Re-encryption, functional re-encryption, and multi-hop re-encryption:

A framework for achieving obfuscation-based security and instantiations from Lattices

Nishanth Chandran[1] Melissa Chase[1] Feng-Hao Liu[2]
Ryo Nishimaki[3] Keita Xagawa[3]

[1]Microsoft Research [2]University of Maryland
[3]NTT Secure Platform Laboratories

PKC 2014 @ Buenos Aires

Our Result

A New Framework for Obfuscating Re-Encryption

- ✔ New relaxed definitions
- ✔ New tools for modular analysis
- ✔ Secure obfuscator from LWE for
  1. (standard) re-encryption
  2. functional re-encryption
  3. multi-hop re-encryption

Outline

Program Obfuscation [BGI+12]

The Goal of Obfuscation

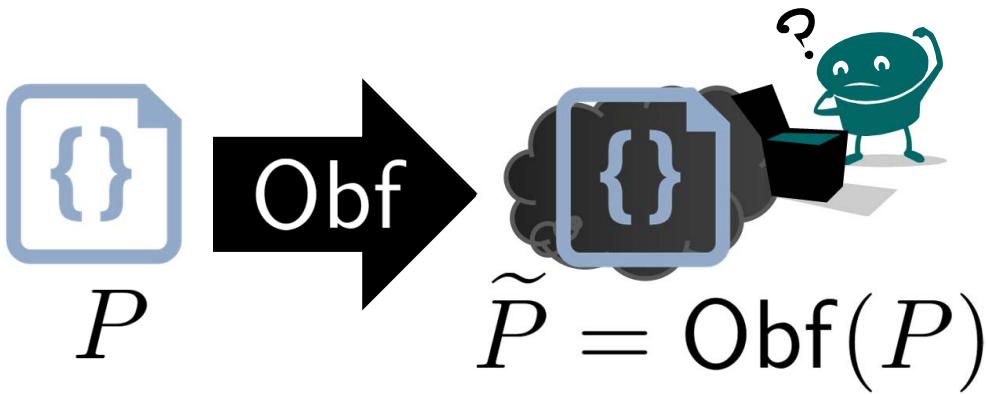
Prevent hacking and reverse engineering

Program Obfuscation [BGI+12]

The Goal of Obfuscation

Prevent hacking and reverse engineering

Obfuscator Obf: Compiler



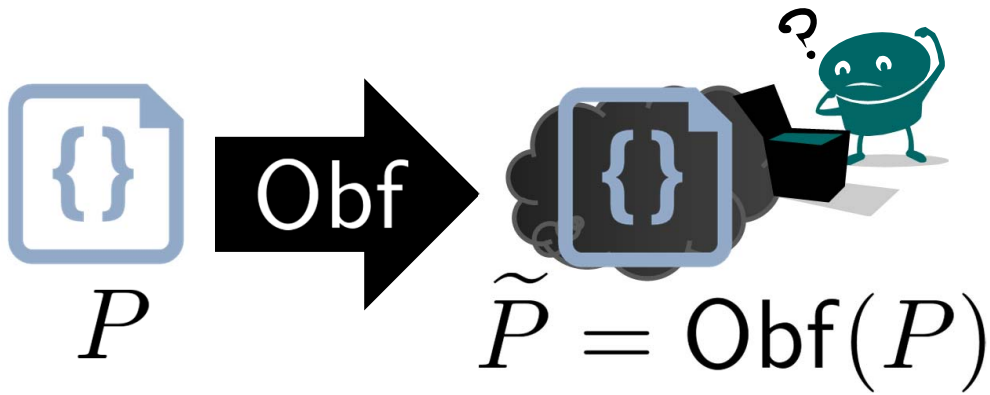
- $\widetilde{P}$: Completely garbled and unintelligible program

Program Obfuscation [BGI+12]

The Goal of Obfuscation

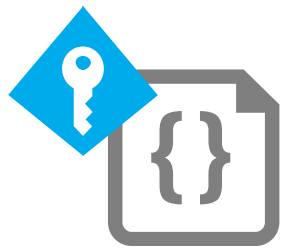
Prevent hacking and reverse engineering

Obfuscator Obf: Compiler



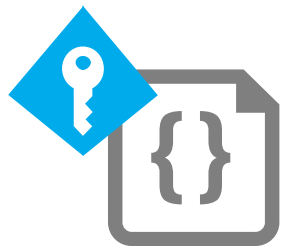
- $\widetilde{P}$: Completely garbled and unintelligible program
- Functionally equivalent: $\widetilde{P}(\text{input}) \equiv P(\text{input})$

Application of Obfuscation



- Protecting software

Application of Obfuscation



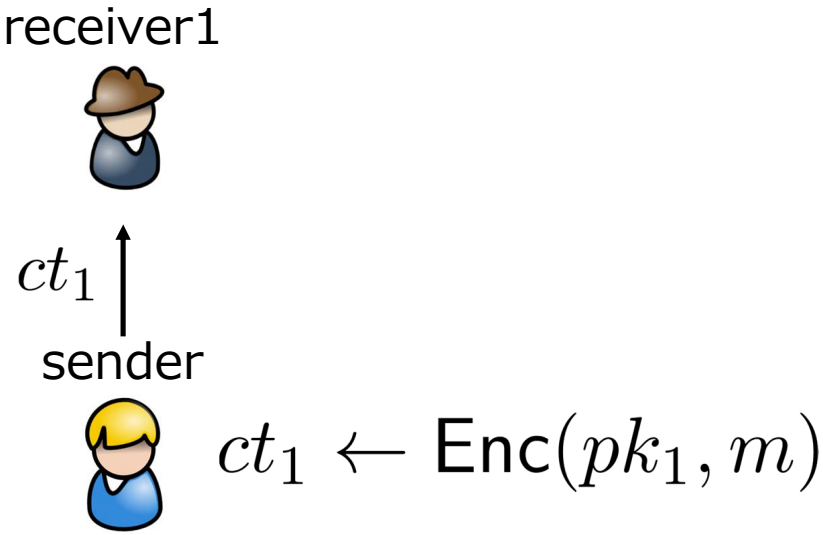
- Protecting software
- Almost all crypto

  Before GGHRSW13: [Can97, CMR98, Hada00, BGI+01, LPS04, Wee05, HLS10, HRsV11, CD08, CRV09, CB10, CCV12, NX13]

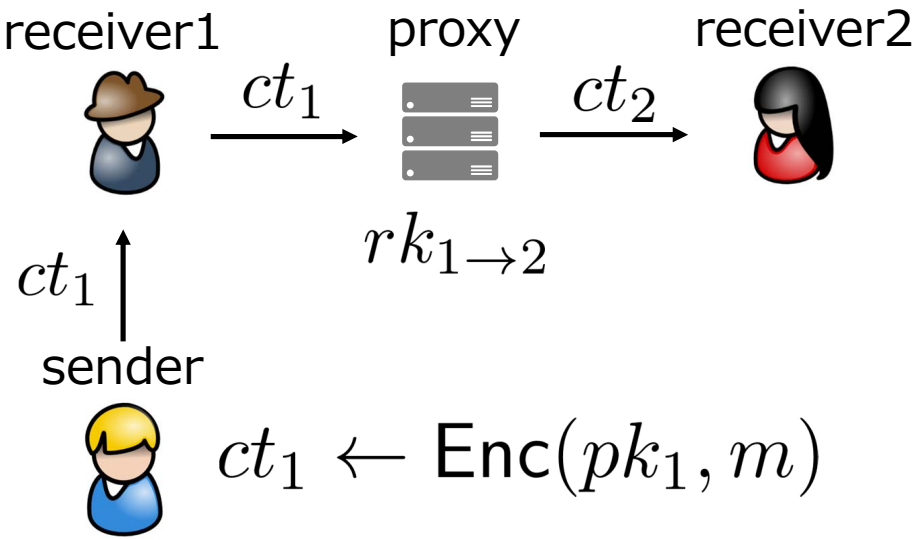
  After GGHRSW13: [ABGSZ13, BBCKPS14, BCP14, BCPR13a, BCPR13b, BR14a, BR14b, BGKPS14, BZ13, CGK13, CV13, GGHRSW13, GGHW13, GGG+14, GGSR14, GJKS13, GK13, HSW14, MO13, MR13, PPS13, PTS13, SW14]

Outline

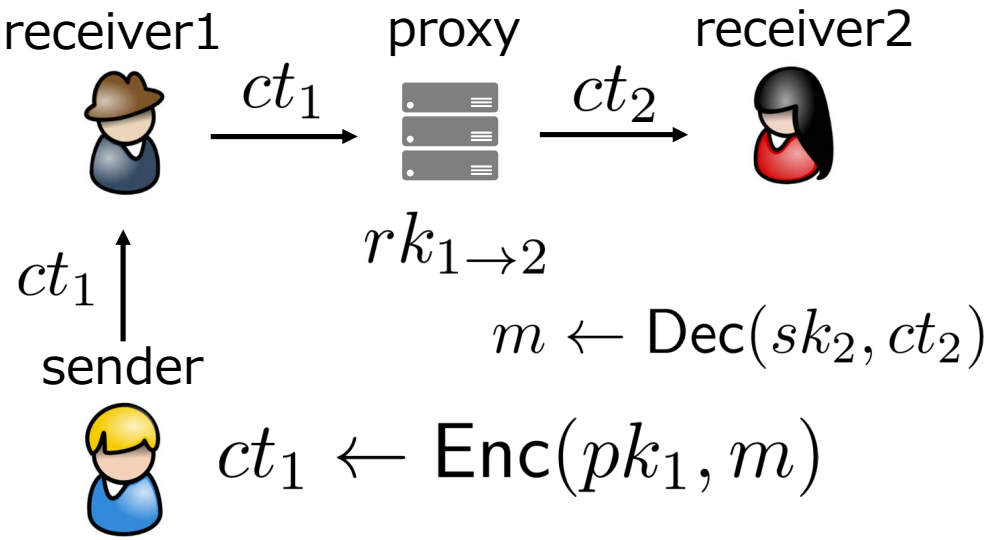
Re-Encryption (standard)



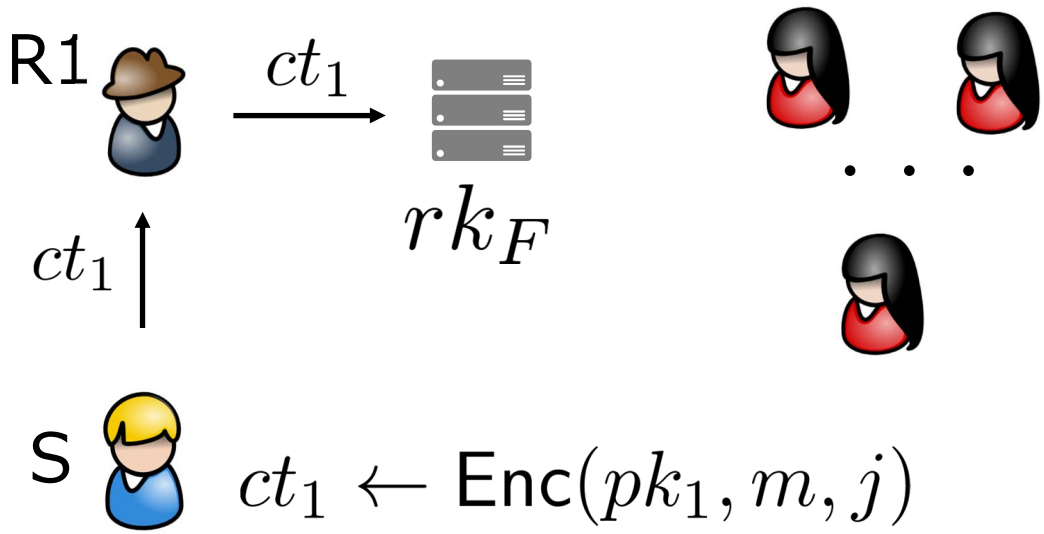
Re-Encryption (standard)



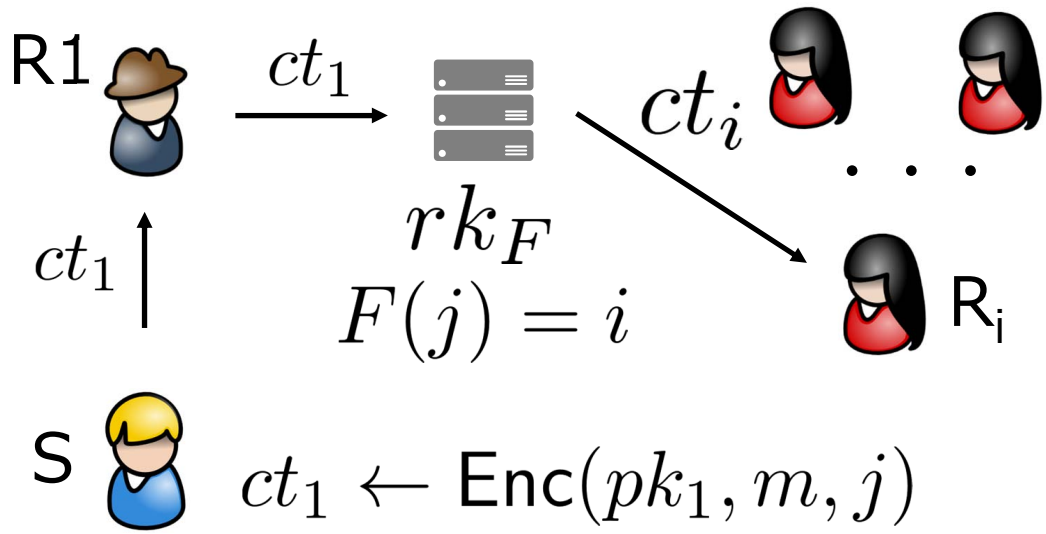
Re-Encryption (standard)



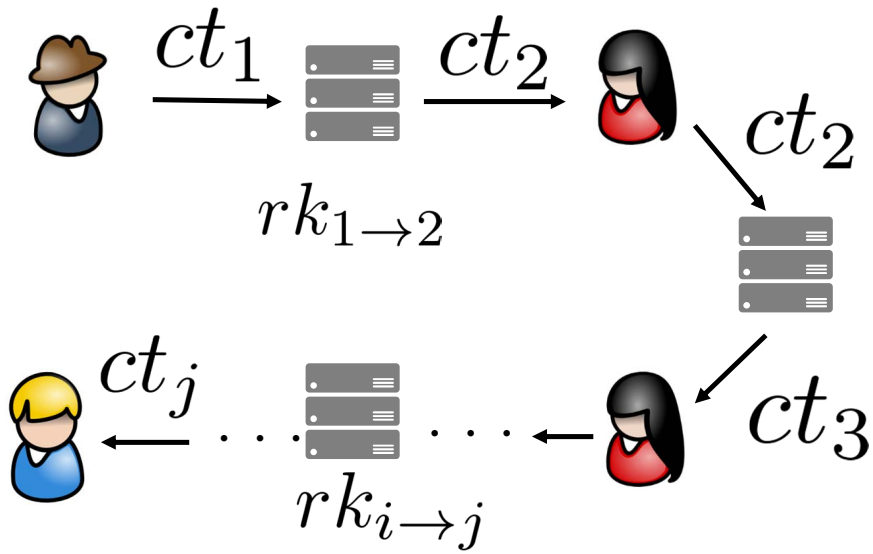
Re-Encryption (functional)



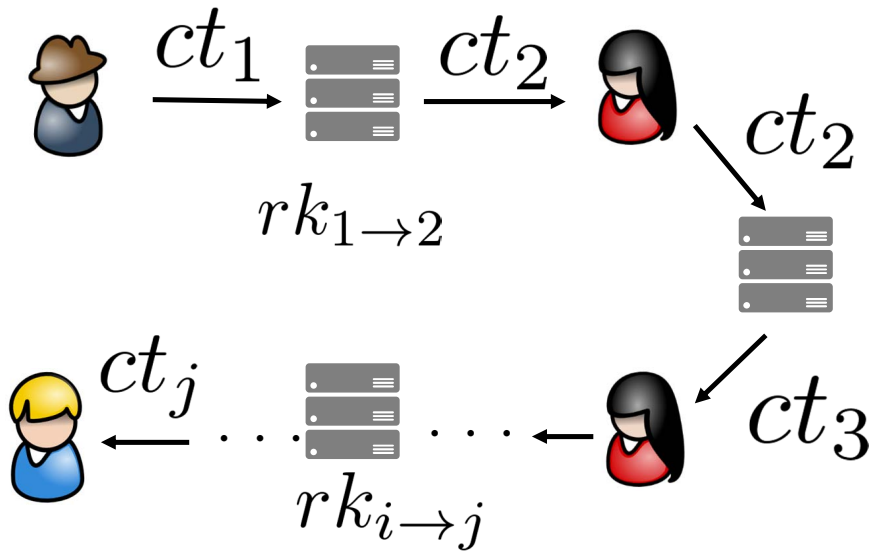
Re-Encryption (functional)



Re-Encryption (multi-hop)



Re-Encryption (multi-hop)



Applications of Re-Encryption

Secure distributed file servers, Outsource filtering of encrypted spam, iTunes DRM system, Constructing FHE, ABE...

Outline

Why Obfuscating Re-Encryption?

- ✔ Strong security

Why Obfuscating Re-Encryption?

- ✔ Strong security
- ✔ Clean and easy definition

Why Obfuscating Re-Encryption?

- ✔ Strong security
- ✔ Clean and easy definition
- ✔ More positive results on obfuscation

Why Obfuscating Re-Encryption?

- ✔ Strong security
- ✔ Clean and easy definition
- ✔ More positive results on obfuscation

NOTE on this talk

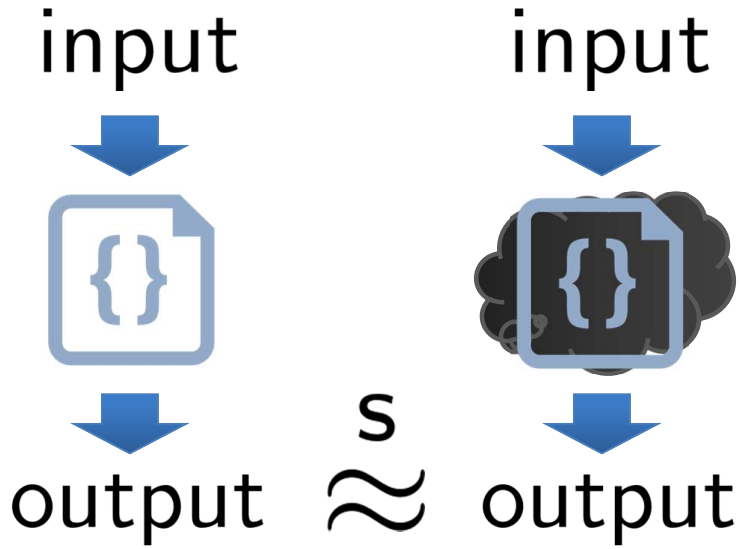
Virtual black-box obfuscation

Not iO

Outline

Results and Comparisons 1

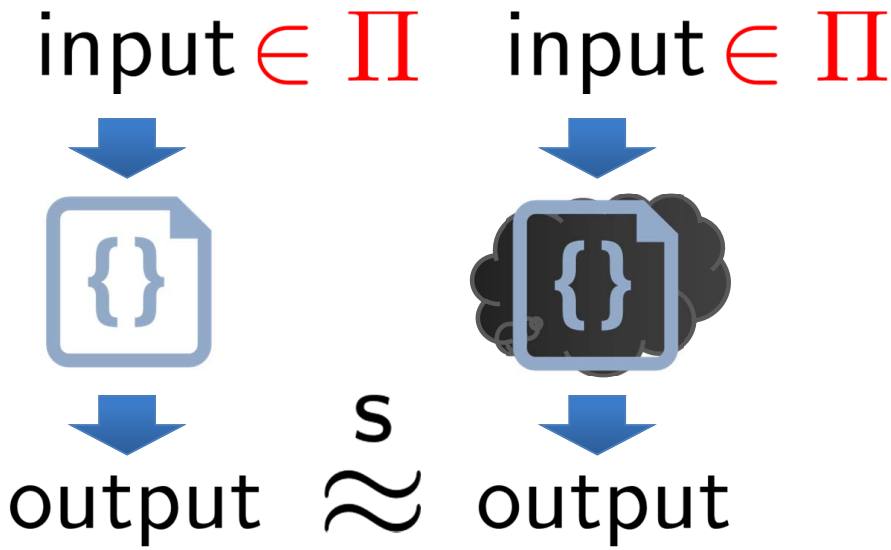
- New Definition for Correctness of Obfuscation



Reference	input	output distribution
[HRsV12]	all	statistically indistinguishable

Results and Comparisons 1

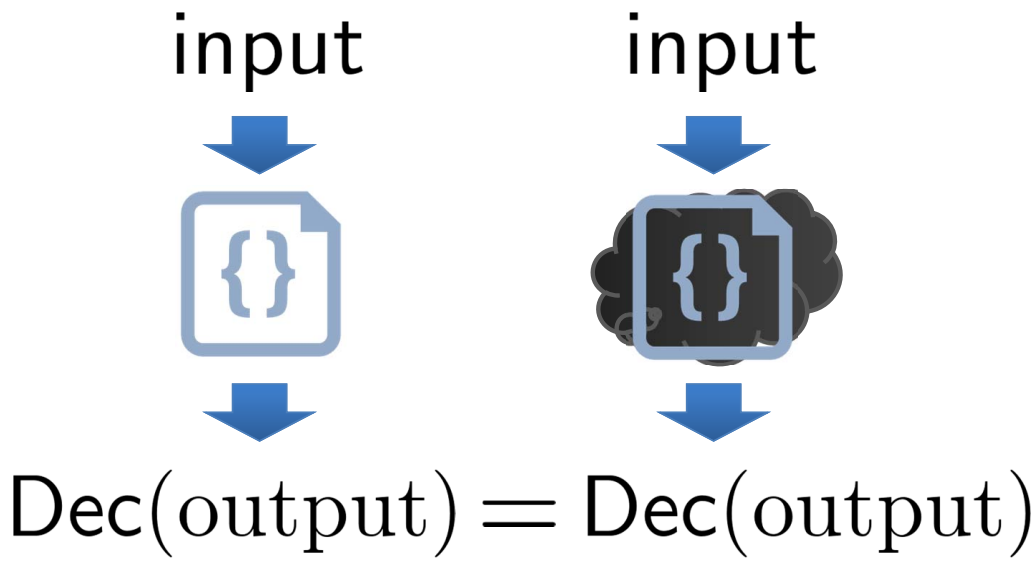
- New Definition for Correctness of Obfuscation



Reference	input	output distribution
[HRsV12]	all	statistically indistinguishable
This work	restricted	statistically indistinguishable

Results and Comparisons 1

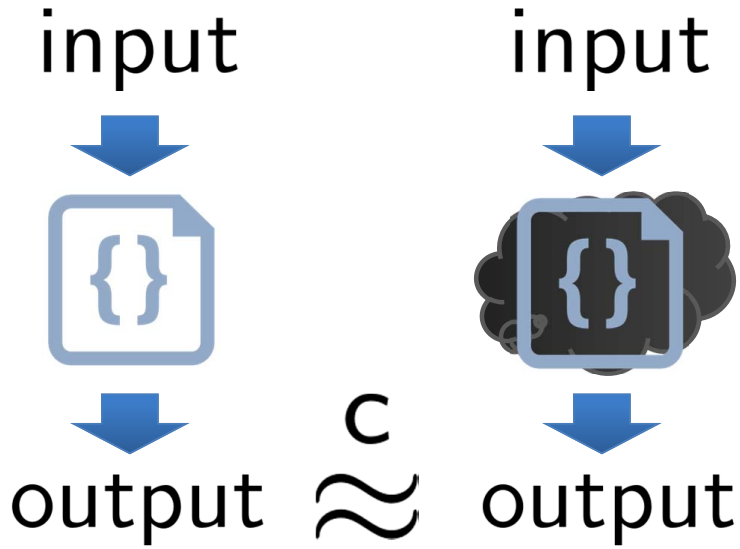
- New Definition for Correctness of Obfuscation



Reference	input	output distribution
[HRsV12]	all	statistically indistinguishable
This work	restricted	statistically indistinguishable
This work	all	same value under Decryption

Results and Comparisons 1

- New Definition for Correctness of Obfuscation



Reference	input	output distribution
[HRsV12]	all	statistically indistinguishable
This work	restricted	statistically indistinguishable
This work	all	same value under Decryption
This work	all	computationally indistinguishable

Results and Comparisons 2

- New Concrete Instantiations

Reference	Type	#Hop	Assumption
[HRsV12]	standard	single	DLIN & SDHI
[CCV12]	functional	signle	SXDH or DLIN
[Gen09,BV11]	standard	multi	LWE w/ FHE

Results and Comparisons 2

- New Concrete Instantiations

Reference	Type	#Hop	Assumption
[HRsV12]	standard	single	DLIN & SDHI
[CCV12]	functional	signle	SXDH or DLIN
[Gen09,BV11]	standard	multi	LWE w/ FHE
This work	standard	single	LWE
This work	functional	single	LWE
This work	standard	multi	LWE w/o FHE

3 instantiations from our new tools and unified framework.

Outline

LWE and Regev PKE

LWE assumption

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{N \times k},\ \boldsymbol{s} \leftarrow \mathbb{Z}_q^k,\ \boldsymbol{e} \leftarrow \chi^N$$

$$(\boldsymbol{A}, \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}) \overset{\mathsf{c}}{\approx} (\boldsymbol{A}, \boldsymbol{u})$$

where $\boldsymbol{u}$ is uniformly random over $\mathbb{Z}_q^N$

LWE and Regev PKE

LWE assumption

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{N \times k},\ \boldsymbol{s} \leftarrow \mathbb{Z}_q^k,\ \boldsymbol{e} \leftarrow \chi^N$$

$$(\boldsymbol{A}, \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}) \overset{\mathsf{c}}{\approx} (\boldsymbol{A}, \boldsymbol{u})$$

where $\boldsymbol{u}$ is uniformly random over $\mathbb{Z}_q^N$

Regev PKE

pk: $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$

$$ct_1 = \boldsymbol{r}^\top \boldsymbol{A}$$
$$ct_2 = \boldsymbol{r}^\top \boldsymbol{b} + m \cdot \lfloor q/2 \rceil$$

LWE and Regev PKE

LWE assumption

$$\boldsymbol{A} \leftarrow \mathbb{Z}_q^{N \times k},\ \boldsymbol{s} \leftarrow \mathbb{Z}_q^k,\ \boldsymbol{e} \leftarrow \chi^N$$

$$(\boldsymbol{A}, \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}) \overset{\mathsf{c}}{\approx} (\boldsymbol{A}, \boldsymbol{u})$$

where $\boldsymbol{u}$ is uniformly random over $\mathbb{Z}_q^N$

Regev PKE

pk: $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$

$$ct_1 = \boldsymbol{r}^\top \boldsymbol{A}$$
$$ct_2 = \boldsymbol{r}^\top \boldsymbol{b} + m \cdot \lfloor q/2 \rceil \quad = \boldsymbol{r}^\top \boldsymbol{A}\boldsymbol{s} + \boldsymbol{r}^\top \boldsymbol{e} + m \cdot \lfloor q/2 \rceil$$

Key point: small error (noise): $\boldsymbol{r}^\top \boldsymbol{e}$

Outline

Correctness (Preserving Functionality)

Original Correctness (specialized for Re-Enc)

$\mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$

Correctness (Preserving Functionality)

Original Correctness (specialized for Re-Enc)

$\mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$

On invalid ciphertexts

- Require "independent" and "small" noise in lattice-based crypto

Correctness (Preserving Functionality)

Original Correctness (specialized for Re-Enc)

$\mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$

On invalid ciphertexts

- Require "independent" and "small" noise in lattice-based crypto
- Arbitrary inputs include invalid ciphertext

New Relaxed Correctness

Restricted inputs

$$ct \in \Pi,\ \mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$$

Concretely, Π is a set of honestly generated ciphertext

New Relaxed Correctness

Restricted inputs

$$ct \in \Pi,\ \mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$$

Concretely, Π is a set of honestly generated ciphertext

Under decryption

$$\mathsf{Dec}(\mathsf{ReEnc}_{rk}(ct)) = \mathsf{Dec}(\mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct))$$

New Relaxed Correctness

Restricted inputs

$$ct \in \Pi,\ \mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{s}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$$

Concretely, Π is a set of honestly generated ciphertext

Under decryption

$$\mathsf{Dec}(\mathsf{ReEnc}_{rk}(ct)) = \mathsf{Dec}(\mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct))$$

Computational indistinguishability

$$\mathsf{ReEnc}_{rk}(ct) \overset{\mathsf{c}}{\approx} \mathsf{Obf}(\mathsf{ReEnc}_{rk})(ct)$$

Note: We can apply to arbitrary programs

Outline

Key-Switching

Key-Switching

$\mathsf{Enc}(pk, m)$

- $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$

Key-Switching

Key-Switching

$\mathsf{Enc}(pk, m) \Rightarrow \mathsf{Switch}(K_{pk \to \widehat{pk}}, \cdot) \Rightarrow \mathsf{Enc}(\widehat{pk}, m)$

- $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$

Key-Switching

Key-Switching

$\mathsf{Enc}(pk, m) \Rightarrow \mathsf{Switch}(K_{pk \to \widehat{pk}}, \cdot) \Rightarrow \mathsf{Enc}(\widehat{pk}, m)$

- $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$
- $\widetilde{K}_{pk \to \widehat{pk}} \leftarrow \mathsf{SimSwitchGen}(\widehat{pk})$

Key-Switching

Key-Switching

$$\mathsf{Enc}(pk, m) \Rightarrow \mathsf{Switch}(K_{pk \to \widehat{pk}}, \cdot) \Rightarrow \mathsf{Enc}(\widehat{pk}, m)$$

- $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$
- $\widetilde{K}_{pk \to \widehat{pk}} \leftarrow \mathsf{SimSwitchGen}(\widehat{pk})$

Security

$$K_{pk \to \widehat{pk}} \overset{\mathsf{c}}{\approx} \widetilde{K}_{pk \to \widehat{pk}}$$

Key-Switching

Key-Switching

$$\mathsf{Enc}(pk, m) \Rightarrow \mathsf{Switch}(K_{pk \to \widehat{pk}}, \cdot) \Rightarrow \mathsf{Enc}(\widehat{pk}, m)$$

- $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$
- $\widetilde{K}_{pk \to \widehat{pk}} \leftarrow \mathsf{SimSwitchGen}(\widehat{pk})$

Security

$$\mathsf{LWE.Enc}(\widehat{pk}, sk) = K_{pk \to \widehat{pk}} \overset{c}{\approx} \widetilde{K}_{pk \to \widehat{pk}} = \mathsf{LWE.Enc}(\widehat{pk}, 0)$$

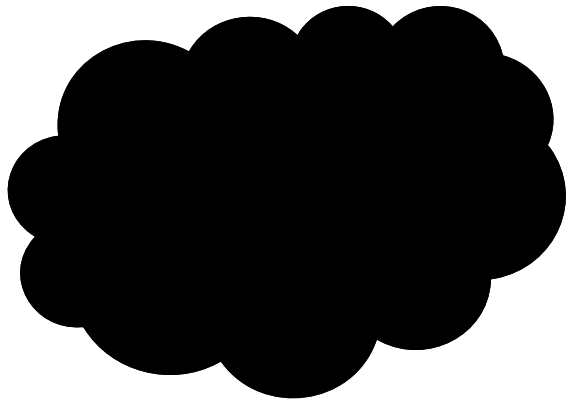
Intuitively, $K_{pk \to \widehat{pk}} = \mathsf{LWE.Enc}(\widehat{pk}, sk)$ (enc of sk) [BV11,B12]

Outline

Blurring (Re-Randomization LWE-based ciphertext)

Arbitrary value: ct

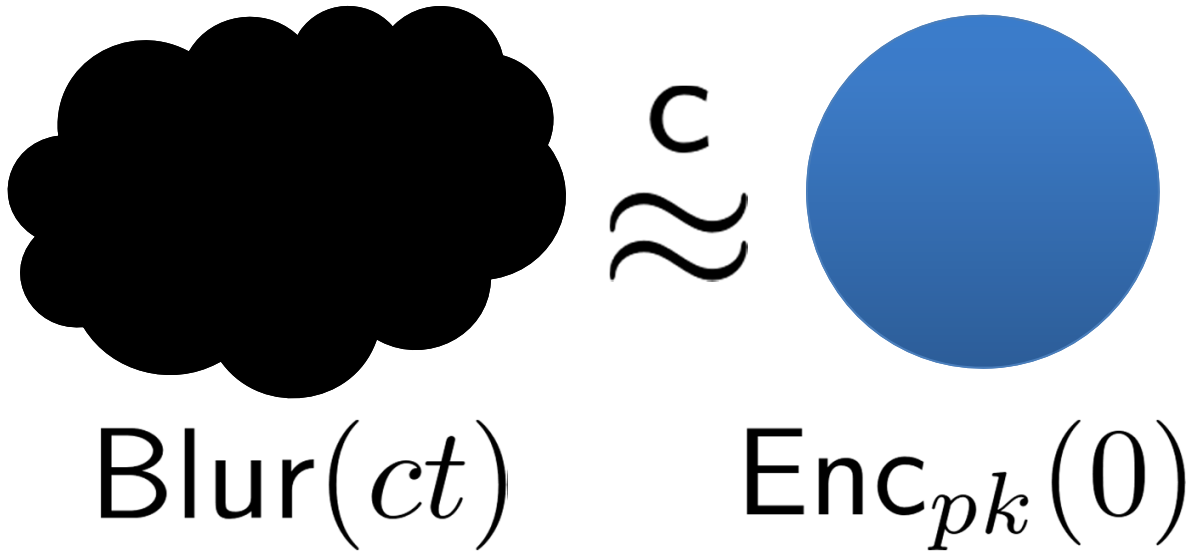
Blurring (Re-Randomization LWE-based ciphertext)



Arbitrary value: ct

Blurring the distribution of ct

Blurring (Re-Randomization LWE-based ciphertext)

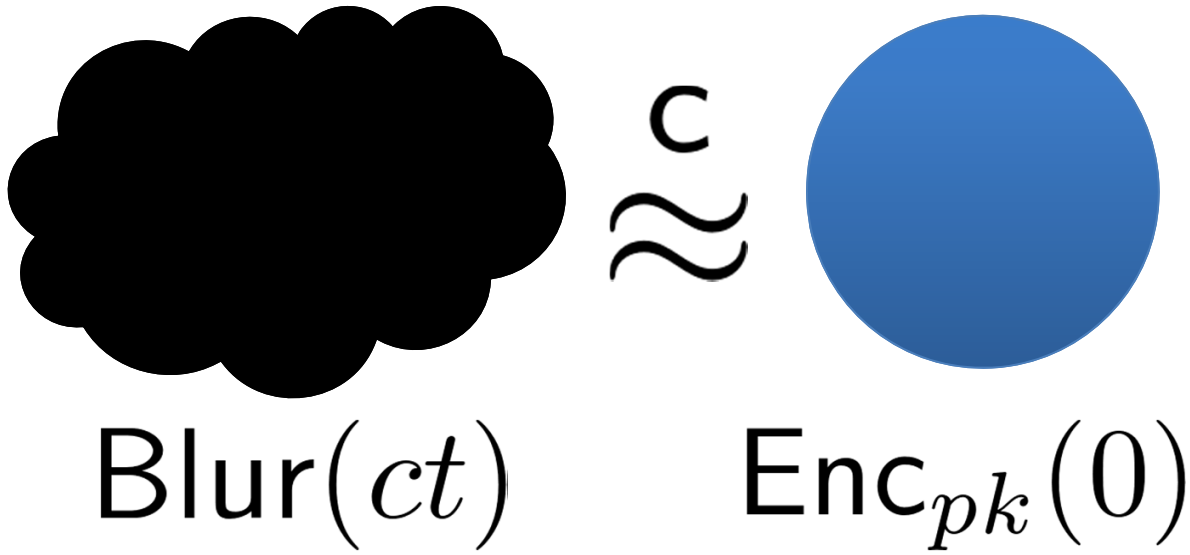


Arbitrary value: ct

Blurring the distribution of ct

$\mathsf{Blur}(pk, ct) := ct + \mathsf{Enc}_{pk}(0)$ $\quad (\mathsf{Dec}(ct) = \mathsf{Dec}(\mathsf{Blur}(ct)))$

Blurring (Re-Randomization LWE-based ciphertext)



Arbitrary value: ct

Blurring the distribution of ct

$\mathsf{Blur}(pk, ct) := ct + \mathsf{Enc}_{pk}(0)$ $\quad (\mathsf{Dec}(ct) = \mathsf{Dec}(\mathsf{Blur}(ct)))$

Strong Blurring: $\mathsf{Blur}(ct) \overset{s}{\approx} \mathsf{Blur}(\mathsf{Enc}_{pk}(m))$

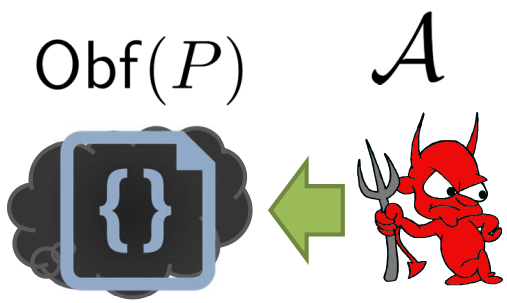
Outline

Obfuscator for Re-Encryption (Basic Idea)

Obfuscator

Input: (pk, sk), $\widehat{pk}$

Output: $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$



Real

Obfuscator for Re-Encryption (Basic Idea)

Obfuscator

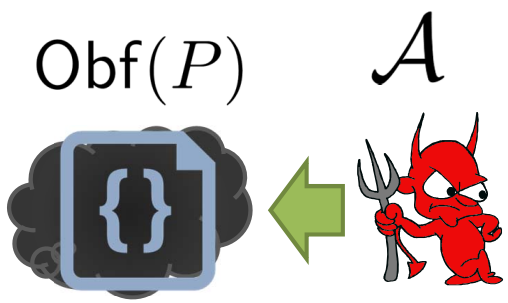
Input: (pk, sk), $\widehat{pk}$

Output: $K_{pk \to \widehat{pk}} \leftarrow \mathsf{SwitchGen}(pk, sk, \widehat{pk})$

Execution of Obfuscated Program

Input: ct

Output: $\widetilde{ct} \leftarrow \mathsf{Blur}(\widehat{pk}, \mathsf{Switch}(K_{pk \to \widehat{pk}}, ct))$



Obfuscator for Re-Encryption (Basic Idea)

Simulated Obfuscator

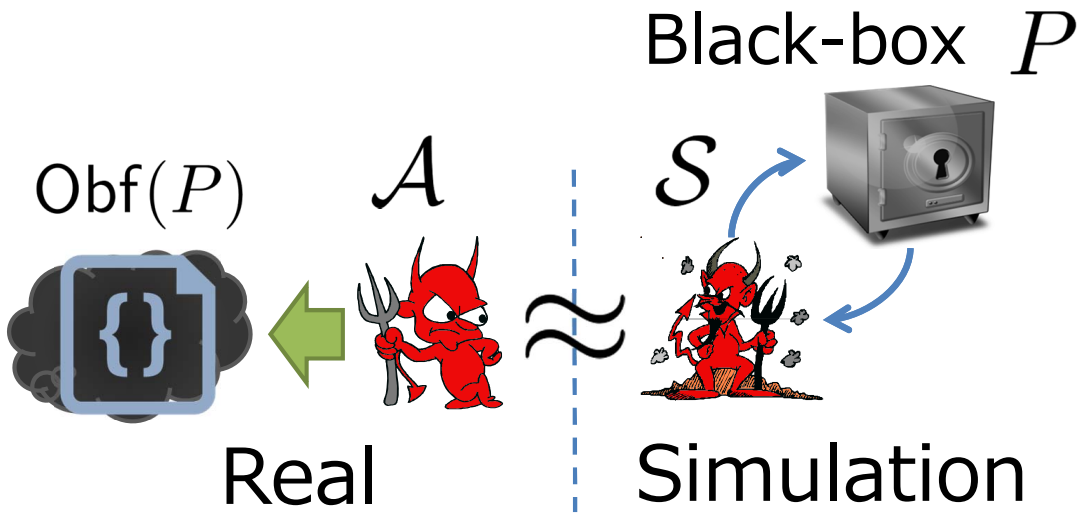
Input: $\widehat{pk}$

Output: $\widetilde{K}_{pk \to \widehat{pk}} \leftarrow \mathsf{SimSwitchGen}(\widehat{pk})$

Execution of Obfuscated Program

Input: ct

Output: $\widetilde{ct} \leftarrow \mathsf{Blur}(\widehat{pk}, \mathsf{Switch}(K_{pk \to \widehat{pk}}, ct))$



Outline

Summary

A New Framework for Obfuscating Re-Encryption

- New definition of correctness
  1. statistical indistinguishability for restricted inputs
  2. same value under decryption for all inputs
  3. computational indistinguishability for all inputs
- Standard , functional, and multi-hop re-encryption
- Key-switching and blurring mechanism
- Instantiations from LWE-based PKE

Summary

A New Framework for Obfuscating Re-Encryption

- New definition of correctness
  1. statistical indistinguishability for restricted inputs
  2. same value under decryption for all inputs
  3. computational indistinguishability for all inputs
- Standard , functional, and multi-hop re-encryption
- Key-switching and blurring mechanism
- Instantiations from LWE-based PKE

Thank you.
Q?