Adaptive and Composable Non-committing Encryptions

Takashi Nishide
joint work with
Huafei Zhu, Tadashi Araragi, Kouichi Sakurai

2010 July 5th

Supported by JAPAN SCIECE AND TECHNOLOGY AGENCY (JST), Strategic Japanese-Indian Cooperative Programme on Multidisciplinary Research Field, which combines Information and Communications Technology with Other Fields, entitled "Analysis of Cryptographic Algorithms and Evaluation on Enhancing Network Security Based on Mathematical Science."

Motivation

- Security against more powerful adversary is more preferable.
- However, constructing protocols that withstand a wider class of adversaries is usually harder to achieve…
- We consider to construct a secure channel protocol against an adaptive (more powerful) adversary in the UC framework.

Adversarial Models in Cryptographic Protocol

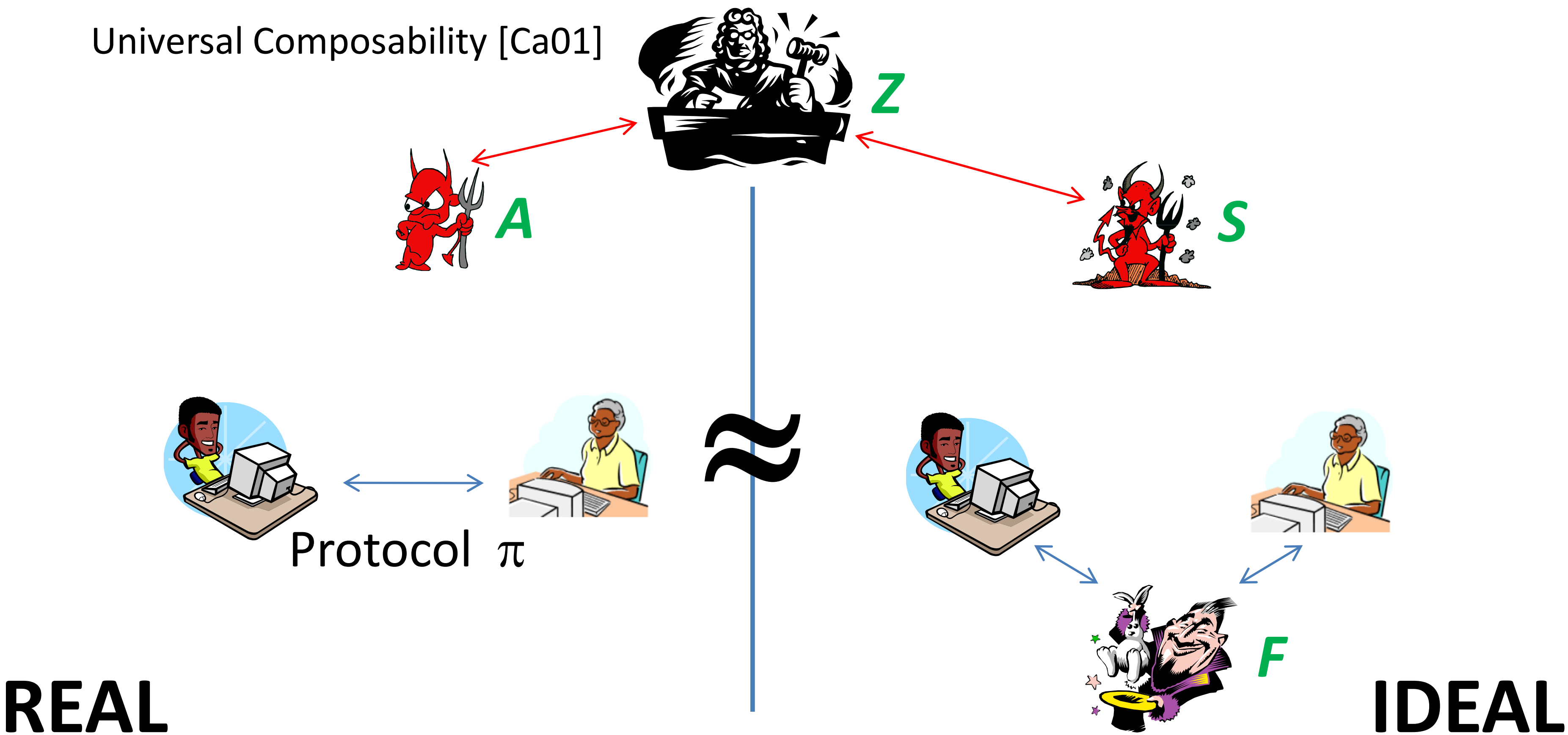
- Static vs Adaptive
  - Static adversary
    - needs to decide the set of players to corrupt prior to the execution of the protocol
  - Adaptive adversary
    - can corrupt players during the execution of the protocol arbitrarily
    - More flexible and realistic
- Erasure vs Non-Erasure
  - In the erasure model, players are assumed to be able to erase the past data when corrupted by an adversary
    - So the adversary cannot get the past computation history even if it corrupt a player
  - The erasure model is not realistic and may be impossible…
- Adversarial models have a large influence on security proof
- In particular, an adaptive adversary in the non-erasure model makes it hard to construct a secure channel

Adaptive Security for Secure Channel

- Secure channel is a basic cryptographic primitive.
- However, to construct a secure channel against an adaptive adversary, **traditional public key encryption** is not sufficient…
- [Nie02] proved that no non-interactive communication protocol can achieve adaptive security without the random oracle(RO) model.
- So we need an interactive protocol to realize a secure channel against an adaptive adversary w/o the RO model.

Security Definition in UC Framework

Universal Composability [Ca01]



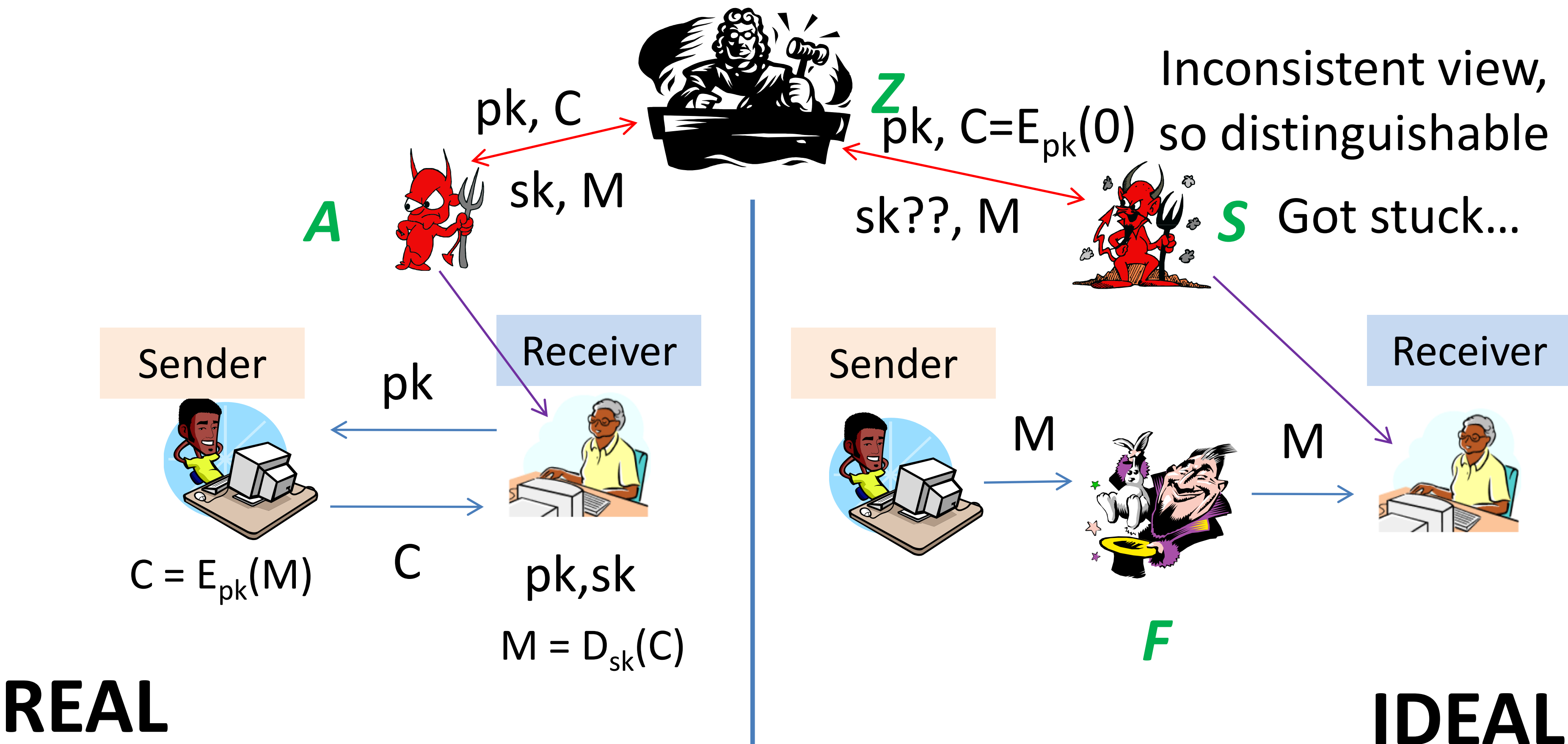
Protocol π is a secure realization of an ideal functionality **F** if

for every real adversary **A**

there exists a simulator **S** s.t.

ideal & real worlds are indistinguishable to any environment **Z**

Secure Channel with Adaptive Adversary?



Non-committing Encryption needed instead of Public Key Encryption

Non-committing Encryption

- With non-committing encryption(NCE), we can construct a secure channel protocol against an adaptive adversary.
- Simulator can run an NCE protocol and create a fake ciphertext that can be opened to any chosen plaintext (0 or 1).
- Encryption is done for each bit of message M
  - inefficient, but same efficiency as other schemes in the non-erasure model
  - Price for adaptive security…

Building Block

- Naor-Pinkas randomizer (NPR) ϕ [NP01]
- Setup: $p = 2q+1$

  $G \subseteq Z_p^*$ is a subgroup of order q
- $\phi((g_1, g_2, h_1, h_2) \times (s,t))$ defined as

  $(u, v) = (g_1^s g_2^t \bmod p,\ h_1^s h_2^t \bmod p)$

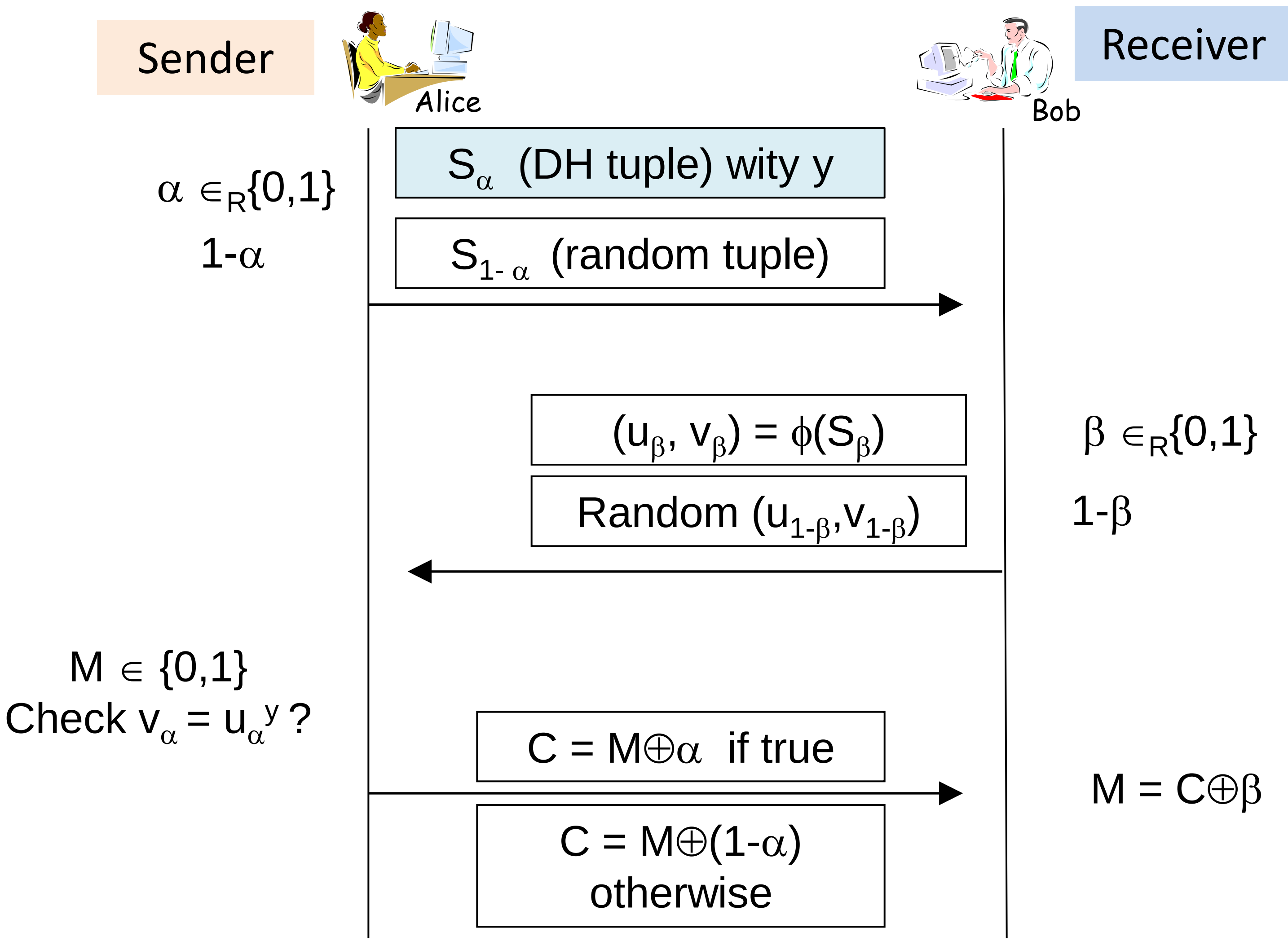
  where $s,t \in_R Z_q$,and $g_i, h_i \in G$
  - If $(g_1, g_2, h_1 = g_1^y, h_2 = g_2^y)$ is a random Diffie-Hellman tuple, we have $v = u^y \bmod p$
  - If (g_1, g_2, h_1, h_2) is a non-DH random tuple, (u,v) is a random tuple in G^2.

Building Block cont'd

- Canetti-Fischlin oblivious sampling & faking algorithms [CF01]

- By using the faking algorithm, the simulator can construct a fake transcript (computation history) to the environment ***Z***
  - in such a way that a Diffie-Hellman tuple looks completely random

Sketch of Construction



More Formal Construction

- Sender generates with secret $\alpha \in_R \{0,1\}$, y
  - $S_0 = (g_{1,0}, g_{2,0}, h_{1,0}, h_{2,0})$
  - $S_1 = (g_{1,1}, g_{2,1}, h_{1,1}, h_{2,1})$
  - where S_α is a DH tuple, $S_{1-\alpha}$ is a random tuple, and $h_{1,\alpha} = g_{1,\alpha}^y$, $h_{2,\alpha} = g_{2,\alpha}^y$
- Receiver generates with secret $\beta \in_R \{0,1\}$
  - $w_\beta = (u_\beta, v_\beta)$ from S_β with Naor-Pinkas randomizer
  - $w_{1-\beta} = (u_{1-\beta}, v_{1-\beta})$ at random
  - Sends w_β , $w_{1-\beta}$ to the sender
- Sender checks $v_\alpha = u_\alpha^y$ mod p?
  - If true, ciphertext $C = M \oplus \alpha$ where $\alpha = \beta$
  - Otherwise, ciphertext $C = M \oplus (1-\alpha)$ where $\alpha \neq \beta$

Proof in UC Framework

- Ideal functionality for non-committing encryption.
- Case analysis based on when the corruption occurs
- Simulator uses the Canetti-Fischling oblivious faking algorithm to show the randomness used in the corrupted player to the environment ***Z***.
- Indistinguishability based on DDH assumption

Functionality F_{NCE}[Ca01]

- Upon receiving an input (send, sid, m), do: If sid = (S, R, sid') for some R then send (send, sid, l(m)) to the adversary, generate a private delayed output (send, sid, m) to R and halt. Else, ignore the input.
- Upon receiving (corrupt, sid, P) from the adversary, where $P \in \{S,R\}$, disclose m to the adversary. Next, if the adversary provides a value m', and P=S, and no output has been yet written to R, then output (send, sid, m') to R and halt.

Summary

- Non-committing encryption protocol secure against an adaptive adversary with the DDH assumption
- Proof given in the UC framework and non-erasure model
- Can be used as a building block realizing secure channel in other protocols that need to be secure against an adaptive adversary

Thank you for you attention!