Lossy Encryption from General Assumptions

Brett Hemenway and Rafail Ostrovsky

Crypto in the Clouds Workshop, MIT

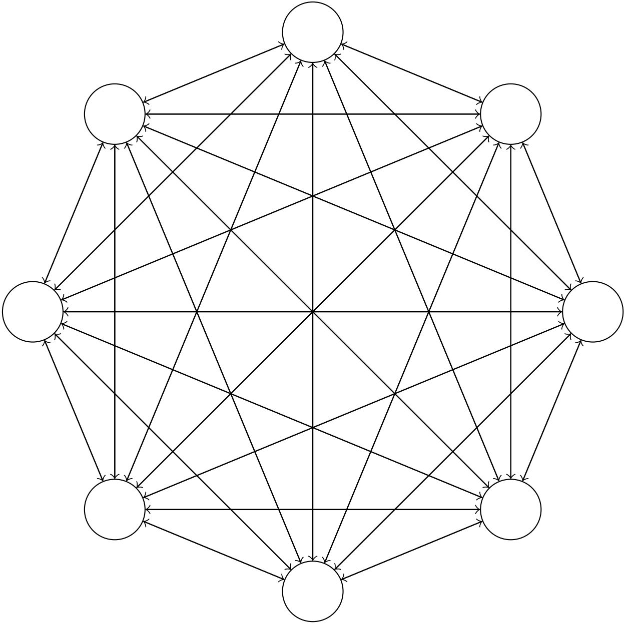
August 5, 2009

Outline

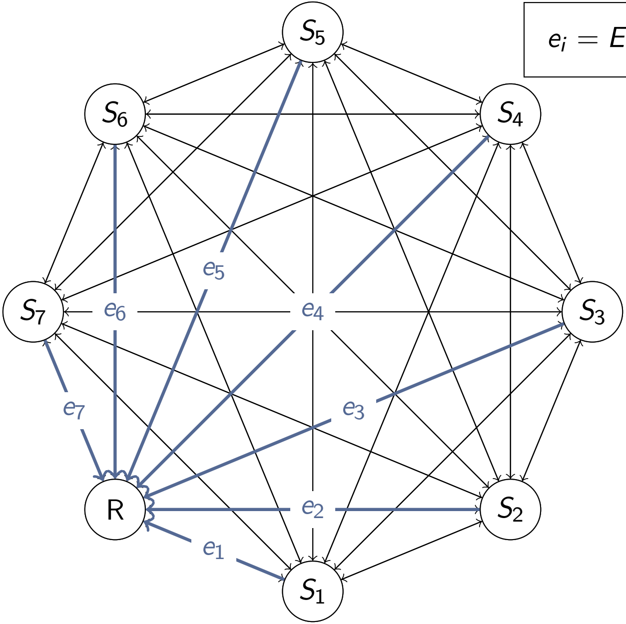
Outline

Motivation

Motivation



Motivation

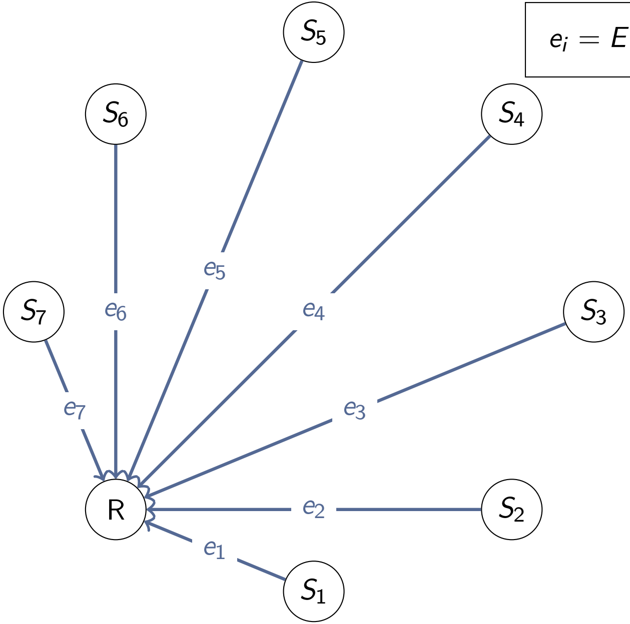


$$e_i = E(pk, m_i, r_i)$$

UCLA

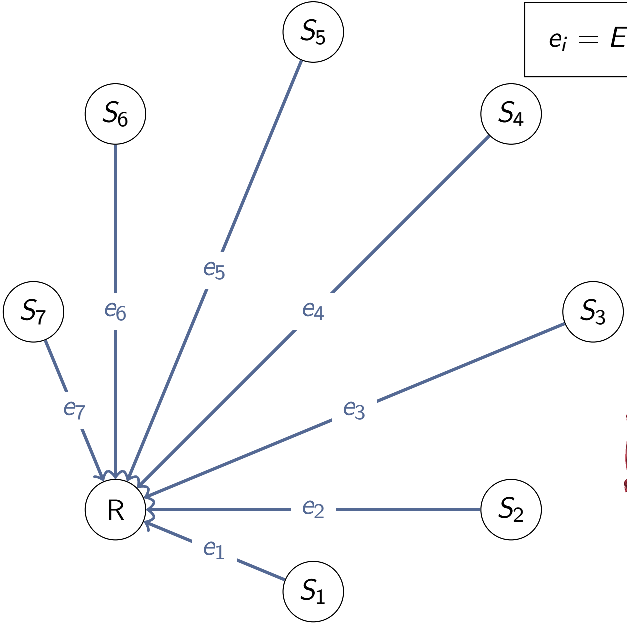
Motivation

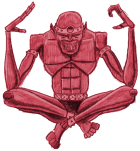
$$e_i = E(pk, m_i, r_i)$$



Motivation

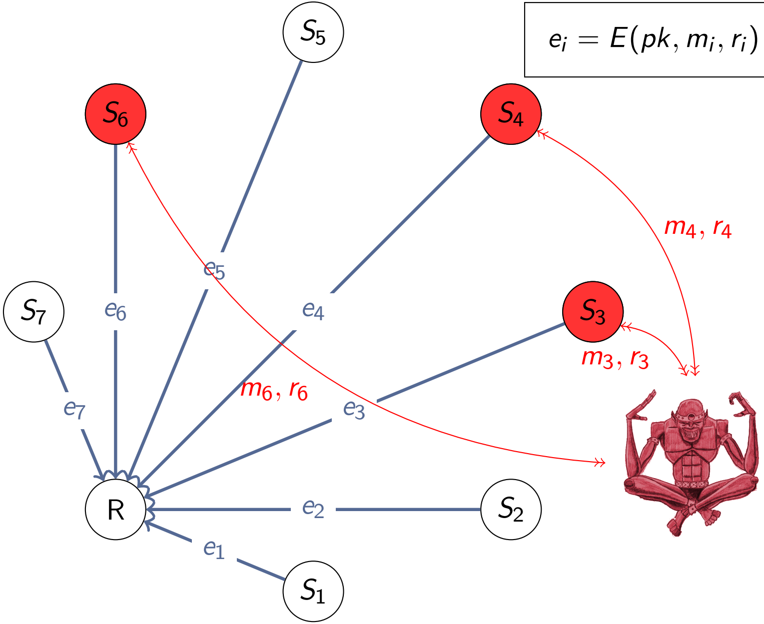
$$e_i = E(pk, m_i, r_i)$$



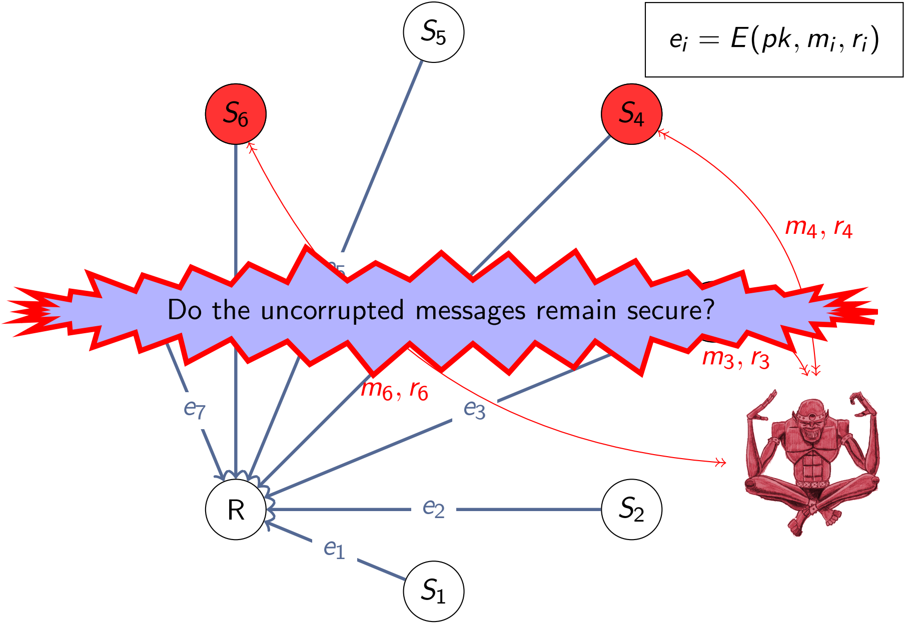


Motivation

$$e_i = E(pk, m_i, r_i)$$



Motivation


$e_i = E(pk, m_i, r_i)$
S_5
S_6
S_4
m_4, r_4
Do the uncorrupted messages remain secure?
m_3, r_3
m_6, r_6
e_7
e_3
R
e_2
S_2
e_1
S_1

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/Meaningless Encryption (KN08)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/Meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/Meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)
- Lossy Encryption (BHY09)

Lossy Encryption

This problem has been attacked by creating encryption protocols that are not always binding.

- Interactive Protocols (BH92)
- Non-committing Encryption (CFGN96)
- Extensions (B97,CHK05)
- Deniable Encryption (CDNO07)
- Meaningful/Meaningless Encryption (KN08)
- Dual-Mode Encryption (PVW08)
- Lossy Encryption (BHY09)

Outline

Motivation

Definitions

Our Results

Selective Opening Security

Selective Opening Security

- This type of security is called Selective Opening Security.

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
- If the messages are independent, then this follows from IND-CPA security.

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
- If the messages are independent, then this follows from IND-CPA security.
- No one has been able to show that IND-CPA security implies IND-SOA security.

Selective Opening Security

- This type of security is called Selective Opening Security.
  - Recognized long ago in folklore.
  - Formalized in [DNRS03],[BHY09]
- If the adversary does not learn the randomness, then this follows from IND-CPA security.
- If the messages are independent, then this follows from IND-CPA security.
- No one has been able to show that IND-CPA security implies IND-SOA security.
- No one has been able to exhibit an IND-CPA secure system that is not IND-SOA security.

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \mathrm{coins}(E)$

UCLA

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$

UCLA

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

UCLA

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \mathrm{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \mathrm{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $(m'_1, \ldots, m'_n) \leftarrow M | M_I$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $(m_1', \ldots, m_n') \leftarrow M | M_I$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1', \ldots, m_n'))$

Selective Opening Security: Indistinguishability [BHY09]

IND-SO-ENC (Real)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m_1, \ldots, m_n))$

IND-SO-ENC (Ideal)

- $(m_1, \ldots, m_n) \leftarrow M$
- $r_1, \ldots, r_n \leftarrow \text{coins}(E)$
- $I \leftarrow A((E(m_1, r_i), \ldots, E(m_n, r_n))$
- $(m'_1, \ldots, m'_n) \leftarrow M | M_I$
- $b \leftarrow A(((m_i, r_i))_{i \in I}, (m'_1, \ldots, m'_n))$

$$\left| \Pr\left[A^{IND-SO-ENC-REAL} = 1\right] - \Pr\left[A^{IND-SO-ENC-IDEAL} = 1\right] \right| < \nu$$

Lossy Encryption in Detail

$$G(1^\lambda, mode), E(pk, m, r), D(sk, c)$$

Correctness:

For all m, r

$$D(E(pk_I, m, r)) = m$$

Lossiness:

For all m_0, m_1

$$\{E(pk_L, m_0, r)\} \approx^s \{E(pk_L, m_1, r)\}$$

Indistinguishability

$$\{pk_I : pk_I \leftarrow G(1^\lambda, Injective)\} \approx^c \{pk_L : pk_L \leftarrow G(1^\lambda, Lossy)\}$$

Lossy Encryption in Detail

$$G(1^\lambda, mode), E(pk, m, r), D(sk, c)$$

Correctness:

For all m, r

$$\boxed{D(E(pk_I, m, r)) = m}$$

Lossiness:

For all m_0, m_1

$$\{E(pk_L, m_0, r)\} \approx^s \{E(pk_L, m_1, r)\}$$

Indistinguishability

$$\{pk_I : pk_I \leftarrow G(1^\lambda, Injective)\} \approx^c \{pk_L : pk_L \leftarrow G(1^\lambda, Lossy)\}$$

Lossy Encryption in Detail

$$G(1^\lambda, mode), E(pk, m, r), D(sk, c)$$

Correctness:

For all m, r

$$D(E(pk_I, m, r)) = m$$

Lossiness:

For all m_0, m_1

$$\{E(pk_L, m_0, r)\} \approx^s \{E(pk_L, m_1, r)\}$$

Indistinguishability

$$\{pk_I : pk_I \leftarrow G(1^\lambda, Injective)\} \approx^c \{pk_L : pk_L \leftarrow G(1^\lambda, Lossy)\}$$

Lossy Encryption in Detail

$$G(1^\lambda, mode), E(pk, m, r), D(sk, c)$$

Correctness:

For all m, r

$$D(E(pk_I, m, r)) = m$$

Lossiness:

For all m_0, m_1

$$\{E(pk_L, m_0, r)\} \approx^s \{E(pk_L, m_1, r)\}$$

Indistinguishability

$$\{pk_I : pk_I \leftarrow G(1^\lambda, Injective)\} \approx^c \{pk_L : pk_L \leftarrow G(1^\lambda, Lossy)\}$$

Lossy Encryption in Detail

$$G(1^\lambda, mode), E(pk, m, r), D(sk, c)$$

Correctness:

For all m, r

$$D(E(pk_I, m, r)) = m$$

Lossiness:

For all m_0, m_1

$$\{E(pk_L, m_0, r)\} \approx^s \{E(pk_L, m_1, r)\}$$

Indistinguishability

$$\{pk_I : pk_I \leftarrow G(1^\lambda, Injective)\} \approx^c \{pk_L : pk_L \leftarrow G(1^\lambda, Lossy)\}$$

Notice: Indistinguishability + Lossiness $\Longrightarrow$ IND-CPA security

Lossy Encryption is IND-SO-ENC Secure (BHY09)

In Lossy mode, the distributions

$$(E(m_1, r_1), \ldots, E(m_n, r_n)) \approx^s (E(m'_1, r_1), \ldots, E(m'_n, r_n))$$

Since the encryptions are statistically independent of the messages, so even after conditioning on certain openings, the rest remain independent of the messages.

ReRandomizable Encryption

ReRandomizable Encryption

- (G, E, D) is semantically secure.

ReRandomizable Encryption

- (G, E, D) is semantically secure.
- There exists a function ReRand such that for all pk, m, r, r'

ReRandomizable Encryption

- (G, E, D) is semantically secure.
- There exists a function ReRand such that for all pk, m, r, r'
  - Correctness:

$$D(\mathrm{ReRand}(E(pk, m, r))) = m$$

UCLA

ReRandomizable Encryption

- (G, E, D) is semantically secure.
- There exists a function ReRand such that for all pk, m, r, r'
  - Correctness:

$$D(\text{ReRand}(E(pk, m, r))) = m$$

  - Statistical rerandomization:

$$\{\text{ReRand}(E(pk, m, r))\} \approx^s \{\text{ReRand}(E(pk, m, r'))\}$$

Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\mathrm{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Caution: this is not necessarily statistically re-randomizing.

Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\mathrm{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Caution: this is not necessarily statistically re-randomizing.

It is statistically re-randomizing for all known homomorphic cryptosystems.

Homomorphic Encryption

If $E(pk, m, r)E(pk, m', r') = E(pk, m + m', r^*)$, then we can re-randomize by doing

$$\text{ReRand}(E(pk, m, r)) = E(pk, m, r)E(pk, 0, r').$$

Caution: this is not necessarily statistically re-randomizing.

It is statistically re-randomizing for all known homomorphic cryptosystems.

If you can sample statistically close to uniformly from the set of encryptions of 0 then homomorphic encryption is statistically rerandomizable

Outline

Motivation

Definitions

Our Results

Our Results

Our Results

- ReRandomizable Encryption “is” Lossy Encryption

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model

ReRandomizable Encryption "is" Lossy Encryption

ReRandomizable Encryption "is" Lossy Encryption

- Let $(G, E, D, \mathrm{ReRand})$ be a ReRandomizable Encryption.

ReRandomizable Encryption "is" Lossy Encryption

- Let $(G, E, D, \mathrm{ReRand})$ be a ReRandomizable Encryption.
- Let $(pk, sk) \leftarrow G$
  $e_0 = E(pk, b_0, r_0)$, $e_1 = E(pk, b_1, r_1)$.
  Define $PK = (pk, e_0, e_1)$, $SK = sk$.

ReRandomizable Encryption “is” Lossy Encryption

- Let (G, E, D, ReRand) be a ReRandomizable Encryption.
- Let $(pk, sk) \leftarrow G$
  $e_0 = E(pk, b_0, r_0)$, $e_1 = E(pk, b_1, r_1)$.
  Define $PK = (pk, e_0, e_1)$, $SK = sk$.
- Encryption of b will be

$$\text{ReRand}(e_b).$$

ReRandomizable Encryption "is" Lossy Encryption

- Let $(G, E, D, \mathrm{ReRand})$ be a ReRandomizable Encryption.
- Let $(pk, sk) \leftarrow G$
  $e_0 = E(pk, b_0, r_0)$, $e_1 = E(pk, b_1, r_1)$.
  Define $PK = (pk, e_0, e_1)$, $SK = sk$.
- Encryption of b will be

$$\mathrm{ReRand}(e_b).$$

- Decryption is the same as for the ReRandomizable scheme.

ReRandomizable Encryption “is” Lossy Encryption

- Let (G, E, D, ReRand) be a ReRandomizable Encryption.
- Let $(pk, sk) \leftarrow G$
  $e_0 = E(pk, b_0, r_0)$, $e_1 = E(pk, b_1, r_1)$.
  Define $PK = (pk, e_0, e_1)$, $SK = sk$.
- Encryption of b will be

$$\text{ReRand}(e_b).$$

- Decryption is the same as for the ReRandomizable scheme.

This is lossy if $b_0 = b_1$, and injective if $b_0 \neq b_1$.

ReRandomizable Encryption "is" Lossy Encryption

- Let (G, E, D, ReRand) be a ReRandomizable Encryption.
- Let $(pk, sk) \leftarrow G$
  $e_0 = E(pk, b_0, r_0)$, $e_1 = E(pk, b_1, r_1)$.
  Define $PK = (pk, e_0, e_1)$, $SK = sk$.
- Encryption of b will be

$$\text{ReRand}(e_b).$$

- Decryption is the same as for the ReRandomizable scheme.

This is lossy if $b_0 = b_1$, and injective if $b_0 \neq b_1$.

The indistinguishability of modes follows immediately from the Semantic Security of (G, E, D).

For Homomorphic Encryption

For Homomorphic Encryption

- If (G, E, D) is homomorphic and $E(pk, 0, r)$ is statistically close to uniform on the set of encryptions of 0, then

For Homomorphic Encryption

- If (G, E, D) is homomorphic and $E(pk, 0, r)$ is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting $PK = (pk, e)$ where $e = E(pk, 0, r)$ in Lossy Mode and $E(pk, 1, r)$ in injective mode.

For Homomorphic Encryption

- If (G, E, D) is homomorphic and $E(pk, 0, r)$ is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting $PK = (pk, e)$ where $e = E(pk, 0, r)$ in Lossy Mode and $E(pk, 1, r)$ in injective mode.
- Encryption of m is just $e^m \cdot E(pk, 0, r)$.

For Homomorphic Encryption

- If (G, E, D) is homomorphic and $E(pk, 0, r)$ is statistically close to uniform on the set of encryptions of 0, then
- We can make lossy encryption, simply by setting $PK = (pk, e)$ where $e = E(pk, 0, r)$ in Lossy Mode and $E(pk, 1, r)$ in injective mode.
- Encryption of m is just $e^m \cdot E(pk, 0, r)$.
- Decryption is the same.

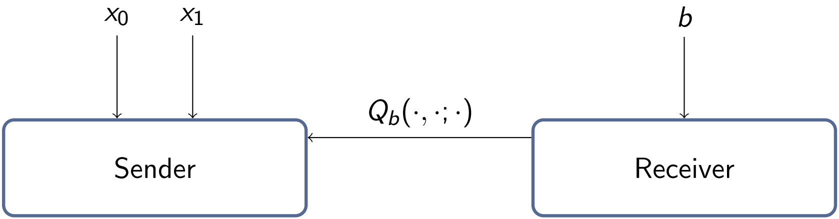
Oblivious Transfer Implies Lossy Encryption

Sender

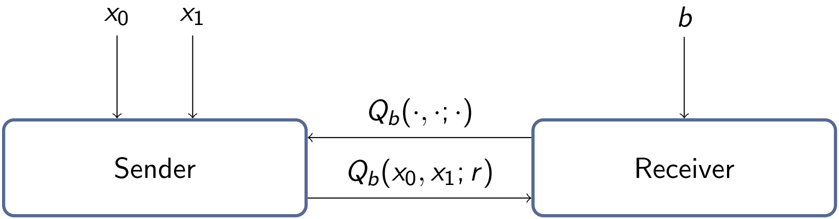
Receiver

Oblivious Transfer Implies Lossy Encryption

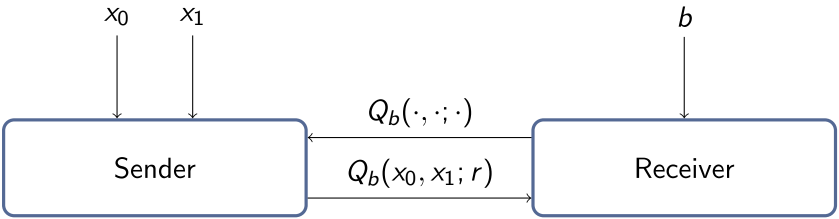
Oblivious Transfer Implies Lossy Encryption



Oblivious Transfer Implies Lossy Encryption



Oblivious Transfer Implies Lossy Encryption

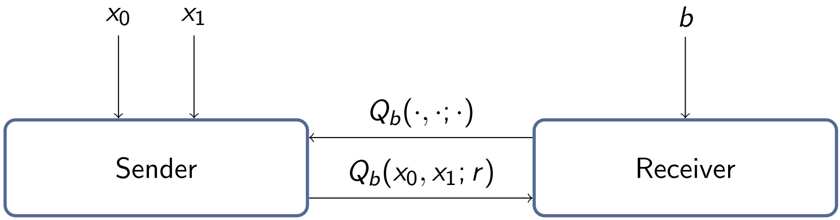


PK_{inj}:
Q_0

PK_{lossy}:
Q_1

$$E(m, r) \equiv Q_b(m, 0; r)$$

Oblivious Transfer Implies Lossy Encryption



PK_{inj}:
Q_0

PK_{lossy}:
Q_1

$$E(m, r) \equiv Q_b(m, 0; r)$$

Computational receiver privacy implies indistinguishability of modes

Statistical sender privacy implies lossiness of lossy branch

Chosen Ciphertext Security in the Selective Opening Setting

IND-SO-CCA2: Definitions

Challenger

Adversary

IND-SO-CCA2: Definitions

Challenger

Adversary

Decryption Queries

IND-SO-CCA2: Definitions

Challenger

Adversary

Decryption Queries

Selective Opening Query

IND-SO-CCA2: Definitions

Challenger

Adversary

Decryption Queries

Selective Opening Query

Decryption Queries

Output b

IND-SO-CCA2: Definitions

Challenger | Adversary

$$\xleftarrow{\quad c \quad}$$

$$\xrightarrow{\quad D(c) \quad}$$

$$\vdots$$

Selective Opening Query

Decryption Queries

IND-SO-CCA2: Definitions

Challenger

Adversary

c

$D(c)$

$\vdots$

$E(m_1, r_1), \ldots, E(m_n, r_n)$

I

$\{m_i, r_i\}_{i \in I}, \quad \{m'_j\}_{j \notin I}$

Decryption Queries

IND-SO-CCA2: Definitions

Challenger | Adversary

$$\xleftarrow{\quad c \quad}$$

$$\xrightarrow{\quad D(c) \quad}$$

$$\vdots$$

$$\xrightarrow{\quad E(m_1, r_1), \ldots, E(m_n, r_n) \quad}$$

$$\xleftarrow{\quad I \quad}$$

$$\xrightarrow{\quad \{m_i, r_i\}_{i \in I}, \quad \{m'_j\}_{j \notin I} \quad}$$

$$\xleftarrow{\quad c \quad}$$

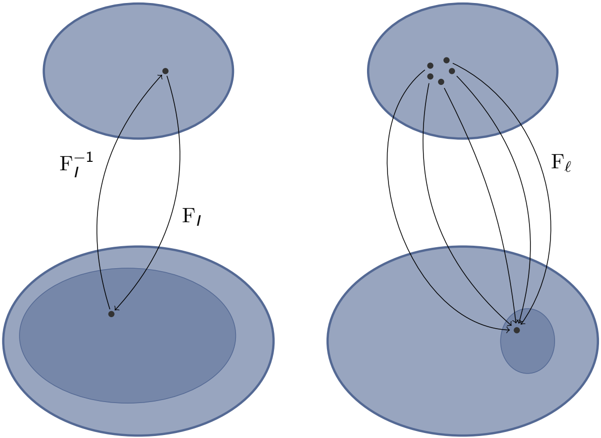
$$\xrightarrow{\quad D(c) \quad}$$

$$\vdots$$

Output b

Lossy Trapdoor Functions [PW08]

$$\mathrm{F}_I \approx \mathrm{F}_\ell$$



Injective Mode

Lossy Mode

Lossy Trapdoor Functions in Detail

$$(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)$$

Lossy Trapdoor Functions in Detail

$(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)$

$(s, \perp) \leftarrow G_{LTDF}(1^\lambda, lossy)$

Lossy Trapdoor Functions in Detail

$(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)$

$(s, \perp) \leftarrow G_{LTDF}(1^\lambda, lossy)$

Trapdoor:

$F^{-1}(t, F(s, x)) = x$

Lossy Trapdoor Functions in Detail

$$(s, t) \leftarrow G_{LTDF}(1^{\lambda}, inj)$$

Trapdoor:

$$F^{-1}(t, F(s, x)) = x$$

$$(s, \perp) \leftarrow G_{LTDF}(1^{\lambda}, lossy)$$

Lossiness:

$$|imF(s, \cdot)| \leq 2^r$$

Lossy Trapdoor Functions in Detail

$(s, t) \leftarrow G_{LTDF}(1^\lambda, inj)$

Trapdoor:
$F^{-1}(t, F(s, x)) = x$

$(s, \perp) \leftarrow G_{LTDF}(1^\lambda, lossy)$

Lossiness:
$|imF(s, \cdot)| \leq 2^r$

The first outputs of $G_{LTDF}(1^\lambda, inj)$, and $G_{LTDF}(1^\lambda, lossy)$ are computationally indistinguishable

All-But-One Functions [PW08]

$$(s, t) \leftarrow G_{ABO}(1^\lambda, b^*)$$

Trapdoor:

For $b \neq b^*$

$$F^{-1}(t, b, F(s, b, x)) = x$$

Lossiness:

$$|imF(s, b^*, \cdot)| \leq 2^r$$

The first outputs of $G_{ABO}(1^\lambda, b_0)$, and $G_{ABO}(1^\lambda, b_1)$ are computationally indistinguishable

All-But-n Functions

$$(s, t) \leftarrow G_{ABN}(1^\lambda, \mathcal{B}) \qquad \text{with } |\mathcal{B}| = n$$

Trapdoor:	Lossiness:
For $b \notin \mathcal{B}$	For $b \in \mathcal{B}$
$F^{-1}(t, b, F(s, b, x)) = x$	$\lvert imF(s, b, \cdot)\rvert \leq 2^r$

The first outputs of $G_{ABN}(1^\lambda, \mathcal{B}_0)$, and $G_{ABN}(1^\lambda, \mathcal{B}_1)$ are computationally indistinguishable.

All-But-n Functions

$$(s,t) \leftarrow G_{ABN}(1^\lambda, \mathcal{B}) \qquad \text{with } |\mathcal{B}| = n$$

Trapdoor:	Lossiness:
For $b \notin \mathcal{B}$	For $b \in \mathcal{B}$
$F^{-1}(t, b, F(s, b, x)) = x$	$\lvert imF(s, b, \cdot)\rvert \leq 2^r$

The first outputs of $G_{ABN}(1^\lambda, \mathcal{B}_0)$, and $G_{ABN}(1^\lambda, \mathcal{B}_1)$ are computationally indistinguishable.

Can be constructed from LTDFs

UCLA

IND-SO-CCA Construction

IND-SO-CCA Construction

- **KeyGen:**

$$(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \qquad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})$$

$$pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).$$

IND-SO-CCA Construction

- **KeyGen:**

$$(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \qquad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})$$

$$pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).$$

- **Encryption:**

IND-SO-CCA Construction

- **KeyGen:**

$$(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \qquad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})$$

$$pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).$$

- **Encryption:**

$$r^{sig} \leftarrow \text{coins}(\text{Sign}), \ x \leftarrow X$$

$$(vk, sk) = \text{G}(r^{sig}).$$

IND-SO-CCA Construction

- **KeyGen:**

$$(s_0, t_0) \leftarrow G_{LTDF}(1^\lambda, inj) \qquad (s_1, t_1) \leftarrow G_{ABN}(1^\lambda, \{1, \ldots, n\})$$

$$pk = (s_0, s_1) \quad \text{and} \quad sk = (t_0, t_1).$$

- **Encryption:**

$$r^{sig} \leftarrow \text{coins}(\text{Sign}), \;\; x \leftarrow X$$

$$(vk, sk) = \mathrm{G}(r^{sig}).$$

For a message m, calculate

$$(F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m)$$

$$\text{sig} = \text{Sign}_{sk}(F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m),$$

output the ciphertext:
$(vk, F_{LTDF}(s_0, x), F_{ABN}(s_1, vk, x), h(x) \oplus m, \text{sig})$

A SEM-SO-CCA Secure Construction

Intuition of our SEM-SO-CCA construction

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for n encryptions, so we need a NIZK which remains secure even after seeing n simulated proofs.

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for n encryptions, so we need a NIZK which remains secure even after seeing n simulated proofs.
    Unduplicatable set selection [S99]

UCLA

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for n encryptions, so we need a NIZK which remains secure even after seeing n simulated proofs.
    Unduplicatable set selection [S99]
  - After we make n simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.

Intuition of our SEM-SO-CCA construction

- To construct SEM-SO-CCA encryption we follow the Naor-Yung paradigm.
- There are difficulties:
  - An encryption query is actually a query for n encryptions, so we need a NIZK which remains secure even after seeing n simulated proofs.
    Unduplicatable set selection [S99]
  - After we make n simulated proofs, for $|I|$ of them, we are forced to reveal the randomness.
  - The statistically hiding property of lossy encryption allows us to prove IND-SO security.
    Statistical NIZKs should allow us to prove IND-SO-CCA security.

Statistical NIZKs [GOS06]

Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.

Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
- **Soundness:** False statements (with witnesses to their falseness) cannot be proven.

Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
- **Soundness:** False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge:** Nothing beyond the truth of the statement is revealed.

Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
- **Soundness:** False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge:** Nothing beyond the truth of the statement is revealed.
- **Proof of Knowledge:** There exists a simulator that can extract a witness from a valid proof.

Statistical NIZKs [GOS06]

- **Completeness:** All true statements can be proven.
- **Soundness:** False statements (with witnesses to their falseness) cannot be proven.
- **Zero-Knowledge:** Nothing beyond the truth of the statement is revealed.
- **Proof of Knowledge:** There exists a simulator that can extract a witness from a valid proof.
- **Honest-Prover State Reconstruction:** There exists a simulator that can create a proof P without a witness, then, given a witness w can produce randomness r such that P appears to have been generated with w and r.

Tools

Tools

- Unduplicatable Set Selector $\mathfrak{g}$.

Tools

- Unduplicatable Set Selector $\mathfrak{g}$.
- SEM-SO-ENC secure encryption (G_{so}, E, D).

Tools

- Unduplicatable Set Selector $\mathfrak{g}$.
- SEM-SO-ENC secure encryption (G_{so}, E, D).
- Statistical NIZKs $(\text{Prover}, \text{Verifier}, \text{Ext}, SR)$.

Tools

- Unduplicatable Set Selector $\mathfrak{g}$.
- SEM-SO-ENC secure encryption (G_{so}, E, D).
- Statistical NIZKs $(\mathsf{Prover}, \mathsf{Verifier}, \mathsf{Ext}, SR)$.
- Strongly Unforgeable One-Time Signatures $(\mathsf{Sign}, \mathsf{Ver})$.

SEM-SO-CCA Construction

SEM-SO-CCA Construction

- **KeyGen:**

$$(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda),\ \ (\sigma_i, \tau_i) \leftarrow \mathrm{Ext}_1(1^\lambda) \text{ for } i \in L$$

$$pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).$$

SEM-SO-CCA Construction

- **KeyGen:**

$$(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \mathrm{Ext}_1(1^\lambda) \text{ for } i \in L$$
$$pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).$$

- **Encryption:**

SEM-SO-CCA Construction

- **KeyGen:**

$$(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \mathrm{Ext}_1(1^\lambda) \text{ for } i \in L$$
$$pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).$$

- **Encryption:**

$$r^{sig} \leftarrow \mathrm{coins}(\mathrm{Sign}), \ r_0, r_1 \leftarrow \mathrm{coins}(E), \ \{r_i^{nizk}\}_{i=1}^{\ell} \leftarrow \mathrm{coins}(\mathrm{Prover}).$$
$$(vk, sk) = \mathrm{G}(r^{sig}).$$

SEM-SO-CCA Construction

- **KeyGen:**

$$(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \text{Ext}_1(1^\lambda) \text{ for } i \in L$$

$$pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).$$

- **Encryption:**

$$r^{sig} \leftarrow \text{coins(Sign)}, \ r_0, r_1 \leftarrow \text{coins}(E), \ \{r_i^{nizk}\}_{i=1}^{\ell} \leftarrow \text{coins(Prover)}.$$

$$(vk, sk) = \text{G}(r^{sig}).$$

For a message m, calculate

$$e_0 = E(pk_0, m, r_0), \quad e_1 = E(pk_1, m, r_1)$$

set $w = (m, r_0, r_1)$.

SEM-SO-CCA Construction

- **KeyGen:**

$$(pk_0, sk_0), (pk_1, sk_1) \leftarrow G_{so}(1^\lambda), \ (\sigma_i, \tau_i) \leftarrow \mathrm{Ext}_1(1^\lambda) \text{ for } i \in L$$

$$pk = (pk_0, pk_1, \{\sigma_i\}_{i \in L}) \quad \text{and} \quad sk = (sk_0, sk_1, \{\tau_i\}_{i \in L}).$$

- **Encryption:**

$$r^{sig} \leftarrow \mathrm{coins}(\mathrm{Sign}), \ r_0, r_1 \leftarrow \mathrm{coins}(E), \ \{r_i^{nizk}\}_{i=1}^{\ell} \leftarrow \mathrm{coins}(\mathrm{Prover}).$$

$$(vk, sk) = \mathrm{G}(r^{sig}).$$

For a message m, calculate

$$e_0 = E(pk_0, m, r_0), \quad e_1 = E(pk_1, m, r_1)$$

set $w = (m, r_0, r_1)$.

$$\overline{\pi} = (\pi_1, \ldots, \pi_\ell) = (\mathrm{Prover}(\sigma_i, (e_0, e_1), w), r_i^{nizk})_{i \in \mathfrak{g}(vk)}$$

$$\mathrm{sig} = \mathrm{Sign}(e_0, e_1, \overline{\pi}),$$

output the ciphertext: $c = (vk, e_0, e_1, \overline{\pi}, \mathrm{sig})$.

Theorem

This construction is SEM-SO-CCA2 Secure

Our Results

Our Results

- ReRandomizable Encryption "is" Lossy Encryption

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
    - A framework for creating Lossy Encryption:

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption

Our Results

- ReRandomizable Encryption “is” Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model

Our Results

- ReRandomizable Encryption "is" Lossy Encryption
  - A framework for creating Lossy Encryption:
  - Applying the results of [BHY09] gives:
    - Goldwasser-Micali
    - El-Gamal
    - Paillier / Damgård-Jurik
  - The first proof that Paillier/Damgård-Jurik is SEM-SO-ENC secure.
    This is the most efficient known SEM-SO-ENC cryptosystem.
- Statistically Hiding-OT implies Lossy Encryption
  - PIR implies Lossy Encryption
  - Homomorphic Encryption implies Lossy Encryption
- CCA2 Selective Opening Secure definitions and constructions
  - Constructions from statistically-hiding NIZKs in the simulation-based model
  - Constructions from Lossy-Trapdoor Functions in the indistinguishability-based model

Open Questions

Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?

Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?
- Can we remove the dependence on n in the CCA constructions.

Open Questions

- Can we construct an IND-CPA secure system that is not IND-SO secure?
- Can we remove the dependence on n in the CCA constructions.
- What about receiver corruption?

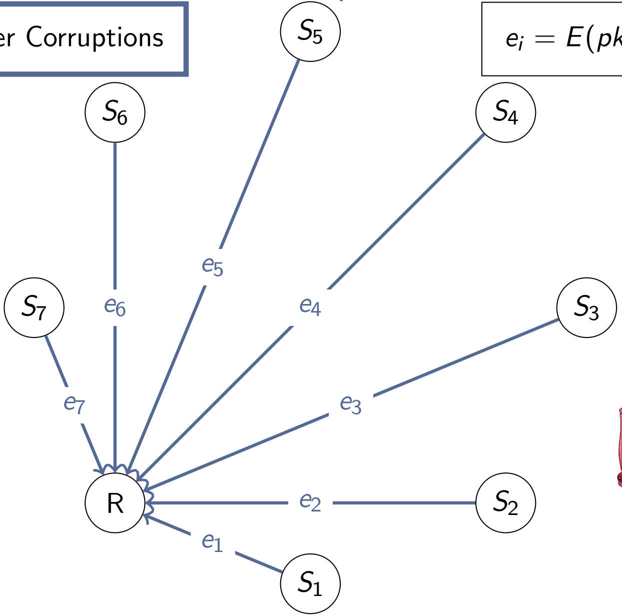
Open Question: Receiver Corruption

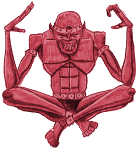
Recall: Sender Corruption Game

Open Question: Receiver Corruption

Sender Corruptions

$$e_i = E(pk, m_i, r_i)$$

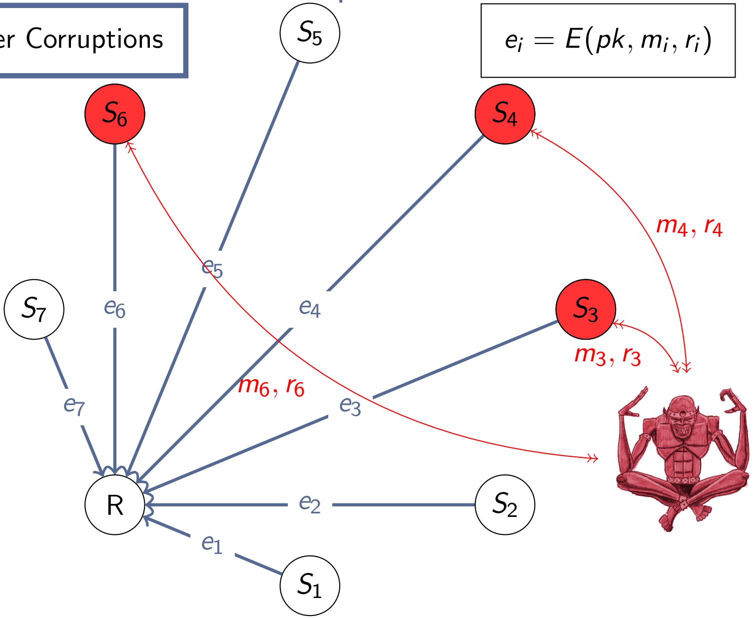




Open Question: Receiver Corruption

Sender Corruptions

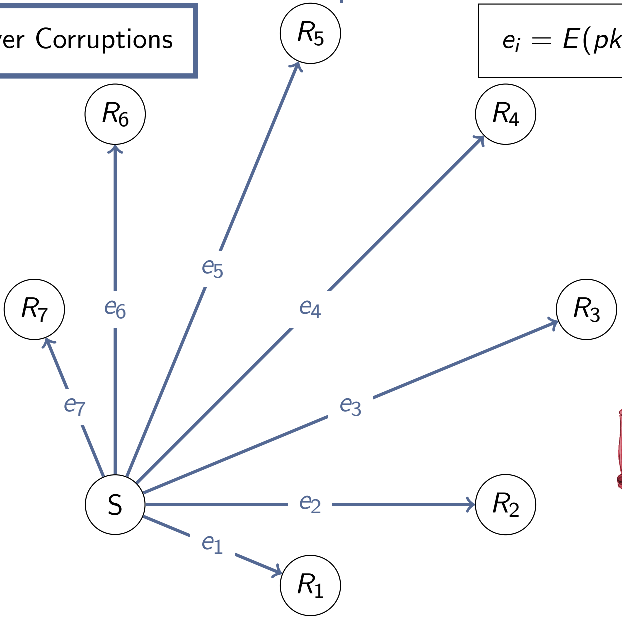
$$e_i = E(pk, m_i, r_i)$$

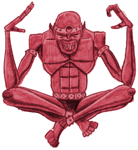


Open Question: Receiver Corruption

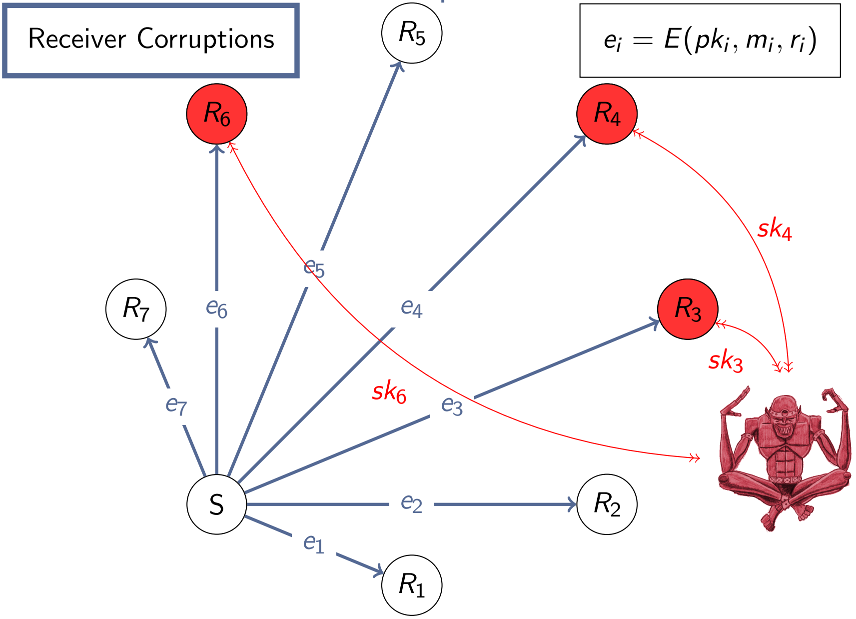
Receiver Corruptions

$$e_i = E(pk_i, m_i, r_i)$$

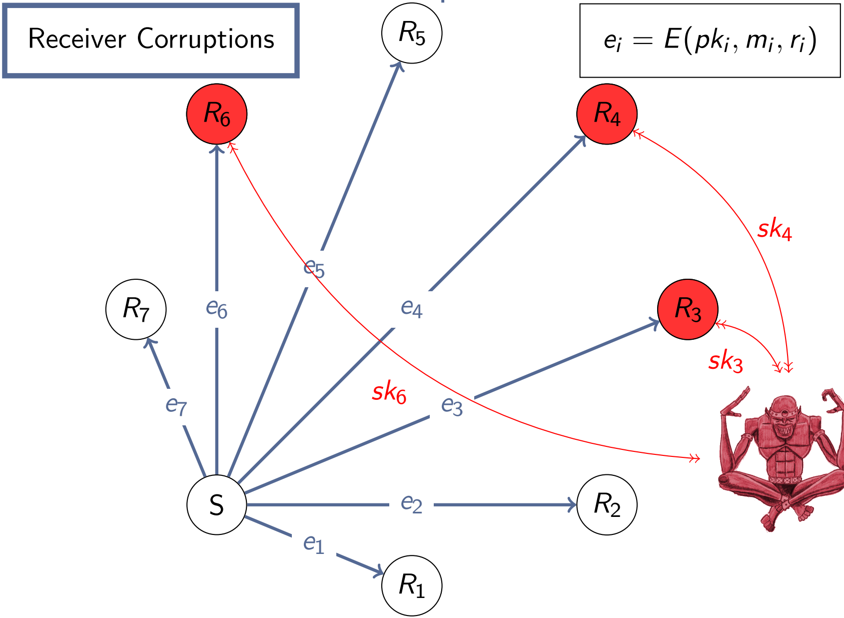




Open Question: Receiver Corruption



Open Question: Receiver Corruption



Thanks!