RUHR-UNIVERSITÄT BOCHUM

Lattice-based Proxy Re-encryption

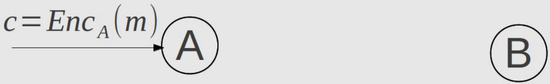
PKC 2014, 26.03.14

Elena Kirshanova
Horst Görtz Institute for IT Security
Ruhr University Bochum

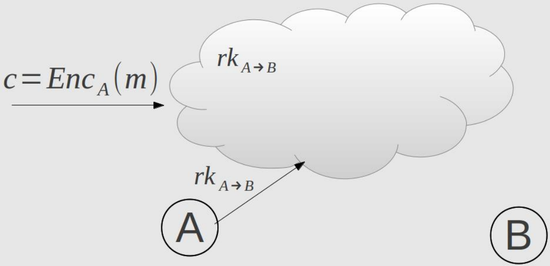
Horst Görtz Institut
für IT-Sicherheit

Outline

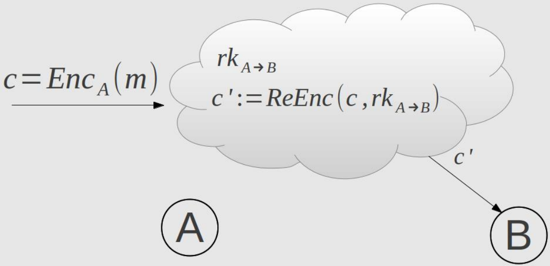
The informal definition of a Proxy Re-Encyption



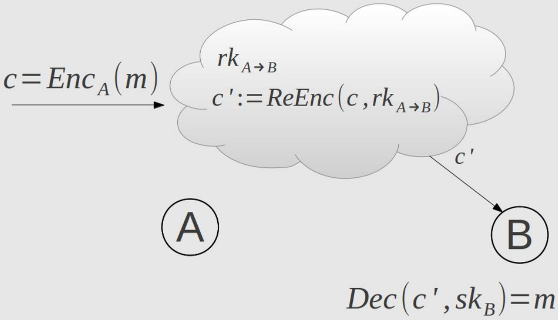
The informal definition of a Proxy Re-Encyption



The informal definition of a Proxy Re-Encyption



The informal definition of a Proxy Re-Encyption



The formal Definition

Definition 1 (Proxy Re-Encryption)

A *unidirectional* Proxy Re-Encryption (PRE) is a tuple of algorithms:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$
- $c_{\mathsf{pk}} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$

The formal Definition

Definition 1 (Proxy Re-Encryption)

A *unidirectional* Proxy Re-Encryption (PRE) is a tuple of algorithms:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$
- $c_{\mathsf{pk}} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
- $\mathsf{rk}_{\mathsf{pk} \rightarrow \mathsf{pk}'} \leftarrow \mathsf{ReKeyGen}(\mathsf{pk}, \mathsf{sk}, \mathsf{pk}')$

The formal Definition

Definition 1 (Proxy Re-Encryption)

A *unidirectional* Proxy Re-Encryption (PRE) is a tuple of algorithms:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$
- $c_{\mathsf{pk}} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
- $m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
- $\mathsf{rk}_{\mathsf{pk} \to \mathsf{pk}'} \leftarrow \mathsf{ReKeyGen}(\mathsf{pk}, \mathsf{sk}, \mathsf{pk}')$
- $c' \leftarrow \mathsf{ReEnc}(\mathsf{rk}_{\mathsf{pk} \to \mathsf{pk}'}, c_{\mathsf{pk}})$

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$	$\mathcal{A}$

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	$\xrightarrow{\mathsf{pk}^*}$	

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	$\xrightarrow{\mathsf{pk}^*}$	
	$\xrightarrow{(\mathsf{pk}, \mathsf{pk}')}$	
	$\xrightarrow{\mathsf{rk}_{\mathsf{pk} \to \mathsf{pk}'}}$	

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	pk^* $\longrightarrow$	
	$(\mathsf{pk}, \mathsf{pk}')$ $\longrightarrow$	
	$\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'}$ $\longrightarrow$	
	$(\mathsf{Dec}(c), \mathsf{pk})$ $\longleftarrow$	
	$\ldots$	

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	pk^* →	
	$(\mathsf{pk}, \mathsf{pk}')$ →	
	$\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'}$ →	
	← $(\mathsf{Dec}(c), \mathsf{pk})$	
	$\dots$	
	← m_0, m_1	$m_0, m_1 \in \mathcal{M}$

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	pk^* $\longrightarrow$	
	$(\mathsf{pk}, \mathsf{pk}')$ $\longrightarrow$	
	$\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'}$ $\longrightarrow$	
	$\longleftarrow$ $(\mathsf{Dec}(c), \mathsf{pk})$	
	$\dots$	
	$\longleftarrow$ m_0, m_1	$m_0, m_1 \in \mathcal{M}$
$b \leftarrow \{0,1\}$ $c^* = \mathsf{Enc}(\mathsf{pk}^*, m_b)$	c^* $\longrightarrow$	

PRE-CCA1 Security (simplified)

$\mathsf{PRE}^{\mathsf{CCA1}}_{\mathcal{A},\Pi}(n)$		$\mathcal{A}$
	$\xrightarrow{\mathsf{pk}^*}$	
	$\xrightarrow{(\mathsf{pk},\mathsf{pk}')}$	
	$\xrightarrow{\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'}}$	
	$\xleftarrow{(\mathsf{Dec}(c),\mathsf{pk})}$	
	$\ldots$	
	$\xleftarrow{m_0, m_1}$	$m_0, m_1 \in \mathcal{M}$
$b \leftarrow \{0,1\}$ $c^* = \mathsf{Enc}(\mathsf{pk}^*, m_b)$	$\xrightarrow{c^*}$	
If $b = b'$ output 1 else output 0	$\xleftarrow{b'}$	$b' \in \{0,1\}$

Desired properties of PRE schemes

- Unidirectional ($\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'} \neq \mathsf{rk}_{\mathsf{pk}'\to\mathsf{pk}}$)

Desired properties of PRE schemes

- Unidirectional ($\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'} \neq \mathsf{rk}_{\mathsf{pk}'\to\mathsf{pk}}$)
- Non-interactive ($\mathsf{ReKeyGen}(\mathsf{pk}, \mathbf{sk}, \mathsf{pk}')$)

Desired properties of PRE schemes

- Unidirectional ($\mathsf{rk}_{\mathsf{pk}\rightarrow\mathsf{pk}'} \neq \mathsf{rk}_{\mathsf{pk}'\rightarrow\mathsf{pk}}$)
- Non-interactive ($\mathsf{ReKeyGen}(\mathsf{pk}, \mathbf{sk}, \mathsf{pk}')$)
- Collusion ‘safe’

Desired properties of PRE schemes

- Unidirectional ($\mathsf{rk}_{\mathsf{pk}\to\mathsf{pk}'} \neq \mathsf{rk}_{\mathsf{pk}'\to\mathsf{pk}}$)
- Non-interactive ($\mathsf{ReKeyGen}(\mathsf{pk}, \mathbf{sk}, \mathsf{pk}')$)
- Collusion 'safe'
- Key optimal
- Non-transitive
- Proxy invisibility

RUB

Outline

PRE overview

	Unidirectional	Non-interactive	Collusion-safe	Assumption	Security Model
[BBS98]	✗	✗	✗	DDH	IND-CPA

PRE overview

	Unidirectional	Non-interactive	Collusion-safe	Assumption	Security Model
[BBS98]	✗	✗	✗	DDH	IND-CPA
[AFGH06]	✓	✓	✓	eDBDH	IND-CPA

PRE overview

	Unidirectional	Non-interactive	Collusion-safe	Assumption	Security Model
[BBS98]	✗	✗	✗	DDH	IND-CPA
[AFGH06]	✓	✓	✓	eDBDH	IND-CPA
[CH07]	✗	✗	✗	DBDH	IND-CCA

PRE overview

	Unidirectional	Non-interactive	Collusion-safe	Assumption	Security Model
[BBS98]	✗	✗	✗	DDH	IND-CPA
[AFGH06]	✓	✓	✓	eDBDH	IND-CPA
[CH07]	✗	✗	✗	DBDH	IND-CCA
[Xag10]	✗	✗	✗	LWE	IND-CPA

PRE overview

	Unidirectional	Non-interactive	Collusion-safe	Assumption	Security Model
[BBS98]	✗	✗	✗	DDH	IND-CPA
[AFGH06]	✓	✓	✓	eDBDH	IND-CPA
[CH07]	✗	✗	✗	DBDH	IND-CCA
[Xag10]	✗	✗	✗	LWE	IND-CPA
This work	✓	✓	✓	LWE	IND-CCA1

Main result

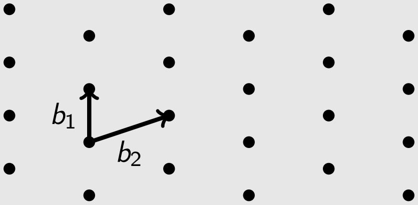
Theorem 2

Our unidirectional Proxy Re-Encryption scheme is IND-CCA1-secure assuming the hardness of decision-LWE.

Outline

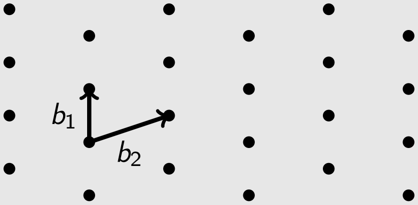
Lattice definition

- Lattice Λ of dimension m is a discrete additive subgroup of $\mathbb{Z}^m$.



Lattice definition

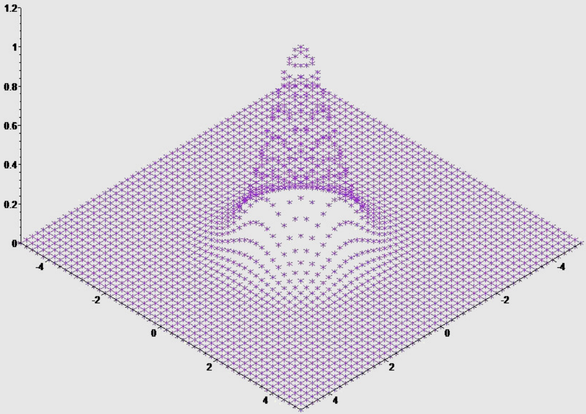
- Lattice Λ of dimension m is a discrete additive subgroup of $\mathbb{Z}^m$.



- Basis $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} : \Lambda(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^k\}$.

Gaussians on Lattices

$$v \leftarrow D_{\Lambda,s} \Leftrightarrow v \propto \rho_s(\mathbf{x}) = \exp(-\frac{\pi\|\mathbf{x}\|^2}{s^2})$$



One-way functions from lattices

- Public $\left[\mathbf{A}\right] \in \mathbb{Z}_q^{n\times m}$, $q = \mathrm{poly}(n), m \approx n \log q$

One-way functions from lattices

- Public $[\mathbf{A}] \in \mathbb{Z}_q^{n \times m}$, $q = \text{poly}(n), m \approx n \log q$

SIS	LWE
$\mathbf{u} := f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \mod q \in \mathbb{Z}_q^n$	$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \mod q \in \mathbb{Z}_q^m$

One-way functions from lattices

- Public $\left[\mathbf{A}\right] \in \mathbb{Z}_q^{n\times m}$, $q = \text{poly}(n), m \approx n \log q$

SIS	LWE
$\mathbf{u} := f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \mod q \in \mathbb{Z}_q^n$	$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \mod q \in \mathbb{Z}_q^m$
$f_{\mathbf{A}}^{-1}$: sample $\mathbf{x}' \leftarrow D_{\Lambda_{\mathbf{u}},s}$ s.t. $\mathbf{A}\mathbf{x}' = \mathbf{u}$	$g_{\mathbf{A}}^{-1}$: find the unique $\mathbf{s}$ (or $\mathbf{e}$)

G-trapdoor [PM12]

- For a uniform $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times \bar{m}}$ and a short $\mathbf{R} \leftarrow \mathbb{Z}^{\bar{nk} \times nk}$ define

$$\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{G}] \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}]$$

for some $\mathbf{G}$ with easy $f_{\mathbf{G}}^{-1}$ and $g_{\mathbf{G}}^{-1}$.

G-trapdoor [PM12]

- For a uniform $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times \bar{m}}$ and a short $\mathbf{R} \leftarrow \mathbb{Z}^{\bar{nk} \times nk}$ define

$$\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{G}] \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}]$$

  for some $\mathbf{G}$ with easy $f_{\mathbf{G}}^{-1}$ and $g_{\mathbf{G}}^{-1}$.
- $[\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{R}]$ is uniform by the leftover hash lemma, so is $\mathbf{A}$.

G-trapdoor [PM12]

- For a uniform $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times \bar{m}}$ and a short $\mathbf{R} \leftarrow \mathbb{Z}^{\bar{nk} \times nk}$ define

$$\mathbf{A} = [\mathbf{A}_0 \mid \mathbf{G}] \begin{bmatrix} \mathbf{I} & -\mathbf{R} \\ & \mathbf{I} \end{bmatrix} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}]$$

  for some $\mathbf{G}$ with easy $f_{\mathbf{G}}^{-1}$ and $g_{\mathbf{G}}^{-1}$.
- $[\mathbf{A}_0 \mid \mathbf{A}_0\mathbf{R}]$ is uniform by the leftover hash lemma, so is $\mathbf{A}$.
- $\mathbf{A} \cdot \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} = \mathbf{G}$

RUB

Outline

Extended G-trapdoor

- Idea: generate multiple $\mathbf{R}$-transformations

$$\mathbf{A} = [\mathbf{A}_0 \mid \underbrace{\mathbf{G} - \mathbf{A}_0\mathbf{R}_1}_{\text{trapdoor for } f_{\mathbf{A}}} \mid \overbrace{\mathbf{G} - \mathbf{A}_0\mathbf{R}_2}^{\text{trapdoor for } g_{\mathbf{A}}}]$$

- $\mathbf{R}_1$ allows to sample short vectors (i.e. generate rk)
- $\mathbf{R}_2$ allows to invert $\mathbf{s}^t\mathbf{A} + \mathbf{e}^t$ (i.e. decrypt)

Encryption

- $\mathsf{pk} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \in \mathbb{Z}_q^{n \times m}$, $\mathsf{sk} := [\mathbf{R}_1 \mid \mathbf{R}_2]$

Encryption

- $\text{pk} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \in \mathbb{Z}_q^{n \times m}$, $\text{sk} := [\mathbf{R}_1 \mid \mathbf{R}_2]$
- $\text{Enc}(mes, \text{pk})$:

$$\mathbf{c}_1 = \mathbf{s}^t \cdot \text{pk} + \mathbf{e}_1^t \mod q,$$
$$\mathbf{c}_2 = \mathbf{s}^t \cdot \mathbf{A}_{aux} + \mathbf{e}_2^t + \texttt{enc}(mes) \mod q,$$

for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_1, \mathbf{e}_2 \leftarrow D_s$, $\mathbf{A}_{aux} \xleftarrow{\$} \mathbb{Z}_q^{n \times nk}$ and $\texttt{enc}(mes) := mes \cdot \lfloor \frac{q}{2} \rfloor$.

Encryption

- $\mathsf{pk} = [\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \in \mathbb{Z}_q^{n \times m}$, $\mathsf{sk} := [\mathbf{R}_1 \mid \mathbf{R}_2]$
- $\mathsf{Enc}(mes, \mathsf{pk})$:

$$\mathbf{c}_1 = \mathbf{s}^t \cdot \mathsf{pk} + \mathbf{e}_1^t \mod q,$$
$$\mathbf{c}_2 = \mathbf{s}^t \cdot \mathbf{A}_{aux} + \mathbf{e}_2^t + \mathsf{enc}(mes) \mod q,$$

for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_1, \mathbf{e}_2 \leftarrow D_s, \mathbf{A}_{aux} \xleftarrow{\$} \mathbb{Z}_q^{n \times nk}$ and $\mathsf{enc}(mes) := mes \cdot \lfloor \frac{q}{2} \rfloor$.

- $\mathsf{Dec}(\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{A}_{aux}), sk)$: recover $\mathbf{s}$ using $\mathbf{R}_2$:

$$\mathbf{c}_1 \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} = \mathbf{s}^t[\mathbf{G}] + \widetilde{\mathbf{e}}^t \mod q.$$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathrm{pk} + \mathbf{e}^t \rightarrow \mathbf{c}_1' = \mathbf{s}^t \cdot \mathrm{pk}' + \widetilde{\mathbf{e}}^t$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathsf{pk} + \mathbf{e}^t \to \mathbf{c}_1' = \mathbf{s}^t \cdot \mathsf{pk}' + \widetilde{\mathbf{e}}^t$

$$\mathbf{c} = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \to$$
$$\mathbf{c}' = \mathbf{s}^t[\mathbf{A}_0' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2'] + \widetilde{\mathbf{e}}^t$$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathrm{pk} + \mathbf{e}^t \rightarrow \mathbf{c}_1' = \mathbf{s}^t \cdot \mathrm{pk}' + \widetilde{\mathbf{e}}^t$

$$\mathbf{c} = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \rightarrow$$
$$\mathbf{c}' = \mathbf{s}^t[\mathbf{A}_0' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2'] + \widetilde{\mathbf{e}}^t$$

- Use $\mathbf{R}_1$ to sample Gaussian $\mathbf{x}$ for a vector $\mathbf{a}$:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot \mathbf{x} = \mathbf{a}$$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathsf{pk} + \mathbf{e}^t \to \mathbf{c}_1' = \mathbf{s}^t \cdot \mathsf{pk}' + \widetilde{\mathbf{e}}^t$

$$\mathbf{c} = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \to$$
$$\mathbf{c}' = \mathbf{s}^t[\mathbf{A}_0' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2'] + \widetilde{\mathbf{e}}^t$$

- Use $\mathbf{R}_1$ to sample Gaussian $\mathbf{x}$ for a vector $\mathbf{a}$:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot \mathbf{x} = \mathbf{a}$$

- Extend to matrices in column-wise way:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot [\mathbf{x}_1, \ldots, \mathbf{x}_n] = \underbrace{[\mathbf{a}_1, \ldots, \mathbf{a}_n]}_{\mathbf{A}_0'}$$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathsf{pk} + \mathbf{e}^t \to \mathbf{c}'_1 = \mathbf{s}^t \cdot \mathsf{pk}' + \widetilde{\mathbf{e}}^t$

$$\mathbf{c} = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \to$$
$$\mathbf{c}' = \mathbf{s}^t[\mathbf{A}'_0 \mid \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_1 \mid \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_2] + \widetilde{\mathbf{e}}^t$$

- Use $\mathbf{R}_1$ to sample Gaussian $\mathbf{x}$ for a vector $\mathbf{a}$:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot \mathbf{x} = \mathbf{a}$$

- Extend to matrices in column-wise way:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot [\mathbf{x}'_1, \ldots, \mathbf{x}'_n] = \underbrace{[\mathbf{a}'_1, \ldots, \mathbf{a}'_n]}_{\mathbf{G} - \mathbf{A}'_0\mathbf{R}'_1}$$

Re-Encrytion key generation

- **Goal:** transform $\mathbf{c}_1 = \mathbf{s}^t \cdot \mathrm{pk} + \mathbf{e}^t \rightarrow \mathbf{c}'_1 = \mathbf{s}^t \cdot \mathrm{pk}' + \widetilde{\mathbf{e}}^t$

$$\mathbf{c} = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \rightarrow$$
$$\mathbf{c}' = \mathbf{s}^t[\mathbf{A}'_0 \mid \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_1 \mid \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_2] + \widetilde{\mathbf{e}}^t$$

- Use $\mathbf{R}_1$ to sample Gaussian $\mathbf{x}$ for a vector $\mathbf{a}$:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot \mathbf{x} = \mathbf{a}$$

- Extend to matrices in column-wise way:

$$[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1] \cdot [\mathbf{x}''_1, \ldots, \mathbf{x}''_n] = \underbrace{[\mathbf{a}''_1, \ldots, \mathbf{a}''_n]}_{\mathbf{G} - \mathbf{A}'_0\mathbf{R}'_2}$$

Re-Encrytion key generation

$$\mathsf{pk} = \underbrace{[\mathbf{A}_0 | \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 |}\ \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \xrightarrow{\mathsf{rk}} \mathsf{pk}' = [\mathbf{A}_0' |\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' |\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2']$$

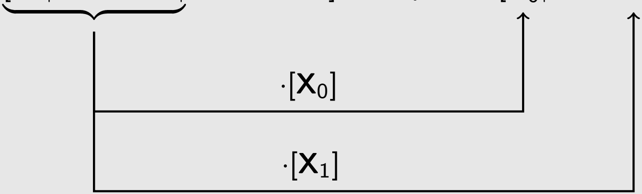
Re-Encrytion key generation

$$\mathsf{pk} = \underbrace{[\mathbf{A}_0 | \mathbf{G} - \mathbf{A}_0\mathbf{R}_1|}\ \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \xrightarrow{\mathsf{rk}} \mathsf{pk}' = [\mathbf{A}'_0|\ \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_1|\ \mathbf{G} - \mathbf{A}'_0\mathbf{R}'_2]$$

$$\cdot[\mathbf{X}_0]$$

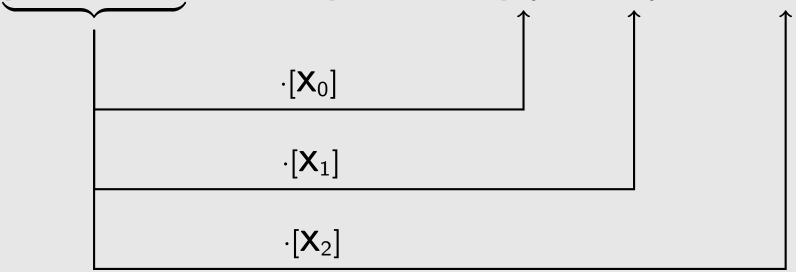
Re-Encrytion key generation

$$pk = \underbrace{[\mathbf{A}_0 | \mathbf{G} - \mathbf{A}_0\mathbf{R}_1|}\ \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \xrightarrow{rk} pk' = [\mathbf{A}_0'|\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1'|\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2']$$



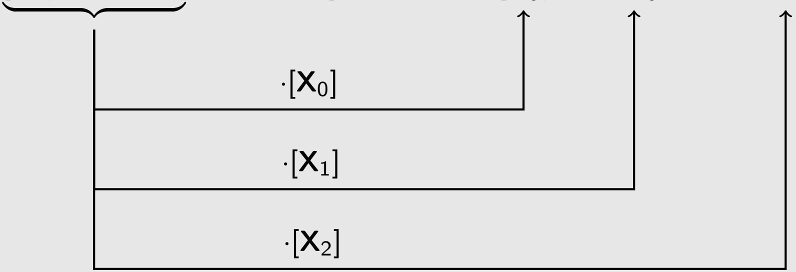
Re-Encrytion key generation

$$\mathsf{pk} = \underbrace{[\mathbf{A}_0 | \mathbf{G} - \mathbf{A}_0\mathbf{R}_1|}\ \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \xrightarrow{\mathsf{rk}} \mathsf{pk}' = [\mathbf{A}_0'|\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1'|\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2']$$



Re-Encrytion key generation

$$\mathrm{pk} = \underbrace{[\mathbf{A}_0 | \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 |}\ \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] \xrightarrow{\mathrm{rk}} \mathrm{pk}' = [\mathbf{A}_0' |\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' |\ \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2']$$



$$\mathrm{rk}_{\mathrm{pk}\to\mathrm{pk}'} = \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \in \mathbb{Z}^{m\times m}, \text{ where all } \mathbf{X} \text{ are gaussian.}$$

Re-Encryption

So for $\mathbf{c}_1 = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \mod q$

Re-Encryption

So for $\mathbf{c}_1 = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \mod q$

- $\mathbf{c}_1' = \mathsf{ReEnc}(\mathbf{c}_{pk}, rk_{\mathsf{pk}\rightarrow\mathsf{pk}'}) = \mathbf{c}_{pk} \cdot \mathsf{rk}_{\mathsf{pk}\rightarrow\mathsf{pk}'}$

Re-Encryption

So for $\mathbf{c}_1 = \mathbf{s}^t[\mathbf{A}_0 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_1 \mid \mathbf{G} - \mathbf{A}_0\mathbf{R}_2] + \mathbf{e}^t \mod q$

- $\mathbf{c}_1' = \mathrm{ReEnc}(\mathbf{c}_{pk}, rk_{\mathrm{pk}\to\mathrm{pk}'}) = \mathbf{c}_{pk} \cdot \mathrm{rk}_{\mathrm{pk}\to\mathrm{pk}'}$
- $\mathbf{c}_1' = \mathbf{s}^t[\mathbf{A}_0' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_1' \mid \mathbf{G} - \mathbf{A}_0'\mathbf{R}_2'] + \widetilde{\mathbf{e}}^t \mod q,$

where $\widetilde{\mathbf{e}}^t = (\mathbf{e}_0, \mathbf{e}_1)^t \cdot \begin{bmatrix} \mathbf{X}_0 & \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$ is as small as

$$\approx \sqrt{3} \cdot \|\mathbf{e}_0\mathbf{X}_2 + \mathbf{e}_1\|.$$

Summary

Proxy re-encryption scheme that

- is based on hard problems on lattices
- is unidirectional
- does not require a trusted party to generate re-encryption keys
- uses the 'Extended $\mathbf{G}$-trapdoor'.

Summary

Proxy re-encryption scheme that

- is based on hard problems on lattices
- is unidirectional
- does not require a trusted party to generate re-encryption keys
- uses the 'Extended $\mathbf{G}$-trapdoor'.

Many thanks for your attention!

Reference I

- Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger.
  Improved proxy re-encryption schemes with applications to secure distributed storage.
  In *ACM TISSEC*, pages 29–43, 2006.
- Matt Blaze, Gerrit Bleumer, and Martin Strauss.
  Divertible protocols and atomic proxy cryptography.
  In *EUROCRYPT*, pages 127–144. Springer-Verlag, 1998.
- Ran Canetti and Susan Hohenberger.
  Chosen-ciphertext secure proxy re-encryption.
  In *Proc. of ACM-CCS'007*, pages 185–194. ACM Press, 2007.

Reference II

Chris Peikert and Daniele Micciancio.
Trapdoors for lattices: Simpler, tighter, faster, smaller.
In *EUROCRYPT*, pages 700–718, 2012.

Keita Xagawa.
Cryptography with Lattices.
PhD thesis, Tokyo Institute of Technology, 2010.