How to Encrypt with the LPN Problem

Henri Gilbert, Matt Robshaw, and Yannick Seurin

Orange Labs

ICALP 2008 – July 9, 2008

the context

- the authentication protocol HB^+ by Juels and Weis [JW05] recently renewed interest in cryptographic protocols based on the LPN (*Learning Parity with Noise*) problem, the problem of learning an unknown vector x given noisy versions of its scalar product $a \cdot x$ with random vectors a
- this problem seems promising to obtain efficient protocols since it implies only basic operations on GF(2)
- in this work, we present a probabilistic symmetric encryption scheme, named LPN-C, whose security against chosen-plaintext attacks can be proved assuming the hardness of the LPN problem

outline

- the LPN problem: a brief survey
- description and analysis of the encryption scheme LPN-C
- concrete parameters, practical optimizations
- conclusion & open problems

the LPN problem

> Given q noisy samples $(a_i, a_i \cdot x \oplus \nu_i)$, where x is a secret k-bit vector, the a_i's are random, and $\Pr[\nu_i = 1] = \eta$, find x.

- similar to the problem of decoding a random linear code (NP-complete)
- best solving algorithms require $T, q = 2^{\Theta(\frac{k}{\log k})}$: Blum, Kalai, Wasserman [BKW03], Levieil, Fouque [LF06]
- a variant by Lyubashevsky [L05] requires $q = \mathcal{O}(k^{1+\epsilon})$ but $T = 2^{\mathcal{O}(\frac{k}{\log \log k})}$
- numerical examples:
  - for $k = 512$ and $\eta = 0.25$, LF requires $T, q \simeq 2^{89}$
  - for $k = 768$ and $\eta = 0.01$, LF requires $T, q \simeq 2^{74}$

previous schemes based on LPN

- PRNG by Blum et al. [BFKL93]
- public-key encryption scheme by Regev [R05] based on the LWE problem, the generalization of LPN to GF(p), $p > 2$
- the HB family of authentication protocols:
  - HB [HB01]
  - HB^{+} [JW05]
  - HB^{++} [BCD06]
  - HB* [DK07]
  - HB$^{\#}$ [GRS08]
  - Trusted-HB [BC07]
  - PUF-HB [HS08]

description of LPN-C

- **public components:** a (linear) error-correcting code $C : \{0,1\}^r \rightarrow \{0,1\}^m$ of parameters $[m, r, d]$ and the corresponding decoding algorithm C^{-1}
- **secret key:** a $k \times m$ binary matrix M
- **encryption:**
  - r-bit plaintext $\mathbf{x}$, encode it to $C(\mathbf{x})$
  - draw a random k-bit vector $\mathbf{a}$ and a noise vector $\mathbf{v}$ where $\Pr[\mathbf{v}[i] = 1] = \eta$
  - ciphertext $(\mathbf{a}, \mathbf{y})$, where $\mathbf{y} = C(\mathbf{x}) \oplus \mathbf{a} \cdot M \oplus \mathbf{v}$
- **decryption:** on input $(\mathbf{a}, \mathbf{y})$, compute $\mathbf{y} \oplus \mathbf{a} \cdot M$ and decode the resulting value, or output $\perp$ if unable to decode

security intuition

- $\mathbf{y} = C(\mathbf{x}) \oplus \mathbf{a} \cdot M \oplus \boldsymbol{\nu}$
- in a chosen-plaintext attack, the adversary only learns $\mathbf{a}_i \cdot M \oplus \boldsymbol{\nu}_i$ for random vectors $\mathbf{a}_i$
- hardness of the LPN problem implies that the adversary cannot guess $\mathbf{a} \cdot M$ for a new random $\mathbf{a}$ better than with *a priori* probability ("MHB puzzle" [GRS08]), hence will have no information on a challenge ciphertext $(\mathbf{a}, C(\mathbf{x}) \oplus \mathbf{a} \cdot M \oplus \boldsymbol{\nu})$

decryption failures

- decryption failures happen when $\mathrm{Hwt}(\boldsymbol{\nu}) > t$, where $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ is the correction capacity of the code
- when the noise is randomly drawn,

$$P_{\mathrm{DF}} = \sum_{i=t+1}^{m} \binom{m}{i} \eta^i (1-\eta)^{m-i}$$

  is negligible for $\eta m < t$

- for eliminating decryption failures, the Hamming weight of the noise vector can be tested before being used and regenerated when $\mathrm{Hwt}(\boldsymbol{\nu}) > t$, but this may impact the security proof

quasi-homomorphic encryption

- the scheme enjoys some kind of “homomorphism” property
- given two plaintexts

$$\begin{aligned}(\mathbf{a}, \mathbf{y}) &= (\mathbf{a}, C(\mathbf{x}) \oplus \mathbf{a} \cdot M \oplus \boldsymbol{\nu}) \\ (\mathbf{a}', \mathbf{y}') &= (\mathbf{a}', C(\mathbf{x}') \oplus \mathbf{a}' \cdot M \oplus \boldsymbol{\nu}'),\end{aligned}$$

  one has:

$$\mathbf{y} \oplus \mathbf{y}' = C(\mathbf{x} \oplus \mathbf{x}') \oplus (\mathbf{a} \oplus \mathbf{a}') \cdot M \oplus (\boldsymbol{\nu} \oplus \boldsymbol{\nu}')$$

  so that $(\mathbf{a} \oplus \mathbf{a}', \mathbf{y} \oplus \mathbf{y}')$ is a valid ciphertext for $\mathbf{x} \oplus \mathbf{x}'$ if $\mathrm{Hwt}(\boldsymbol{\nu} \oplus \boldsymbol{\nu}') \leqslant t$
- $\boldsymbol{\nu} \oplus \boldsymbol{\nu}'$ is a noise vector with noise parameter $\eta' = 2\eta(1-\eta)$; if $\eta' m < t$, the homomorphism property holds with overwhelming probability

security notions

- security goals: indistinguishability (IND) and non-malleability (NM)
- adversaries run in two phases; at the end of the first phase they output a distribution on the plaintexts and receive a ciphertext challenge
- they are denoted PX-CY according to the oracles (P for encryption, C for decryption) they can access
  - $X, Y = 0$: the adversary can never access the oracle
  - $X, Y = 1$: the adversary can only access the oracle during phase 1 (non-adaptive)
  - $X, Y = 2$: the adversary can access the oracle during phases 1 and 2, *i.e.* after having seen the challenge ciphertext (adaptive)

security notions

- relations between different types of attacks have been studied by Katz and Yung [KY06]:
- IND-P1-C Y $\Leftrightarrow$ IND-P2-C Y and NM-P1-C Y $\Leftrightarrow$ NM-P2-C Y
- IND-P2-C2 $\Leftrightarrow$ NM-P2-C2

security proof: a useful lemma

- notations:
  - U_{k+1} will be the oracle returning uniformly random $(k+1)$-bit strings
  - $\Pi_{s,\eta}$ will be the oracle returning the $(k+1)$-bit string $(\mathbf{a}, \mathbf{a} \cdot \mathbf{s} \oplus \nu)$, where $\mathbf{a}$ is uniformly random and $\Pr[\nu = 1] = \eta$
- we have the following decision-to-search lemma (Regev [R05], Katz and Shin [KS06]):

  lemma: if there is an efficient oracle adversary distinguishing between the two oracles U_{k+1} and $\Pi_{s,\eta}$, then there is an efficient adversary solving the LPN problem

IND-P2-C0 security proof

- P2-C0 adversary $\mathcal{A}$ breaking the indistinguishability of the scheme
- we use it to distinguish between U_{k+1} and $\Pi_{s,\eta}$ as follows:
  - draw a random $j \in [1..m]$ and a random $k \times (m-j)$ binary matrix M'
  - use the following method to encrypt:
    - get a sample $(\mathbf{a}, z)$ from the oracle $\mathcal{O}$
    - form the m-bit masking vector $\mathbf{b} = \mathbf{r} \| z \| (\mathbf{a} \cdot M' \oplus \boldsymbol{\nu})$ where $\mathbf{r}$ is a random $(j-1)$-bit string and $\boldsymbol{\nu}$ an $(m-j)$-bit noise vector
    - return the ciphertext $(\mathbf{a}, C(\mathbf{x}) \oplus \mathbf{b})$
  - play the indistinguishability game with $\mathcal{A}$; if $\mathcal{A}$ distinguishes, return 1, otherwise return 0

IND-P2-C0 security proof

- masking vector $\mathbf{b} = \mathbf{r} \| z \| (\mathbf{a} \cdot M' \oplus \mathbf{v})$
- when $\mathcal{O} = U_{k+1}$, the j first bits of $\mathbf{b}$ are random and the $m - j$ last ones are distributed according to an LPN distribution; for $j = m$ the ciphertexts are completely random
- when $\mathcal{O} = \Pi_{s,\eta}$, the $j - 1$ first bits of $\mathbf{b}$ are random and the $m - j + 1$ last ones are distributed according to an LPN distribution; for $j = 1$ the encryption is perfectly simulated
- when expressing the advantage of this distinguisher, the terms for $j = 2$ to $(m - 1)$ cancel and we obtain advantage δ/m if the advantage of the original distinguisher $\mathcal{A}$ was δ

malleability

- as is, the scheme is clearly malleable (P0-C0 attack):
- given a ciphertext (a, y) corresponding to some plaintext x, the adversary can simply modify it to $(a, y \oplus C(x'))$, which will correspond to the plaintext $x \oplus x'$
- since IND-P2-C2 $\Leftrightarrow$ NM-P2-C2, the scheme cannot be IND-P2-C2 or even IND-P0-C2 either
- what about non-adaptive ciphertext attacks?

an IND-P0-C1 attack

- idea: query the decryption oracle on (a, y_i) many times with the same a and random y_i's to get approximate equations on $a \cdot M$
- when $y_i \oplus a \cdot M$ is at Hamming distance less than t from a codeword, the decryption oracle will return x_i such that $\mathrm{Hwt}(C(x_i) \oplus y_i \oplus a \cdot M) \leqslant t$
- this will give an approximation of each bit of $a \cdot M$ with noise parameter less than t/m; repeating the experiment sufficiently many times with the same a enables to retrieve $a \cdot M$ with high probability, hence to retrieve the secret key M
- this attack works only if the probability that a random m-bit string is decodable is sufficiently high, *i.e.* if the code is good enough

P2-C2 security

- one can obtain an IND/NM-P2-C2 scheme by appending a MAC to the ciphertext (*Encrypt-then-MAC* paradigm studied by Bellare et al. [BN00])
- we propose the following MAC based on the LPN problem:
  - let M be a $l \times l'$ secret binary matrix and H be a one-way function
  - for $\mathbf{X} \in \{0,1\}^*$ define $\mathrm{MAC}_M(\mathbf{X}) = H(\mathbf{X}) \cdot M \oplus \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a noise vector of parameter η
- one can prove the security of this MAC in the random oracle model for H, using the hardness of the "MHB puzzle" [GRS08]

Given q noisy samples $(\mathbf{a}_i, \mathbf{a}_i \cdot M \oplus \boldsymbol{\nu}_i)$, where M is a secret $k \times m$ matrix and $\Pr[\boldsymbol{\nu}_i[j] = 1] = \eta$, and a random challenge $\mathbf{a}$, find $\mathbf{a} \cdot M$.

example parameters

- expansion factor $\sigma = \frac{|\text{ciphertext}|}{|\text{plaintext}|} = \frac{m+k}{r}$

k	η	m	r	d	expansion factor	key size	key size (Toeplitz)	P_{DF}
512	0.125	80	27	21	21.9	40,960	591	0.42
512	0.125	160	42	42	16	81,920	671	0.44
768	0.05	80	53	9	16	61,440	847	0.37
768	0.05	160	99	17	9.4	122,880	927	0.41
768	0.05	160	75	25	12.4	122,880	927	0.06

possible variants and optimizations

- use of Toeplitz matrices to reduce the key size
- Toeplitz matrices have good randomization properties: $(x \to x \cdot T)_T$ is a $1/2^m$-balanced function family (for any non-zero vector a, $a \cdot T$ is uniformly distributed)

$$\begin{pmatrix} & t_3 & t_2 & t_1 \\ & & t_3 & t_2 \\ & \iddots & & t_3 \\ t_{k+m-1} & & & \end{pmatrix}$$

- possibility to pre-share the random vectors a used to encrypt, or to re-generate them from a PRNG and a small seed; then $\sigma = \frac{m}{r}$, the expansion factor of the error-correcting code

conclusion & open problems

- we presented LPN-C, a probabilistic symmetric encryption scheme whose security relies on the LPN problem
- it extends the range of cryptographic protocols based on the LPN problem
- implementation would be quite efficient but practical problems remain: expansion of the ciphertext, high key size
- open problems include:
  - understand the impact of the use of Toeplitz matrices on the security of the scheme
  - devise an efficient MAC whose security relies only on the LPN problem to obtain an IND/NM-P2-C2 secure encryption scheme

thanks for your attention!

comments $\vee$ questions?