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"

$$\boldsymbol{a_1} \xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_1 =< \boldsymbol{a_1} , \boldsymbol{s} > \oplus e_1$$

$$\boldsymbol{a_2} \xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_2 =< \boldsymbol{a_2} , \boldsymbol{s} > \oplus e_2$$

$$\vdots$$

$$\boldsymbol{a_q} \xleftarrow{R} \mathbb{Z}_2^\ell \quad , \quad b_q =< \boldsymbol{a_q} , \boldsymbol{s} > \oplus e_q$$

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 $(\boldsymbol{a_i}, b_i)$

Alekhnovich Public Key Encryption

Key Generation

The sender S chooses

- a secret key $\boldsymbol{s} \xleftarrow{R} \mathbb{Z}_2^{\ell}$
- $\boldsymbol{A} \xleftarrow{R} \mathbb{Z}_2^{q\times\ell}$ and the error $\boldsymbol{e} \leftarrow \mathrm{Ber}_{\tau}^{q}$
  to compute the public key $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} \oplus \boldsymbol{e})$

Encryption of a message bit $m \in \mathbb{Z}_2$

Sender S | Receiver R

choose a vector $\boldsymbol{f} \leftarrow \mathrm{Ber}_{\tau}^{q}$

compute $\boldsymbol{u} = \boldsymbol{f} \cdot \boldsymbol{A}$

$c = \langle \boldsymbol{f}, \boldsymbol{b} \rangle \oplus m \xrightarrow{(\boldsymbol{u}, c)}$

Decryption

The receiver R computes $d = c \oplus \langle \boldsymbol{s}, \boldsymbol{u} \rangle$

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

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

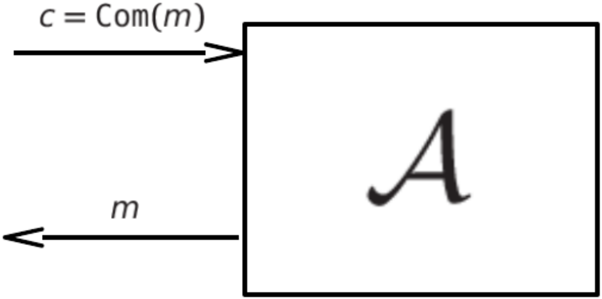
Receiver $\underline{\mathtt{R_i}}$		Receiver $\underline{\mathtt{R_j}}$
$\boldsymbol{d_i} \leftarrow \mathtt{ThLPN.Pdec}(\boldsymbol{C_1}, \boldsymbol{c_2}, \boldsymbol{s_i})$	$\xrightarrow{\boldsymbol{d_i}}$	

Partial Decryption

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

Partial Decryption

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


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