gemalto

Can RSA Keys be (Non-)Malleable?

Pascal Paillier

Gemplus/Gemalto, Cryptography & Innovation

IPAM Workshop I, October 10, 2006

Outline

1. Preliminaries
   - Impossibility Results for RSA-based Cryptography
   - RSA and Related Computational Problems
   - The Example of RSA Encryption
2. Instance-Malleability of RSA Key Generation
   - A Definition
   - The Fundamental Property
3. Impossibility Results for RSA Encryption
   - One-Wayness kills Chosen-Ciphertext Security
   - Indistinguishability kills Plaintext-Checking Security
   - IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]
4. Non-Malleability in Other Settings
5. Computational Number Theoretic Challenges

Impossibility Results for RSA Cryptography

We recently found evidence that

Impossibility Results for RSA Cryptography

We recently found evidence that

RSA Signatures

the unforgeability of RSA-based signatures such as FDH, PSS, PSS-R and many others cannot be equivalent to inverting RSA in the standard model.

Impossibility Results for RSA Cryptography

We recently found evidence that

RSA Signatures

the unforgeability of RSA-based signatures such as FDH, PSS, PSS-R and many others cannot be equivalent to inverting RSA in the standard model.

RSA Encryption

the CCA security of RSA-based encryption *e.g.* RSA-OAEP and similar schemes cannot be equivalent to inverting RSA in the standard model.

Impossibility Results for RSA Cryptography

Factoring-based cryptography:

Impossibility Results for RSA Cryptography

Factoring-based cryptography:

Factoring-Based Encryption and Signatures

- Rabin/RW-SAEP[+]/OAEP[+][+], EPOC-2, etc. are not equivalent to factoring
- Rabin signatures (+ padding) are not equivalent to factoring

Impossibility Results for RSA Cryptography

Factoring-based cryptography:

Factoring-Based Encryption and Signatures

- Rabin/RW-SAEP[+]/OAEP[+][+], EPOC-2, etc. are not equivalent to factoring
- Rabin signatures (+ padding) are not equivalent to factoring

These results contradict standard proofs

standing in the RO model

Impossibility Results for RSA Cryptography

These results explicitly assume:

Impossibility Results for RSA Cryptography

These results explicitly assume:

Single-Key Encryption/Signatures

The public key is formed of only one RSA modulus. Schemes using multiple keys (NY/DDN//GMR/DN) are immune.

Impossibility Results for RSA Cryptography

These results explicitly assume:

Single-Key Encryption/Signatures

The public key is formed of only one RSA modulus. Schemes using multiple keys (NY/DDN//GMR/DN) are immune.

Unkeyed Hash functions in Paddings

It is unclear whether replacing hash functions by families of hash functions really help, but it is likely.

Impossibility Results for RSA Cryptography

These results explicitly assume:

Single-Key Encryption/Signatures

The public key is formed of only one RSA modulus. Schemes using multiple keys (NY/DDN//GMR/DN) are immune.

Unkeyed Hash functions in Paddings

It is unclear whether replacing hash functions by families of hash functions really help, but it is likely.

Non-Malleable Key Generation

Non-malleability: Given $n \leftarrow \mathrm{Gen}(1^k)$, a factoring oracle $\mathrm{FACT}(n' \neq n)$ does not help to factor n

Instance Non-malleability: Given $(n, e) \leftarrow \mathrm{Gen}(1^k)$, a root extraction oracle $\mathrm{RSA}((n', e') \neq (n, e))$ does not help to extract an e-th root modulo n.

RSA Instance Generators

RSA Instance Generators

RSA

consists in computing $x = y^{1/e} \bmod n$ given random integers $n, e \leftarrow \mathbb{N}$ such that $\gcd(e, \phi(n)) = 1$ where $\phi(n) = |\mathbb{Z}_n^*|$ and $y \leftarrow \mathbb{Z}_n$.

RSA Instance Generators

RSA

consists in computing $x = y^{1/e} \bmod n$ given random integers $n, e \leftarrow \mathbb{N}$ such that $\gcd(e, \phi(n)) = 1$ where $\phi(n) = |\mathbb{Z}_n^*|$ and $y \leftarrow \mathbb{Z}_n$.

Hard Instances

We define an instance generator Gen. Given $k > 0$, $\mathrm{Gen}(1^k)$ generates a hard instance (n, e), as well as the side information $d = e^{-1} \bmod \phi(n)$.

RSA Instance Generators

RSA

consists in computing $x = y^{1/e} \bmod n$ given random integers $n, e \leftarrow \mathbb{N}$ such that $\gcd(e, \phi(n)) = 1$ where $\phi(n) = |\mathbb{Z}_n^*|$ and $y \leftarrow \mathbb{Z}_n$.

Hard Instances

We define an instance generator Gen. Given $k > 0$, $\mathrm{Gen}(1^k)$ generates a hard instance (n, e), as well as the side information $d = e^{-1} \bmod \phi(n)$.

(ε, τ)-Inverting Gen

A probabilistic algorithm $\mathcal{A}$ is said to (ε, τ)-invert Gen when

$$\Pr\left[(n, e, d) \leftarrow \mathrm{Gen}(1^k), y \leftarrow \mathbb{Z}_n \ : \ \mathcal{A}(n, e, y) = y^d \bmod n\right] \geq \varepsilon \ ,$$

where $\mathcal{A}$ halts after τ steps.

RSA Instance Generators

RSA

consists in computing $x = y^{1/e} \bmod n$ given random integers $n, e \leftarrow \mathbb{N}$ such that $\gcd(e, \phi(n)) = 1$ where $\phi(n) = |\mathbb{Z}_n^*|$ and $y \leftarrow \mathbb{Z}_n$.

Hard Instances

We define an instance generator Gen. Given $\mathrm{k} > 0$, $\mathrm{Gen}(1^{\mathrm{k}})$ generates a hard instance (n, e), as well as the side information $d = e^{-1} \bmod \phi(n)$.

(ε, τ)-Inverting Gen

A probabilistic algorithm $\mathcal{A}$ is said to (ε, τ)-invert Gen when

$$\Pr\left[(n, e, d) \leftarrow \mathrm{Gen}(1^{\mathrm{k}}), y \leftarrow \mathbb{Z}_n \ : \ \mathcal{A}(n, e, y) = y^d \bmod n\right] \geq \varepsilon \ ,$$

where $\mathcal{A}$ halts after τ steps.

Breaking INV [Gen] means (ε, τ)-inverting Gen for $\tau = \mathrm{poly}(\mathrm{k})$ and $\varepsilon > 1/\mathrm{poly}(\mathrm{k})$

The One-More RSA Problem

A natural generalization of INV [Gen]

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,
- output $y_0^d \bmod n, \ldots, y_\ell^d \bmod n$

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,
- output $y_0^d \bmod n, \ldots, y_\ell^d \bmod n$

Properties

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,
- output $y_0^d \bmod n, \ldots, y_\ell^d \bmod n$

Properties

- 0-OM [Gen] $\triangleq$ INV [Gen]

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^k)$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,
- output $y_0^d \bmod n, \ldots, y_\ell^d \bmod n$

Properties

- 0-OM [Gen] $\triangleq$ INV [Gen]
- ℓ_2-OM [Gen] $\Leftarrow$ ℓ_1-OM [Gen] if $\ell_2 \geq \ell_1$

The One-More RSA Problem

A natural generalization of INV [Gen]

ℓ-OM [Gen] for $\ell \geq 0$

- given $(n, e) \leftarrow \mathrm{Gen}(1^{\mathsf{k}})$,
- $\ell + 1$ integers $y_0, y_1, \ldots, y_\ell$ modulo n,
- oracle access to $\mathrm{RSA}(n, e)$ at most ℓ times,
- output $y_0^d \bmod n, \ldots, y_\ell^d \bmod n$

Properties

- 0-OM [Gen] $\triangleq$ INV [Gen]
- ℓ_2-OM [Gen] $\Leftarrow$ ℓ_1-OM [Gen] if $\ell_2 \geq \ell_1$

OM [Gen] means

(ε, τ)-breaking ℓ-OM [Gen] for $\ell, \tau = \mathrm{poly}(\mathsf{k})$ and $\varepsilon > 1/\mathrm{poly}(\mathsf{k})$.

Relations among RSA-Related Problems

Relations among RSA-Related Problems

FACT [Gen]

compute d from $(n, e, d) \leftarrow \mathrm{Gen}(1^k)$.

Relations among RSA-Related Problems

FACT [Gen]

compute d from $(n, e, d) \leftarrow \mathrm{Gen}(1^k)$.

GAP [Gen]

compute d from (n, e) for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ given unlimited access to $\mathrm{RSA}(n, e, \cdot)$.

Relations among RSA-Related Problems

FACT [Gen]

compute d from $(n, e, d) \leftarrow \mathrm{Gen}(1^k)$.

GAP [Gen]

compute d from (n, e) for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ given unlimited access to $\mathrm{RSA}(n, e, \cdot)$.

Relations among RSA-related problems

For any instance generator Gen, one has

$$\begin{array}{ccc} \mathrm{GAP}\,[\mathrm{Gen}] & \Leftarrow & \mathrm{FACT}\,[\mathrm{Gen}] \\ \Downarrow & & \Downarrow \\ \mathrm{OM}\,[\mathrm{Gen}] & \Leftarrow & \mathrm{INV}\,[\mathrm{Gen}] \end{array}$$

The Example of RSA Encryption

The Example of RSA Encryption

An RSA-based encryption scheme $\mathcal{E}$

combination of an instance generator Gen with a padding function μ

The Example of RSA Encryption

An RSA-based encryption scheme $\mathcal{E}$

combination of an instance generator Gen with a padding function μ

Padding functions

$$\begin{array}{rcl} \mu : \{0,1\}^{|\mathsf{m}|} \times \{0,1\}^{|\mathsf{r}|} & \rightarrow & \{0,1\}^{\mathsf{k}} \\ (m, r) & \rightarrow & \mu(m, r) \end{array}$$

where $|\mathsf{m}| \geq 1$ and $|\mathsf{r}| \geq 0$

The Example of RSA Encryption

An RSA-based encryption scheme $\mathcal{E}$

combination of an instance generator Gen with a padding function μ

Padding functions

$$\begin{array}{rcl} \mu : \{0,1\}^{|\mathsf{m}|} \times \{0,1\}^{|\mathsf{r}|} & \rightarrow & \{0,1\}^{\mathsf{k}} \\ (m, r) & \rightarrow & \mu(m, r) \end{array}$$

where $|\mathsf{m}| \geq 1$ and $|\mathsf{r}| \geq 0$

μ is invertible

there is a function μ^{-1} such that $\forall\, m \in \{0,1\}^{|\mathsf{m}|}, \forall\, y \in \{0,1\}^{\mathsf{k}}$,

$$\begin{array}{rcl} \mu^{-1}(y) & = & m \quad \text{iff} \quad \exists\, r \in \{0,1\}^{|\mathsf{r}|} \quad \text{s.t.} \quad y = \mu(m, r) \\ & = & \perp \quad \text{otherwise} \end{array}$$

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext
- *root extractable* (RE) when the adversary extracts an e-th root modulo n

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext
- *root extractable* (RE) when the adversary extracts an e-th root modulo n

Attack models, where the adversary is given

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext
- *root extractable* (RE) when the adversary extracts an e-th root modulo n

Attack models, where the adversary is given

- *chosen-plaintext attack* (CPA), nothing

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext
- *root extractable* (RE) when the adversary extracts an e-th root modulo n

Attack models, where the adversary is given

- *chosen-plaintext attack* (CPA), nothing
- *plaintext-checking attack* (PCA), a plaintext-checking oracle

Security Notions for RSA Encryption

Adversarial goals: $\mathcal{E}$ is

- *breakable* (BK) when the adversary extracts the secret key
- *one-way* (OW) when the adversary $\mathcal{A}$ returns the plaintext matching a given ciphertext
- *indistinguishable* (IND) when the adversary $\mathcal{A}$ decides whether a given ciphertext encrypts a given plaintext
- *root extractable* (RE) when the adversary extracts an e-th root modulo n

Attack models, where the adversary is given

- *chosen-plaintext attack* (CPA), nothing
- *plaintext-checking attack* (PCA), a plaintext-checking oracle
- *chosen-ciphertext attack* (CCA), a decryption oracle

Relations among Security Notions

Relations among Security Notions

$$
\begin{array}{ccccc}
\text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}]
\end{array}
$$

Relations among Security Notions

$$
\begin{array}{ccccc}
\text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \equiv \text{FACT}\,[\text{Gen}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \\
\Downarrow & & \Downarrow & & \Downarrow \\
\text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}]
\end{array}
$$

Relations among Security Notions

$$
\begin{array}{ccccccc}
\text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] & \equiv & \text{FACT}\,[\text{Gen}] \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] & \equiv & \text{INV}\,[\text{Gen}] \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] & & \\
\Downarrow & & \Downarrow & & \Downarrow & & \\
\text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}] & &
\end{array}
$$

Relations among Security Notions

$$
\begin{array}{ccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \equiv \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}]
\end{array}
$$

Relations among Security Notions

$$
\begin{array}{ccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \equiv \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}]
\end{array}
$$

Our results

Relations among Security Notions

$$
\begin{array}{ccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \equiv \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]} \\
 & & \Downarrow & & \Downarrow & & \equiv \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}]
\end{array}
$$

Our results

1. if OW-CPA $[\mathcal{E}] \equiv$ INV [Gen]

Relations among Security Notions

$$
\begin{array}{ccccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] & \equiv & \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow & & \\
0 & \equiv & \text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] & \equiv & \text{INV [Gen]} \\
 & & \equiv & & \Downarrow & & \equiv & & \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] & \equiv & \text{INV [Gen]} \\
 & & \equiv & & \Downarrow & & \Downarrow & & \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}] & &
\end{array}
$$

Our results

1. if OW-CPA $[\mathcal{E}] \equiv$ INV [Gen] then RE-CCA $[\mathcal{E}]$ is polynomial

Relations among Security Notions

$$
\begin{array}{ccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] \equiv \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow \\
 & & \text{RE-CCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]} \\
 & & \Downarrow & & \Downarrow & & \equiv \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \Leftarrow & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] \\
 & & \Downarrow & & \Downarrow & & \equiv \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}] \equiv \text{INV [Gen]}
\end{array}
$$

Our results

1. if OW-CPA $[\mathcal{E}] \equiv$ INV [Gen] then RE-CCA $[\mathcal{E}]$ is polynomial
2. if IND-CPA $[\mathcal{E}] \equiv$ INV [Gen]

Relations among Security Notions

$$
\begin{array}{ccccccccc}
\text{GAP [Gen]} & \Leftarrow & \text{BK-CCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-PCA}\,[\mathcal{E}] & \Leftarrow & \text{BK-CPA}\,[\mathcal{E}] & \equiv & \text{FACT [Gen]} \\
 & & \Downarrow & & \Downarrow & & \Downarrow & & \\
0 & \equiv & \text{RE-CCA}\,[\mathcal{E}] & \equiv & \text{RE-PCA}\,[\mathcal{E}] & \Leftarrow & \text{RE-CPA}\,[\mathcal{E}] & \equiv & \text{INV [Gen]} \\
 & & \equiv & & \equiv & & \equiv & & \\
 & & \text{OW-CCA}\,[\mathcal{S}] & \equiv & \text{OW-PCA}\,[\mathcal{E}] & \Leftarrow & \text{OW-CPA}\,[\mathcal{E}] & & \\
 & & \equiv & & \equiv & & \equiv & & \\
 & & \text{IND-CCA}\,[\mathcal{E}] & \equiv & \text{IND-PCA}\,[\mathcal{E}] & \Leftarrow & \text{IND-CPA}\,[\mathcal{E}] & \equiv & \text{INV [Gen]}
\end{array}
$$

Our results

1. if OW-CPA $[\mathcal{E}] \equiv$ INV [Gen] then RE-CCA $[\mathcal{E}]$ is polynomial
2. if IND-CPA $[\mathcal{E}] \equiv$ INV [Gen] then RE-PCA $[\mathcal{E}]$ is polynomial

Instance-Malleability of Gen: Intuition

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

$$\mathrm{RSA}(n, e_1 e_2) \Leftarrow \mathrm{RSA}(n, e_1) + \mathrm{RSA}(n, e_2)$$

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

$$\mathrm{RSA}(n, e_1 e_2) \Leftarrow \mathrm{RSA}(n, e_1) + \mathrm{RSA}(n, e_2)$$

- $|n|$ and $\sum_{p|n} 1$ vary

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

$$\mathrm{RSA}(n, e_1 e_2) \Leftarrow \mathrm{RSA}(n, e_1) + \mathrm{RSA}(n, e_2)$$

- $|n|$ and $\sum_{p|n} 1$ vary

$$\mathrm{RSA}(n, e) \Leftarrow \mathrm{RSA}(\alpha n, e)$$

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

$$\mathrm{RSA}(n, e_1 e_2) \Leftarrow \mathrm{RSA}(n, e_1) + \mathrm{RSA}(n, e_2)$$

- $|n|$ and $\sum_{p|n} 1$ vary

$$\mathrm{RSA}(n, e) \Leftarrow \mathrm{RSA}(\alpha n, e)$$

- Combination of the above

Instance-Malleability of Gen: Intuition

We say that Gen is instance-malleable if

solving RSA for $(n, e) \leftarrow \mathrm{Gen}(1^k)$ is easier when given unlimited access to $\mathrm{RSA}(n', e', \cdot)$ for $(n', e') \neq (n, e)$ and $(n', e') \in \mathrm{Range}\left(\mathrm{Gen}(1^k)\right)$

Examples

- public exponents are easy-to-factor integers

$$\mathrm{RSA}(n, e_1 e_2) \Leftarrow \mathrm{RSA}(n, e_1) + \mathrm{RSA}(n, e_2)$$

- $|n|$ and $\sum_{p|n} 1$ vary

$$\mathrm{RSA}(n, e) \Leftarrow \mathrm{RSA}(\alpha n, e)$$

- Combination of the above

Can you see non-trivial examples?

Is Instance-Malleability for Real?

Is Instance-Malleability for Real?

Most instance generators tend to avoid instance-malleability

Is Instance-Malleability for Real?

Most instance generators tend to avoid instance-malleability

Plain-RSA.

$p, q \leftarrow \text{Primes}(\lceil k/2 \rceil)$
$e \leftarrow \text{Primes}(k)$
Set $n = pq$ and $d = e^{-1} \bmod \phi(n)$

Is Instance-Malleability for Real?

Most instance generators tend to avoid instance-malleability

Plain-RSA.

$p, q \leftarrow \text{Primes}(\lceil k/2 \rceil)$
$e \leftarrow \text{Primes}(k)$
Set $n = pq$ and $d = e^{-1} \bmod \phi(n)$

Low-exponent-RSA.

$e = 3$ or $F_4 = 2^{16} + 1$
$p, q \leftarrow \text{Primes}(\lceil k/2 \rceil)$
must have $\gcd(e, p-1) = \gcd(e, q-1) = 1$
Set $n = pq$ and $d = e^{-1} \bmod \phi(n)$.

Is Instance-Malleability for Real?

Most instance generators tend to avoid instance-malleability

Plain-RSA.

$p, q \leftarrow \text{Primes}(\lceil k/2 \rceil)$
$e \leftarrow \text{Primes}(k)$
Set $n = pq$ and $d = e^{-1} \bmod \phi(n)$

Low-exponent-RSA.

$e = 3$ or $F_4 = 2^{16} + 1$
$p, q \leftarrow \text{Primes}(\lceil k/2 \rceil)$
must have $\gcd(e, p-1) = \gcd(e, q-1) = 1$
Set $n = pq$ and $d = e^{-1} \bmod \phi(n)$.

Do these RSA generators feature instance-non-malleability in a strong sense?

Defining Instance-Malleability

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$
- $\mathcal{A} \Leftarrow \mathsf{INV}\,[\mathsf{Gen}]$ and can be $\equiv 0$

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$
- $\mathcal{A} \Leftarrow \mathsf{INV}\,[\mathsf{Gen}]$ and can be $\equiv 0$

Game 0

$$\begin{array}{c}(n, e, d) \leftarrow \mathsf{Gen}(1^{\mathsf{k}}) \\ y \leftarrow \mathbb{Z}_n\end{array} \quad : \quad \mathcal{R}^{\mathcal{A}(n,e,\cdot)}(n, e, y) = y^d \bmod n$$

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$
- $\mathcal{A} \Leftarrow \mathsf{INV}\,[\mathsf{Gen}]$ and can be $\equiv 0$

Game 0

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} \;:\; \mathcal{R}^{\mathcal{A}(n,e,\cdot)}(n,e,y) = y^d \bmod n\right]$$

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$
- $\mathcal{A} \Leftarrow \mathsf{INV}\,[\mathsf{Gen}]$ and can be $\equiv 0$

Game 0

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} : \mathcal{R}^{\mathcal{A}(n,e,\cdot)}(n,e,y) = y^d \bmod n\right]$$

$$\| $$

$$\mathsf{Succ}^{\text{Game 0}}(\mathcal{R}, \mathcal{A}, \tau, q_{\mathcal{A}})$$

Defining Instance-Malleability

Let $\mathcal{R}$ be a root extractor

- $\mathcal{R}$ attempts to compute an e-th root modulo n
- computational resources of $\mathcal{R}$ = some perfect oracle $\mathcal{A}(n, e, \cdot)$
- $\mathcal{A} \Leftarrow \mathsf{INV}\,[\mathsf{Gen}]$ and can be $\equiv 0$

Game 0

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} : \mathcal{R}^{\mathcal{A}(n,e,\cdot)}(n,e,y) = y^d \bmod n\right]$$

$$\| $$

$$\mathsf{Succ}^{\text{Game 0}}(\mathcal{R}, \mathcal{A}, \tau, q_{\mathcal{A}})$$

We define

$$\mathsf{Succ}^{\text{Game 0}}(\mathcal{A}, \tau, q_{\mathcal{A}}) = \max_{\mathcal{R}} \mathsf{Succ}^{\text{Game 0}}(\mathcal{R}, \mathcal{A}, \tau, q_{\mathcal{A}})$$

Defining Instance-Malleability

Defining Instance-Malleability

Game 1 = Game 0 + RSA$(n', e', \cdot)$ for $(n', e') \neq (n, e)$

Defining Instance-Malleability

Game 1 = Game 0 + RSA$(n', e', \cdot)$ for $(n', e') \neq (n, e)$

$$(n, e, d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n \quad : \quad \mathcal{R}^{\mathcal{A}(n,e,\cdot),\mathsf{RSA}(n',e',\cdot)}(n, e, y) = y^d \bmod n$$

Defining Instance-Malleability

Game 1 = Game 0 + RSA$(n', e', \cdot)$ for $(n', e') \neq (n, e)$

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} : \mathcal{R}^{\mathcal{A}(n,e,\cdot),\mathsf{RSA}(n',e',\cdot)}(n,e,y) = y^d \bmod n\right]$$

Defining Instance-Malleability

Game 1 = Game 0 + RSA$(n', e', \cdot)$ for $(n', e') \neq (n, e)$

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} : \mathcal{R}^{\mathcal{A}(n,e,\cdot),\mathsf{RSA}(n',e',\cdot)}(n,e,y) = y^d \bmod n\right]$$

$$\| $$

$$\mathsf{Succ}^{\text{Game 1}}(\mathcal{R}, \mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

Defining Instance-Malleability

Game 1 = Game 0 + RSA$(n', e', \cdot)$ for $(n', e') \neq (n, e)$

$$\Pr\left[\begin{array}{c}(n,e,d) \leftarrow \mathsf{Gen}(1^k) \\ y \leftarrow \mathbb{Z}_n\end{array} : \mathcal{R}^{\mathcal{A}(n,e,\cdot),\mathsf{RSA}(n',e',\cdot)}(n,e,y) = y^d \bmod n\right]$$

$$\| $$

$$\mathsf{Succ}^{\text{Game 1}}(\mathcal{R},\mathcal{A},\tau,q_{\mathcal{A}},q_{\mathsf{RSA}})$$

We define

$$\mathsf{Succ}^{\text{Game 1}}(\mathcal{A},\tau,q_{\mathcal{A}},q_{\mathsf{RSA}}) = \max_{\mathcal{R}} \mathsf{Succ}^{\text{Game 1}}(\mathcal{R},\mathcal{A},\tau,q_{\mathcal{A}},q_{\mathsf{RSA}})$$

Defining Instance-Malleability

Defining Instance-Malleability

Distance from perfect non-malleability

Defining Instance-Malleability

Distance from perfect non-malleability

$$\left|\mathsf{Succ}^{\mathrm{Game\ 1}}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game\ 0}}(\mathcal{A}, \tau, q_{\mathcal{A}})\right|$$

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV}[\mathsf{Gen}]} \left| \mathsf{Succ}^{\mathrm{Game}\ 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game}\ 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV}[\mathsf{Gen}]} \left| \mathsf{Succ}^{\mathrm{Game}\ 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game}\ 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \quad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV}[\mathsf{Gen}]} \left| \mathsf{Succ}^{\mathrm{Game}\ 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game}\ 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \quad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

We have

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV[Gen]}} \left| \mathsf{Succ}^{\mathrm{Game\ 1}}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game\ 0}}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \quad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

We have

- Game 1 = Game 0 when $q_{\mathsf{RSA}} = 0$

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV}[\mathsf{Gen}]} \left| \mathsf{Succ}^{\text{Game } 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\text{Game } 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \qquad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

We have

- Game 1 = Game 0 when $q_{\mathsf{RSA}} = 0$

$$\Delta(\tau, q_{\mathcal{A}}, 0) = 0$$

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV}[\mathsf{Gen}]} \left| \mathsf{Succ}^{\text{Game } 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\text{Game } 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \quad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

We have

- Game 1 = Game 0 when $q_{\mathsf{RSA}} = 0$

$$\Delta(\tau, q_{\mathcal{A}}, 0) = 0$$

- Gen is **instance-non-malleable** if

Defining Instance-Malleability

Distance from perfect non-malleability

$$\max_{\mathcal{A} \Leftarrow \mathsf{INV[Gen]}} \left| \mathsf{Succ}^{\mathrm{Game}\ 1}(\mathcal{A}, \tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) - \mathsf{Succ}^{\mathrm{Game}\ 0}(\mathcal{A}, \tau, q_{\mathcal{A}}) \right|$$

$$\| \quad \Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}})$$

We have

- Game 1 = Game 0 when $q_{\mathsf{RSA}} = 0$

$$\Delta(\tau, q_{\mathcal{A}}, 0) = 0$$

- Gen is **instance-non-malleable** if

$$\Delta(\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}}) = \mathsf{negl}(k)$$

when $\tau, q_{\mathcal{A}}, q_{\mathsf{RSA}} = \mathsf{poly}(k)$

Properties of Instance-Non-Malleability

Properties of Instance-Non-Malleability

Allows to relate
(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2$ = solve an instance of P_1 with an oracle solving any instance of P_2

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

$\mathcal{P}$ is said to be $\tau_{\mathcal{P}}$-perfect if

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

$\mathcal{P}$ is said to be $\tau_{\mathcal{P}}$-perfect if

- it runs in at most $\tau_{\mathcal{P}}$ steps

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

$\mathcal{P}$ is said to be $\tau_{\mathcal{P}}$-perfect if

- it runs in at most $\tau_{\mathcal{P}}$ steps
- makes at most 1 call to RSA$(n, e, \cdot)$

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

$\mathcal{P}$ is said to be $\tau_{\mathcal{P}}$-perfect if

- it runs in at most $\tau_{\mathcal{P}}$ steps
- makes at most 1 call to RSA$(n, e, \cdot)$
- to solve $\mathcal{A}(n, e, \text{aux})$ with probability 1

Properties of Instance-Non-Malleability

Allows to relate

(unconstrained) reduction problems INV [Gen] $\Leftarrow \mathcal{A}$ to solving Game 1

(Unconstrained) reduction problems

$P_1 \Leftarrow P_2 =$ solve an instance of P_1 with an oracle solving any instance of P_2

But $\mathcal{A}$ must be perfectly reducible to INV [Gen]

$\mathcal{P}$ is said to be $\tau_{\mathcal{P}}$-perfect if

- it runs in at most $\tau_{\mathcal{P}}$ steps
- makes at most 1 call to RSA$(n, e, \cdot)$
- to solve $\mathcal{A}(n, e, \text{aux})$ with probability 1, $\forall$ aux

Properties of Instance-Non-Malleability

A Fundamental Lemma

Properties of Instance-Non-Malleability

A Fundamental Lemma

Let $\mathcal{A}$ be a computational problem with a $\tau_{\mathcal{P}}$-perfect reduction $\mathcal{P}$ to INV [Gen].

Properties of Instance-Non-Malleability

A Fundamental Lemma

Let $\mathcal{A}$ be a computational problem with a $\tau_{\mathcal{P}}$-perfect reduction $\mathcal{P}$ to INV [Gen]. Then for any positive integers $\tau, q_{\mathcal{A}}$,

Properties of Instance-Non-Malleability

A Fundamental Lemma

Let $\mathcal{A}$ be a computational problem with a $\tau_{\mathcal{P}}$-perfect reduction $\mathcal{P}$ to INV [Gen]. Then for any positive integers $\tau, q_{\mathcal{A}}$,

$$\mathsf{Succ}^{\text{Game 0}}(\mathcal{A}, \tau, q_{\mathcal{A}}) \leq \mathsf{Succ}(\mathsf{INV}\,[\mathsf{Gen}] \Leftarrow \mathcal{A}, \tau, q_{\mathcal{A}})$$

Properties of Instance-Non-Malleability

A Fundamental Lemma

Let $\mathcal{A}$ be a computational problem with a $\tau_{\mathcal{P}}$-perfect reduction $\mathcal{P}$ to INV [Gen]. Then for any positive integers $\tau, q_{\mathcal{A}}$,

$$\begin{gathered}\mathsf{Succ}^{\text{Game 0}}(\mathcal{A}, \tau, q_{\mathcal{A}}) \\ \leq \\ \mathsf{Succ}\left(\mathsf{INV}\left[\mathsf{Gen}\right] \Leftarrow \mathcal{A}, \tau, q_{\mathcal{A}}\right) \\ \leq\end{gathered}$$

Properties of Instance-Non-Malleability

A Fundamental Lemma

Let $\mathcal{A}$ be a computational problem with a $\tau_{\mathcal{P}}$-perfect reduction $\mathcal{P}$ to INV [Gen]. Then for any positive integers $\tau, q_{\mathcal{A}}$,

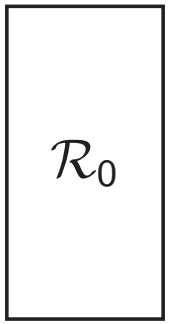
$$\begin{gathered}\mathsf{Succ}^{\text{Game 0}}(\mathcal{A}, \tau, q_{\mathcal{A}}) \\ \leq \\ \mathsf{Succ}\left(\mathsf{INV}\left[\mathsf{Gen}\right] \Leftarrow \mathcal{A}, \tau, q_{\mathcal{A}}\right) \\ \leq \\ \mathsf{Succ}^{\text{Game 1}}(\mathcal{A}, \tau + q_{\mathcal{A}} \cdot \tau_{\mathcal{P}}, q_{\mathcal{A}}, q_{\mathcal{A}}) \ .\end{gathered}$$

Proof (Sketchy and Visual)

Skip proof

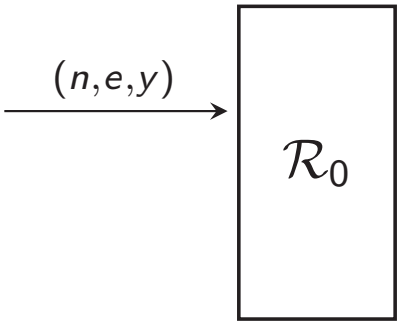
Proof (Sketchy and Visual)

Skip proof



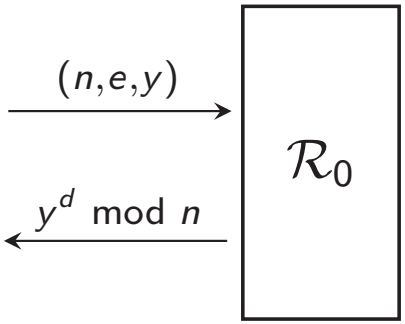
Proof (Sketchy and Visual)

Skip proof



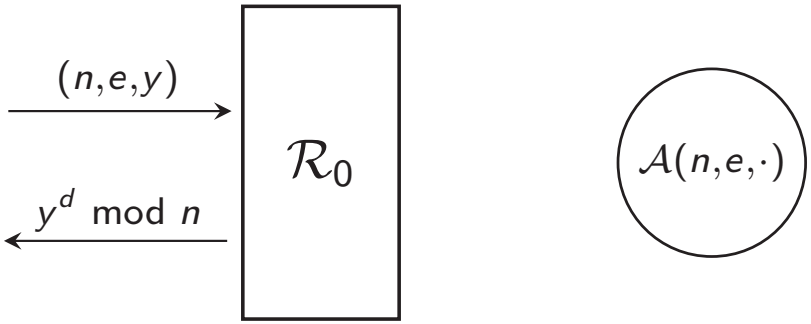
Proof (Sketchy and Visual)

Skip proof



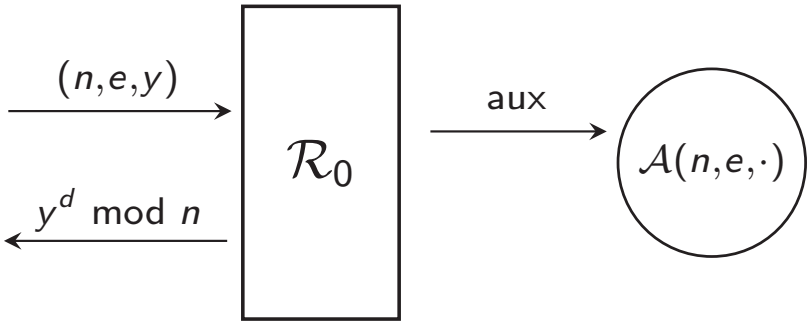
Proof (Sketchy and Visual)

Skip proof



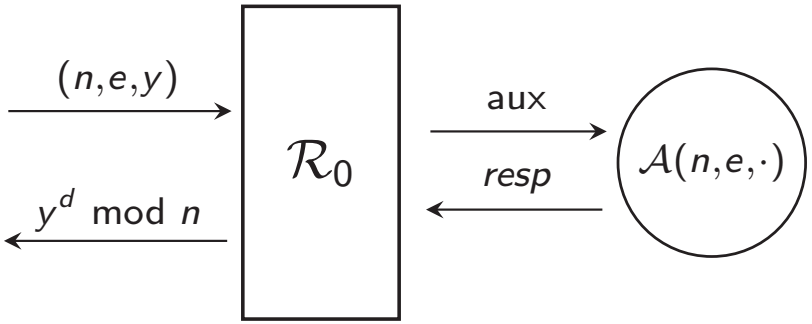
Proof (Sketchy and Visual)

Skip proof



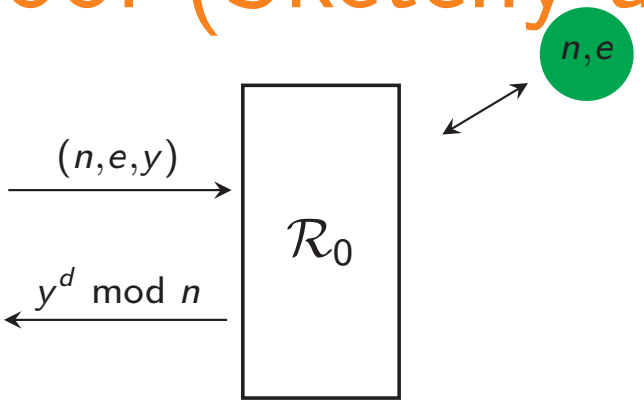
Proof (Sketchy and Visual)

Skip proof



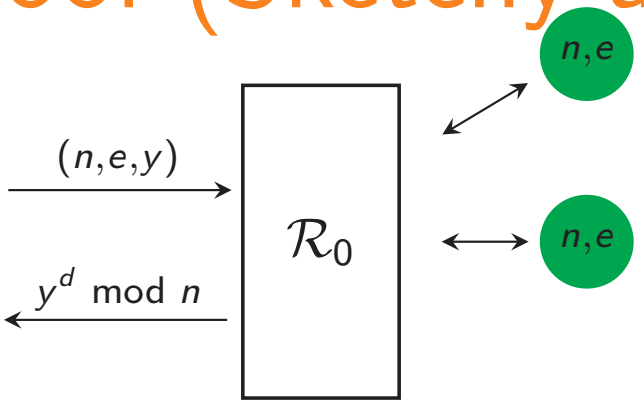
Proof (Sketchy and Visual)

Skip proof



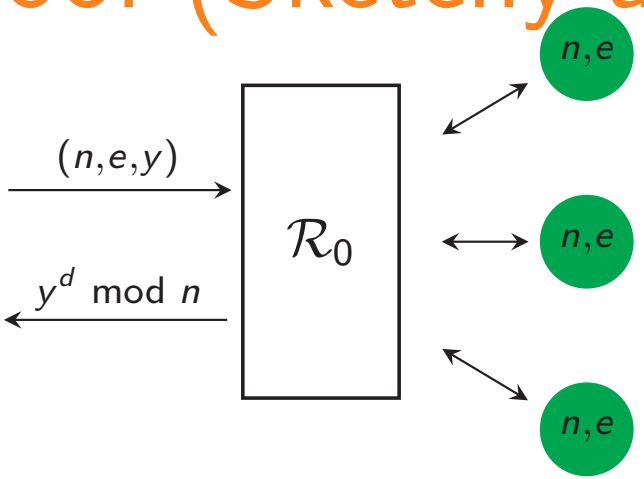
Proof (Sketchy and Visual)

Skip proof



Proof (Sketchy and Visual)

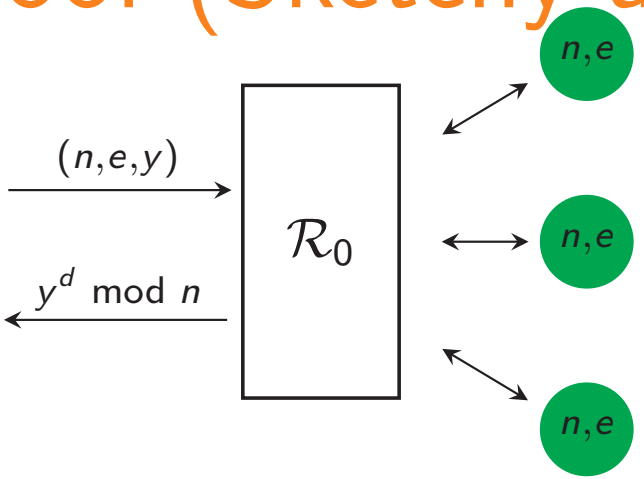
Skip proof



Proof (Sketchy and Visual)

Skip proof

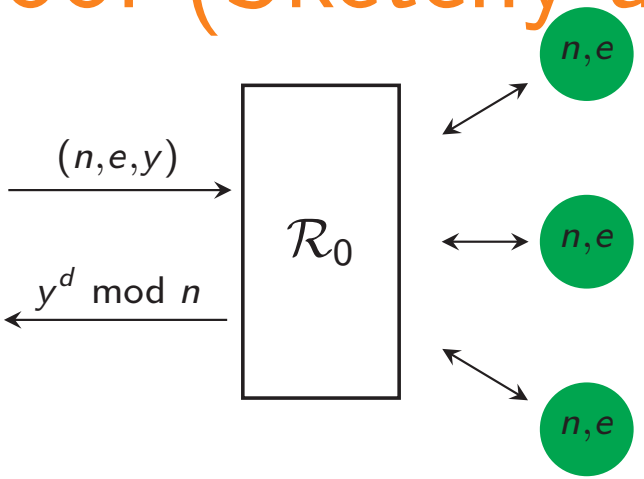
GAME 0



Proof (Sketchy and Visual)

Skip proof

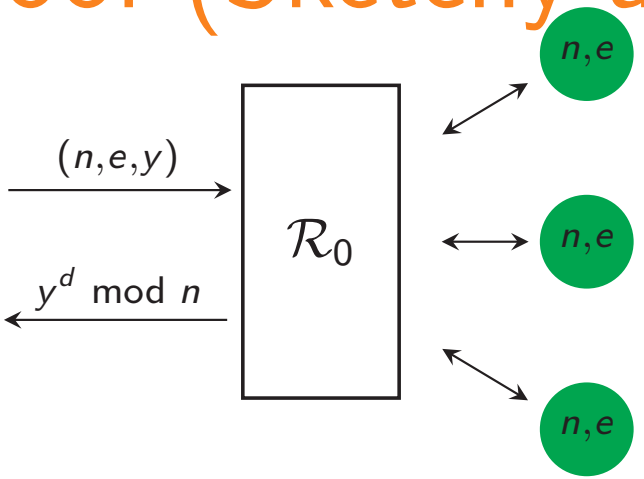
GAME 0

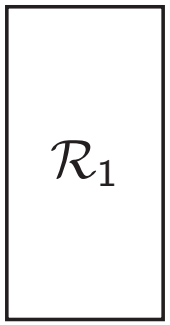


Proof (Sketchy and Visual)

Skip proof

GAME 0

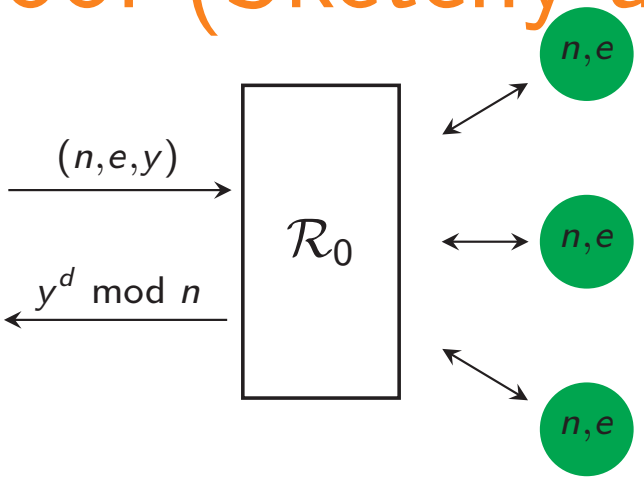


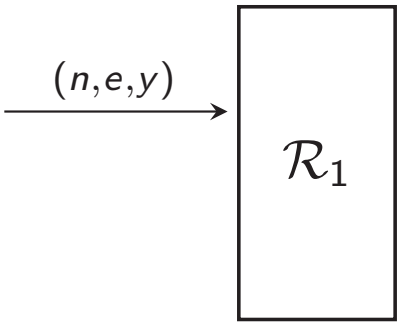


Proof (Sketchy and Visual)

Skip proof

GAME 0

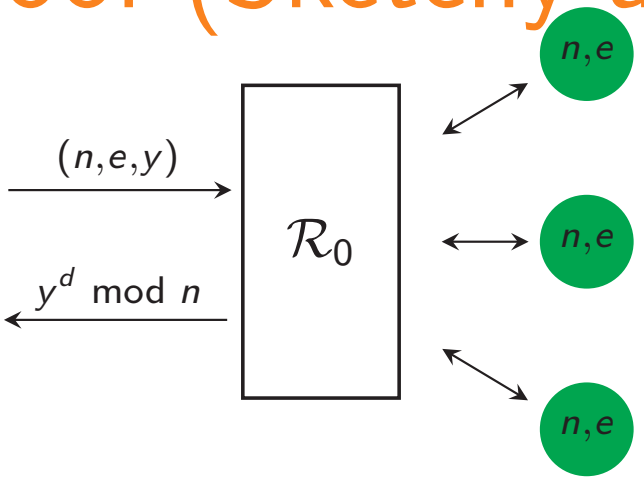


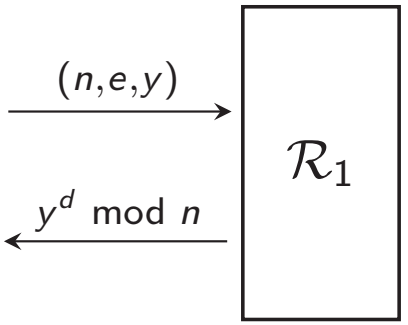


Proof (Sketchy and Visual)

Skip proof

GAME 0

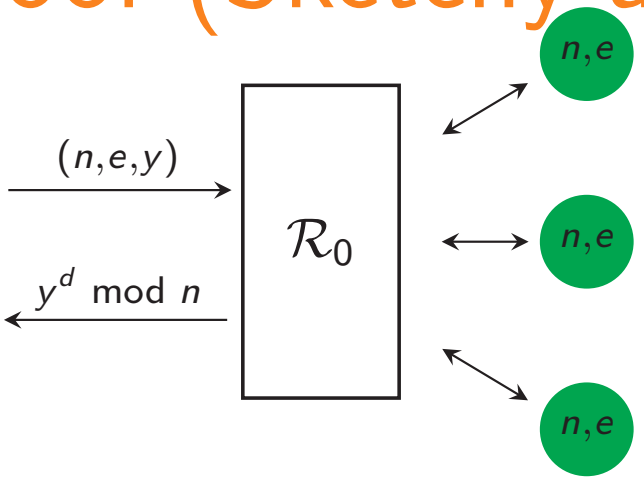


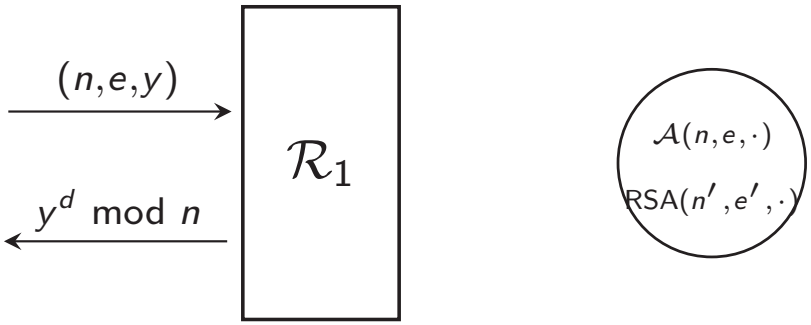


Proof (Sketchy and Visual)

Skip proof

GAME 0

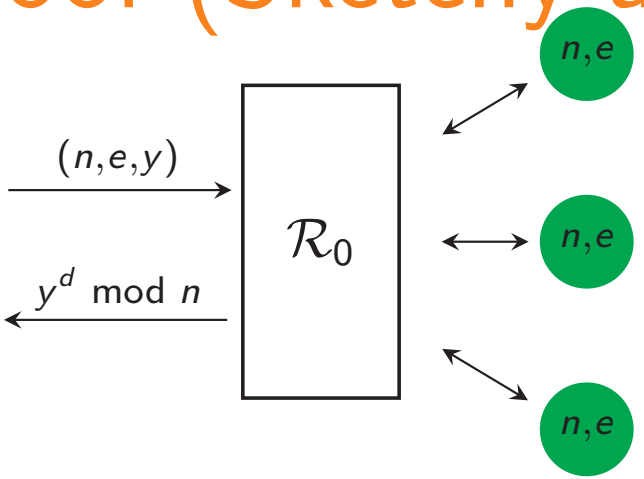


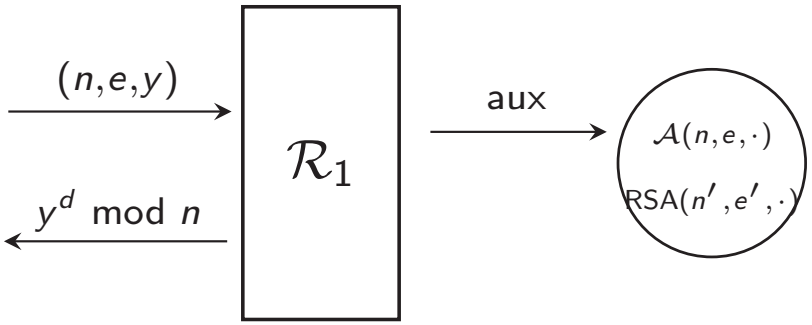


Proof (Sketchy and Visual)

Skip proof

GAME 0

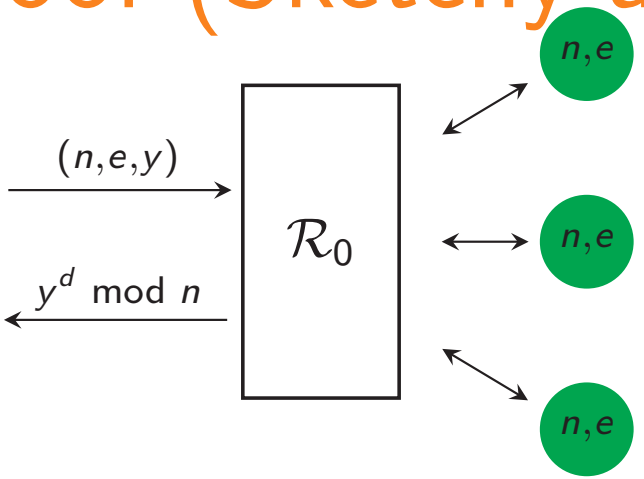


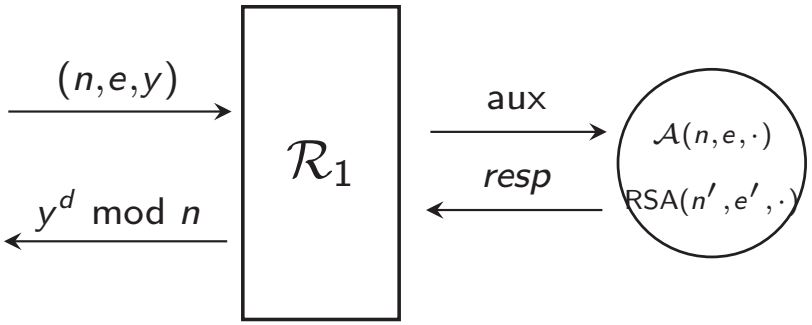


Proof (Sketchy and Visual)

Skip proof

GAME 0

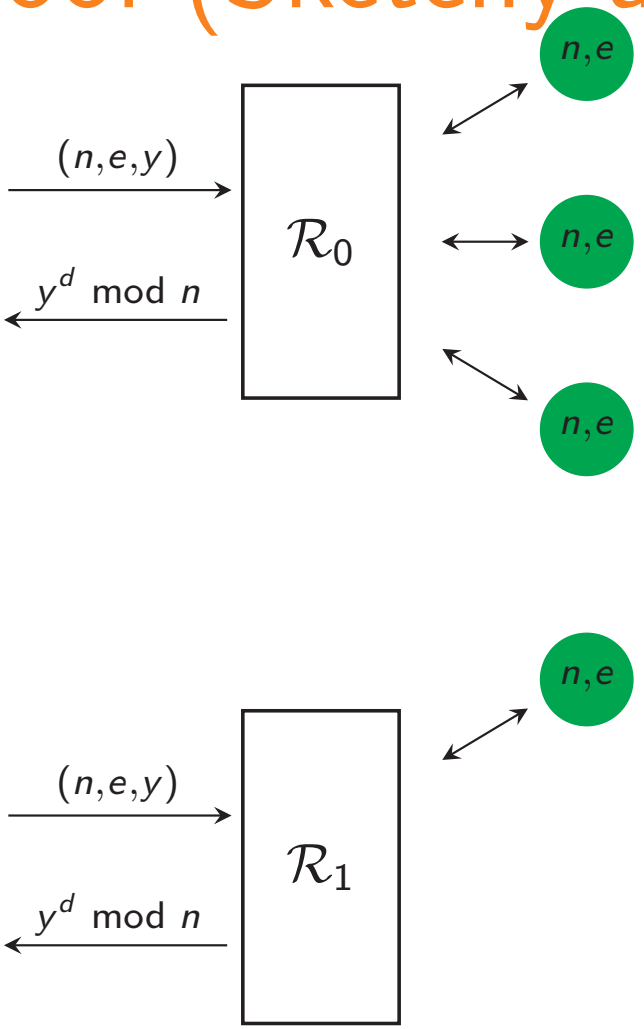




Proof (Sketchy and Visual)

Skip proof

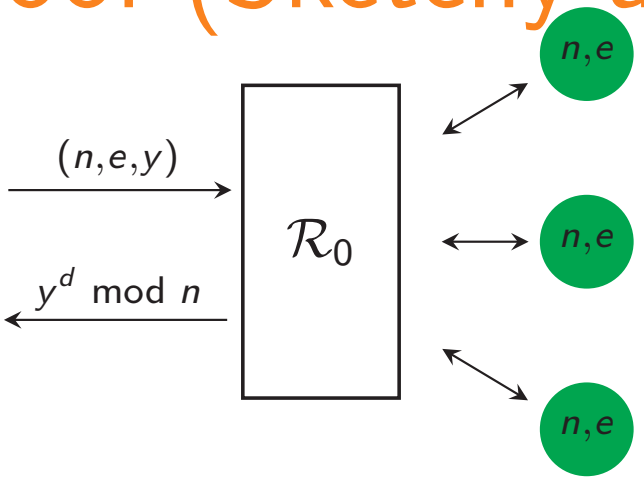
GAME 0

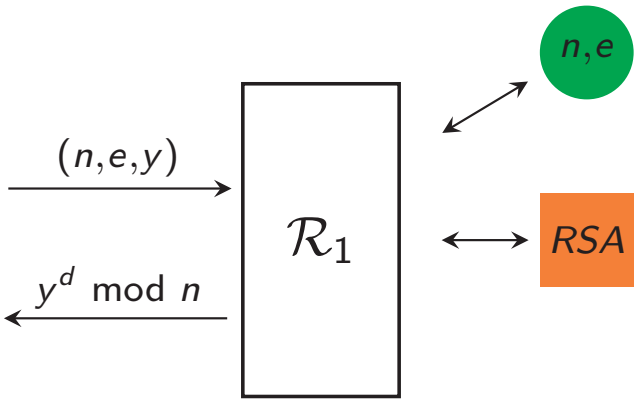


Proof (Sketchy and Visual)

Skip proof

GAME 0

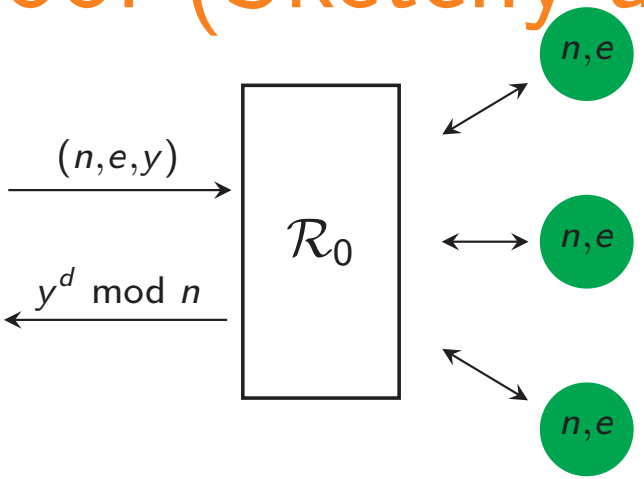


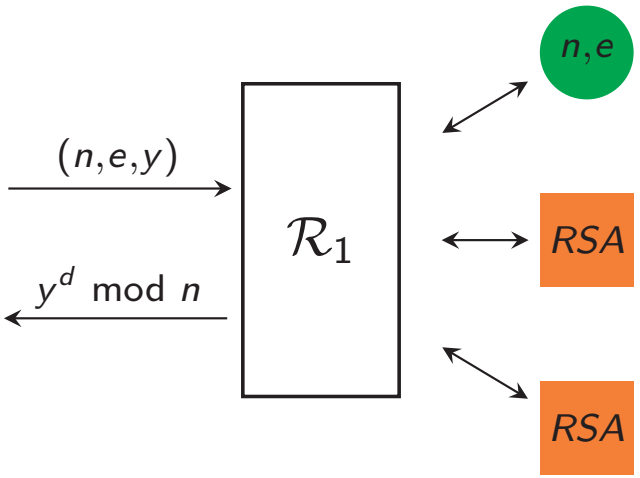


Proof (Sketchy and Visual)

Skip proof

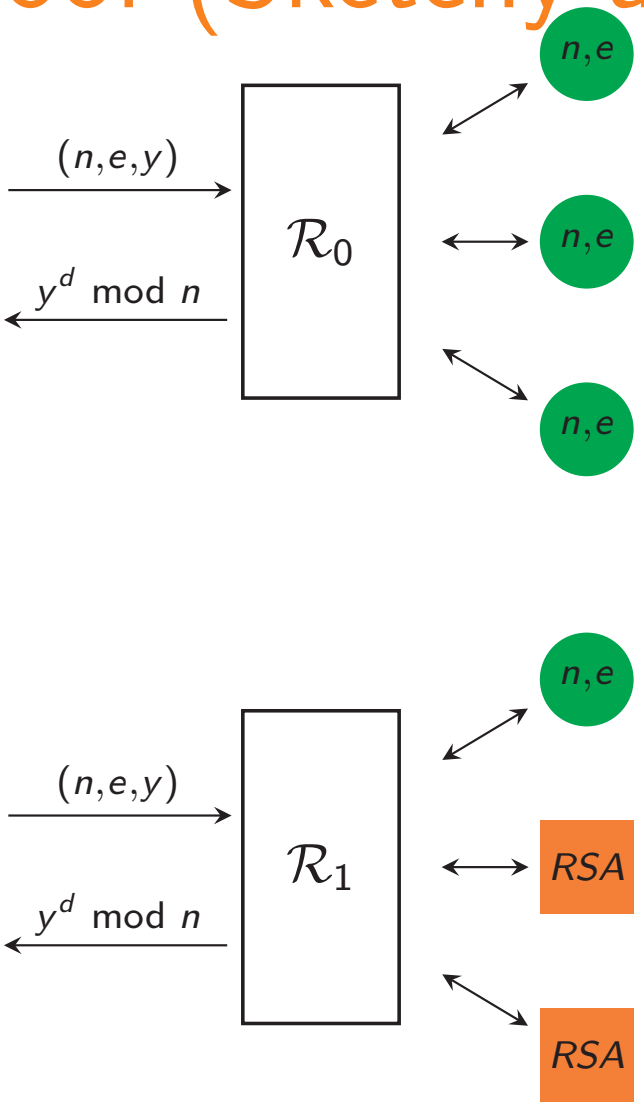
GAME 0





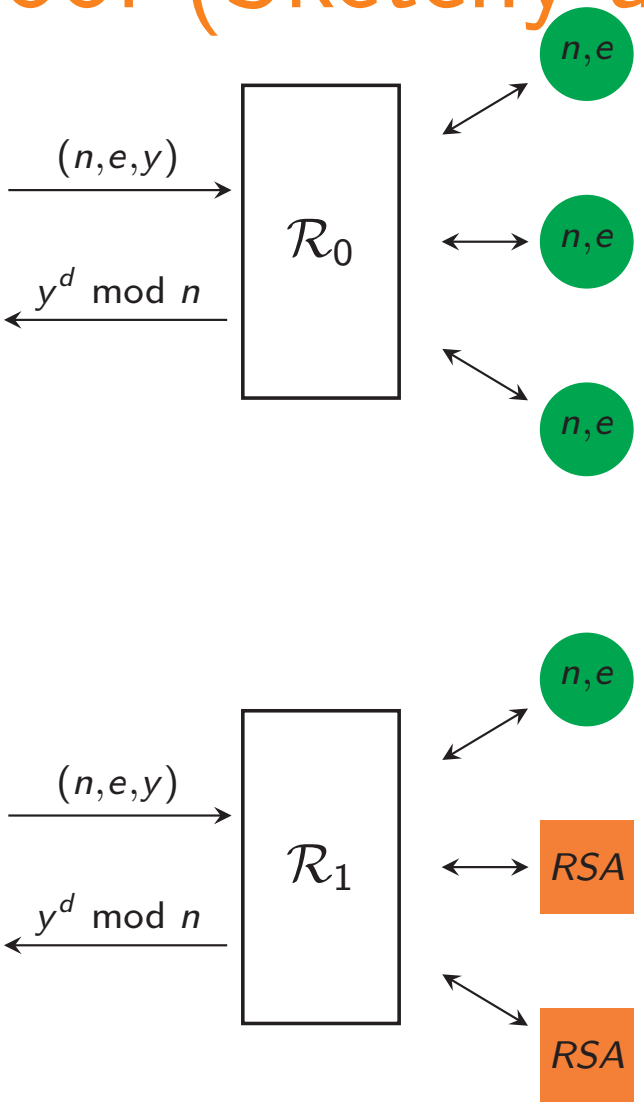
Proof (Sketchy and Visual)

Skip proof



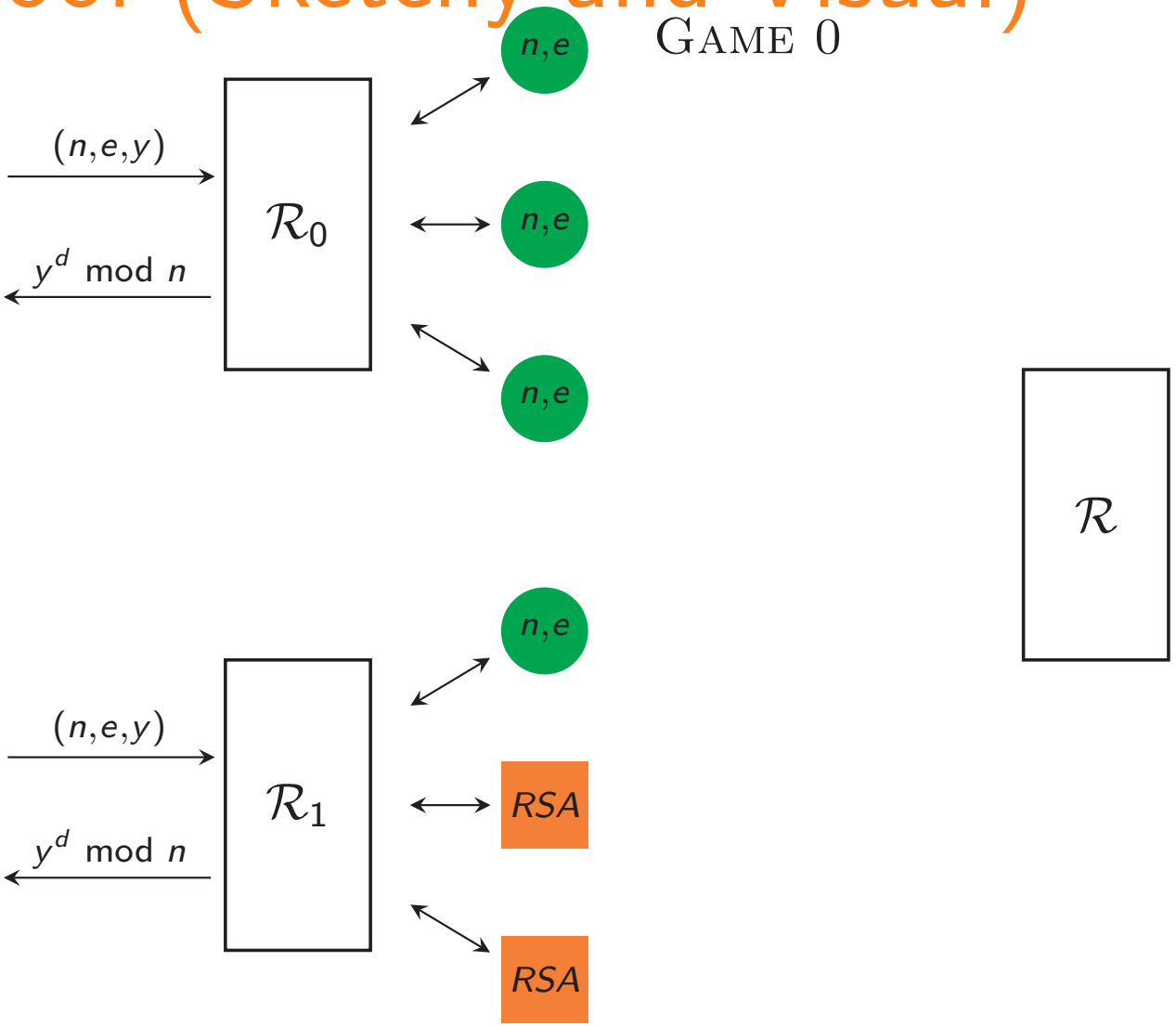
Proof (Sketchy and Visual)

Skip proof



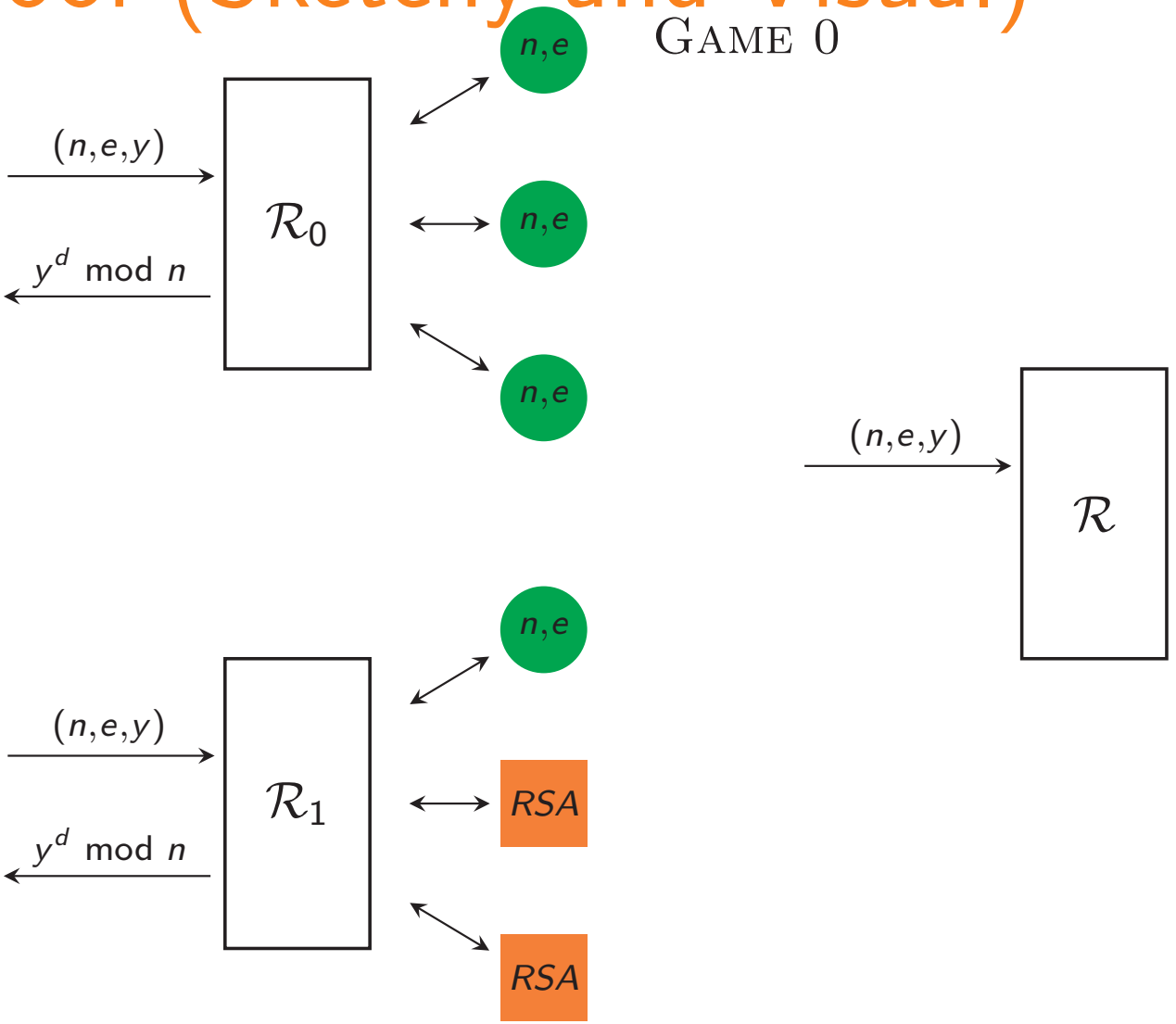
Proof (Sketchy and Visual)

Skip proof



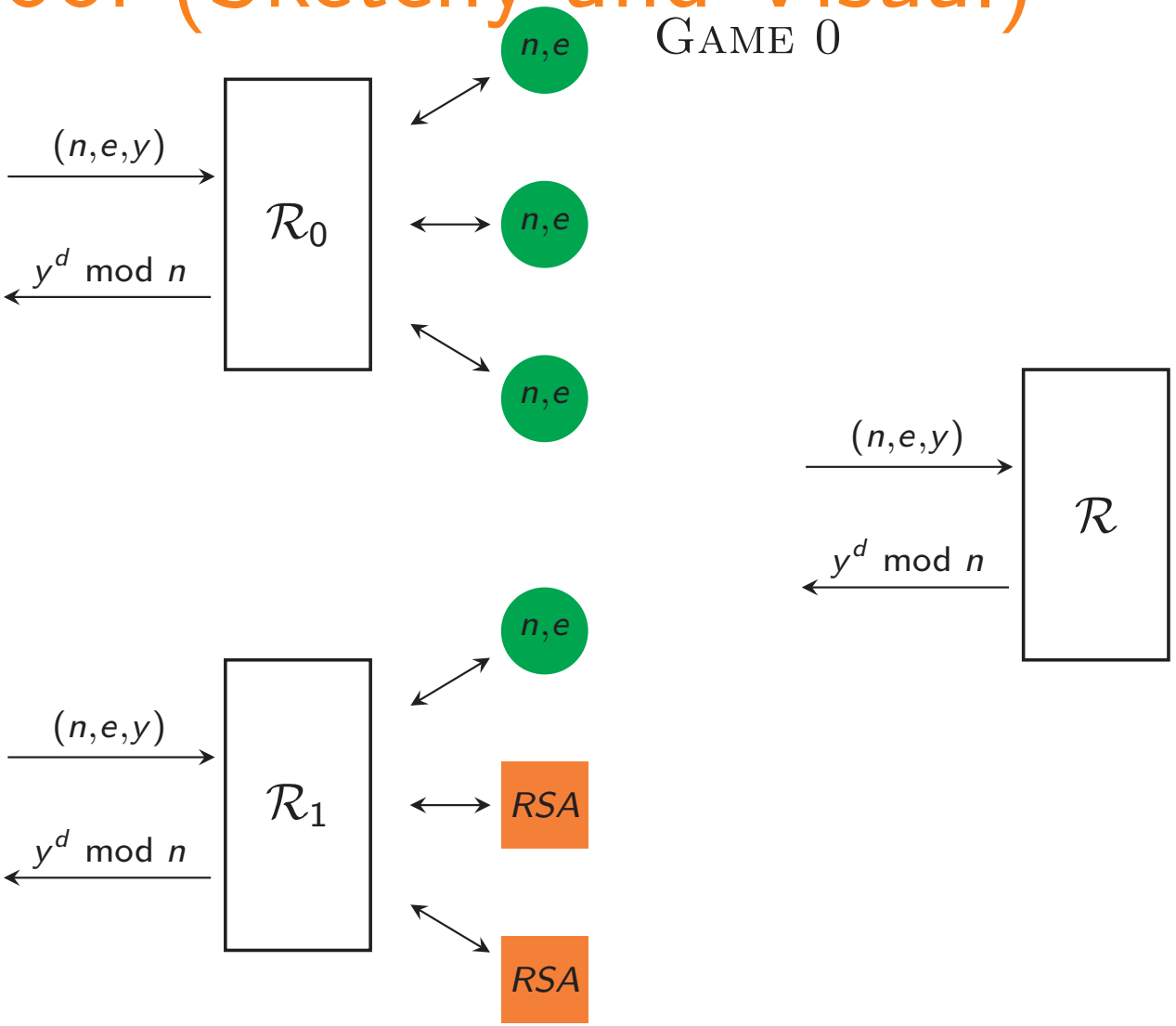
Proof (Sketchy and Visual)

Skip proof



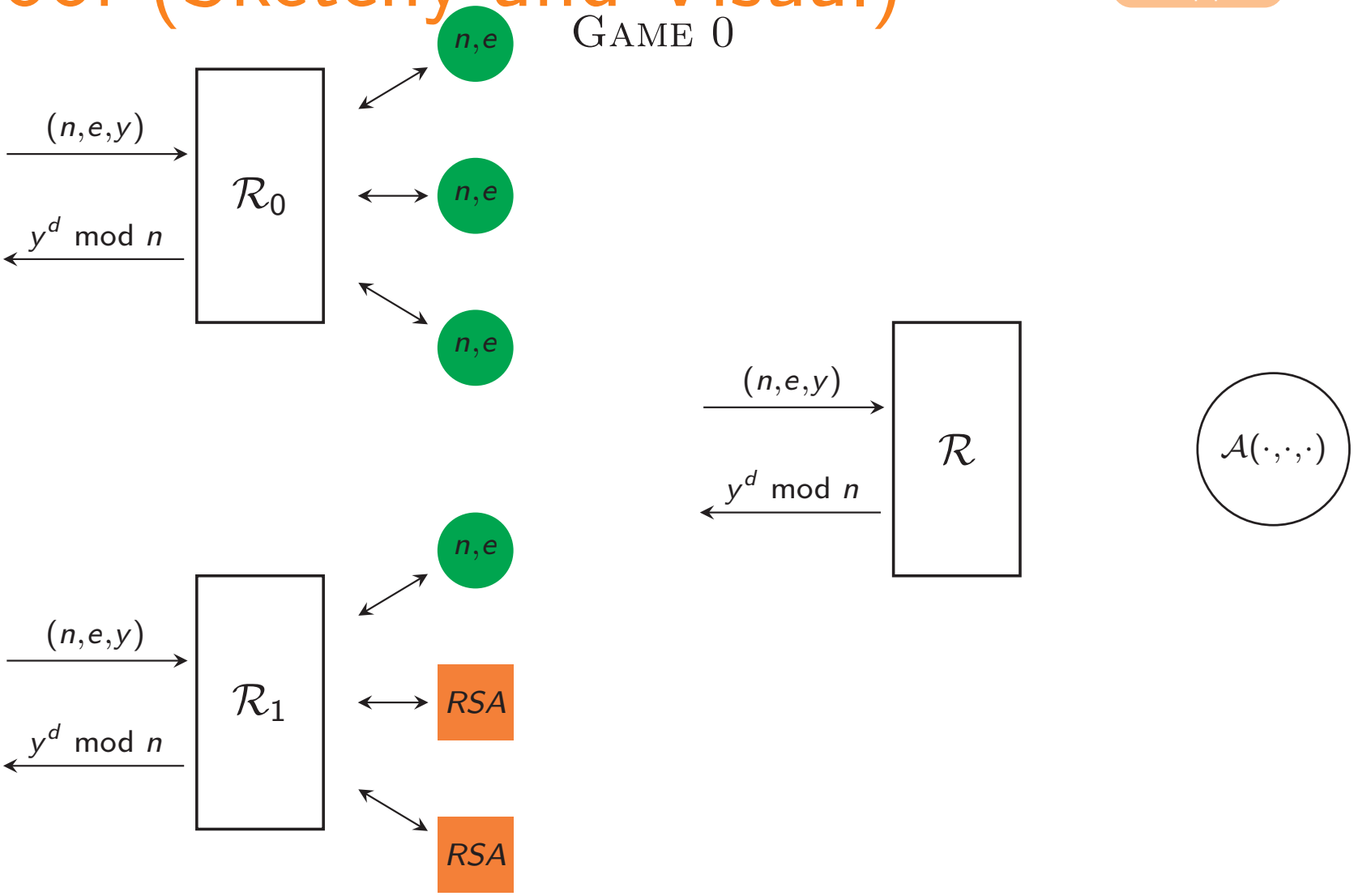
Proof (Sketchy and Visual)

Skip proof



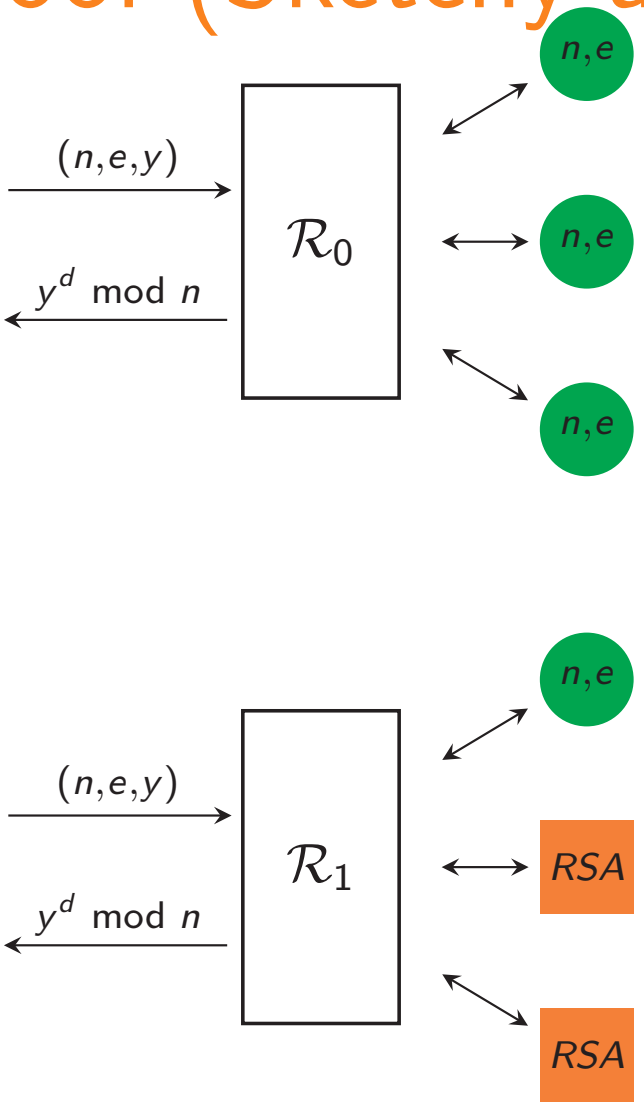
Proof (Sketchy and Visual)

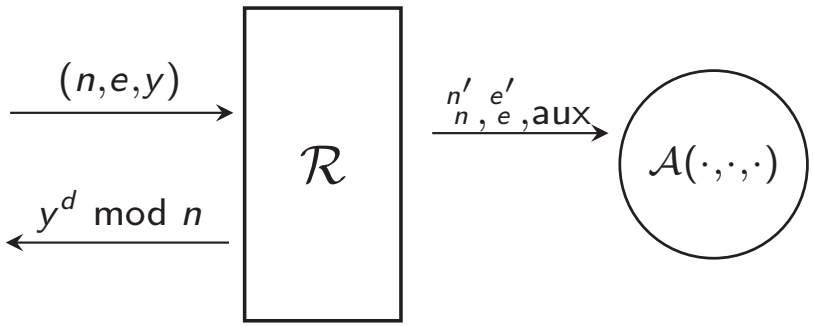
Skip proof



Proof (Sketchy and Visual)

Skip proof

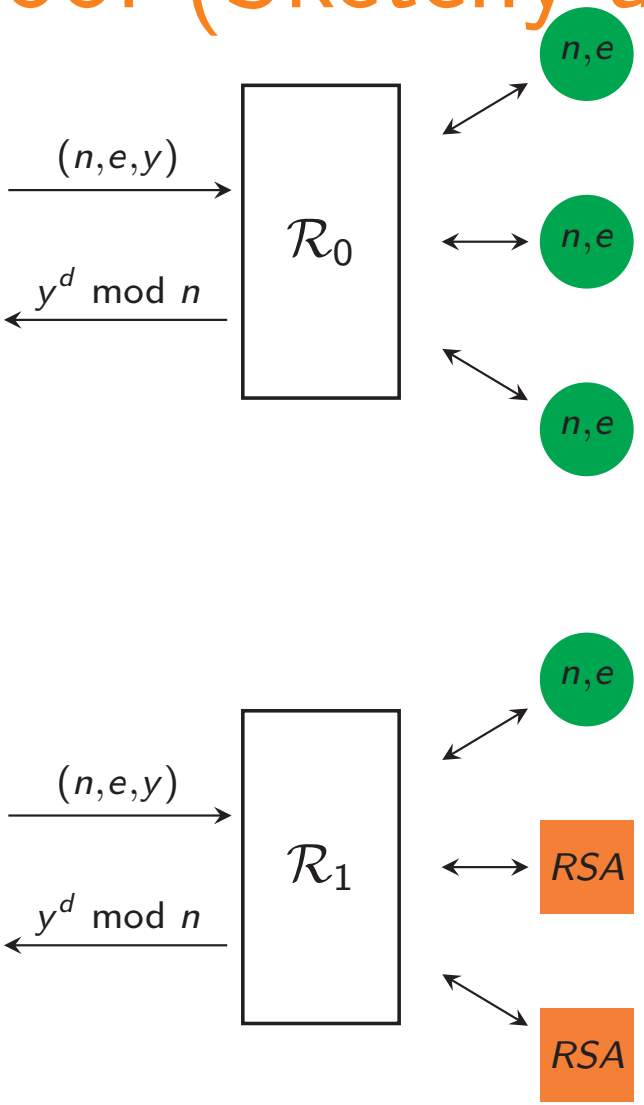


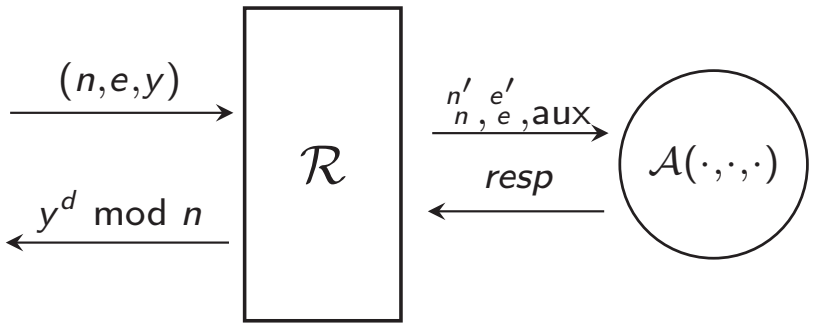


GAME 1

Proof (Sketchy and Visual)

Skip proof

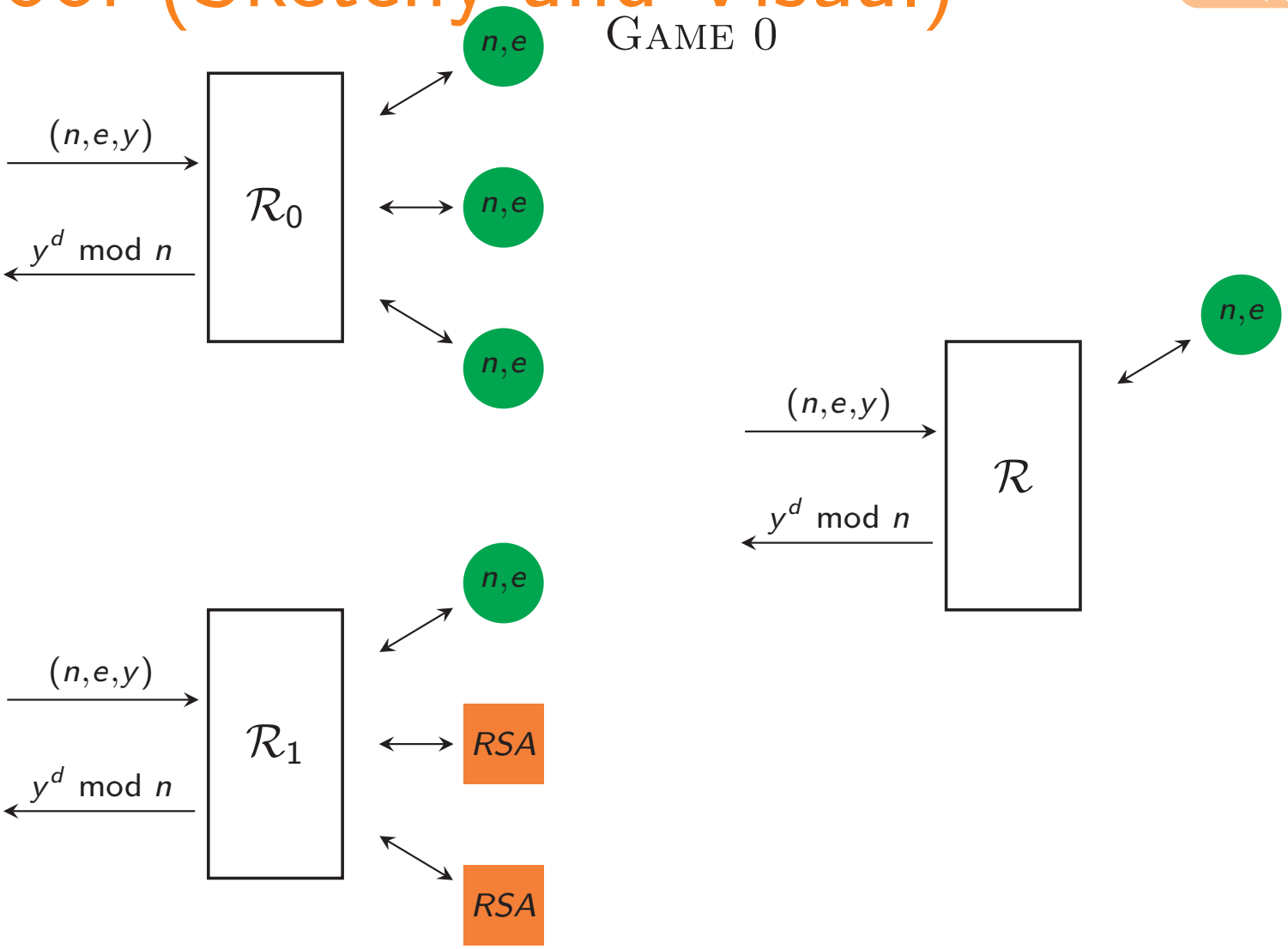




GAME 1

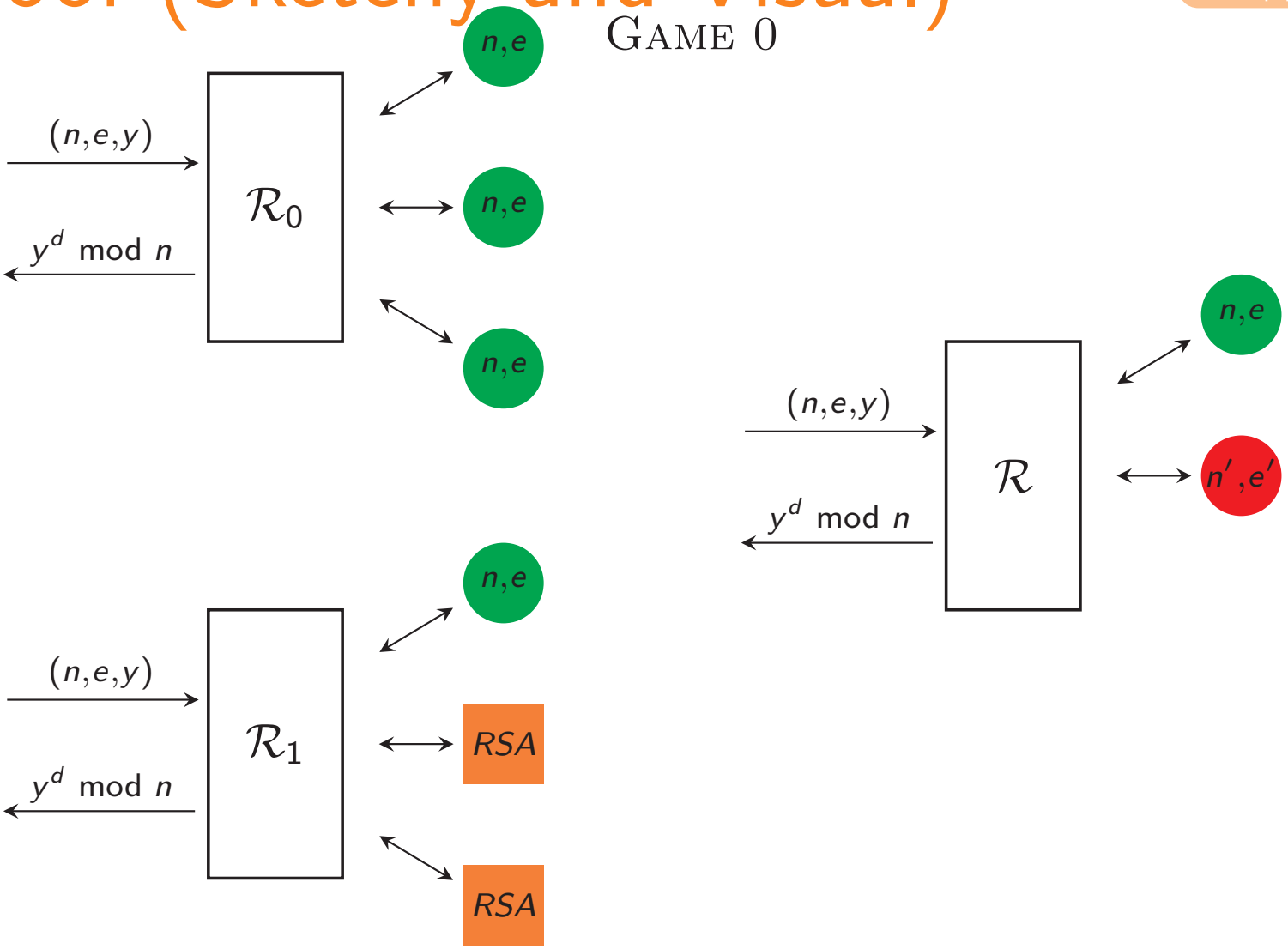
Proof (Sketchy and Visual)

Skip proof



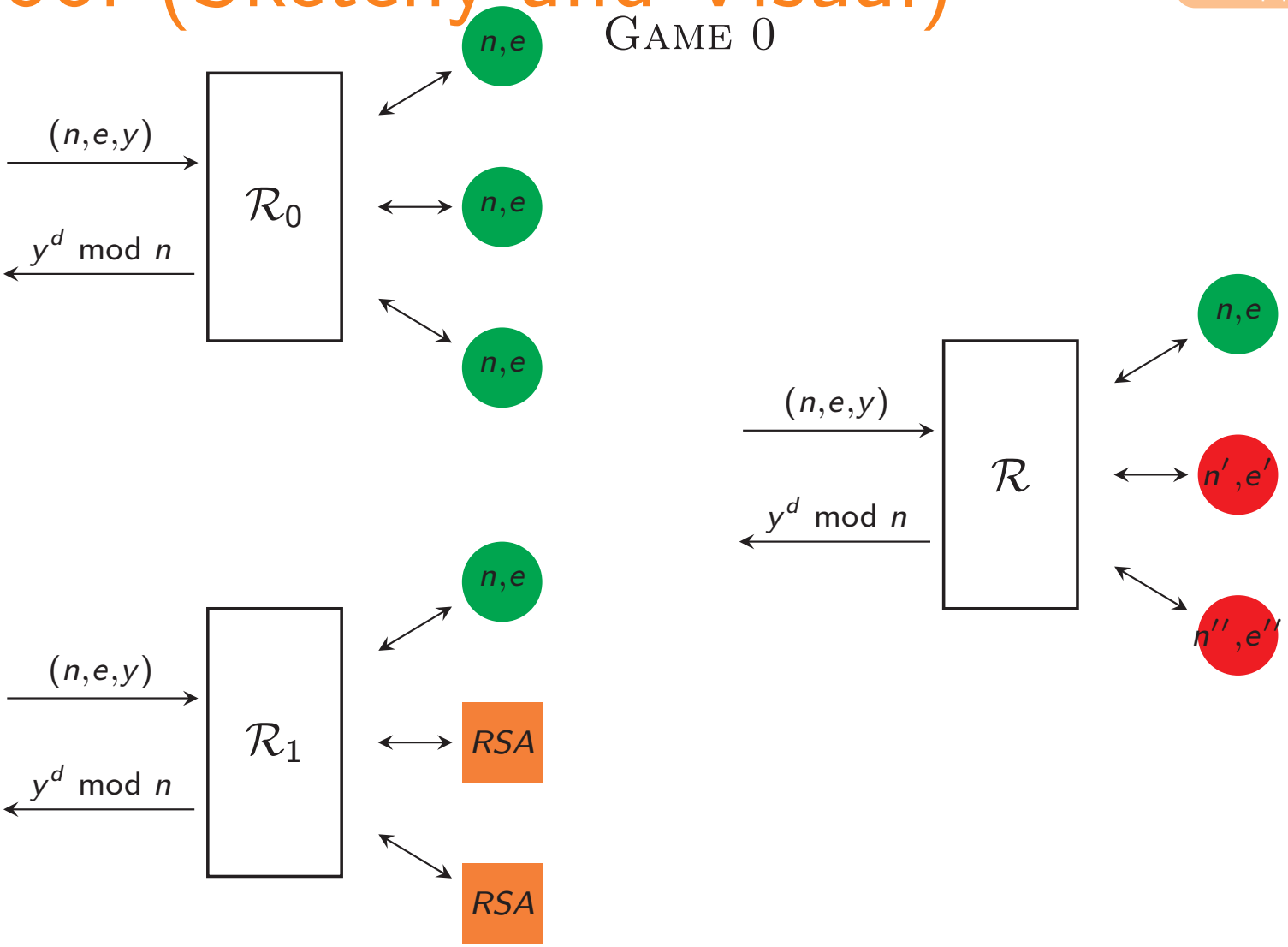
Proof (Sketchy and Visual)

Skip proof



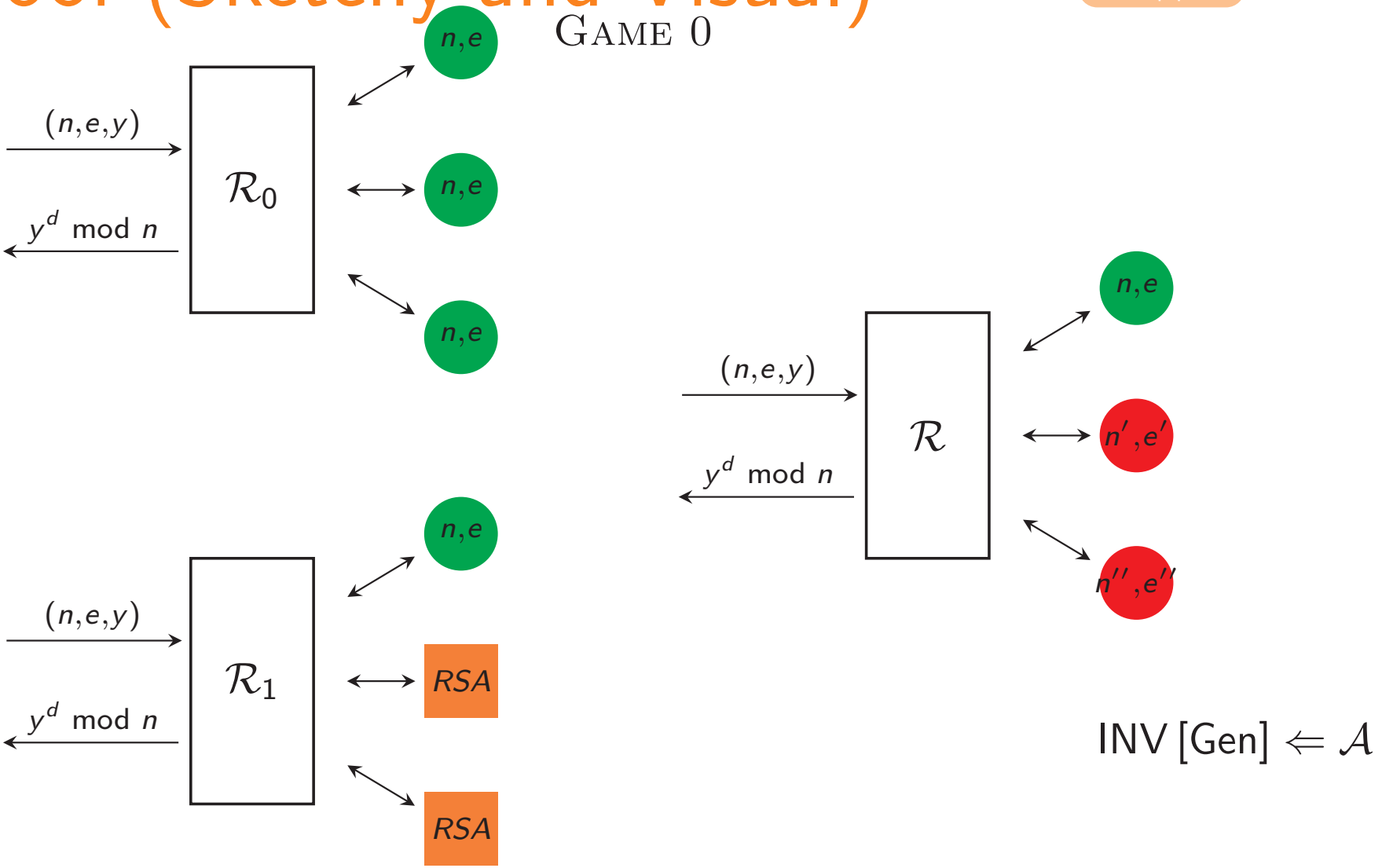
Proof (Sketchy and Visual)

Skip proof



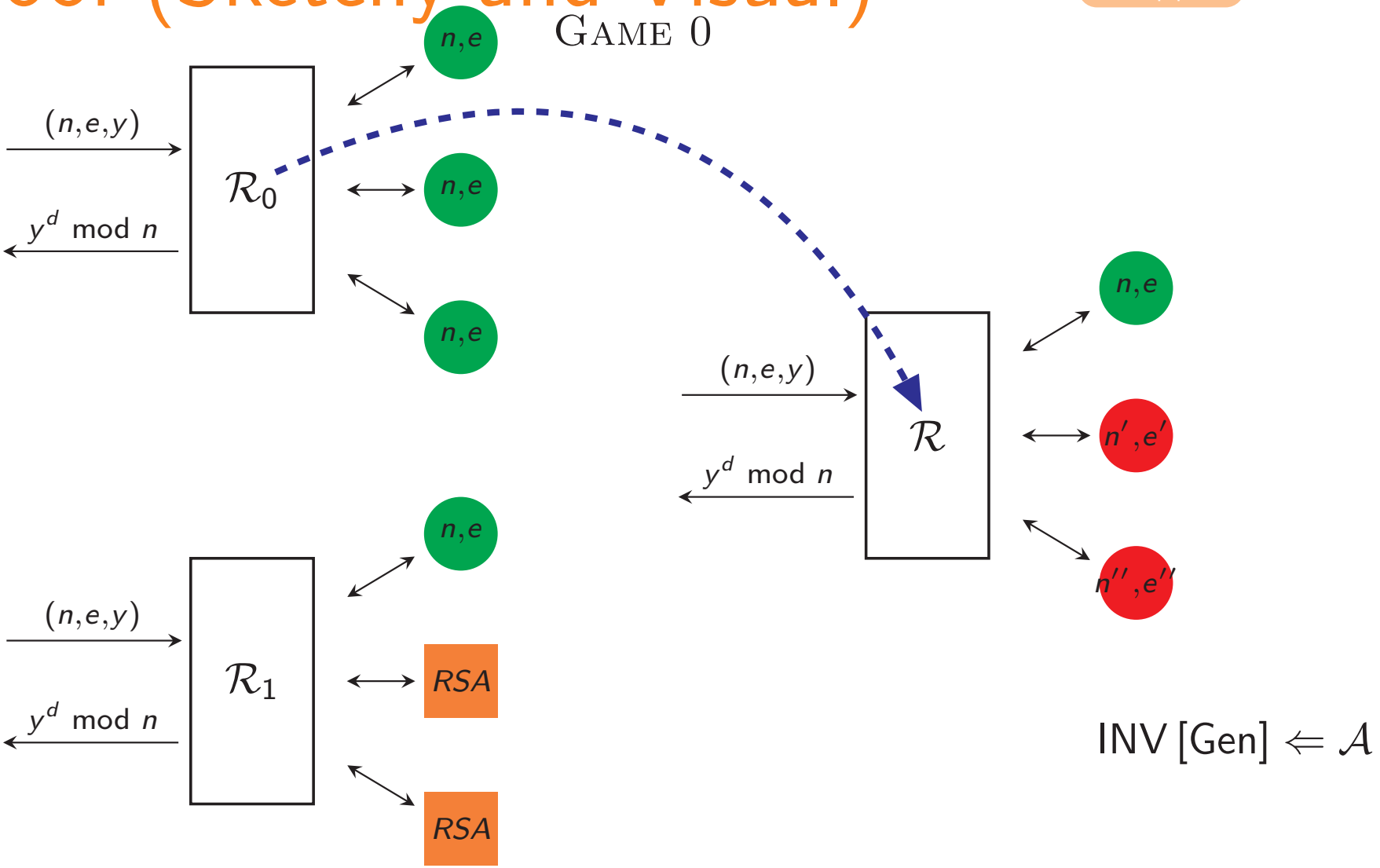
Proof (Sketchy and Visual)

Skip proof



Proof (Sketchy and Visual)

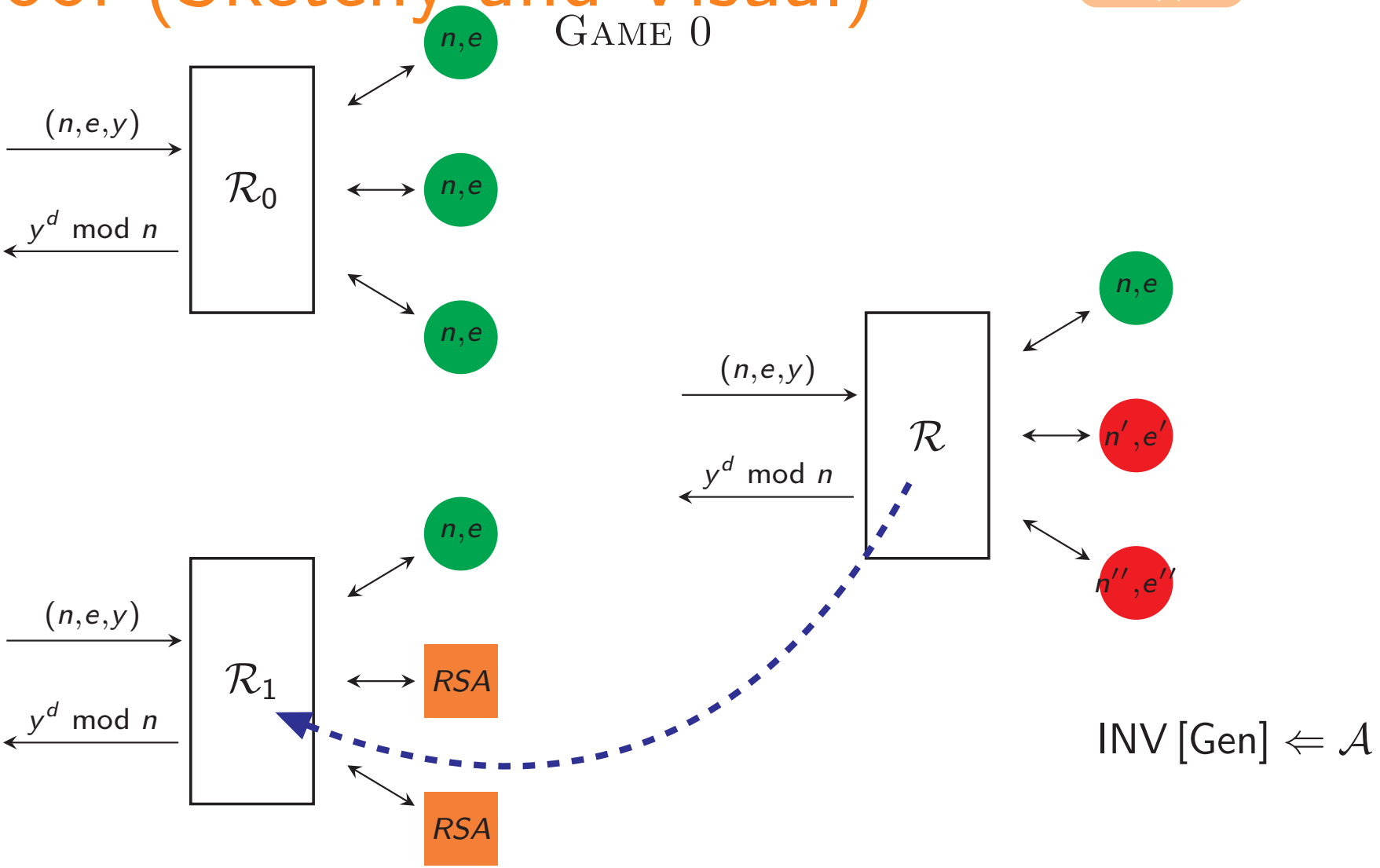
Skip proof



INV [Gen] $\Leftarrow \mathcal{A}$

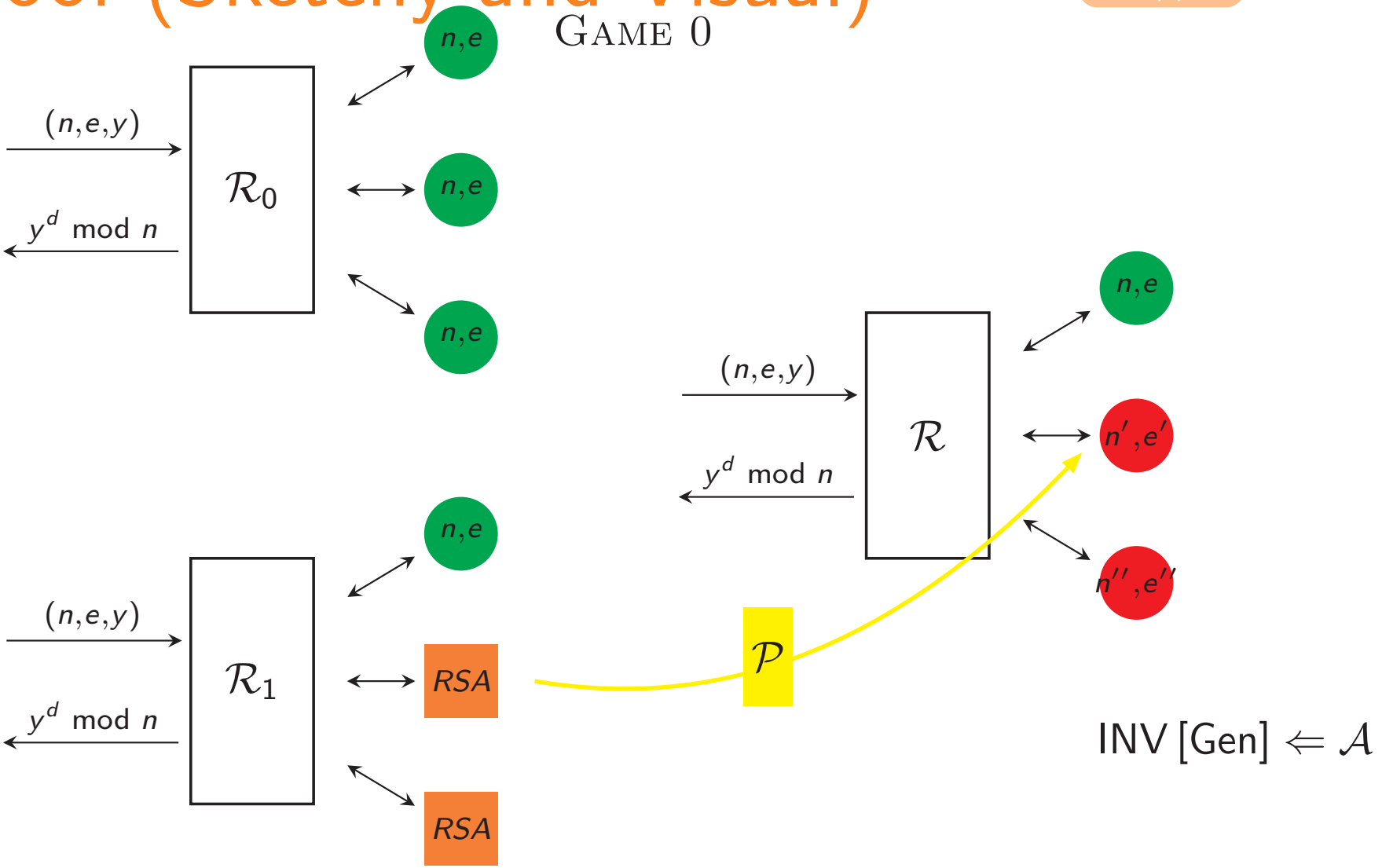
Proof (Sketchy and Visual)

Skip proof



Proof (Sketchy and Visual)

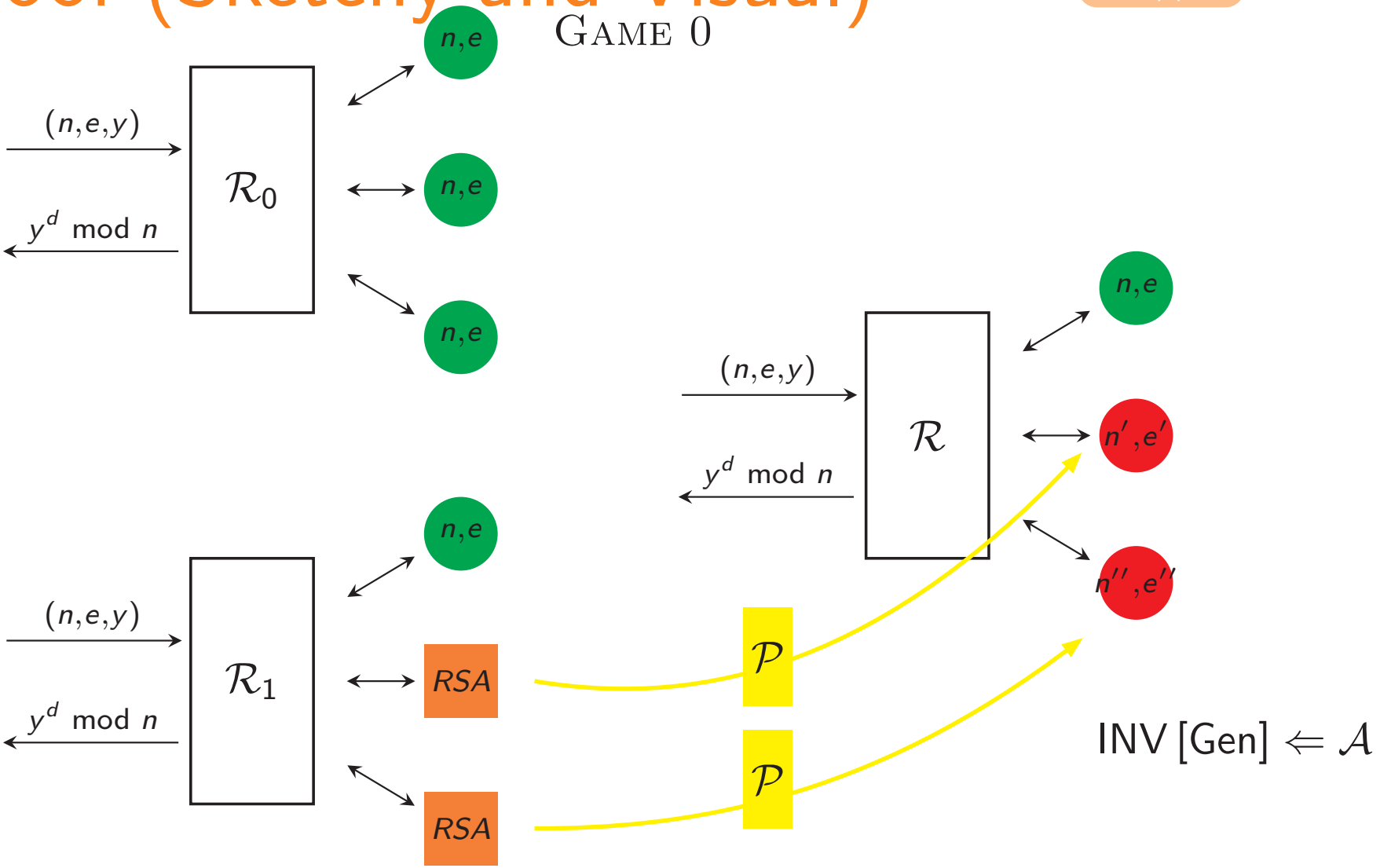
Skip proof



INV [Gen] $\Leftarrow \mathcal{A}$

Proof (Sketchy and Visual)

Skip proof



What does this mean?

What does this mean?

Essentially

$$\textsc{Game}\ 0 \leq \mathsf{INV}\,[\mathsf{Gen}] \Leftarrow \mathcal{A} \leq \textsc{Game}\ 1$$

What does this mean?

Essentially

$$\textsc{Game}\ 0 \leq \mathsf{INV}\,[\mathsf{Gen}] \Leftarrow \mathcal{A} \leq \textsc{Game}\ 1$$

Assuming Gen is instance-non-malleable

$$|\textsc{Game}\ 0 - \textsc{Game}\ 1| = \mathsf{negl}\,(\mathsf{k})$$

What does this mean?

Essentially

$$\textsc{Game}\ 0 \leq \mathsf{INV}\,[\mathsf{Gen}] \Leftarrow \mathcal{A} \leq \textsc{Game}\ 1$$

Assuming Gen is instance-non-malleable

$$|\textsc{Game}\ 0 - \textsc{Game}\ 1| = \mathsf{negl}\,(\mathsf{k})$$

Consequently

$$\textsc{Game}\ 0 \simeq \mathsf{INV}\,[\mathsf{Gen}] \Leftarrow \mathcal{A} \simeq \textsc{Game}\ 1$$

One-Wayness kills CCA Security

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$]

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]

One-Wayness kills CCA Security

Theorem

Assume that INV $[\mathrm{Gen}] \Leftarrow_{\mathcal{R}}$ OW-CPA $[\mathcal{E}]$ where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA $[\mathcal{E}]$
- runs $\mathcal{A}$ at most ℓ times

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$]

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

Proof technique

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

Proof technique

- note that $\mathcal{R}$ solves INV [Gen] $\Leftarrow \mathcal{A}$ with $\mathcal{A} \triangleq$ OW-CPA [$\mathcal{E}$]

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

Proof technique

- note that $\mathcal{R}$ solves INV [Gen] $\Leftarrow \mathcal{A}$ with $\mathcal{A} \triangleq$ OW-CPA [$\mathcal{E}$]
- obviously $\mathcal{A}$ is perfectly reducible to INV [Gen]

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

Proof technique

- note that $\mathcal{R}$ solves INV [Gen] $\Leftarrow \mathcal{A}$ with $\mathcal{A} \triangleq$ OW-CPA [$\mathcal{E}$]
- obviously $\mathcal{A}$ is perfectly reducible to INV [Gen]
- hence winning $\textsc{Game}\ 0$ is easy too if Gen is instance-non-malleable

One-Wayness kills CCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ OW-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks OW-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-CCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, \mathsf{k})}$$

Proof technique

- note that $\mathcal{R}$ solves INV [Gen] $\Leftarrow \mathcal{A}$ with $\mathcal{A} \triangleq$ OW-CPA [$\mathcal{E}$]
- obviously $\mathcal{A}$ is perfectly reducible to INV [Gen]
- hence winning $\textsc{Game}\ 0$ is easy too if Gen is instance-non-malleable
- finally note that winning $\textsc{Game}\ 0$ is exactly breaking RE-CCA [$\mathcal{E}$]

Indistinguishability kills PCA Security

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$]

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-PCA [$\mathcal{E}$]

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-PCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, k)}$$

Indistinguishability kills PCA Security

Theorem

Assume that INV [Gen] $\Leftarrow_{\mathcal{R}}$ IND-CPA [$\mathcal{E}$] where $\mathcal{R}$

- is given $\mathcal{A}$ which $(\varepsilon_{\mathcal{A}}, \tau_{\mathcal{A}})$-breaks IND-CPA [$\mathcal{E}$]
- runs $\mathcal{A}$ at most ℓ times
- $(\varepsilon_{\mathcal{R}}, \tau_{\mathcal{R}})$-inverts Gen

We build $\mathcal{M}$ that $(\varepsilon_{\mathcal{M}}, \tau_{\mathcal{M}})$-breaks ℓ-RE-PCA [$\mathcal{E}$] where

$$\boxed{\varepsilon_{\mathcal{M}} \geq \varepsilon_{\mathcal{R}} \text{ and } \tau_{\mathcal{M}} \leq \tau_{\mathcal{R}} + \text{poly}(\ell, k)}$$

Proof technique

Identical.

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen]

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen] ▸ why?

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- OW-CCA $[\mathcal{E}] \equiv$ INV [Gen]

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- OW-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- IND-PCA $[\mathcal{E}] \equiv$ INV [Gen]

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- OW-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- IND-PCA $[\mathcal{E}] \equiv$ INV [Gen]

The ROM is way too optimistic

IND-CCA $[\mathcal{E}] \not\equiv$ INV [Gen]

There is no hope for having

- IND-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- OW-CCA $[\mathcal{E}] \equiv$ INV [Gen]
- IND-PCA $[\mathcal{E}] \equiv$ INV [Gen]

The ROM is way too optimistic

In the ROM, there are encryptions *e.g.* RSA-OAEP for which

$$\boxed{\text{IND-CCA}\left[\mathcal{E}^{H}\right] \equiv \text{INV [Gen]} \text{ !!}}$$

Conclusion

Conclusion

The ROM is useful but way too optimistic

Conclusion

The ROM is useful but way too optimistic

What went wrong?

Conclusion

The ROM is useful but way too optimistic

What went wrong?

The programmability of random oracles makes them too powerful and unrealistic

Conclusion

The ROM is useful but way too optimistic

What went wrong?

The programmability of random oracles makes them too powerful and unrealistic

What is the real security level of RSA-OAEP?

Conclusion

The ROM is useful but way too optimistic

What went wrong?

The programmability of random oracles makes them too powerful and unrealistic

What is the real security level of RSA-OAEP?

We don't know!

Non-Malleability in Other Settings

Assume $\Delta(\text{poly}(k)) = \text{negl}(k)$.

RSA Signatures

the unforgeability of RSA-based signatures such as FDH, PSS, PSS-R and many others cannot be equivalent to INV [Gen]

Non-Malleability in Other Settings

Assume $\Delta(\mathrm{poly}(k)) = \mathrm{negl}(k)$.

RSA Signatures

the unforgeability of RSA-based signatures such as FDH, PSS, PSS-R and many others cannot be equivalent to INV [Gen]

Transposable Signatures *e.g.* PSS

Non-Malleability in Other Settings

Assume $\Delta(\text{poly}(k)) = \text{negl}(k)$.

RSA Signatures

the unforgeability of RSA-based signatures such as FDH, PSS, PSS-R and many others cannot be equivalent to INV [Gen]

Transposable Signatures *e.g.* PSS

Even worse as UF-KOA $[\mathcal{S}] \not\equiv$ OM [Gen]!

Simple Non-Malleability

Simple Non-Malleability

Non-malleability: Given $n \leftarrow \mathrm{Gen}(1^k)$, a factoring oracle $\mathrm{FACT}(n' \neq n)$ does not help to factor n. Defined in a similar way.

Simple Non-Malleability

Simple Non-Malleability

Non-malleability: Given $n \leftarrow \mathrm{Gen}(1^k)$, a factoring oracle $\mathrm{FACT}(n' \neq n)$ does not help to factor n. Defined in a similar way.

Factoring-Based Encryption

The CCA security of Rabin/RW-SAEP[+]/OAEP[+][+], EPOC-2, etc. is $\not\equiv$ FACT [Gen]

Simple Non-Malleability

Simple Non-Malleability

Non-malleability: Given $n \leftarrow \mathrm{Gen}(1^k)$, a factoring oracle $\mathrm{FACT}(n' \neq n)$ does not help to factor n. Defined in a similar way.

Factoring-Based Encryption

The CCA security of Rabin/RW-SAEP[+]/OAEP[+][+], EPOC-2, etc. is $\not\equiv$ FACT [Gen]

Factoring-Based Signatures

The unforgeability of Rabin signatures, etc. is $\not\equiv$ FACT [Gen]

Computational Number Theoretic Challenges

Can we prove non-malleability?

Real-life Generators seem non-malleable

But can one use number theory to come up with a formal proof?

Computational Number Theoretic Challenges

Can we prove non-malleability?

Real-life Generators seem non-malleable

But can one use number theory to come up with a formal proof?

Can we build **malleable** RSA key generation?

Computational Number Theoretic Challenges

Can we prove non-malleability?

Real-life Generators seem non-malleable

But can one use number theory to come up with a formal proof?

Can we build **malleable** RSA key generation?

Malleable RSA moduli

$(n, n') \leftarrow \mathrm{Gen}(1^k)$ such that

- n' can be recovered from n alone
- factoring n' allows to factor n easily