On the Learning Parity with Noise Problem

Luca Melis

Università degli Studi di Firenze

Århus Universitet

19 Aprile 2013

Advisors:
Prof. Alessandro Piva
Prof. Fabrizio Argenti

Co-advisors:
Dr. Claudio Orlandi
Prof. Ivan Damgård

Scenario

Protezione dei contenuti digitali

- Il commercio elettronico non è ancora percepito come sicuro
- Risulta difficile proteggere il diritto d'autore
- tecnologie disponibili: protocolli Buyer-Seller

Learning Parity with Noise Problem LPN

- Dimension ℓ (security parameter)
- **Search**: find $\boldsymbol{s} \in \mathbb{Z}_2^\ell$ given "noisy random inner products"

$$\begin{aligned}
\boldsymbol{a_1} &\xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_1 =< \boldsymbol{a_1} , \boldsymbol{s} > \oplus e_1 \\
\boldsymbol{a_2} &\xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_2 =< \boldsymbol{a_2} , \boldsymbol{s} > \oplus e_2 \\
&\vdots \\
\boldsymbol{a_q} &\xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_q =< \boldsymbol{a_q} , \boldsymbol{s} > \oplus e_q
\end{aligned}$$

Errors $e_i \leftarrow \chi$ = Bernoulli over $\mathbb{Z}_2$, param $\tau \in \left(0, \frac{1}{2}\right]$

- **Decision**: distinguish $(\boldsymbol{a_i}, \boldsymbol{b_i})$ from uniform $(\boldsymbol{a_i}, b_i)$

Learning Parity with Noise Problem LPN

- Dimension ℓ (security parameter)
- **Search**: find $\boldsymbol{s} \in \mathbb{Z}_2^\ell$ given "noisy random inner products"

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a_1} \\ \vdots \\ \boldsymbol{a_q} \end{pmatrix}, \boldsymbol{b} = \boldsymbol{A} \cdot \boldsymbol{s} \oplus \boldsymbol{e}$$

Errors $e_i \leftarrow \chi$ = Bernoulli over $\mathbb{Z}_2$, param $\tau \in \left(0, \frac{1}{2}\right]$

- **Decision**: distinguish $(\boldsymbol{a_i}, \boldsymbol{b_i})$ from uniform $(\boldsymbol{a_i}, b_i)$

Protocol phases:

- **Key Generation**
- Key Assembly
- Encryption
- Partial Decryption
- Finish Decryption

Protocol phases:

- **Key Generation**
- **Key Assembly**
- Encryption
- Partial Decryption
- Finish Decryption

Protocol phases:

- **Key Generation**
- **Key Assembly**
- **Encryption**
- **Partial Decryption**
- **Finish Decryption**

Protocol phases:

- **Key Generation**
- **Key Assembly**
- **Encryption**
- **Partial Decryption**
- **Finish Decryption**

Protocol phases:

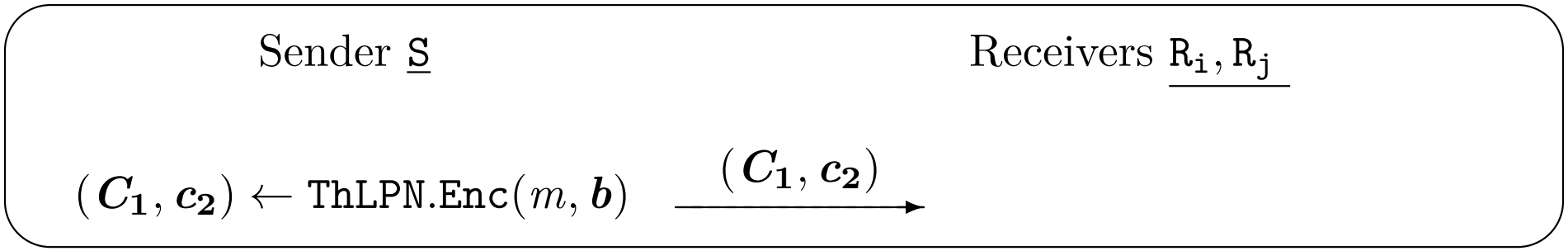
- **Key Generation**
- **Key Assembly**
- **Encryption**
- **Partial Decryption**
- **Finish Decryption**

Encryption

Encryption

Sender $\underline{\mathtt{S}}$	Receivers $\underline{\mathtt{R_i}, \mathtt{R_j}}$
$(\boldsymbol{C_1}, \boldsymbol{c_2}) \leftarrow \mathtt{ThLPN.Enc}(m, \boldsymbol{b})$	

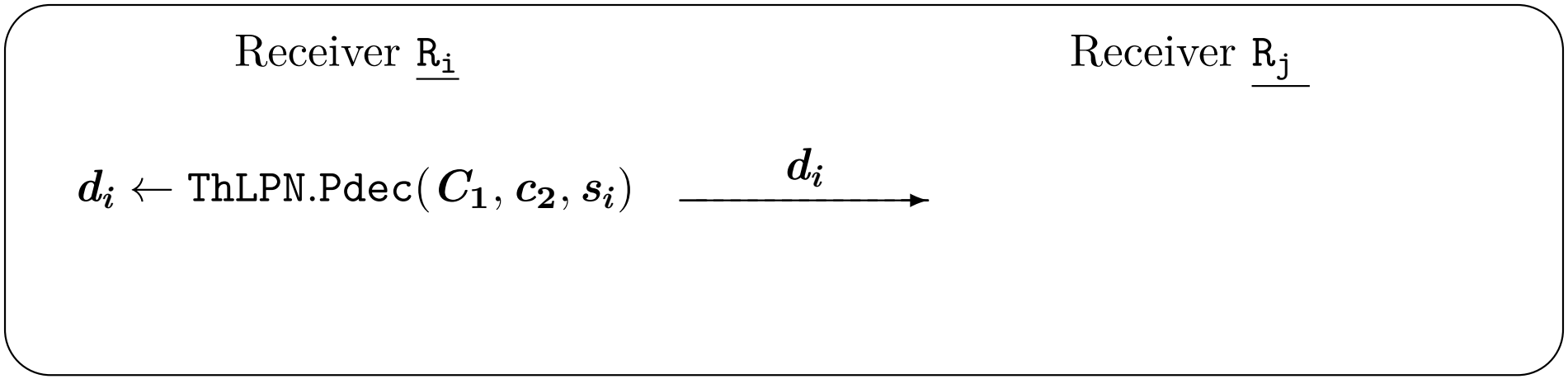
Encryption


Sender S
Receivers $\mathtt{R_i}, \mathtt{R_j}$
$(\boldsymbol{C_1}, \boldsymbol{c_2}) \leftarrow \mathtt{ThLPN.Enc}(m, \boldsymbol{b})$
$(\boldsymbol{C_1}, \boldsymbol{c_2})$

Partial Decryption

Receiver $\underline{\mathtt{R_i}}$	Receiver $\underline{\mathtt{R_j}}$
$\boldsymbol{d_i} \leftarrow \mathtt{ThLPN.Pdec}(\boldsymbol{C_1}, \boldsymbol{c_2}, \boldsymbol{s_i})$	

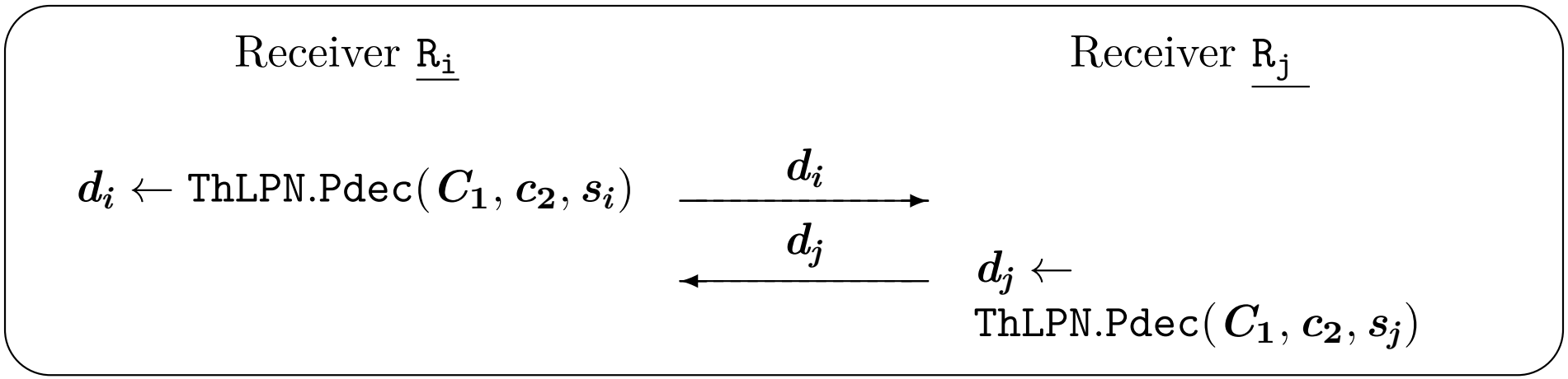
Partial Decryption

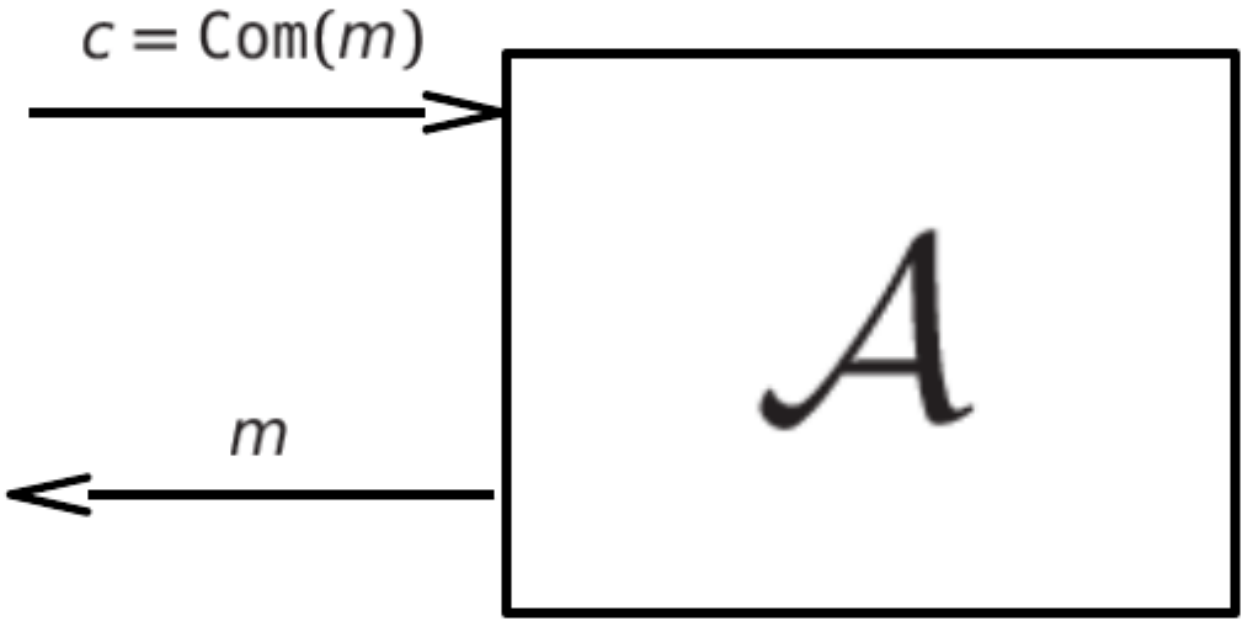


Partial Decryption

Receiver $\underline{\mathtt{R_i}}$		Receiver $\underline{\mathtt{R_j}}$
$d_i \leftarrow \mathtt{ThLPN.Pdec}(C_1, c_2, s_i)$	$\xrightarrow{\quad d_i \quad}$	$d_j \leftarrow \mathtt{ThLPN.Pdec}(C_1, c_2, s_j)$

Partial Decryption




$c = \mathsf{Com}(m)$
$\mathcal{A}$
m