Learning Parity with Noise

Botond Molnár

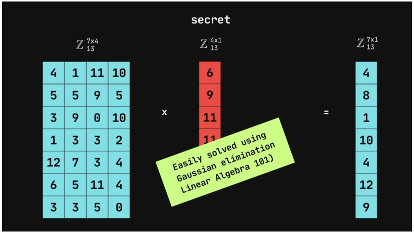
University of Debrecen

molnarbotond.eagle@gmail.com

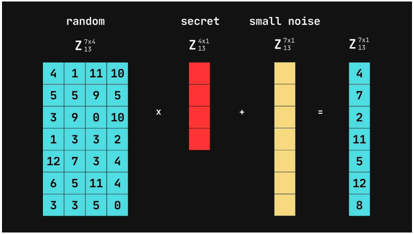
March 21, 2024

Overview

The LPN Problem



(a) Matrix Multiplication



(b) Add Noise

The LPN Problem

- The LPN problem is a computational problem in the field of cryptography [**LPNluke2022medium**].
- It is a generalization of the Learning with Errors (LWE) problem.
- The problem is to find the secret key s from the public key A and the noisy output b.
- The public key A is a matrix of size $m \times n$ and the secret key s is a vector of size n.
- The noisy output b is a vector of size m.
- The noise is added to the output by taking the dot product of the public key and the secret key and adding a vector of noise.
- It is a hard problem to find the secret key from the public key and the public parameters

The LPN Problem

Figure: LPN Formally

Definition 1 (search/decisional LPN Problem). *For* $\tau \in]0, 1/2[$, $\ell \in \mathbb{N}$, *the* decisional $\mathsf{LPN}_{\tau,\ell}$ *problem is* (q, t, ϵ)*-hard if for every distinguisher* D *running in time* t

$$\left| \Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathsf{D}(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e}) = 1] - \Pr_{\mathbf{r},\mathbf{A}}[\mathsf{D}(\mathbf{A}, \mathbf{r}) = 1] \right| \leq \epsilon \quad (1)$$

Where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^{\ell}$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_2^{q \times \ell}$, $\mathbf{e} \leftarrow \mathsf{Ber}_{\tau}^{q}$ *and* $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^{q}$. *The search* $\mathsf{LPN}_{\tau,\ell}$ *problem is* (q, t, ϵ)*-hard if for every* D *running in time* t

$$\Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathsf{D}(\mathbf{A}, \mathbf{A}.\mathbf{s} \oplus \mathbf{e}) = \mathbf{s}] \leq \epsilon \quad (2)$$

Public-key Encryption Scheme

[**base**]'s encryption scheme is a improved version of [**damgard**] scheme.

- Reducing the DLPN variety problem with $S \leftarrow Ber_r^{n\times n}$ to the normal DLPN problem.
- New single-bit public key encryption algorithm where the plaintext will be converted into a bit vector involved in cryptographic operations.
- The probability of the hamming weight exceeding expectations will exponentially decay rapidly to a value the is negligible.
- This ensures even if $r = 1/\sqrt{n}$ parameter is large the decription error can be ignored, therefore thr size of the public key is smaller than in [**damgard**]'s scheme.
- Decryption and encryption time of the algorythm is greatly reduced.
-

Enrcrption Scheme

The scheme consists of three PPT algorithms:

- Key generation $\rightarrow KeyGen(1^n, r)$
- Enccryption $\rightarrow Enc(pk, m)$
- Decryption $\rightarrow Dec(sk, c)$

Key Generation

The key generation of the cryptosytem $KeyGen(1^{\ltimes}, r)$

- n integer
- r noise rate
- Choose matrix $A \leftarrow \mathbb{Z}_2^{n \times n}$ randomly
- Choose $S \leftarrow Ber_r^{n \times n}$, $E \leftarrow Ber_r^{n \times n}$
- Compute $B = AS + E$
- Public key: $pk = (A, B)$
- Secret key: $sk = (S)$

Encryption

The encryption of the cryptosytem $Enc(pk, m)$

- Input is pk and $m \in Z_2$
- Compute $c_1 = r^T A + e_1^T, c_2 = r^T B + e_2^T$.
- Returns ciphertext $c = (c_1, c_2)$

Decryption

The decryption of the cryptosytem $Dec(sk, c)$

- Input secret key sk and ciphertext c
- Compute $d = c_1 \times S + c_2$
- If $h(d) << n/2$, It returns $m = 0$, else it return $m = 1$

Comparison with RSA and Damgard's Scheme

- It is faster than RSA
- It fells short of Damgard's scheme, while having the same public key size
- Decryption error is negligible.
- Not CCA secure, only IND-CPA secure.

Comparison

Figure: Comparison

	Time per encryption (ms)			Time per decryption		
Security level (bits)	80	112	128	80	112	128
RSA scheme(not padding)	0.010	0.030	0.060	0.140	0.940	2.890
Damgård's multi-bit	25.80	128.40	241.70	0.052	0.098	0.128
Our multi-bit scheme	15.60	45.30	102.10	0.11	0.221	0.258

Making the Scheme CCA Secure

[**CCA**] extended the public key scheme to be IND-CCA1 and IND-CCA2 secure. To achieve IND-CCA1 security the author extended the scheme with an instance-key derivation step that assigns a tag to each ciphertext and derives an instance public or secret key for each the tag. These instance keys are used as keys for public key scheme. To achieve IND-CCA2 security the author introduced one-time signatures

Base IND-CPA PKE Scheme Construction

- Key generation: $KeyGen(1^k)$
- Encryption: $Enc(pk, m)$
- Decryption: $Dec(sk, c)$
- k security parameter
- $n \in O(k^{2/(1-2\epsilon)})$
- $l_1, l_2, l_3 \in O(k^{2/(1-2\epsilon)})$
- $\rho = O(k^{-(1+2\epsilon)/(1-2\epsilon)})$
- $G \in \mathbb{F}_2^{l_2 \times n}$ is the generator matrix of a binary linear error-correcting code $\mathcal{C}$
- $Decode_{\mathcal{C}}$ an efficient decoding procodeure for $\mathcal{C}$ that corrects up to αl_2 errors (α is a constant)
- $\mathcal{D} \subseteq \mathbb{F}_2^{l_3}$ is a binary error correcting code with efficient encoding $Encode_{\mathcal{D}}$ and error-correction $Decode_{\mathcal{C}}$ which corrects up to λl_3 errors

Base IND-CPA PKE Scheme Key Generation

$KeyGen(1^k)$:

- Sample matrix $A \in \mathbb{F}_2^{l_1 \times n}$ uniformly at random.
- Sample matrix $C \in \mathbb{F}_2^{l_3 \times n}$ uniformly at random.
- Sample the matrix T from $\mathcal{X}_p^{l_2 \times l_1}$
- Sample matrix X from $\mathcal{X}_p^{l_2 \times n}$
- Set $B = G + T \cdot A + X$
- $pk = (A, B, C)$
- $sk = T$

Base IND-CPA PKE Scheme Encryption

$Enc_{pk}(m)$:

- Takes $pk = (A, B, C)$ and plaintext $m \in \mathbb{F}_2^n$
- Sample s from $\mathcal{X}_\rho^n$, e_1 from $\mathcal{X}_\rho^{l_1}$, e_2 from $\mathcal{X}_\rho^{l_2}$,e_3 from $\mathcal{X}_\rho^{l_3}$
- Set $c_1 = A \cdot s + e_1, c_2 = B \cdot s + e_2, c_3 = C \cdot s + e_3 + Encode_{\mathcal{D}}(m)$
- Output $c = (c_1, c_2, c_3)$

Base IND-CPA PKE Scheme Decryption

$Dec_{sk}(c)$:

- Takes $sk = T$ and ciphertext $c = (c_1, c_2, c_3)$
- Computes $y = c_2 - T \cdot c_1$
- Runs error correcting $s = Decode_{\mathcal{C}}(c_3 - C \cdot d)$, if succeeds run $m = Decode_{\mathcal{D}}(c_3 - C \cdot s)$
- Outputs m

Correctness of the scheme

Decryption only fails if one of the two error decoding operations fails. Probability of faliure:

- It is sufficient to bound the hamming-weight of the error-term $v = X \cdot s + e_2 - T \cdot e_1$
- Fix constants $\beta, \gamma > 0$ such that $2\beta + \gamma\rho < \alpha$ and $\gamma\rho < \lambda$
- By a Chernoff-bound it holds that $|s| < \gamma\rho n, e_1 < \gamma\rho l_1, e_2 < \gamma\rho l_2, e_3 < \gamma\rho l_3$ with owerwhelming probability k
- The decoding procedure can correct up to αl_2 errors
- Altogether it holds that

$$|v| \leq |Xs| + |e_2| + |Te_1| \leq 2\beta l_2 + \gamma\rho l_2 < \alpha l_2$$

- Therefore, the decoding algorithm $Decode_{\mathcal{C}}$ will successfully recover s and $Decode_{\mathcal{D}}$ will successfully recover m as

$$|e_3| < \gamma\rho \cdot l_3 < \lambda l_3$$

Expansion of the Base IND-CPA PKE Scheme

[**CCA**] expanded the previous scheme with an instance-key derivation step that assigns a tag to each ciphertext and derives a instance public or secret key for each the tag. These keys will be used as the keys for the PKE.

IND-CCA1 Secure PKE Scheme Construction

- k security parameter
- $n \in O(k^{2/(1-2\epsilon)})$
- $l_1, l_2, l_3 \in O(k^{2/(1-2\epsilon)})$
- $\rho = O(k^{-(1+2\epsilon)/(1-2\epsilon)})$
- $G \in \mathbb{F}_2^{l_2 \times n}$ is the generator matrix of a binary linear error-correcting code $\mathcal{C}$
- $Decode_{\mathcal{C}}$ an efficient decoding procodeure for $\mathcal{C}$ that corrects up to αl_2 errors (α is a constant)
- $\mathcal{D} \subseteq \mathbb{F}_2^{l_3}$ is a binary error correcting code with efficient encoding $Encode_{\mathcal{D}}$ and error-correction $Decode_{\mathcal{C}}$ which corrects up to λl_3 errors
- $\mathcal{E} \in \Sigma^{l_2}$ be a q-ary code over alphabet Σ ($q = |\Sigma|$) with relative minimum-distance δ and dimension n
- It is sufficient to choose $\delta < 1 - 1/q$ such that $2\beta + \gamma\rho + 1 - \delta < \alpha$, since α must be big enough to correct the decryption error which has a hamming weight $\leq (2\beta + \gamma\rho)l_2, \beta > 0$ and the additional error included by erasures will have a hamming weight $\leq (1-\delta)l_2$

IND-CCA1 Secure PKE Scheme Construction

- Since β and γ can be chosen arbitrarily small, we can always find q and δ such that the requirements are met.
- Therefore, fix β, γ, q, δ so that dor sufficiently large n it holds that

$$2\beta + \delta\rho + 1 - \delta < \alpha$$

IND-CCA1 Secure PKE Scheme Key Generation

$KeyGen(1^k)$:

- Sample matrix $A \in \mathbb{F}_2^{l_1 \times n}$ uniformly at random
- Sample matrix $C \in \mathbb{F}_2^{l_3 \times n}$ uniformly at random
- For every $j \in \Sigma$ sample a matrix T_j from $\mathcal{X}_\rho^{l_2 \times l_1}$ and matrix X_j from $\mathcal{X}_\rho^{l_2 \times n}$
- Set $B_j = G + T_j \cdot A + X_j$
- Set $pk = (A, (B_j)_{j \in \Sigma}, C)$
- Set $sk = (T_j)_{j \in \Sigma}$

IND-CCA1 Secure PKE Scheme Encryption

$Enc_{pk}(m)$:

- Takes $pk = (A, (B_j)_{j \in \Sigma}, C)$ and plaintext $m \in \mathbb{F}_2^n$
- Write each B_j as $B_j = (b_{j,1}, ..., b_{j,l_2})^T$
- Smaple a tag $\tau \in \Sigma^n$ uniformly at random and set $\hat{\tau} = Encode_{\mathcal{E}}(\tau)$
- Set $B_{\hat{\tau}} = (b_{\hat{\tau},1}, ..., b_{\hat{\tau}_{l_2},l_2})^T$
- Encryption samples s from $\mathcal{X}_\rho^{l_1}$, e_1 from $\mathcal{X}_\rho^{l_1}$, e_2 from $\mathcal{X}_\rho^{l_2}$, e_3 from $\mathcal{X}_\rho^{l_3}$
- Set $c_1 = A \cdot s + e_1, c_2 = B_{\hat{\tau}} \cdot s + e_2, c_3 = C \cdot s + e_3 + Encode_{\mathcal{D}}(m)$
- Output $c = (c_1, c_2, c_3, \tau)$

IND-CCA1 Secure PKE Scheme Decryption

$Dec_{sk}(c)$:

- Takes $sk = (T_j)_{j \in \Sigma}$ and ciphertext $c = (c_1, c_2, c_3, \tau)$
- Write each T_j as $T_j = (t_{j,1}, ..., t_{j,l_2})^T$
- Compute $\hat{\tau} = Encode_{\mathcal{E}}$ and $T_{\hat{\tau}} = (t_{\hat{\tau},1}, ...,)$
- Compute $y = c_2 - T_{\hat{\tau}} \cdot c_1$ and $s = Decode_{\mathcal{C}}(y)$
- Outputs nil if the decoding fails, else computes $m = Decode_{\mathcal{D}}(c_3 - C \cdot s)$
- Computes $e_1 = c_1 - A \cdot s, e_2 = c_2 - B_{\hat{\tau}} \cdot s, e_3 = c_3 - C \cdot s - Encode_{\mathcal{D}}(m)$
- Checks if $|s| < \gamma\rho n$, $|e_1| < \gamma\rho l_1$, $|e_2| < \gamma\rho l_2$, $|e_3| < \gamma\rho l_3$
- If all coditions met outputs m, else outputs nil.

Correctness of the IND-CCA1 Secure PKE Scheme

Correcness can be derived from the previous scheme with the only additional step of checking the hamming weights $|s|, |e_1|, |e_2|, |e_3|$

Room for Improvement

- The scheme is IND-CCA1 secure, but not IND-CCA2 secure
- [**CCA**] extended the scheme to be IND-CCA2 secure by introducing one-time signatures

Observations

- The previous CCA1 secure scheme can be improved to be CCA2 secure.
- The scheme can be extended with one-time signatures to achieve CCA2 security.
- [**CCA**] mentions that it is not necessary to choose tag $\tau \in \Sigma^n$ uniformly at random in the ecryption procedure of the prevous PKE scheme.
- The scheme must only guarantee that a PPT adversary $\mathcal{A}$ will have a negligible probablity of guessing the secret tag τ^* correctly if it is granted a polynomial number of trials.
- Therefore it is sufficient to sample the tags τ from a distribution with high min-entorpy.

IND-CCA2 Secure PKE Scheme Construction

- $SIG = (Gen, Sign, Verify)$ be an EUF-CMA secure one time signature scheme.
- Key generation is identical to the previous CCA1 secure scheme
- Enc first computes a pair of verification keys a pair of verification and signture-keys $(vk, sk) = SIG.Gen(1^k)$
- Then it runs the encryption procedure of the previous scheme $PKE.Enc$.
- The only difference that it sets $\tau = vk$ instead of choosing τ uniformly at random.

IND-CCA2 Secure PKE keygen

$KeyGen(1^k)$: The same as the previous CCA1 secure scheme

IND-CCA2 Secure PKE Encryption

$Enc_{pk}(m)$:

- Generate $(vk, sgk) = SIG.Gen(1^k)$
- Encrypt $c' = Enc'_{pk}(m, vk)$
- Sign $\sigma = SIG.Sign_{sgk}(c')$, output $c = (c'.\sigma)$

IND-CCA2 Secure PKE Decryption

$Dec_{sk}(c)$:

- $c = (c', \sigma), c' = (\tau, c_1, c_2, c_3)$
- Set $vk = \tau$
- Check if $SIG.Verify_{vk}(c', \sigma) = 1$, if not abort
- Compute $m = PKE.Dec_{sk}(c')$

Proving IND-CCA2 Security

If *SIG* is an EUF-CMA secure one-time signature scheme and the security level of the previous PKE scheme stands the scheme is IND-CCA2 secure.

Aim of Our Research

Our research goal is to make [**base**]'s public key encryption scheme IND-CCA2 secure, and to integrate it into a lightwegiht adhoc mixnet which can be used in a drone network prividing anonymity and confidentiality. We turned for inspiration to [**CCA**] to achieve IND-CCA2 in [**base**]'s cryptosystem.

Thanks for your attention!

References