Public-Key Encryption Based on LPN

Li Chen

lichens.xd at gmail.com

Xidian University

November 3, 2013
Outline

1 Basic LPN cryptosystem
2 Multi-bit LPN cryptosystem
3 Ring-LPN cryptosystem
4 Discussion


References


**Claim:** Our slides are based on reference [1]
1 Basic LPN cryptosystem

Notations

- $\text{Ber}_\tau$ denotes the Bernoulli distribution with parameter $\tau$.

- $\text{Ber}_\tau^k$ denotes the distribution of vectors in $\mathbb{Z}_2^k$, where each entry is drawn independently from $\text{Ber}_\tau$.

- $\text{Bin}_{n,\tau}$ denotes the binomial distribution with $n$ trials, each with success probability $\tau$.

- We use a bold lower case character $\mathbf{z}$ to denote a column vector, use a bold upper case character $\mathbf{Z}$ to denote a matrix.
Definition 1.1  Decisional LPN Problem Take parameters $n \in \mathbb{N}$ and $\tau \in \mathbb{R}$ with $0 < \tau < 0.5$ (the noise rate). A distinguisher $D$ is said to $(q, t, \varepsilon)$-solve the decisional LPN$_{n, \tau}$ problem if

$$\left| \Pr_{A,mathbf{s},e}[D(A, As + e) = 1] - \Pr_{A,r}[D(A, r) = 1] \right| \geq \varepsilon$$

where $A \leftarrow \mathbb{Z}_{2}^{q \times n}$, $s \leftarrow \mathbb{Z}_{2}^{n}$, $e \leftarrow \text{Ber}_{\tau}$, $r \leftarrow \mathbb{Z}_{2}^{q}$, and the distinguisher runs in time at most $t$.

Lemma 1.2 (Lemma 1 from []) If there exists a distinguisher $D$ that $(q, t, \varepsilon)$-solve the decisional LPN$_{n, \tau}$ problem, then there exists a distinguisher $D'$ that $(q', t', \varepsilon')$-solve the search LPN$_{n, \tau}$ problem.

Definition 1.3 (Decisional LPN Assumption, DLPN) For any probabilistic algorithm $D$ that $(q, t, \varepsilon)$-solve the decisional LPN$_{n, \tau}$ problem for all large enough $n$, where $\tau$ is $\Theta(1/\sqrt{n})$, $t$ is polynomial in $n$, and $q$ is $O(n)$, it holds that $\varepsilon$ is negligible as a function of $n$. 
**Definition 1.4 (Basic LPN Cryptosystem)** The basic LPN cryptosystem is a 3-tuple \((\text{BasicLPNKenGen}, \text{BasicLPNEnc}, \text{BasicLPNDec})\), with the parameters \(n \in \mathbb{N}\), the length of the secret key, and \(\tau \in \mathbb{R}\), the noise rate. All operations are performed over \(\mathbb{Z}_2\).

- **BasicLPNKenGen()**: Choose a secret key \(sk = s \in \mathbb{Z}_2^n\). The public key is \(pk = (A, b)\), where \(A \leftarrow \mathbb{Z}_2^{2n \times n}\), \(b = As + e\), \(e \leftarrow \text{Ber}_{\tau^{2n}}\).

- **BasicLPNEnc(pk = (A, b), v)**: To encrypt a message bit \(v \in \mathbb{Z}_2\), choose \(f \leftarrow \text{Ber}_{\tau^{2n}}\) and output ciphertext \((u, c)\), where \(u = A^T f\) and \(c = < b, f > + v\).

- **BasicLPNDec(sk = s, (u, v))**: The decryption is \(d = c + < u, s >\).

**Note:**

\[
d = < b, f > + v + < u, s > = b^T f + s^T u = (s^T A^T + e^T)f + s^T A^T f + v = e^T f + v
\]
**Correctness:** Only need to show $e^T f = 0$. To show this, we need some lemmas as follows.

**Lemma 1.5** Let $X \sim \text{Bin}_{n,\tau}$, then the probability that $X$ is even is $\frac{1}{2} + \frac{(1-2\tau)^n}{2}$.

**Proof**

\[\Box\]

**Lemma 1.6** For any $k$ such that $\lim_{n \to \infty} \frac{n}{k} = \infty$, then it holds that $\lim_{n \to \infty} (1 + \frac{k}{n})^n = e^k$.

**Proof**

\[\Box\]
Theorem 1.7 (Correctness) For any constant \( \varepsilon > 0 \), it holds that \( \tau \) can be chosen with \( \tau = \Theta(\frac{1}{\sqrt{n}}) \) such that the probability of correct decryption by BasicLPNDec is at least \( 1 - \varepsilon \).

Proof
As we show above that \( d = e^T f + v \). Let \( e_i \) and \( f_i \) denote the entries of \( e \) and \( f \) respectively. Define \( C_i = e_i f_i \) and \( C = \sum_i C_i \), then \( e^T f = 0 \iff C \) is even. Since each \( C_i \sim \text{Ber}_{\tau^2} \), independently and identically, so \( C \sim \text{Bin}_{2n, \tau^2} \). By Lemma 1.5, then \( \Pr[e^T f = 0] = \frac{1}{2} + \frac{(1-2\tau)^{2n}}{2} \). Take \( 0 < \tau < O(\frac{1}{\sqrt{n}}) \), then \( \tau^2 n = O(1) \), so \( \lim_{n \to \infty} \frac{n}{\tau^2 n} = \infty \). Applying Lemma 1.6 yields \( \lim_{n \to \infty} (1 - 2\tau^2)^{2n} = e^{-2\tau^2(2n)} \). Hence, for large \( n \), \( \Pr[e^T f = 0] \approx \frac{1 + e^{-2\tau^2(2n)}}{2} \).

If \( \tau \leq \frac{c}{\sqrt{n}} \) for some constant \( c > 0 \), then \( \| -2\tau^2(2n) \| \leq 4c^2 \), \( \lim_{c \to 0} -2\tau^2(2n) = 0 \), so \( \lim_{c \to 0} 1 + e^{-2\tau^2(2n)} = 1 \). It follows that take \( \tau = \Theta(\frac{c}{\sqrt{n}}) \), for any \( \varepsilon > 0 \), the probability of correct decryption by BasicLPNDec is at least \( 1 - \varepsilon \) provided by choosing \( c \) sufficiently close to 0. \( \square \)
2 Multi-bit LPN cryptosystem

Definition 2.1 (Multi-bit LPN Cryptosystem) The multi-bit LPN cryptosystem is a 3-tuple (MultiLPNKenGen, MultiLPNEnc, MultiLPNDec), with the parameters $n$ and $\tau$ as in Definition 2.1, $l = O(n)$, the length of plaintext that can be encrypted in a single operation.

- MultiLPNKenGen(): Choose a secret key $sk = S \in \mathbb{Z}_{n}^{2n \times l}$. The public key is $pk = (A, B)$, where $A \leftarrow \mathbb{Z}_{2}^{2n \times n}, B = AS + E$, $E \leftarrow \text{Ber}_{\tau}^{2n \times l}$.

- MultiLPNEnc($pk = (A, B), v$): To encrypt a message $v \in \mathbb{Z}_{2}^{l}$, choose $f \leftarrow \text{Ber}_{\tau}^{2n}$ and output ciphertext $(u, c)$, where $u = A^T f$ and $c = B^T f + v$.

- MultiLPNDec($sk = s, (u, v)$): The decryption is $d = c + S^T u$.

Note:

$$d = B^T f + v + S^T u = S^T A^T f + E^T f + S^T A^T f + v = E^T f + v$$
3 Ring-LPN cryptosystem

Notations: For a polynomial ring $R = GF(2)[x]/(g(x))$, the distribution $Ber^R_\tau$ denotes the distribution over $R$, where each of the coefficients of the polynomial is drawn independently from $Ber_\tau$. For a polynomial $r \in R$, let $|r|$ denote the weight of $r$, i.e. the number of nonzero coefficients $r$ has. Let $r[i]$ denote the coefficient of $x_i$ in $r$.

For matrix $A \in \mathbb{Z}_2^{m \times n}$, $B \in \mathbb{Z}_2^{m' \times n}$, let $A//B \in \mathbb{Z}_2^{(m+m') \times n}$ denote the vertical concatenation of $A$ and $B$, i.e. $A//B$ is the matrix whose rows are those of $A$ followed by those of $B$.

For any polynomial $r \in R$ with degree $n - 1$, let $\text{vec}(r) \in \mathbb{Z}_2^n$ denote the column vector whose $i^{th}$ entry is $r[i]$, for all $0 \leq i \leq n$. And let $\text{mat}(r) \in \mathbb{Z}_2^{n \times n}$ be the matrix such that for all $r' \in R$, $\text{mat}(r) \text{vec}(r') = \text{vec}(r \cdot r')$. Note that the $i^{th}$ column vector of the matrix $\text{mat}(r)$ is exactly $\text{vec}(rx^{i-1})$. 
Definition 3.1 (Ring LPN Cryptosystem) The ring LPN cryptosystem is a 3-tuple (RingLPNKenGen, RingLPNEnc, RingLPNDec), with the parameters $n \in \mathbb{N}$, the length of the secret key, and $\tau \in \mathbb{R}$, the noise rate, and the ring $R = GF(2)[x]/ < g(x) >$, with $g(x)$ an irreducible polynomial of degree $n$.

- **RingLPNKenGen()**: Choose a secret key $sk = s \xleftarrow{} \mathbb{Z}_2^n$. The public key is $pk = (a_1, a_2, b)$, where $a_1, a_2 \xleftarrow{} R$, $b = As + e$, for $A = (\text{mat}(a_1))^T // (\text{mat}(a_2))^T$, $e \xleftarrow{} \text{Ber}_{\tau}^{2n}$.

- **RingLPNEnc(pk = (a_1, a_2, b), v)**: To encrypt a message bit $v \in \mathbb{Z}_2$, choose $f_1, f_2 \xleftarrow{} \text{Ber}_{R, n}$, define $f = \text{vec}(f_1) // \text{vec}(f_1)$, and output ciphertext $(u, c)$, where $u = A^T f$ and $c = < b, f > + v$.

- **RingLPNDec(sk = s, (u, v))**: The decryption is $d = c + < u, s >$.

**Note:**

1. $d = b^T f + v + s^T u = s^T A^T f + e^T f + s^T A^T f + v = e^T f + v$
2. $u = A^T f = \text{vec}(a_1 f_1 + a_2 f_2)$
4 Discussion

To be continued :)
Thanks! & Questions?