Public-Key Encryption Based on LPN

Li Chen

lichen.xd at gmail.com

Xidian University

November 3, 2013

Outline

References

[1] Ivan Damgård and Sunoo Park. Is public-key encryption based on lpn practical? In *IACR Cryptology ePrint Archive*, 2012.

Claim: Our slides are based on reference [1]

1 Basic LPN cryptosystem

Notations

- Ber_τ denotes the Bernoulli distribution with parameter τ.
- Ber_τ^k denotes the distribution of vectors in $\mathbb{Z}_2^k$, where each entry is drawn independently from Ber_τ.
- $\mathsf{Bin}_{n,\tau}$ denotes the binomial distribution with n trials, each with success probability τ.
- 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, ε)-solve the decisional $\mathsf{LPN}_{n,\tau}$ problem if

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

where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{q\times n}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^q$, $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^q$, and the distinguisher runs in time at most t.

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

Definition 1.3 (Decisional LPN Assumption, DLPN) For any probabilistic algorithm D that (q, t, ε)-solve the decisional $\mathsf{LPN}_{n,\tau}$ problem for all large enough n, where τ is $\Theta(1/\sqrt{n})$, t is polynomial in n, and q is $O(n)$, it holds that ε is negligible as a function of n.

Definition 1.4 (Basic LPN Cryptosystem) The basic LPN cryptosystem is a 3-tuple (BasicLPNKenGen, BasicLPNEnc, 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 $sk = \mathbf{s} \in \mathbb{Z}_2^n$. The public key is $pk = (\mathbf{A}, \mathbf{b})$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{2n \times n}$, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^{2n}$.
- BasicLPNEnc$(pk = (\mathbf{A}, \mathbf{b}), v)$: To encrypt a message bit $v \in \mathbb{Z}_2$, choose $\mathbf{f} \xleftarrow{\$} \mathsf{Ber}_\tau^{2n}$ and output cipertext $(\mathbf{u}, c)$, where $\mathbf{u} = \mathbf{A}^T\mathbf{f}$ and $c =< \mathbf{b}, \mathbf{f} > +v$.
- BasicLPNDec$(sk = \mathbf{s}, (\mathbf{u}, v))$: The decryption is $d = c+ < \mathbf{u}, \mathbf{s} >$.

Note:

$$d =< \mathbf{b}, \mathbf{f} > +v+ < \mathbf{u}, \mathbf{s} >= \mathbf{b}^T\mathbf{f} + \mathbf{s}^T\mathbf{u} = (\mathbf{s}^T\mathbf{A}^T + \mathbf{e}^T)\mathbf{f} + \mathbf{s}^T\mathbf{A}^T\mathbf{f} + v = \mathbf{e}^T\mathbf{f} + v$$

Correctness: Only need to show $\mathbf{e}^T\mathbf{f} = 0$. To show this, we need some lemmas as follows.

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

Proof

... $\square$

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

Proof

... $\square$

Theorem 1.7 (Correctness) For any constant $\varepsilon > 0$, it holds that τ 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 $d == \mathbf{e}^T\mathbf{f} + v$. Let e_i and f_i denote the entries of $\mathbf{e}$ and $\mathbf{f}$ respectively. Define $C_i = e_i f_i$ and $C = \sum_i C_i$, then $\mathbf{e}^T\mathbf{f} = 0 \iff C$ is even. Since each $C_i \sim \mathsf{Ber}_{\tau^2}$, independently and identically, so $C \sim \mathsf{Bin}_{2n,\tau^2}$. By Lemma 1.5, then $\Pr[\mathbf{e}^T\mathbf{f} = 0] = \frac{1}{2} + \frac{(1-2\tau)^{2n}}{2}$. Take $0 < \tau < O(\frac{1}{\sqrt{n}})$, then $\tau^2 n = O(1)$, so $\lim\limits_{n\to\infty} \frac{n}{\tau^2 n} = \infty$. Applying Lemma 1.6 yields $\lim\limits_{n\to\infty} (1 - 2\tau^2)^{2n} = e^{-2\tau^2(2n)}$. Hence, for large n, $\Pr[\mathbf{e}^T\mathbf{f} = 0] \approx \frac{1+e^{-2\tau^2(2n)}}{2}$. If $\tau \leq \frac{c}{\sqrt{n}}$ for some constant $c > 0$, then $\|-2\tau^2(2n)\| \leq 4c^2$, $\lim\limits_{c\to 0} -2\tau^2(2n) = 0$, so $\lim\limits_{c\to 0} 1 + e^{-2\tau^2(2n)} = 1$. 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 τ as in Definition 2.1, $l = O(n)$, the length of plaintxt that can be encrypted in a single operation.

- MultiLPNKenGen(): Choose a secret key $sk = \mathbf{S} \in \mathbb{Z}_2^{n\times l}$. The public key is $pk = (\mathbf{A}, \mathbf{B})$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{2n\times n}$, $\mathbf{B} = \mathbf{AS} + \mathbf{E}$, $\mathbf{E} \leftarrow \mathsf{Ber}_\tau^{2n\times l}$.
- MultiLPNEnc$(pk = (\mathbf{A}, \mathbf{B}), v)$: To encrypt a message $\mathbf{v} \in \mathbb{Z}_2^l$, choose $\mathbf{f} \xleftarrow{\$} \mathsf{Ber}_\tau^{2n}$ and output cipertext $(\mathbf{u}, \mathbf{c})$, where $\mathbf{u} = \mathbf{A}^T\mathbf{f}$ and $\mathbf{c} = \mathbf{B}^T\mathbf{f} + \mathbf{v}$.
- MultiLPNDec$(sk = \mathbf{s}, (\mathbf{u}, \mathbf{v}))$: The decryption is $\mathbf{d} = \mathbf{c} + \mathbf{S}^T\mathbf{u}$.

Note:

$$\mathbf{d} = \mathbf{B}^T\mathbf{f} + \mathbf{v} + \mathbf{S}^T\mathbf{u} = \mathbf{S}^T\mathbf{A}^T\mathbf{f} + \mathbf{E}^T\mathbf{f} + \mathbf{S}^T\mathbf{A}^T\mathbf{f} + \mathbf{v} = \mathbf{E}^T\mathbf{f} + \mathbf{v}$$

3 Ring-LPN cryptosystem

Notations: For a polynomial ring $R = GF(2)[x]/(g(x))$, the distribution Ber_τ^R denotes the distribution over R, where each of the coefficients of the polynomial is drawn independently from Ber_τ. 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 $\mathsf{vec}(r) \in \mathbb{Z}_2^n$ denote the column vector whose i^{th} entry is $r[i]$, for all $0 \le i \le n$. And let $\mathsf{mat}(r) \in \mathbb{Z}_2^{n\times n}$ be the matrix such that for all $r' \in R$, $\mathsf{mat}(r)\mathsf{vec}(r') = \mathsf{vec}(r \cdot r')$. Note that the i^{th} column vector of the matrix $\mathsf{mat}(r)$ is exactly $\mathsf{vec}(rx^{i-1})$.

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 $sk = \mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^n$. The public key is $pk = (a_1, a_2, \mathbf{b})$, where $a_1, a_2 \xleftarrow{\$} R$, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, for $\mathbf{A} = (\mathsf{mat}(a_1))^T // (\mathsf{mat}(a_2))^T$, $\mathbf{e} \leftarrow \mathsf{Ber}_\tau^{2n}$.
- RingLPNEnc($pk = (a_1, a_2, \mathbf{b}), v$): To encrypt a message bit $v \in \mathbb{Z}_2$, choose $f_1, f_2 \xleftarrow{\$} \mathsf{Ber}_\tau^{R,n}$, define $\mathbf{f} = \mathsf{vec}(f_1)//\mathsf{vec}(f_1)$, and output cipertext $(\mathbf{u}, c)$, where $\mathbf{u} = \mathbf{A}^T\mathbf{f}$ and $c =< \mathbf{b}, \mathbf{f} > +v$.
- RingLPNDec($sk = \mathbf{s}, (\mathbf{u}, v)$): The decryption is $d = c+ < \mathbf{u}, \mathbf{s} >$.

Note:

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

4 Discussion

To be continued :)

Thanks! & Questions?