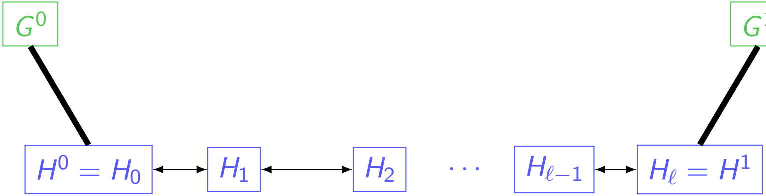
- Observation: *Can play a hybrid game if the pebbled gates in the corresponding configuration are known*

Adaptive Security: Avoiding Exponential Loss



- Observation: *Can play a hybrid game if the pebbled gates in the corresponding configuration are known*
- The level-by-level pebbling requires uses too many pebbles!

Adaptive Security: Avoiding Exponential Loss



- Observation: *Can play a hybrid game if the pebbled gates in the corresponding configuration are known*
- The level-by-level pebbling requires uses too many pebbles!
- Devise a new sequence of hybrids/pebbling sequence
  - A pebbling strategy with fewer pebbles requires less information (and hence less guessing)

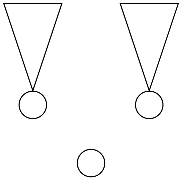
Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

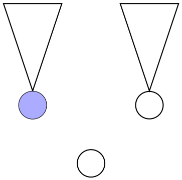


Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

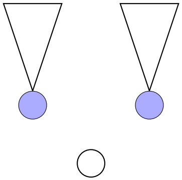
  1. Pebble left parent



Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:
  1. Pebble left parent
  2. Pebble right parent

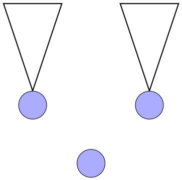


Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

  1. Pebble left parent
  2. Pebble right parent
  3. Pebble vertex

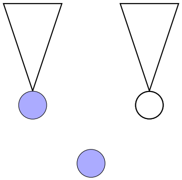


Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

  1. Pebble left parent
  2. Pebble right parent
  3. Pebble vertex
  4. Unpebble right parent

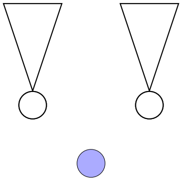


Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

  1. Pebble left parent
  2. Pebble right parent
  3. Pebble vertex
  4. Unpebble right parent
  5. Unpebble left parent



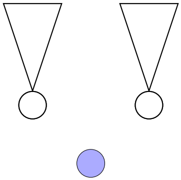
- $\#\text{moves}(d) = \#\text{moves}(d-1) \cdot 2\delta$
- $\#\text{pebbles}(d) = \#\text{pebbles}(d-1) + \delta$

Adaptive Security: Avoiding Exponential Loss...

Lemma 2: A DAG of degree δ and of depth d can be pebbled using $\delta \cdot d$ pebbles and $\approx (2\delta)^d$ moves

- To pebble a vertex, recursively:

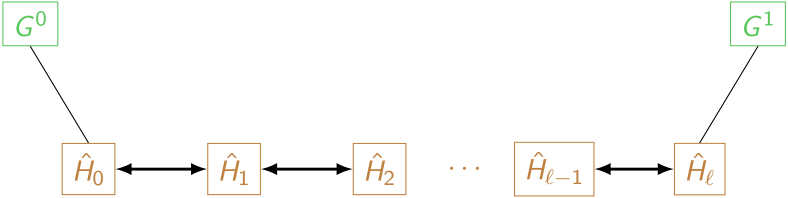
  1. Pebble left parent
  2. Pebble right parent
  3. Pebble vertex
  4. Unpebble right parent
  5. Unpebble left parent



- $\#\text{moves}(d) = \#\text{moves}(d-1) \cdot 2\delta$
- $\#\text{pebbles}(d) = \#\text{pebbles}(d-1) + \delta$

- Denoted by $\hat{P}_0, \ldots, \hat{P}_\ell$

Adaptive Security: Avoiding Exponential Loss...

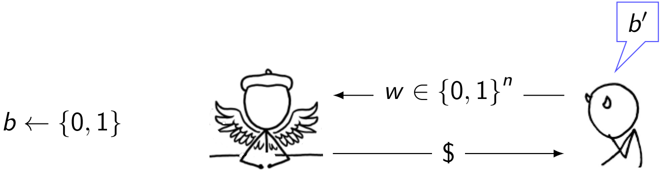


- $\hat{P}_0, \ldots, \hat{P}_\ell$ yields partially-selective hybrids $\hat{H}_0, \ldots, \hat{H}_\ell$
  - Adversary committed to a pebbling configuration instead of corrupt participants: apply random guessing
  - A pebbling configuration $\hat{P}_i$ has at most $\delta \cdot d$: probability of guessing is $2^{-(\delta \cdot d) \cdot \log s} = s^{-\delta \cdot d}$

- Theorem 1 ($$): If the encryption is ϵ-secure, then for any access structure described by a Boolean circuit of size s, depth d and fan-in/fan-out δ Yao's scheme is $\approx \epsilon \cdot (2\delta)^d \cdot s^{\delta \cdot d}$ adaptively-secure

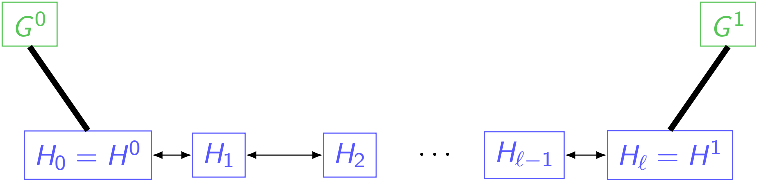
The Framework

In General



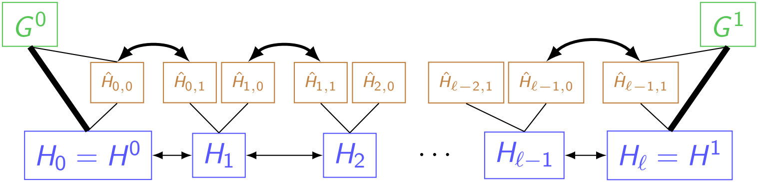
- Consider selective games where adversary commits to some information w
- Challenger checks if w consistent with observed w

In General...

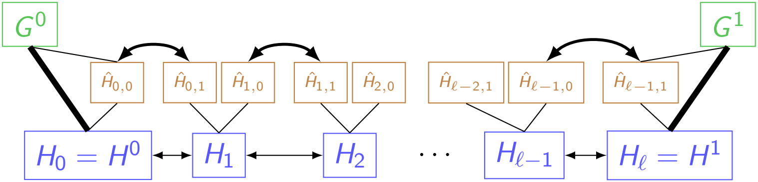


- Theorem 3 (main): If the sequence of selective hybrid games $H^0 = H_0, H_1, \cdots, H_\ell = H^1$ (with $H_i \leftrightarrow H_{i+1}$) satisfy the condition that $H_i \leftrightarrow H_{i+1}$ uses only $w_i = h_i(w) \in \{0,1\}^m$ then ϵ-selective security implies $\epsilon \cdot \ell \cdot 2^m$-adaptive security

In General...



- Theorem 3 (main): If the sequence of selective hybrid games $H^0 = H_0, H_1, \cdots, H_\ell = H^1$ (with $H_i \leftrightarrow H_{i+1}$) satisfy the condition that $H_i \leftrightarrow H_{i+1}$ uses only $w_i = h_i(w) \in \{0,1\}^m$ then ϵ-selective security implies $\epsilon \cdot \ell \cdot 2^m$-adaptive security



- Theorem 3 (main): If the sequence of selective hybrid games $H^0 = H_0, H_1, \cdots, H_\ell = H^1$ (with $H_i \leftrightarrow H_{i+1}$) satisfy the condition that $H_i \leftrightarrow H_{i+1}$ uses only $w_i = h_i(w) \in \{0,1\}^m$ then ϵ-selective security implies $\epsilon \cdot \ell \cdot 2^m$-adaptive security

- Results captured
  - Generalized selective decryption [P,FJP]
  - Constrained pseudo-random functions [FKPR]
  - Yao's garbled circuits [JW]

Open Questions

- Derive lower bounds from pebbling lower bounds
- Find more proofs that fit the framework

References

[BL] Benaloh and Lichter. *Generalized secret sharing and monotone functions*. Crypto'88

[FKPR] Fuchsbauer et al.. *Adaptive security of constrained PRFs*. Asiacrypt'14

[FJP] Fuchsbauer et al.. *A quasipolynomial reduction for generalized selective decryption on trees*. Crypto'15

[JKK+] Jafargohli et al.. *Be Adaptive, Avoid Overcommitting*. Crypto'17

[JW] Jafargholi and Wichs. Adaptive security of Yao's garbled circuits. TCC'16

[KNY] Komargodski et al.. *Secret-sharing for NP*. JoC'17

[P] Panjwani. *Tackling adaptive corruptions in multicast encryption protocols*. TCC'07

[S] Shamir. *How to share a secret*. CACM'79.

[VNS+] Vinod et al.. *On the power of computational secret sharing.*, Indocrypt'03

[Y] Yao. *How to generate and exchange secrets*. FOCS'86.

Thank you!