Somewhat Non-Committing Encryption and Efficient Adaptively Secure Oblivious Transfer

Hong-Sheng Zhou
University of Connecticut

Joint work with
Juan Garay (AT&T) and **Daniel Wichs** (NYU)

CRYPTO 2009

Outline

- Background
- New Approach to Adaptive Security
- Application: Efficient and Adaptively Secure Oblivious Transfer

Our Mission: "Strong" Security

- Protocols that withstand wide variety of adversarial attacks
- The *simulation paradigm* [GMW'87];
  arbitrary environments (Universal Composability [Canetti'01])
- Static vs. Adaptive security
  - Corruptions before computation starts vs. on-the-fly
  - Adaptive security models: Erasure vs. Non-Erasure

Our Mission: “Strong” Security

- Protocols that withstand wide variety of adversarial attacks
- The *simulation paradigm* [GMW’87];
  arbitrary environments (Universal Composability [Canetti’01])
- Static vs. Adaptive security
  - Corruptions before computation starts vs. on-the-fly
  - Adaptive security models: Erasure vs. Non-Erasure

"Strong" Security: Partial History

- Feasibility results: Possible to design adaptively secure UC protocols for almost any task, assuming some trusted setup (e.g., CRS) [CLOS'02]
- Alternative **efficient** approaches by sacrificing some aspect of security [DN'03, KO'04, GMY'04, DI'05, JS'07, LP'07, Lindell'09, …]
  - static UC security
  - adaptive UC security in the erasure model
  - adaptive UC security for honest majority
  - ….

“Strong” Security: Partial History (cont’d)

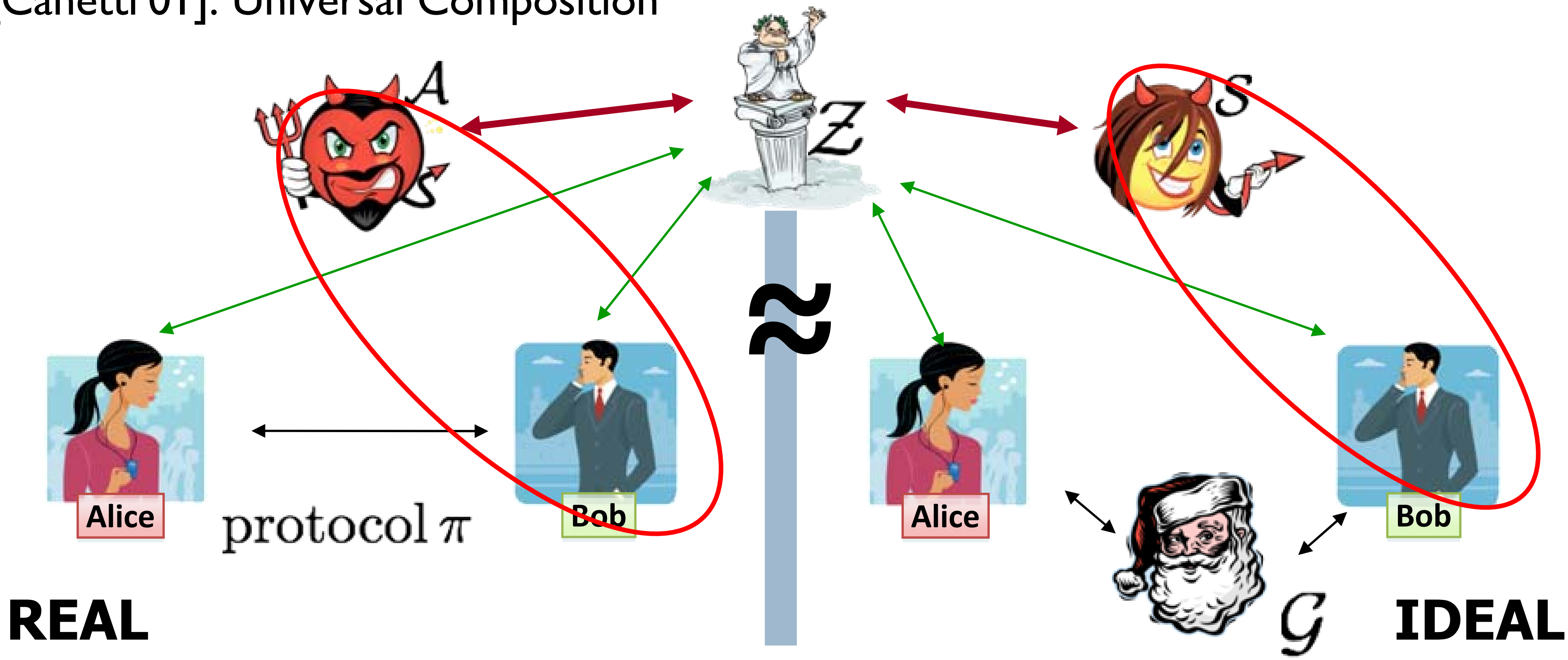
- Adaptive UC security can be achieved efficiently, *given an efficient adaptively secure string-OT protocol* [IPS’08]

Our Results

- Efficient (constant-round, constant public-key op's per bit) adaptively UC secure bit- and string-OT protocols based on standard number-theoretic assumptions
- "Semi-Adaptive" security for two-party tasks
  - Not allowed: Both parties start out honest and then become corrupted
- Compilers: Semi-Adaptive security ⇨ Adaptive security
  - Secure channels ("fully equivocal;" non-committing encryption)
  - "Somewhat equivocal" channels
- *Somewhat* Non-Committing Encryption
  - Limited "equivocation," much more efficient!

Simulation Paradigm: UC Security

[Canetti'01]: Universal Composition



Definition: protocol π is a secure realization of task $\mathcal{G}$ if:

For every real-world adversary $\mathcal{A}$

There exists an ideal-world adversary (simulator) $\mathcal{S}$

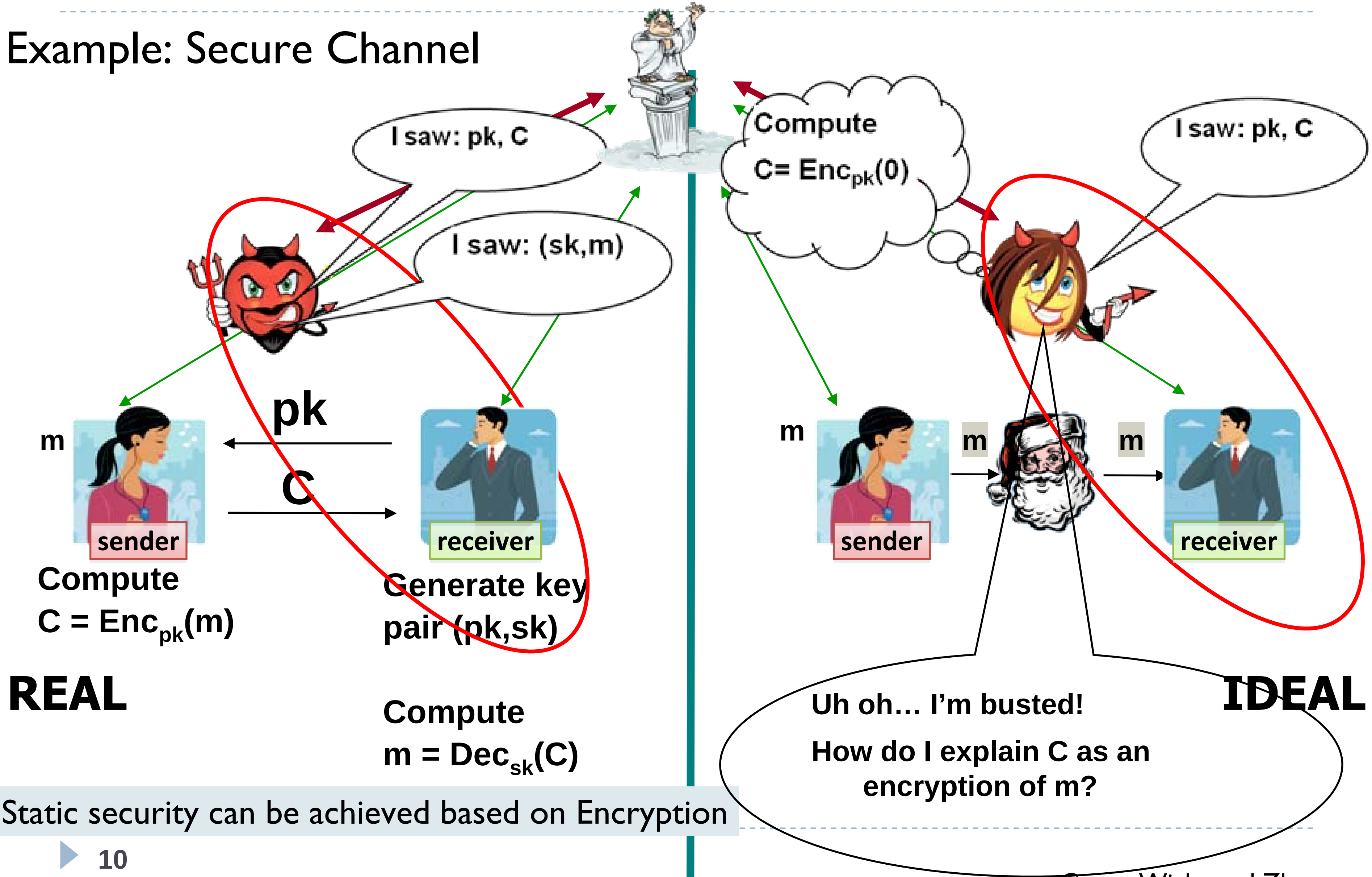
Two worlds indistinguishable to all environments $\mathcal{Z}$

Why is adaptive security hard?

- No constant round adaptively secure general 2-PC or MPC protocol is known
- Adaptive security hard even for basic tasks like "secure channels"
  - Basic public-key encryption is not enough.

Why is adaptive security hard?

Example: Secure Channel



Static security can be achieved based on Encryption

Garay, Wichs and Zhou

Why is adaptive security hard?

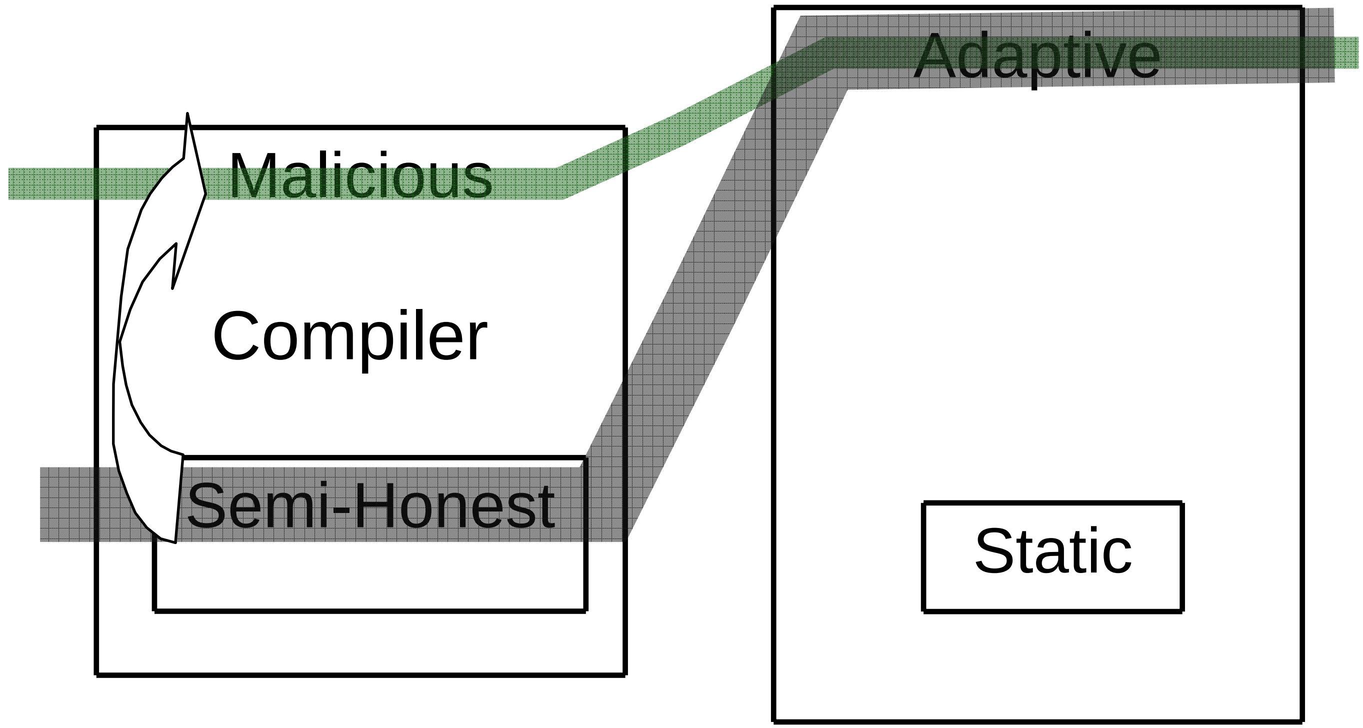
- No constant round adaptively secure general 2-PC or MPC protocol is known
- Adaptive security hard even for basic tasks like "secure channels"
  - Basic public-key encryption is not enough.
  - Extend encryption to Non-Committing Encryption [CFGN'96]
    - Simulator can run a "fake" encryption protocol to produce a ciphertext, and later explain the ciphertext as an encryption of some arbitrarily chosen plaintext
    - Done bit by bit [Beaver'97, DN'00]
    - Very expensive for encrypting long message: $O(1)$ public key operations per bit of message

Outline

- Background
- New Approach to Adaptive Security
- Application: Efficient and Adaptively Secure Oblivious Transfer

Previous Approach to Adaptive Security

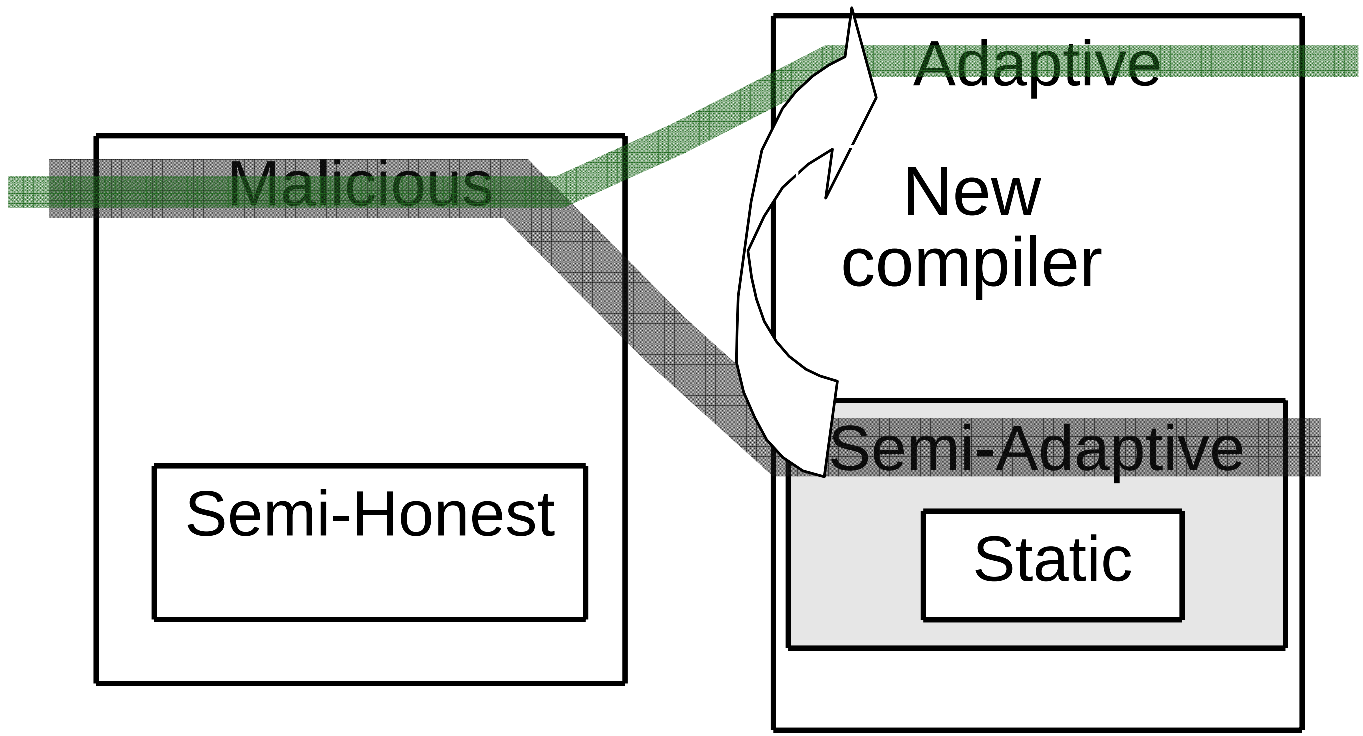
[CLOS'02] for multi-party tasks
[CDMW'09] for oblivious transfer



How?
Use expensive generic zero-knowledge proofs or cut-and-choose techniques

New Approach to Adaptive Security

This work: two-party tasks



1, Introduce Semi-Adaptive Security

2, Develop a new compiler

Semi-Adaptive Security for 2-Party Tasks

Adversary

Case 1: If no party is corrupted at the very beginning, then the adversary can't corrupt any parties.

Case 2: If there is a party corrupted at the very beginning, then the other party can be corrupted adaptively.

Missing case: If no party is corrupted at the very beginning, either party (or both) can be corrupted during the protocol execution.

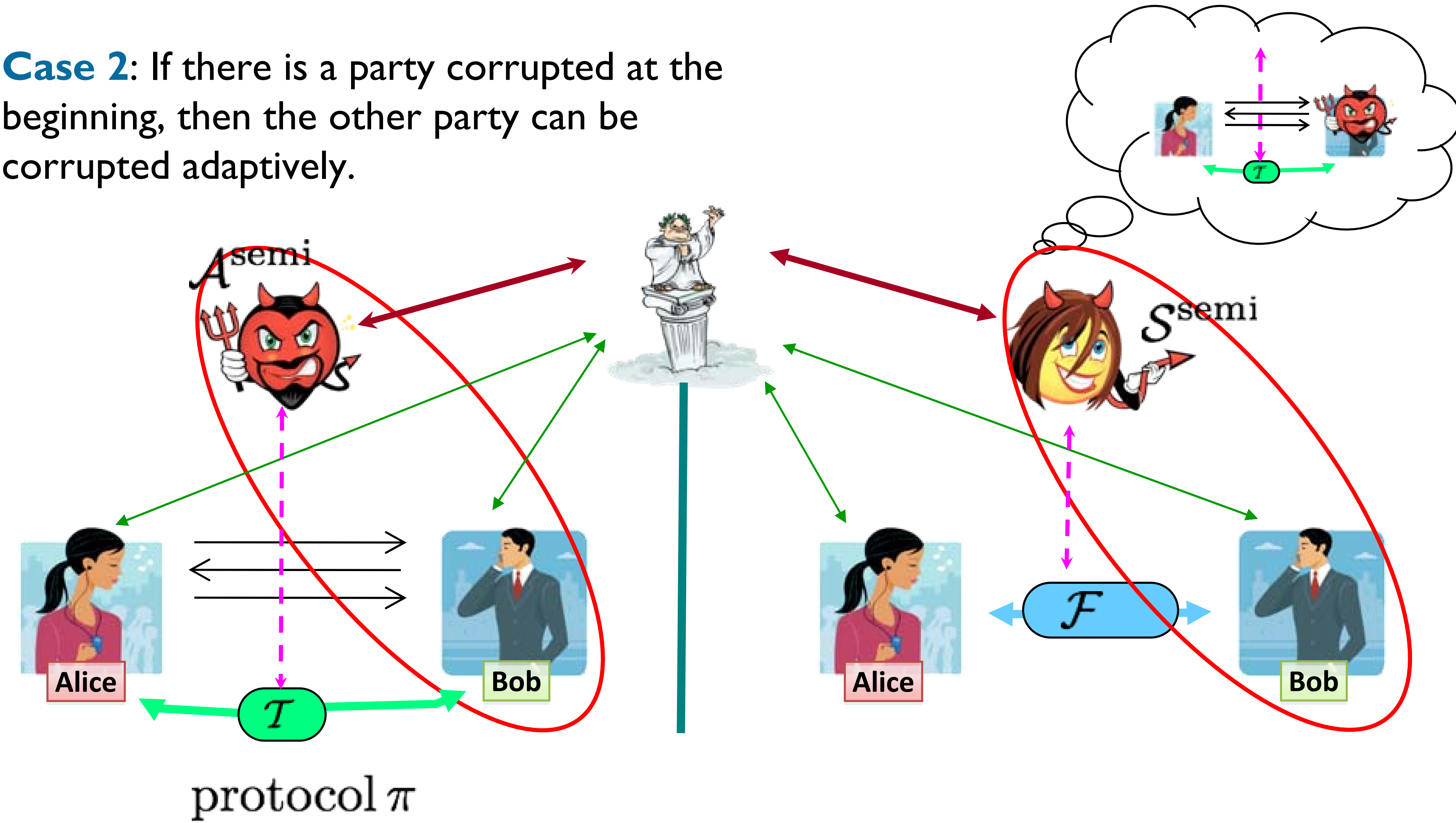
Simulator (Ideal World Adversary)

Trusted setup can be simulated **without** knowing which party is corrupted.

Take care of the corruptions in Cases 1 and 2.

Semi-Adaptive Security: Simulator

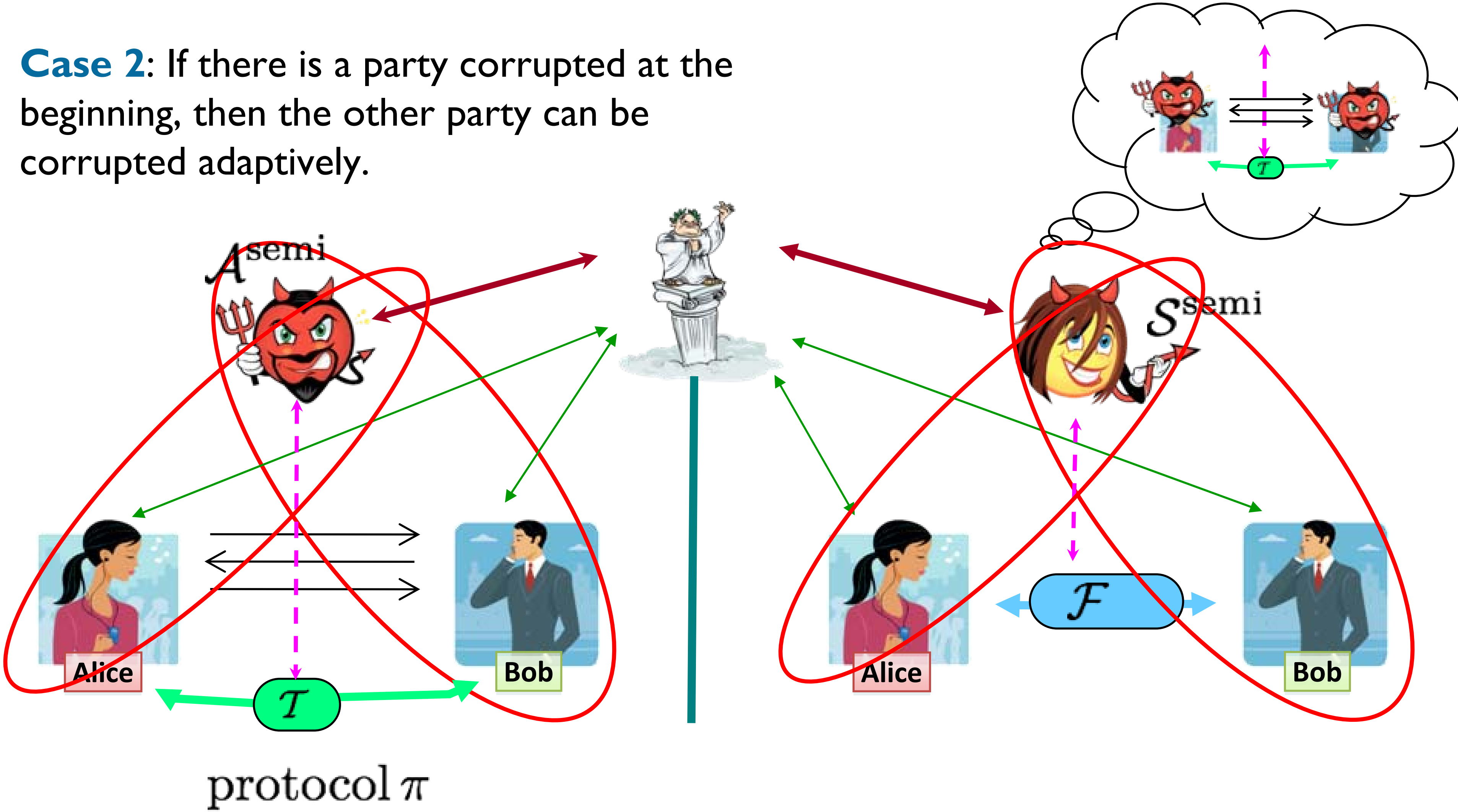
Case 2: If there is a party corrupted at the beginning, then the other party can be corrupted adaptively.



Garay, Wichs and Zhou

Semi-Adaptive Security: Simulator

Case 2: If there is a party corrupted at the beginning, then the other party can be corrupted adaptively.



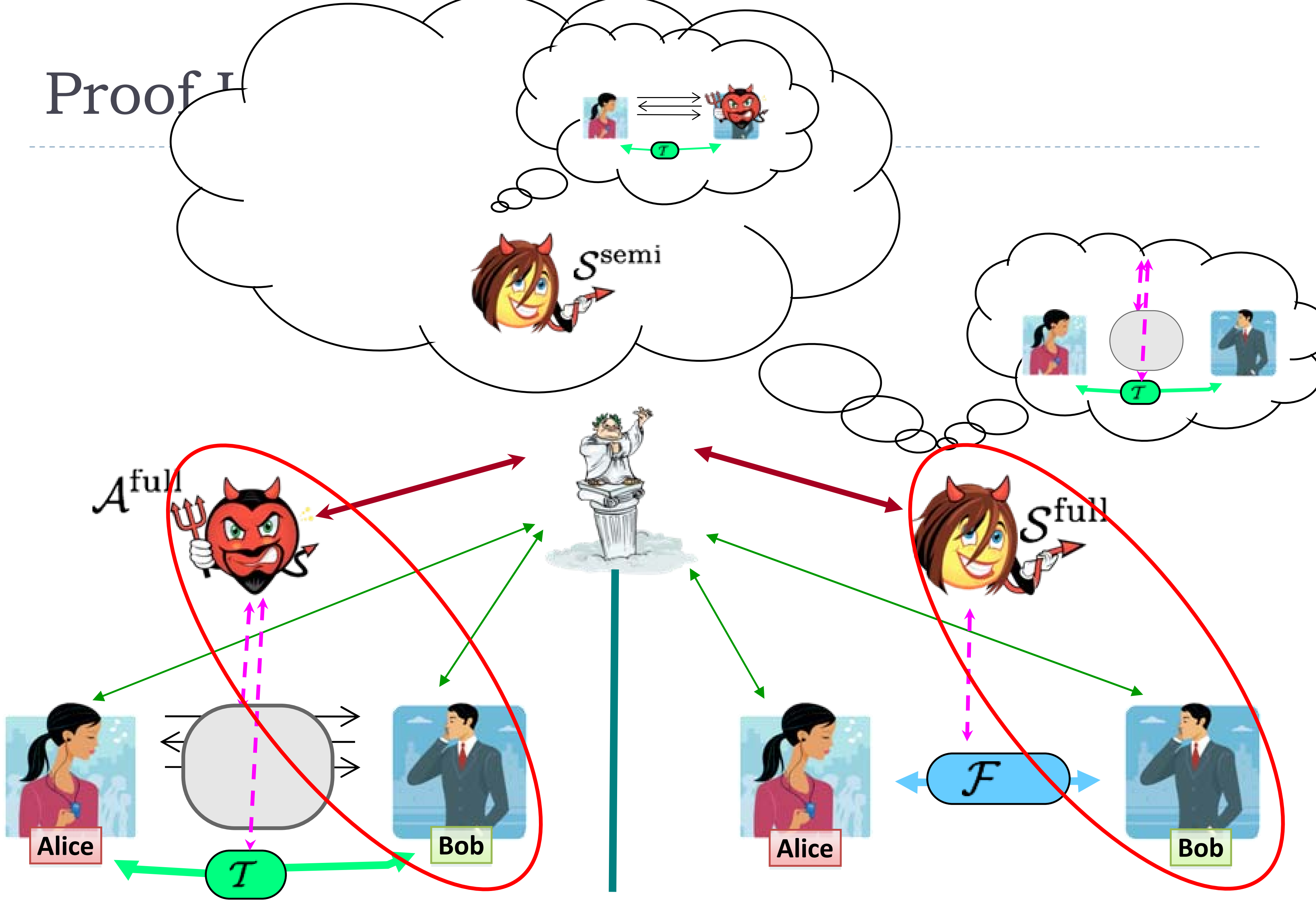
Garay, Wichs and Zhou

Compiler #1

- Conceptually simple: Use **secure channels** to protect communication transcripts between parties.

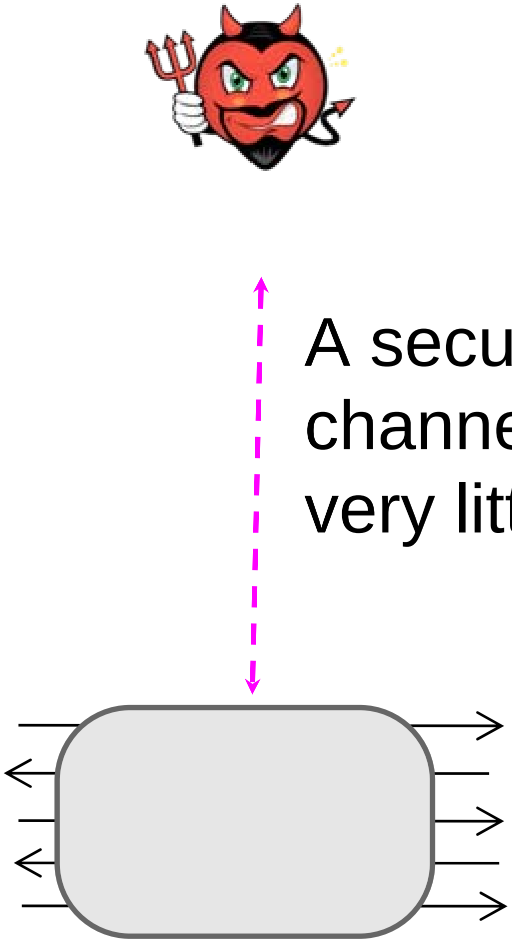
- **Theorem**: A semi-adaptively secure two-party protocol with communication protected by secure channels is fully adaptively secure.

Proof I

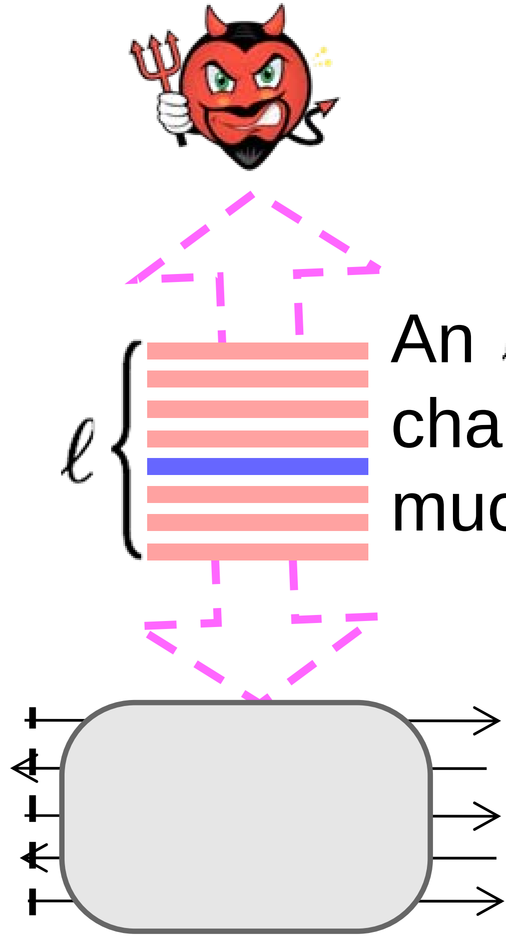


protocol π'
$= \pi +$ secure channel

ℓ-Equivocal Channel: Much Cheaper!



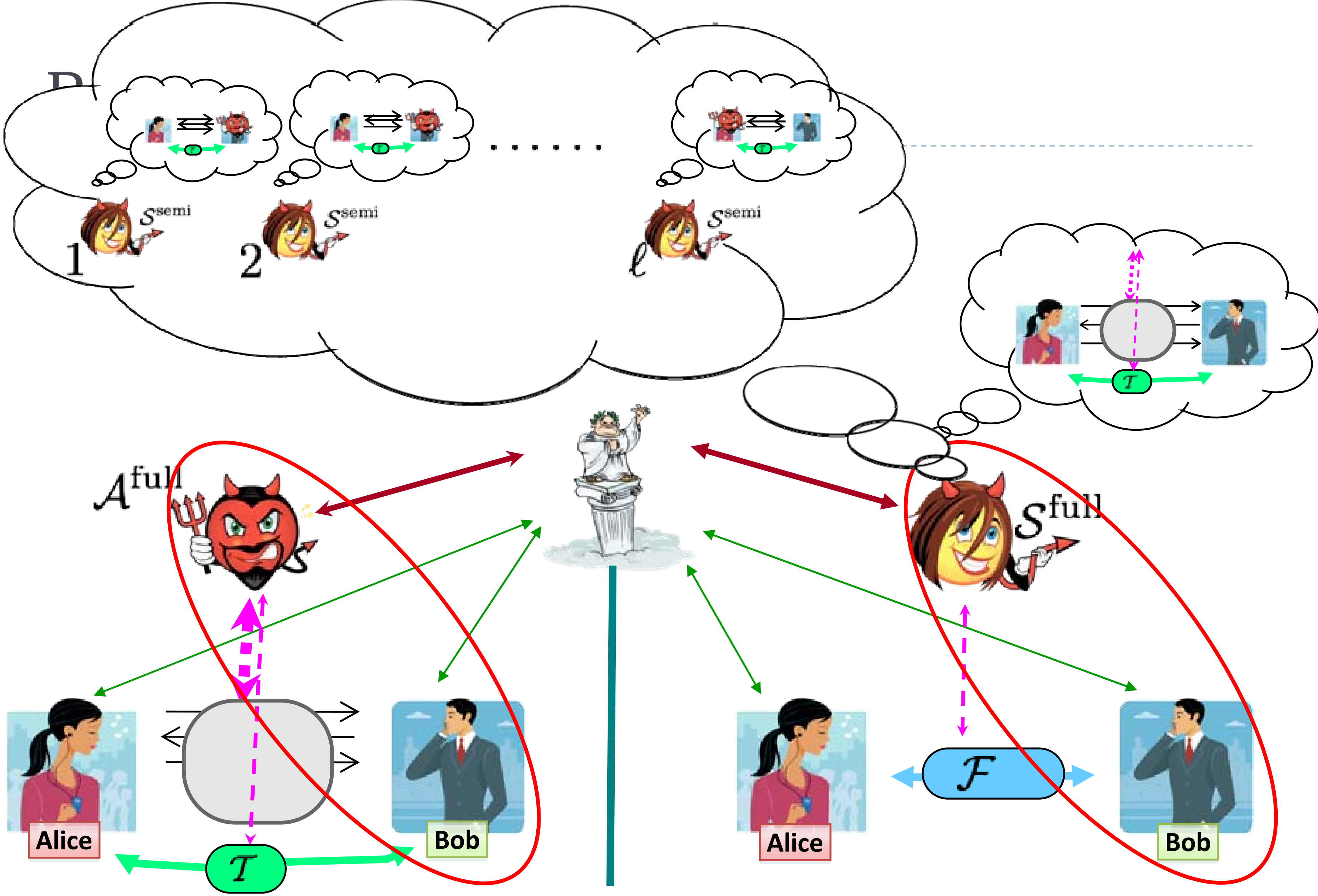
A secure channel leaks very little info



An ℓ-equivocal channel leaks much more info

Compiler #2

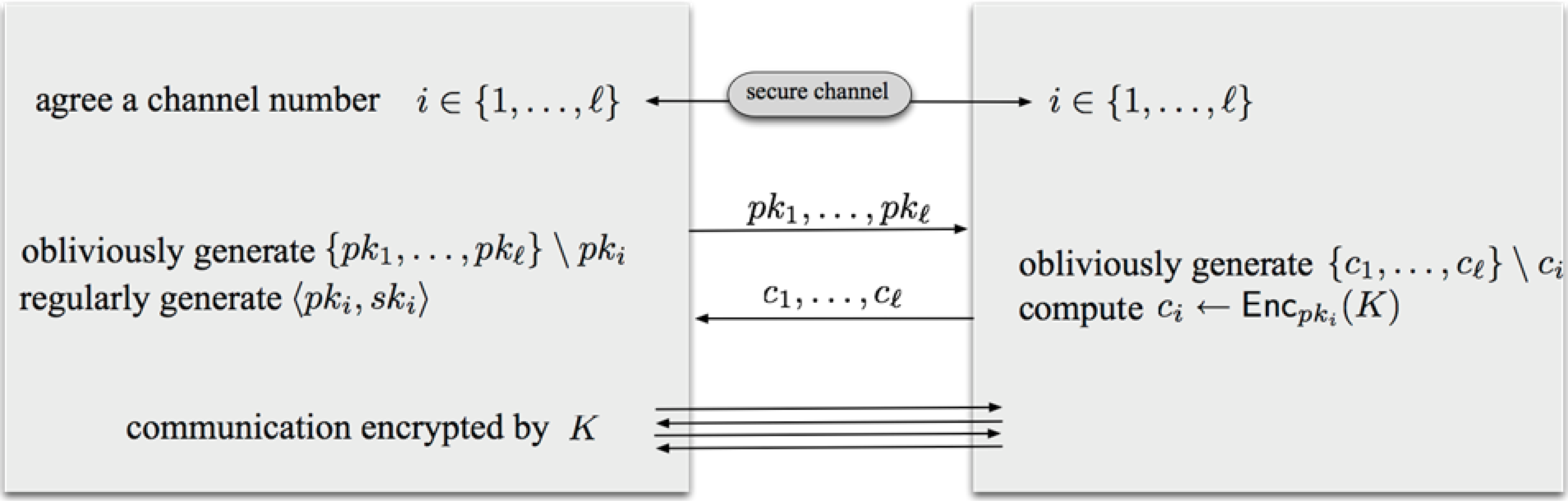
- New compiler: Use ℓ**-equivocal channels** to protect protocol communication
- **Theorem**: A semi-adaptively secure protocol for function $f = X_I \times X_R \rightarrow Y_I \times Y_R$ with communication protected by ℓ-equivocal channels is fully adaptively secure. Here $\ell = |X_I||Y_I| + |X_R||Y_R|$
- Very efficient with small input/output sizes (e.g., bit-OT)
- Proof idea: Communication between honest parties can be explained as any one of the ℓ possible "protocol executions" that may have occurred.



protocol π'
$= \pi + \ell$ equivocal channel

Garay, Wichs and Zhou

ℓ-Equivocal Channel: Implementation



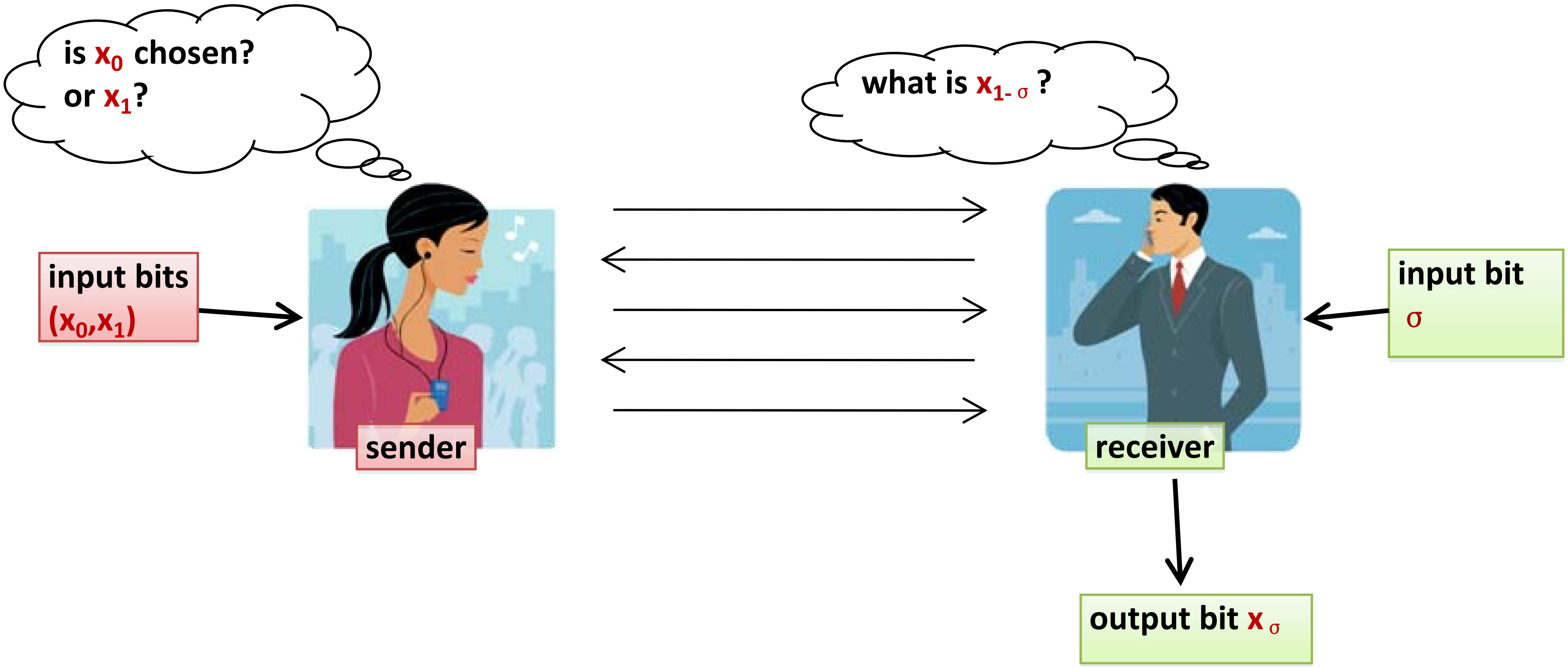
Garay, Wichs and Zhou

Outline

- Background
- New Approach to Adaptive Security
- Application: Efficient and Adaptively Secure Oblivious Transfer

1-out-of-2 Oblivious Transfer

[Rabin'81, EGL'85,Crepau'87]



Garay, Wichs and Zhou

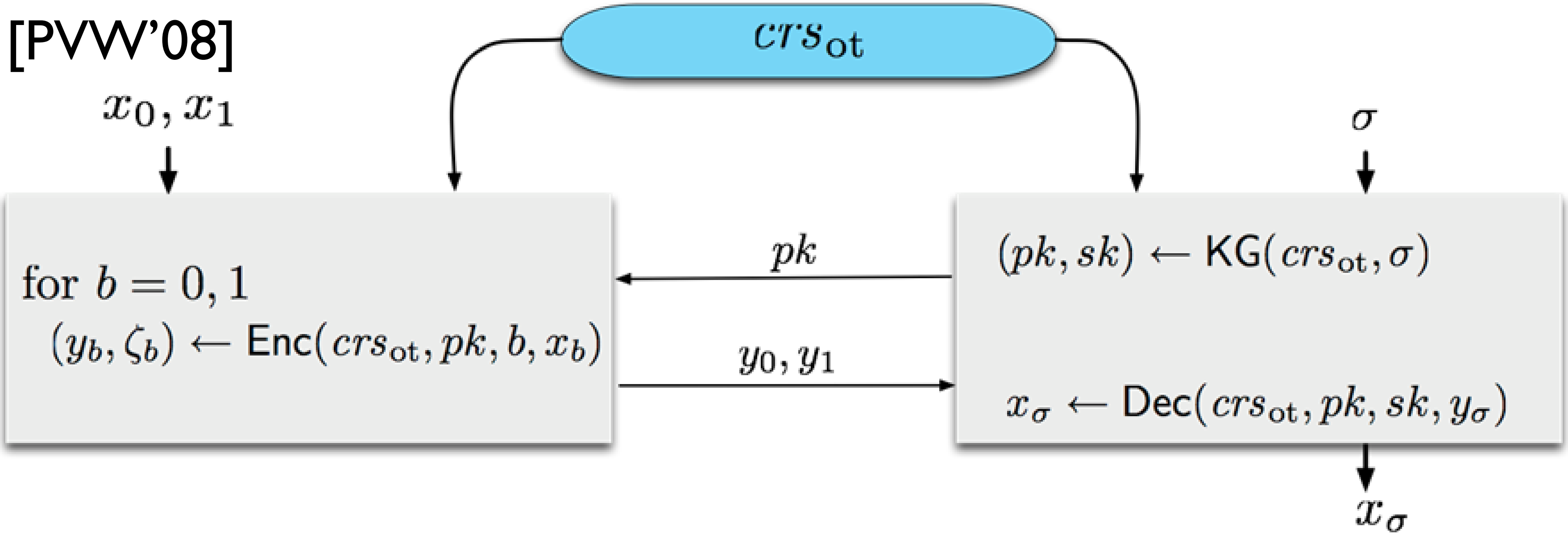
Why OT?

- OT is the cornerstone of secure computation [Yao'82,GMW'87,...,CLOS'02,...]
- OT is complete [Kilian'88]
- Founding secure computation on OT efficiently [IPS'08]

- No efficient adaptively UC-secure OT until recently (comparison later)

PVW OT (Malicious+Static Adversary)

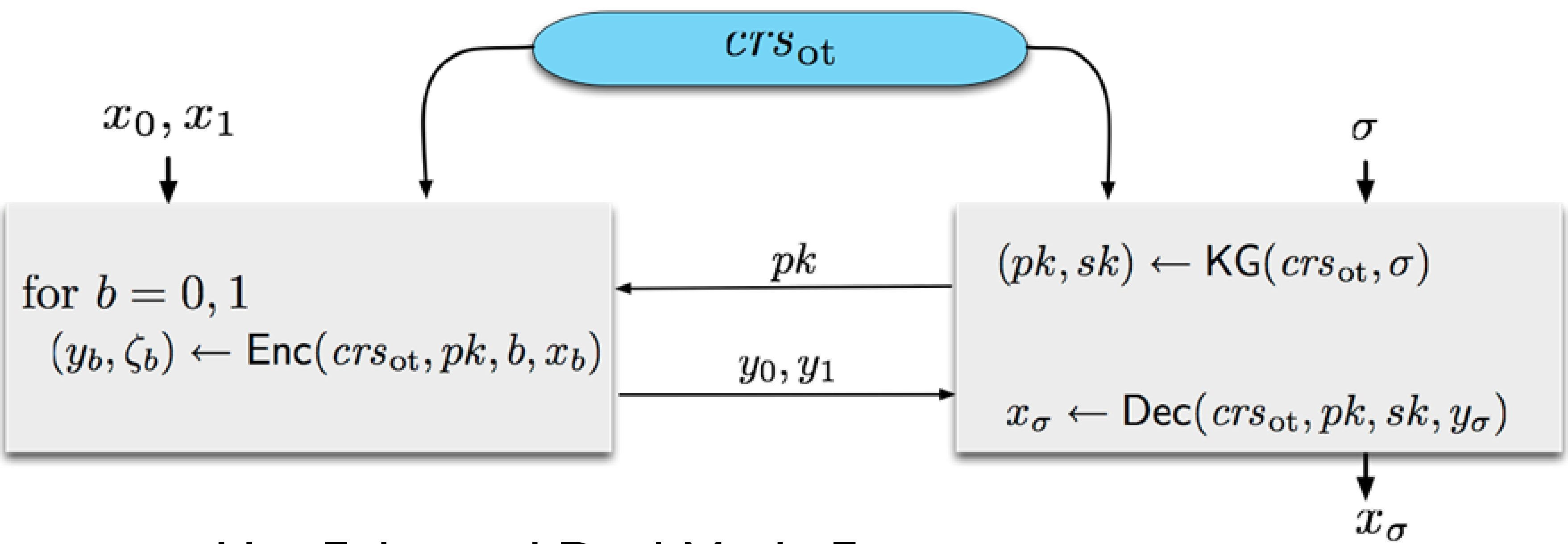
[PVW'08]



- Underlying building block: Dual Mode Encryption
- First truly efficient OT against malicious and **static** adversaries in the UC framework
- How to defend against **adaptive** adversaries?

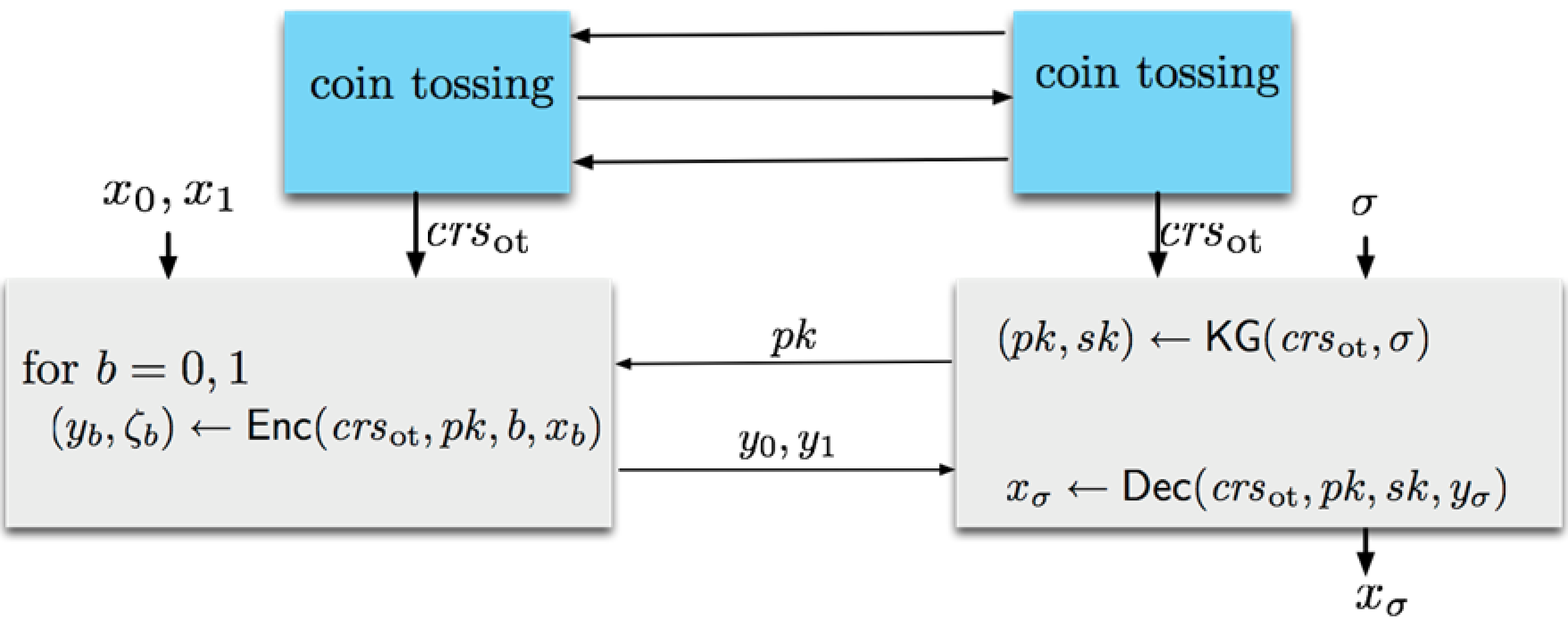
Our Approach to Adaptively Secure OT

- Step 1: Make PVW OT Semi-Adaptively Secure
  - Extend Dual Mode Encryption to support adaptive security: ***Enhanced Dual Mode Encryption***
  - Change the CRS setup to be simulated without knowing which party is corrupted
    - Coin-tossing protocol



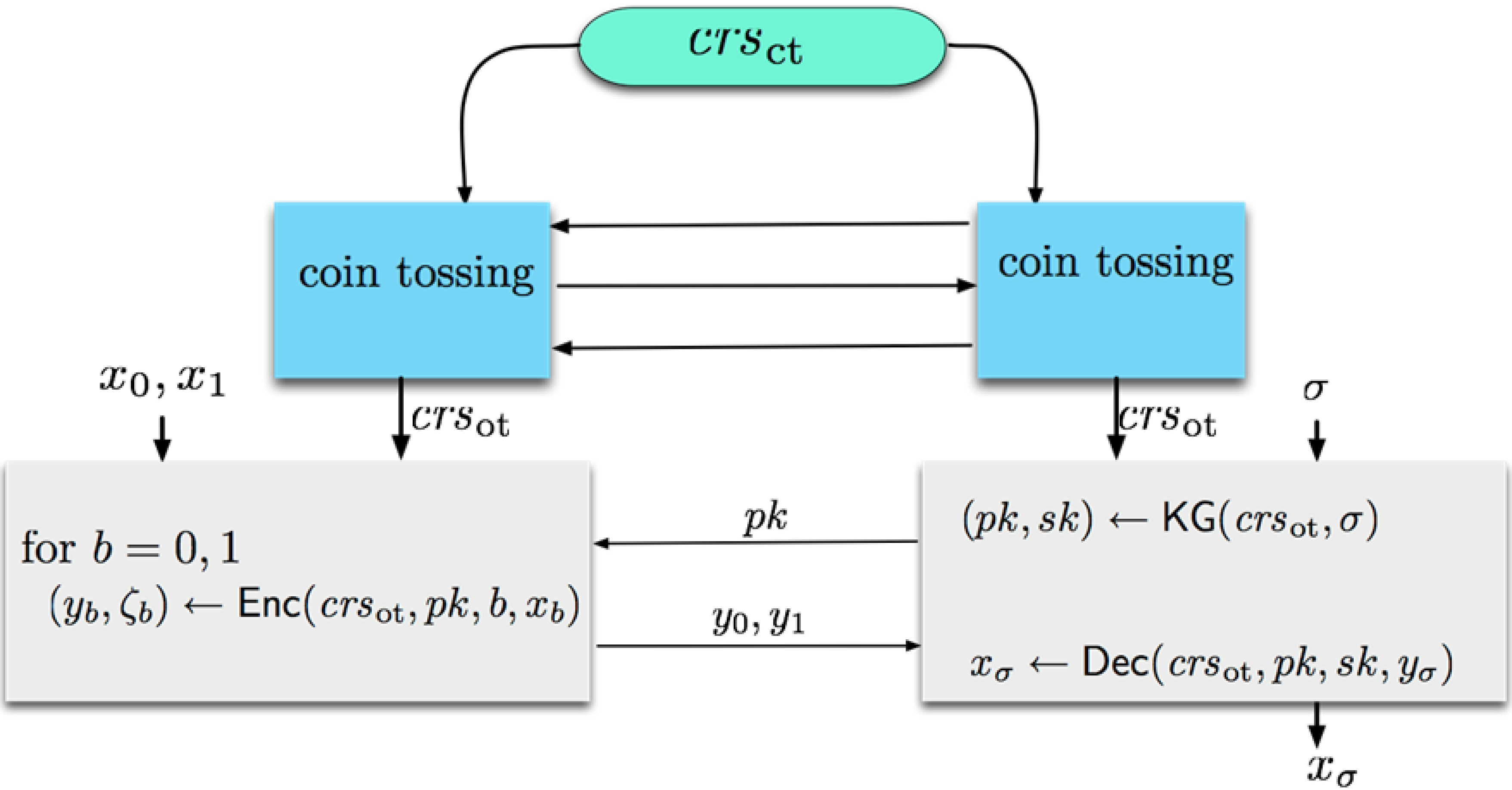
Use Enhanced Dual Mode Encryption

Use coin-tossing protocol to obtain the CRS for enhanced PVW

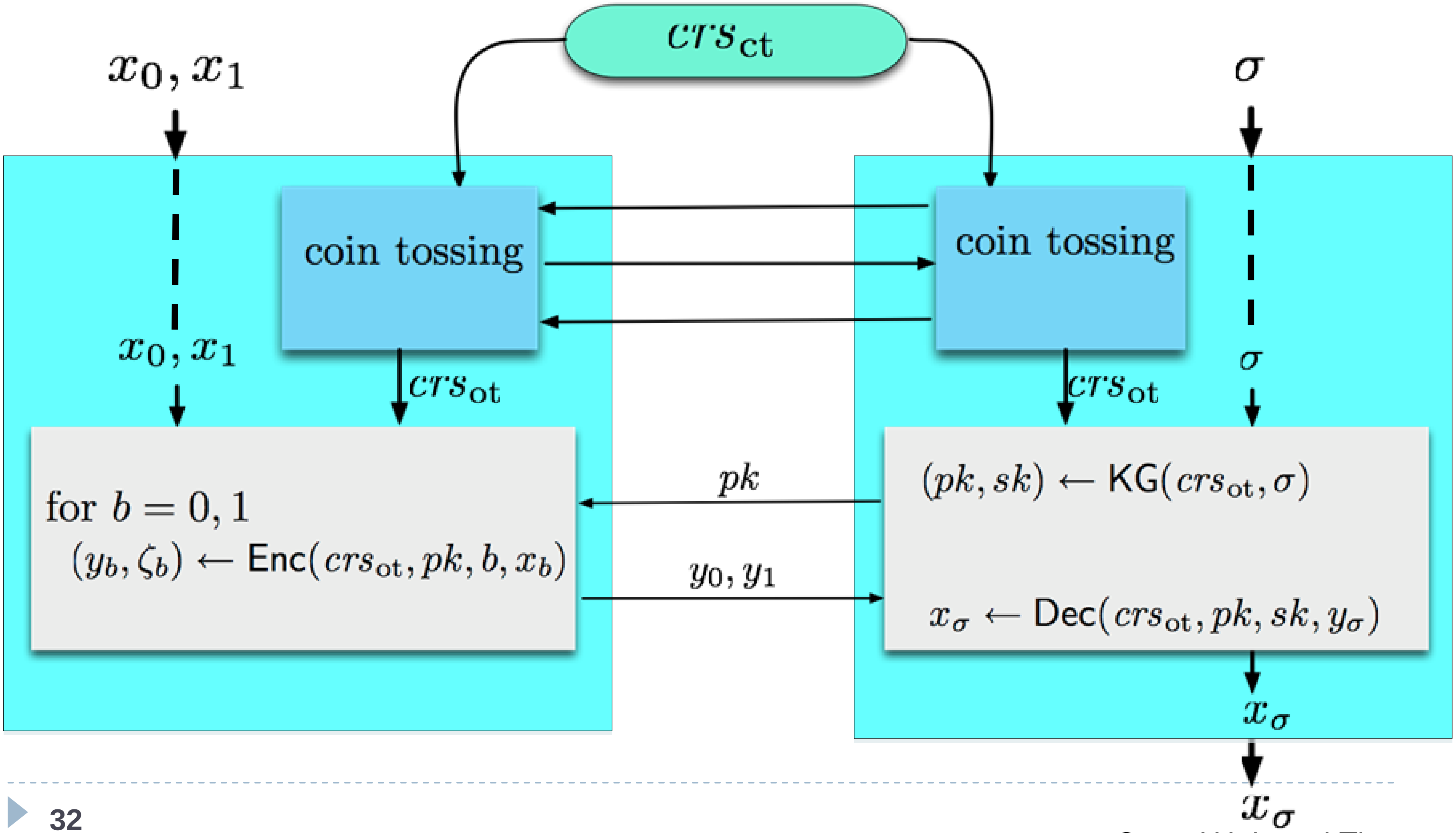


Garay, Wichs and Zhou

Such coin tossing protocol is based on a CRS which can be simulated without knowing which party is corrupted



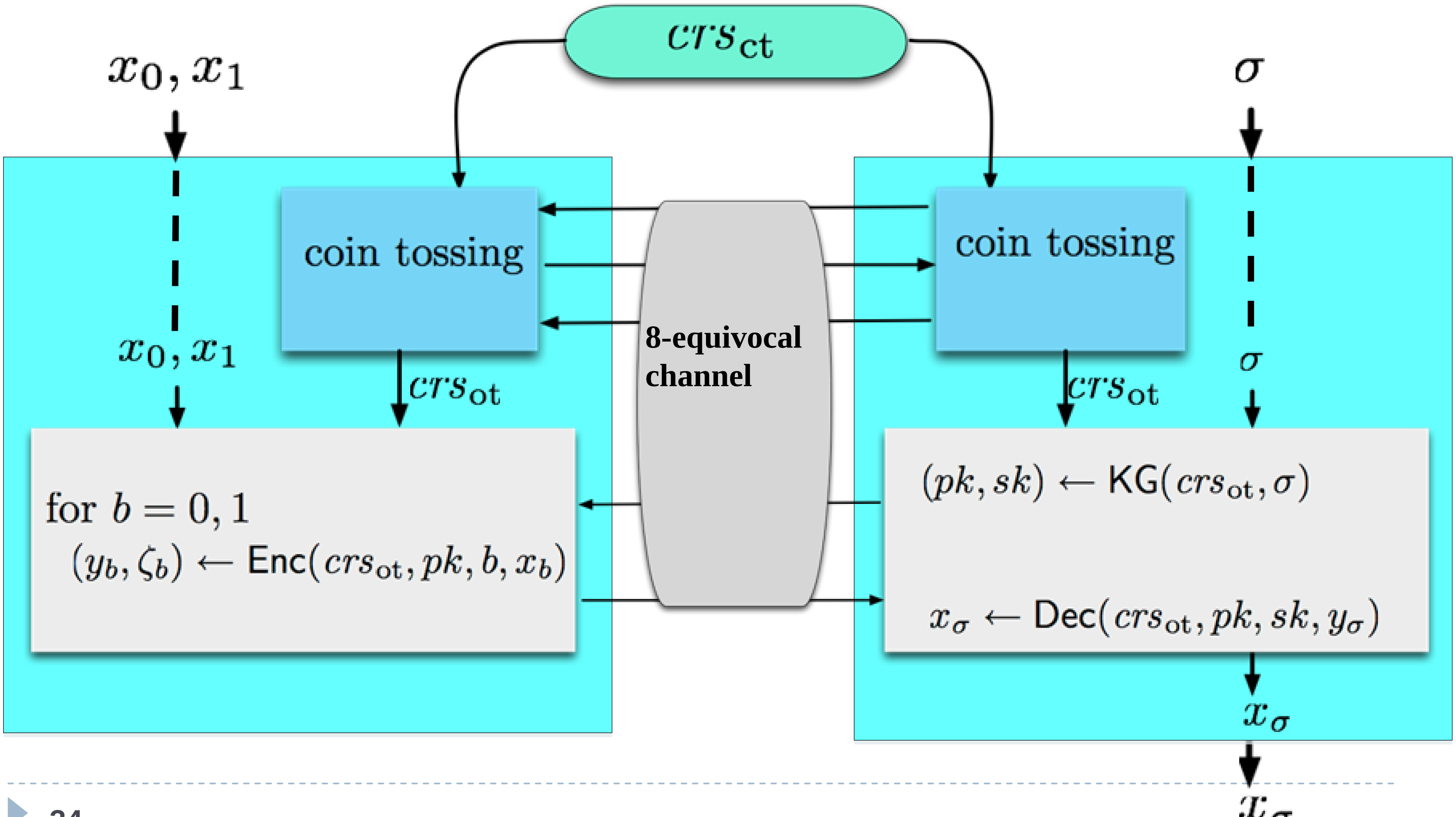
Garay, Wichs and Zhou



Garay, Wichs and Zhou

Our Approach to Adaptively Secure OT

- Step 1: Improve PVW OT to be Semi-Adaptively Secure
- Step 2:
  - Use an equivocal channel to protect the communication. Equivocality parameter is $\ell = 8$



Garay, Wichs and Zhou

Comparison with [CDMW'09]

Assumptions:

[CDMW'09]: general

Ours: DDH and DCR

Efficiency:

No. of public-key operations	bit-OT	string-OT (n bits)
[CDMW'09]	$O(\lambda^2)$	$O(\lambda^2 n)$
Ours: based on Secure Channel	$O(\lambda)$	$O(\lambda n)$
Ours: based on Equivocal Channel	$O(1)$	$O(n)$

Somewhat full version available at
eprint.iacr.org/2008/534

Thanks!