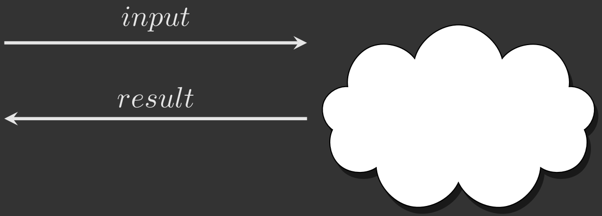
Reconciling Non-malleability & Homomorphic Encryption

Anna Lisa Ferrara · Manoj Prabhakaran · Mike Rosulek

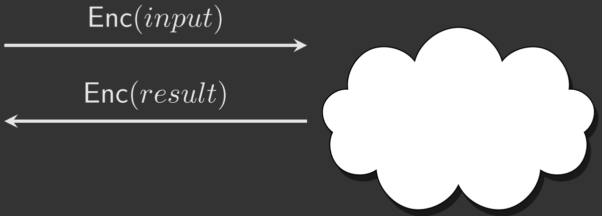
Crypto in the Clouds · August 4, 2009

Computing on Encrypted Data



Typical "Computing on Encrypted Data" Approach:

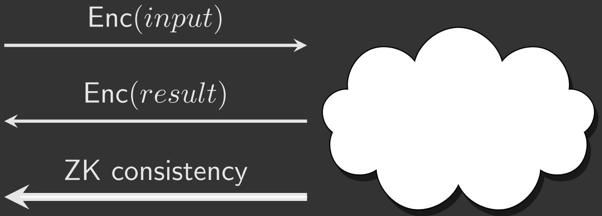
Computing on Encrypted Data



Typical "Computing on Encrypted Data" Approach:

1. Encrypt input/output for secrecy
   - Use homomorphic encryption to allow blind computation

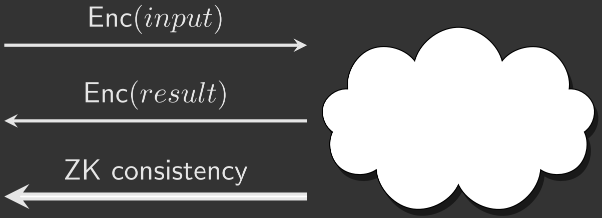
Computing on Encrypted Data



Typical "Computing on Encrypted Data" Approach:

1. Encrypt input/output for secrecy
   - Use homomorphic encryption to allow blind computation
2. Require proof of correct computation. Why?

Computing on Encrypted Data



Typical "Computing on Encrypted Data" Approach:

1. Encrypt input/output for secrecy
   - Use homomorphic encryption to allow blind computation
2. Require proof of correct computation. Why?
   - Only have CPA security

A Difficult Tradeoff

Expressivity:

- Encrypted data can be blindly manipulated
- Homomorphic / computational feature

Integrity:

- Result should reflect correct computation
- Actually a non-malleability requirement

A Difficult Tradeoff

Expressivity:

- Encrypted data can be blindly manipulated
- Homomorphic / computational feature

Integrity:

- Result should reflect correct computation
- Actually a non-malleability requirement

Can we get both in a single encryption scheme?

New Definitions

Consider case of unary operations: $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$

Complementary Definitions [PR08]

1. Scheme allows operations $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$, where f in prescribed set $\mathcal{F}$.
2. Other than those features, scheme is non-malleable.

New Definitions

Consider case of unary operations: $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$

Complementary Definitions [PR08]

1. Scheme allows operations $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$, where f in prescribed set $\mathcal{F}$.

2. Other than those features, scheme is non-malleable.
   - Given unknown $\mathsf{Enc}(m)$, cannot generate C such that $\mathsf{Dec}(C)$ depends on m...

New Definitions

Consider case of unary operations: $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$

Complementary Definitions [PR08]

1. Scheme allows operations $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$, where f in prescribed set $\mathcal{F}$.
2. Other than those features, scheme is non-malleable.
   - Given unknown $\mathsf{Enc}(m)$, cannot generate C such that $\mathsf{Dec}(C)$ depends on m...
   - ... unless $\mathsf{Dec}(C) = f(m)$ for an allowed $f \in \mathcal{F}$

Contrast with Fully Homomorphic Encryption:

Fully homomorphic encryption [G09]:

- Sole focus is maximum expressivity
- Binary operations: $\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(m_1 + m_2)$

This work:

- Focus on sharp tradeoff in homomorphic operations:

  $\in \mathcal{F}$: available as highly expressive full feature

  $\notin \mathcal{F}$: computationally infeasible
- Difficult regardless of expressivity
  - E.g.: $\mathcal{F}$ contains only one operation

$\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ Available as Feature

Correctness Requirement

$\mathsf{Dec}(\mathsf{Trans}(C, f)) = f(\mathsf{Dec}(C))$

$\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ Available as Feature

Correctness Requirement

$\mathsf{Dec}(\mathsf{Trans}(C, f)) = f(\mathsf{Dec}(C))$

New Definition(s): Unlinkability [PR08]

$\mathsf{Trans}(\mathsf{Enc}(m), f)$ "looks like" $\mathsf{Enc}(f(m))$

Weak: $\mathsf{Enc}(f(m)) \approx \mathsf{Trans}(\mathsf{Enc}(m), f)$

(Indistinguishabilities in presence of Dec oracle)

$\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ Available as Feature

Correctness Requirement

$\mathsf{Dec}(\mathsf{Trans}(C, f)) = f(\mathsf{Dec}(C))$

New Definition(s): Unlinkability [PR08]

$\mathsf{Trans}(\mathsf{Enc}(m), f)$ "looks like" $\mathsf{Enc}(f(m))$

Weak: $\mathsf{Enc}(f(m)) \approx \mathsf{Trans}(\mathsf{Enc}(m), f)$

Medium: $(C, \mathsf{Enc}(f(m))) \approx (C, \mathsf{Trans}(C, f))$, where $C \leftarrow \mathsf{Enc}(m)$

Strong: $(C, \mathsf{Enc}(f(m))) \approx (C, \mathsf{Trans}(C, f))$, where C adversarially chosen, $\mathsf{Dec}(C) = m$.

(Indistinguishabilities in presence of Dec oracle)

Non-malleable Except For Desired Operations

Suppose no adversary can distinguish between 2 worlds:

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{Enc}_{PK}(m^*)$.
5. Provide Dec oracle.

Non-malleable Except For Desired Operations

Suppose no adversary can distinguish between 2 worlds:

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{Enc}_{PK}(m^*)$.
5. Provide Dec oracle.

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{RigEnc}(PK)$.
5. Provide Dec oracle, except:
   - If $f \leftarrow \mathsf{RigExtract}_{SK}(C)$, then answer $f(m^*)$.

Non-malleable Except For Desired Operations

Suppose no adversary can distinguish between 2 worlds:

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{Enc}_{PK}(m^*)$.
5. Provide Dec oracle.

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{RigEnc}(PK)$.
5. Provide Dec oracle, except:
   - If $f \leftarrow \mathsf{RigExtract}_{SK}(C)$, then answer $f(m^*)$.

Intuition: suppose some adversary can change $\mathsf{Enc}(m^*)$ into related ciphertext C; $\mathsf{Dec}(C^*) = f(m^*)$ (unknown m^*)

Non-malleable Except For Desired Operations

Suppose no adversary can distinguish between 2 worlds:

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{Enc}_{PK}(m^*)$.
5. Provide Dec oracle.

1. Generate keypair, give PK.
2. Provide $\mathsf{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \mathsf{RigEnc}(PK)$.
5. Provide Dec oracle, except:
   - If $f \leftarrow \mathsf{RigExtract}_{SK}(C)$, then answer $f(m^*)$.

Intuition: suppose some adversary can change $\mathsf{Enc}(m^*)$ into related ciphertext C; $\mathsf{Dec}(C^*) = f(m^*)$ (unknown m^*)

Submit C to Dec oracle, get back answer $\mathsf{Dec}(C) = f(m^*)$

Non-malleable Except For Desired Operations

Suppose no adversary can distinguish between 2 worlds:

1. Generate keypair, give PK.
2. Provide $\text{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \text{Enc}_{PK}(m^*)$.
5. Provide Dec oracle.

1. Generate keypair, give PK.
2. Provide $\text{Dec}_{SK}(\cdot)$ oracle.
3. Adversary chooses m^*.
4. Give $C^* \leftarrow \text{RigEnc}(PK)$.
5. Provide Dec oracle, except:
   - If $f \leftarrow \text{RigExtract}_{SK}(C)$, then answer $f(m^*)$.

Intuition: suppose some adversary can change $\text{Enc}(m^*)$ into related ciphertext C; $\text{Dec}(C^*) = f(m^*)$ (unknown m^*)

Submit C to Dec oracle, get back answer $\text{Dec}(C) = f(m^*)$

Submit C to oracle; RigExtract must output f

A Limit on Malleability

Observation

Operation $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ possible (perhaps adversarially)
$\implies$ RigExtract must be allowed to output f

A Limit on Malleability

Observation

Operation $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ possible (perhaps adversarially)

$\implies$ RigExtract must be allowed to output f

RigExtract never allowed to output f

$\implies$ $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ impossible (even adversarially)

A Limit on Malleability

Observation

Operation $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ possible (perhaps adversarially)
$\implies$ RigExtract must be allowed to output f

RigExtract never allowed to output f
$\implies$ $\mathsf{Enc}(m) \rightsquigarrow \mathsf{Enc}(f(m))$ impossible (even adversarially)

HCCA Security Definition [PR08]

Scheme is non-malleable except for operations $\mathcal{F}$ if there are suitable RigEnc, RigExtract, with range(RigExtract) $\subseteq \mathcal{F}$.

A Limit on Malleability

Observation

Operation $\text{Enc}(m) \rightsquigarrow \text{Enc}(f(m))$ possible (perhaps adversarially)
$\implies$ RigExtract must be allowed to output f

RigExtract never allowed to output f
$\implies$ $\text{Enc}(m) \rightsquigarrow \text{Enc}(f(m))$ impossible (even adversarially)

HCCA Security Definition [PR08]

Scheme is non-malleable except for operations $\mathcal{F}$ if there are suitable RigEnc, RigExtract, with range(RigExtract) $\subseteq \mathcal{F}$.

- RigEnc, RigExtract needed only for security analysis
- Can obtain CCA, RCCA [CKN03], gCCA [S01,ADR02] as special cases by further restricting RigEnc, RigExtract
- Implicitly rules out all malleability not of form $\text{Enc}(m) \rightsquigarrow \text{Enc}(f(m))$

Constructions

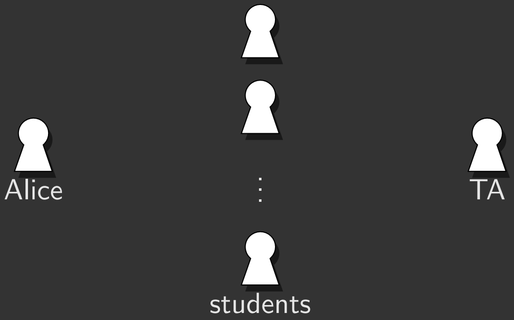
Strong, slightly inefficient construction [PR08]

- DDH $\implies$ strong unlinkability + HCCA
- Expressivity: group operations in DDH group
- Ciphertext is 20 group elements

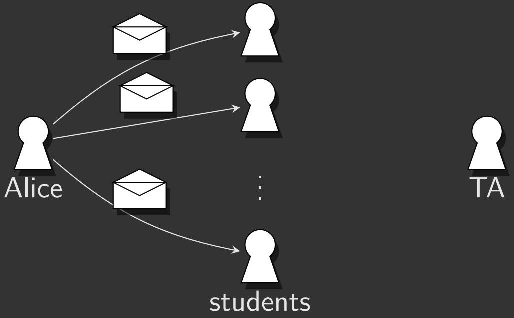
Weak, efficient construction [FPR09]

- CCA $\implies$ weak unlinkability + HCCA
- Expressivity: arbitrary group operations
- Using Cramer-Shoup DDH, ciphertext has 5 group elements

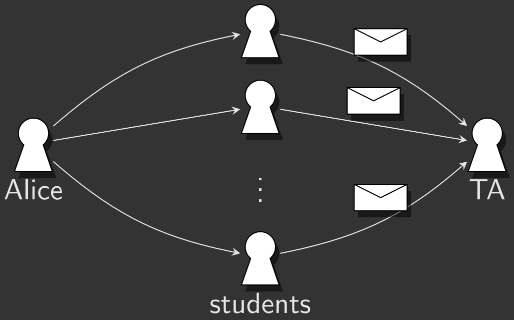
Application: Teaching Evaluations [PR08b]



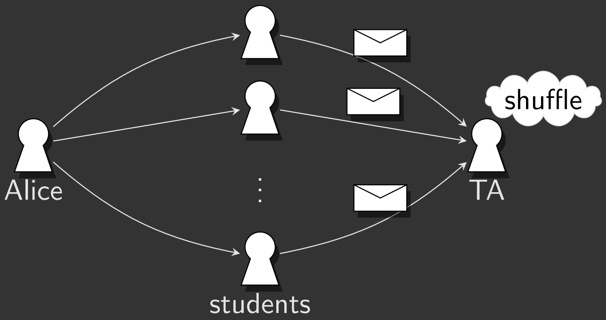
Application: Teaching Evaluations [PR08b]



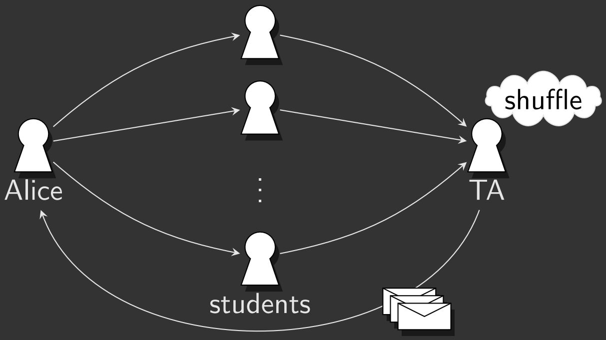
Application: Teaching Evaluations [PR08b]



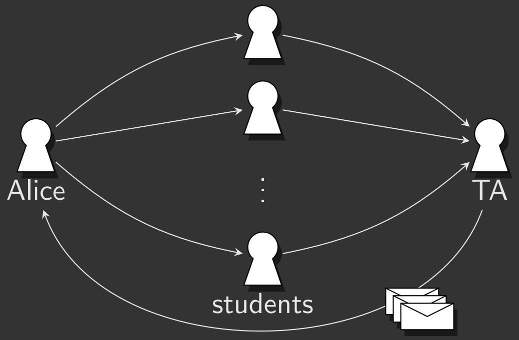
Application: Teaching Evaluations [PR08b]



Application: Teaching Evaluations [PR08b]



Application: Teaching Evaluations [PR08b]

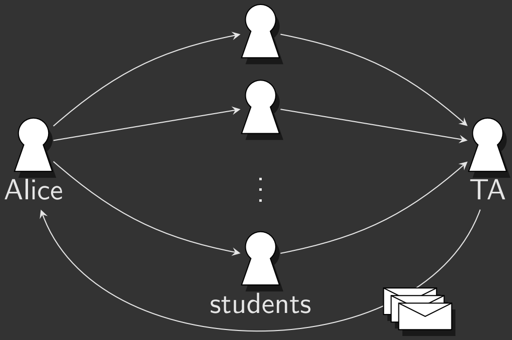


Privacy: TA can't see responses

Functionality: TA must be able to anonymize (shuffle)

Integrity: TA can't modify/replace responses

Application: Teaching Evaluations [PR08b]



Privacy: TA can't see responses

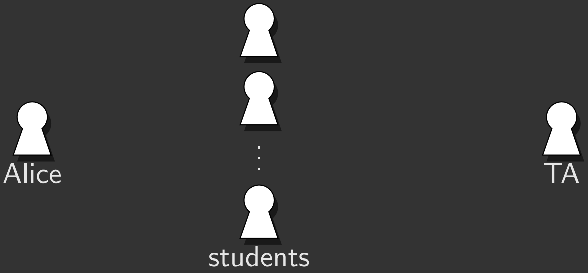
Functionality: TA must be able to anonymize (shuffle)

Integrity: TA can't modify/replace responses

Verifiable ciphertext shuffle [G02,GL07a,GL07b]

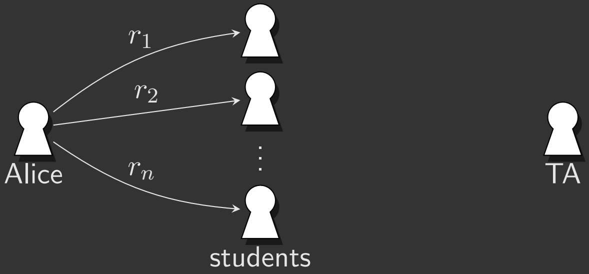
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose <u>only</u> operations are $\mathsf{Enc}(m, r) \rightsquigarrow \mathsf{Enc}(m, rs)$ for r, s in a group.



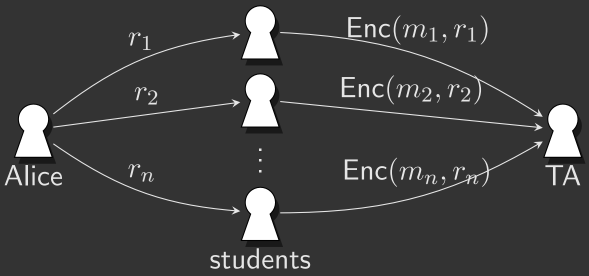
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are $\mathsf{Enc}(m, r) \rightsquigarrow \mathsf{Enc}(m, rs)$ for r, s in a group.



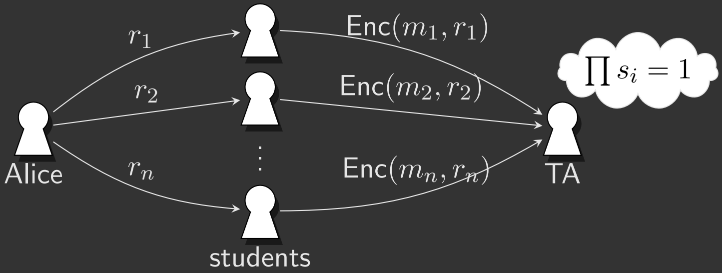
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are Enc$(m, r) \rightsquigarrow$ Enc(m, rs) for r, s in a group.



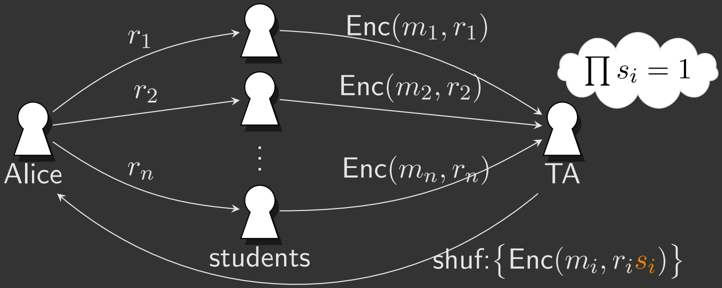
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are Enc$(m, r) \rightsquigarrow$ Enc(m, rs) for r, s in a group.



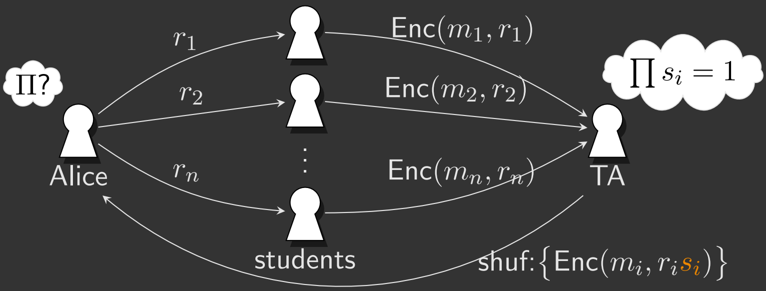
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are $\mathsf{Enc}(m, r) \rightsquigarrow \mathsf{Enc}(m, rs)$ for r, s in a group.



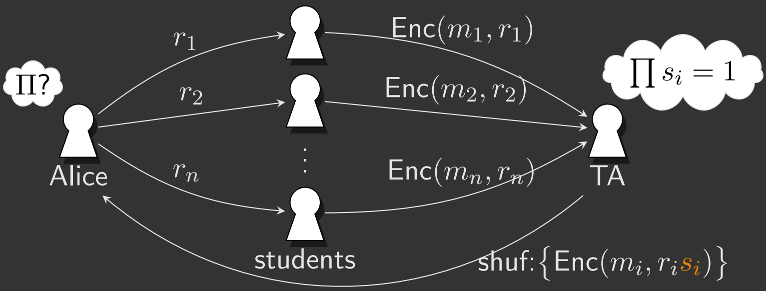
Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are $\mathsf{Enc}(m, r) \rightsquigarrow \mathsf{Enc}(m, rs)$ for r, s in a group.



Protocol Using New Notion

Use non-malleable homomorphic encryption, whose <u>only</u> operations are Enc$(m, r) \rightsquigarrow$ Enc(m, rs) for r, s in a group.

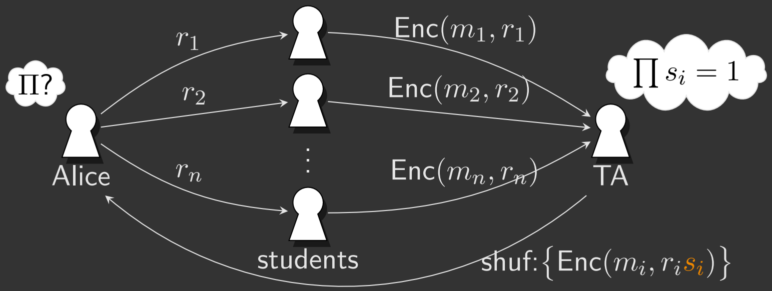


Security proof:

- TA must give $\{$Enc$(m_i', r_i')\}$, where $\prod r_i' = \prod r_i$

Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are Enc$(m, r) \rightsquigarrow$ Enc(m, rs) for r, s in a group.

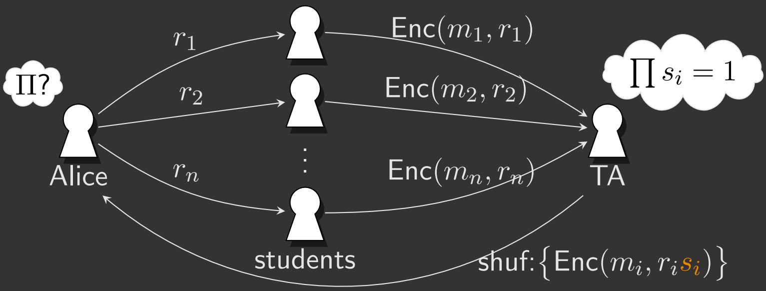


Security proof:

- TA must give $\{$Enc$(m'_i, r'_i)\}$, where $\prod r'_i = \prod r_i$
- Each r'_i is multiple of a single r_j, or independent of all r_i's
- Must depend on each r_i once, else $\prod r'_i$ independent of $\prod r_i$

Protocol Using New Notion

Use non-malleable homomorphic encryption, whose only operations are $\mathsf{Enc}(m, r) \rightsquigarrow \mathsf{Enc}(m, rs)$ for r, s in a group.



Security proof:

- TA must give $\{\mathsf{Enc}(m_i', r_i')\}$, where $\prod r_i' = \prod r_i$
- Each r_i' is multiple of a single r_j, or independent of all r_i's
- Must depend on each r_i once, else $\prod r_i'$ independent of $\prod r_i$
- Can't get dependence on an r_i without its m_i intact

Overview

Non-malleable homomorphic encryption useful in distributed protocols:

- Intuitively simple protocol; avoids ZK
- UC-secure without setups!
- Practical (only need weak unlinkability)
- Can also get distributed OR, group operation protocols

Binary Operations

What about binary operations?

$$\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(f(m_1, m_2))$$

Binary Operations

What about binary operations?

$$\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(f(m_1, m_2))$$

Theorem [PR08b]

Non-malleable homomorphic encryption impossible for a group operation over message space.

Binary Operations

What about binary operations?

$$\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(f(m_1, m_2))$$

Theorem [PR08b]

Non-malleable homomorphic encryption impossible for a group operation over message space.

Proof.

- Transformed ciphertexts look like regular ciphertexts
  - ciphertexts have a-priori length bound
- Simulator must be able to extract ciphertext "history"
- There can be more histories than possible ciphertexts:
  - Given n ciphertexts, each $\prod_{i \in I} m_i$ is a history ($I \subseteq [n]$).

□

A Glimmer of Hope

Length bound crucial in impossibility result!

- What if transformed ciphertexts allowed to grow in size?
- "Cryptocomputing" paradigm [SYY99]

A Glimmer of Hope

Length bound crucial in impossibility result!

- What if transformed ciphertexts allowed to grow in size?
- "Cryptocomputing" paradigm [SYY99]

Theorem [PR08b]

Under DDH, can construct a scheme with following requirements:

- Allows group operation $\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(m_1 m_2)$
- Non-malleable otherwise
- Ciphertext leaks only the $\#$ of operations applied

A Glimmer of Hope

Length bound crucial in impossibility result!

- What if transformed ciphertexts allowed to grow in size?
- "Cryptocomputing" paradigm [SYY99]

Theorem [PR08b]

Under DDH, can construct a scheme with following requirements:

- Allows group operation $\mathsf{Enc}(m_1), \mathsf{Enc}(m_2) \rightsquigarrow \mathsf{Enc}(m_1 m_2)$
- Non-malleable otherwise
- Ciphertext leaks only the # of operations applied

Comparison to SYY99:

- Ciphertext size grows linearly, not exponentially
- Only one group operation, not both ring operations
- Non-malleability property

Moral of the Story

- Non-malleability need not be all-or-nothing
- Can achieve sharp tradeoff between features/non-malleability
  - Future direction: beyond encryption? NIZK? Signatures?
- Non-malleable homomorphic encryption helps for protocols
  - UC security with elementary protocols, no ZK machinery
- Impossible for binary group operations
  - Not all is lost if ciphertext allowed to leak a little

Thanks for your attention!

fin.

Supported Operations

[PR08] construction:

- Message space $= \mathbb{G}^n$ for DDH group $\mathbb{G}$, fixed n
- Parameter $= \mathbb{H}$, subgroup of $\mathbb{G}^n$
- Allowed operations:

$$\mathsf{Enc}(x_1, \ldots, x_n) \rightsquigarrow \mathsf{Enc}(x_1 h_1, \ldots, x_n h_n) \text{ for } \vec{h} \in \mathbb{H}$$

- Note: cannot exponentiate, separate components, etc..

Example instantiations:

- $\mathbb{H} = \{\mathbf{1}\}$: Cannot change plaintext, only rerandomize (Rerandomizable RCCA [CKN03,G04,PR07])
- $\mathbb{H} = \mathbb{G}^n$: Can "multiply" by anything
- $\mathbb{H} = \{\mathbf{1}\} \times \mathbb{G}$: Only first component non-malleable
- $\mathbb{H} = \{(h, \ldots, h) \mid h \in \mathbb{G}\}$: "Scalar multiplication" of vector