Notes on Alekhnovich's cryptosystems

Gilles Zémor

November 2016

Decisional Decoding Hypothesis with parameter t. Let $0 < R_1 < R_2 < 1$. There is no polynomial-time decoding algorithm $\mathcal{A}$ such that: Given k, n such that $R_1 \leqslant k/n \leqslant R_2$, given a random code C defined by a uniform random generating $k \times n$ matrix (or a uniform random parity-check $(n-k) \times n$ matrix), and a vector which is either

(i) a uniformly random vector $\mathbf{u}$

(ii) $\mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in C$ is a uniformly random codeword and $\mathbf{e}$ a uniformly random vector of weight t, independent of $\mathbf{c}$.

$\mathcal{A}$ decides between (i) and (ii) with a non-negligible advantage over random choice.

We remark that we may replace the probability distribution for choosing C by the uniform distribution over all codes of dimension k. To obtain it choose a random generating matrix, and if it is not full-rank discard it and choose again. Since the probability of a $k \times n$ random matrix being singular is at most $1/2^{n-k}$, the two probability distributions (uniform $k \times n$ random generating matrix and uniform random code of dimension k) are computationally indistinguishable. Note also that the uniform distribution over all codes of dimension k may be obtained by choosing an $(n-k) \times n$ parity-check matrix with uniform distribution, discarding it and sampling it again if ever it is not full-rank.

Alekhnovich's first cryptosystem

We set the parameter t to be $o(n^{1/2})$. Let $\mathbf{A}$ be a random $k \times n$ matrix and let $\boldsymbol{\varepsilon}$ be a random vector in $\mathbb{F}_2^n$ of weight t. Let $\mathbf{H}$ be the $(k+1) \times n$ matrix obtained by appending to $\mathbf{A}$ an additional row consisting of the vector

$$\mathbf{y} = \mathbf{x}\mathbf{A} + \boldsymbol{\varepsilon}$$

where $\mathbf{x}$ is a uniform vector in $\mathbb{F}_2^k$ independent of $\boldsymbol{\varepsilon}$. In other words, $\mathbf{y}$ is the sum of the error vector $\boldsymbol{\varepsilon}$ and a random codeword of the code generated by the matrix $\mathbf{A}$. Let C be the code with parity-check matrix $\mathbf{H}$. The public key is a generating matrix for the code C. In this cryptosystem the message space consists of a single bit $\mathcal{M} = \{0, 1\}$.

Encryption. To encrypt 0, output

$$\mathcal{C}(0) = \mathbf{c} + \mathbf{e}$$

where $\mathbf{c}$ is a random codeword of C and $\mathbf{e}$ is a random vector of weight t. To encrypt 1, output

$$\mathcal{C}(1) = \mathbf{u}$$

where $\mathbf{u}$ is a uniform random vector of $\mathbb{F}_2^n$.

Decryption. The secret key is the vector $\boldsymbol{\varepsilon}$. To decrypt the encrypted bit $\mathcal{C}(m)$, $m \in \{0, 1\}$, compute

$$b = \langle \boldsymbol{\varepsilon}, \mathcal{C}(m) \rangle$$

and declare b to be the decrypted plaintext. We have that

$$\begin{aligned} \langle \boldsymbol{\varepsilon}, \mathcal{C}(0) \rangle &= \langle \boldsymbol{\varepsilon}, \mathbf{c} + \mathbf{e} \rangle = \langle \boldsymbol{\varepsilon}, \mathbf{c} \rangle + \langle \boldsymbol{\varepsilon}, \mathbf{e} \rangle \\ &= \langle \boldsymbol{\varepsilon}, \mathbf{e} \rangle \end{aligned}$$

because $\boldsymbol{\varepsilon}$, being a row of the parity-check matrix $\mathbf{H}$, is orthogonal to all codewords of C. Since we have imposed on both $\boldsymbol{\varepsilon}$ and $\mathbf{e}$ to be of weight $t = o(\sqrt{n})$, the probability that $\langle \boldsymbol{\varepsilon}, \mathbf{e} \rangle = 0$ is close to 1. Besides, since $\mathcal{C}(1)$ is a random vector, we have that when $m = 1$, decryption succeeds with probability exactly $1/2$. To obtain a reliable cryptosystem, use an error-correcting code, e.g. encrypt the secret bit m several times.

Security reduction. Suppose there exists a decryption algorithm $\mathcal{D}$ that extracts m from $\mathcal{C}(m)$ given only knowledge of the code C (and not $\boldsymbol{\varepsilon}$). Then this algorithm should work without any noticeable difference if the code C is replaced by a random code C' defined by a uniform random parity-check matrix $\mathbf{H}'$, i.e. if the vector $\mathbf{y} = \mathbf{x} + \boldsymbol{\varepsilon}$ used in defining the last row of $\mathbf{H}$ is uniformly random, equivalently if $\boldsymbol{\varepsilon}$ is taken to be uniformly random rather than random of weight t. If there were a noticeable difference, this would yield a way of distinguishing whether $\mathbf{y}$ is uniformly random or at distance t from the random code generated by the matrix $\mathbf{A}$, contradicting the decisional decoding hypothesis of parameter t.

Suppose therefore that we are now using the encryption scheme with the random code C' rather than the original code C. Again, the decryption algorithm $\mathcal{D}$ should work just as well when the error vector $\mathbf{e}$ used to encrypt the message $m = 0$ is replaced by

a uniform random vector. Otherwise we again have a distinguisher between a uniform random vector and a vector of the form $\mathbf{c}+\mathbf{e}$ where $\mathbf{e}$ is of weight t and $\mathbf{c}$ is a random codeword of the *random* code C'. Again this would contradict the decisional decoding hypothesis. But we now have an absurd result, which is that the decryption algorithm $\mathcal{D}$ should somehow be able to decrypt in the situation when the encryption of both 0 and 1 are uniform random vectors of $\mathbb{F}_2^n$, which clearly cannot be achieved with a success probability different from $1/2$.

Alekhnovich's second cryptosystem

Let $\mathbf{A}$ be a uniform random $n/2 \times n$ matrix, let $\mathbf{X}$ be a uniform random $n \times n/2$ matrix, and let $\mathbf{E}$ be a random $n \times n$ matrix, chosen uniformly among matrices such that every row of $\mathbf{E}$ is of weight t. Set $\mathbf{M} = \mathbf{XA} + \mathbf{E}$. Every row of $\mathbf{M}$ is therefore obtained by adding a random vector of weight t to a random codeword of the code generated by the rows of the matrix $\mathbf{A}$. We add the requirement that $\mathbf{M}$ is invertible: if this is not the case we throw away the matrix $\mathbf{M}$ and choose another one in the same way until we obtain an invertible matrix $\mathbf{M}$.

Let C_0 be an error-correcting code of length n that comes with a polynomial-time decoding algorithm that almost always decodes correctly codewords that have been submitted to a binary symmetric channel of transition probability $p = t^2/n$. We should have $\dim C_0 > n/2$, for example suppose $\dim C_0 = 9n/10$.

Let $\phi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be the linear map defined by the matrix $\mathbf{M}$, i.e. for a column vector $\mathbf{x}$, $\phi(\mathbf{x}) = \mathbf{Mx}$. Let $C_1 = \phi^{-1}(C_0)$, i.e. $C_1 = \{\mathbf{x} \in \mathbb{F}_2^n, \phi(\mathbf{x}) \in C_0\}$. Denote by C_2 the code for which $\mathbf{A}$ is a parity-check matrix. Finally, define the code $C = C_1 \cap C_2$. Let $\mathbf{G}$ be a generating matrix for this code. The matrix $\mathbf{G}$ is the public key of the cryptosystem. Let $k = \dim C$. Without loss of generality we can suppose k is even and set $k = 2m$. The message (cleartext) space is $\mathcal{M} = \mathbb{F}_2^m$.

Encryption. To encrypt a message $\mathbf{m} \in \mathbb{F}_2^m$, append to it a random m-bit vector $\mathbf{r}$ to create $\mathbf{x} \in \mathbb{F}_2^k$. The ciphertext is:

$$\mathcal{C}(\mathbf{m}) = \mathbf{xG} + \mathbf{e}$$

where $\mathbf{e}$ is a random vector of weight t in $\mathbb{F}_2^n$.

Decryption. The secret key is the matrix $\mathbf{E}$. To decrypt, start by computing the vector $\mathbf{y}$, such that $\mathbf{y^T} = \mathbf{E}\mathcal{C}(\mathbf{m})^\mathbf{T}$. We note that $\mathcal{C}(\mathbf{m}) = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} = \mathbf{xG}$ is

a codeword of C. Since C is a subcode of C_2, all rows of $\mathbf{A}$ are orthogonal to all codewords of C, so that $\mathbf{A}\mathbf{c}^\mathbf{T} = 0$. Therefore

$$\begin{aligned}\mathbf{y}^\mathbf{T} &= \mathbf{E}(\mathbf{c}+\mathbf{e})^\mathbf{T} = \mathbf{E}\mathbf{c}^\mathbf{T} + \mathbf{E}\mathbf{e}^\mathbf{T} \\ &= \mathbf{X}\mathbf{A}\mathbf{c}^\mathbf{T} + \mathbf{E}\mathbf{c}^\mathbf{T} + \mathbf{E}\mathbf{e}^\mathbf{T} \\ &= \mathbf{M}\mathbf{c}^\mathbf{T} + \mathbf{E}\mathbf{e}^\mathbf{T} \\ &= \mathbf{c}_0^\mathbf{T} + \mathbf{E}\mathbf{e}^\mathbf{T}\end{aligned}$$

where $\mathbf{c}_0^\mathbf{T} = \mathbf{M}\mathbf{c}^\mathbf{T}$. But since $\mathbf{c} \in C \subset C_1$ and $\phi(C_1) = C_0$, we have $\mathbf{c}_0 \in C_0$. Also, since every row $\mathbf{E}_i$, $1 \leqslant i \leqslant n$, of $\mathbf{E}$ has weight t, we have that $\langle \mathbf{E}_i, \mathbf{e} \rangle = 1$ with probability at most $p = t^2/n$. Therefore, applying the decoding algorithm for C_0 to $\mathbf{y}$ will yield $\mathbf{c}_0$ with a vanishing probability of error. To finish decryption,

solve the linear system $\mathbf{x}\mathbf{G} = \phi(\mathbf{c}_0)$, (i.e. $\mathbf{x}\mathbf{G}(\mathbf{M}^\mathbf{T})^{-1} = \mathbf{c}_0$) and throw away the last m bits of $\mathbf{x}$ to recover $\mathbf{m}$. The matrix $\mathbf{G}(\mathbf{M}^\mathbf{T})^{-1}$ can be chosen in systematic form to avoid solving the linear system. This implies that $\mathbf{M}^\mathbf{T}$ or simply $\mathbf{G}(\mathbf{M}^\mathbf{T})^{-1}$ is known to the decryption algorithm, but these quantities could just as well be public, we will see that they do not help cryptanalysis.

Security reduction. Suppose there is a decryption algorithm $\mathcal{D}$ that decrypts without the secret key $\mathbf{E}$. Then we first argue, as for the first cryptosystem, that the decryption algorithm must behave in the same way when the matrix $\mathbf{M}$ is replaced by a random uniform full-rank matrix $\mathbf{M}'$. Otherwise, by feeding the decryption algorithm $\mathcal{D}$ with messages encrypted either with the genuine cryptosystem, or by the modified cryptosystem where $\mathbf{M}$ is replaced by $\mathbf{M}'$ and everything else constructed in the same way, we would have a way of differentiating between pairs of vectors $\mathbf{u}, \mathbf{v}$, where $\mathbf{u}$ is uniformly random and $\mathbf{v} = \mathbf{a} + \boldsymbol{\varepsilon}$, where $\mathbf{a}$ is a random sum of rows of $\mathbf{A}$ and $\boldsymbol{\varepsilon}$ is a random vector of weight t. This would contradict the decisional decoding hypothesis.

We suppose the decryption algorithm works in the IND-CPA model, this means that $\mathcal{D}$ first chooses two plaintexts m_0 and m_1, and asks for an encryption of $\mathbf{m}_i$: it then returns i with a probability π such that $\pi - 1/2$ is non-negligible. We now work towards a contradiction by showing how to use $\mathcal{D}$ to break the decisional decoding hypothesis.

Let V be a uniform random code of length n and dimension m. Suppose $\mathbf{z}$ is either a uniform random vector of length n, or equal to $\mathbf{r} + \mathbf{e}$ with $\mathbf{r}$ a random vector of V and $\mathbf{e}$ a random vector of weight t. We will call upon the deciphering algorithm $\mathcal{D}$ to decide what category $\mathbf{z}$ belongs to.

Let U be another uniform random code of length n and dimension m. With overwhelming probability, the codes U and V have trivial intersection and the code $C = U \oplus V$ has dimension $k = 2m$. Randomly extend the code C by adding random vectors so as to obtain a code C_1 of the required dimension (say $9n/10$) and similarly extend the code C by adding random vectors so as to obtain a code C_2 of the required dimension $(n/2)$. By construction C_1 and C_2 are uniform random codes of the required dimensions and $C = C_1 \cap C_2$. Let $\mathbf{G}$ be a generating matrix of C of the form $\mathbf{G} = \left[\begin{smallmatrix} \mathbf{G}_U \\ \mathbf{G}_V \end{smallmatrix}\right]$ Let ϕ be a random one-to-one linear mapping that maps C_0 to C_1. Since C_1 is a random linear code of the same dimension as C_0, ϕ is simply a random one-to-one linear map that we associate to the random matrix $\mathbf{M}'$. The matrix $\mathbf{M}'$ and the codes C, C_1, C_2 have the same distribution as in the cryptosystem defined by the matrix $\mathbf{M}'$ and the condition that none of the codes C, C_1, C_2 are degenerate, which happens with overwhelming probability when defining the original cryptosystem.

Now we call upon the algorithm $\mathcal{D}$. Algorithm $\mathcal{D}$ gives us the plaintexts $\mathbf{m}_0$ and $\mathbf{m}_1$, and we choose $i \in \{0, 1\}$ randomly. We then give algorithm $\mathcal{D}$ the ciphertext

$$\mathbf{m}_i \mathbf{G}_U + z.$$

Algorithm $\mathcal{D}$ returns its guess ι of the bit i. If $\iota = i$ we declare $\mathbf{z}$ to be of the form $\mathbf{z} = \mathbf{r} + \mathbf{e}$. Otherwise we declare $\mathbf{z}$ to be equal to the uniform random vector $\mathbf{u}$. We see that whenever $\mathbf{z}$ is of the form $\mathbf{z} = \mathbf{r} + \mathbf{e}$, then the ciphertext given to algorithm $\mathcal{D}$ has exactly the form of a valid encryption of $\mathbf{m}_i$, hence the non-negligible advantage in telling apart whether $\mathbf{z}$ is uniform or at distance t from a random codeword of V.

Reduction to the search decoding problem

Difficulty of Decoding Hypothesis with parameter t. Let $0 < R_1 < R_2 < 1$. There is no polynomial-time decoding algorithm $\mathcal{A}$ such that:

- Given k, n such that $R_1 \leqslant k/n \leqslant R_2$,
- given a random code C defined by a uniform random generating $k \times n$ matrix (or a uniform random parity-check $(n-k) \times n$ matrix),
- and a vector

  $$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

  where $\mathbf{c}$ is a random codeword of C and $\mathbf{e}$ is a random vector of weight t, chosen uniformly and independently of $\mathbf{c}$,

algorithm $\mathcal{A}$ returns $\mathbf{c}$ with non-negligible probability.

Theorem 1. *If the difficulty of decoding hypothesis with parameter t is satisfied, then there is no polynomial-time algorithm $\mathcal{A}$ that computes $\langle \mathbf{r}, \mathbf{x} \rangle$ with a non-negligible advantage over random choice (outputting 0 or 1 with probability $1/2$) given*

- *two real numbers R_1, R_2, with $0 < R_1 < R_2 < 1$,*
- *a $k \times n$ uniform random matrix $\mathbf{G}$ with $R_1 < k/n < R_2$,*
- *an instance $\mathbf{xG} + \mathbf{e}$ of the decoding problem for the code C generated by $\mathbf{G}$, where $\mathbf{x} \in \mathbb{F}_2^k$ is randomly chosen, and a random vector $\mathbf{e}$ of weight t.*
- *a uniform random vector $\mathbf{r}$ of length n,*

Before proving Theorem 1 we show how to use it to deduce the full difficulty of decoding hypothesis from the purely decisional version.

Proposition 2. *The difficulty of decoding hypothesis with parameter t implies the decisional decoding hypothesis with the same parameter t. In other words, if there exists an algorithm that efficiently tells the difference between a vector at distance t from the code and a uniform random vector, then there exists an algorithm that efficiently decodes vectors at distance t from the code.*

Proof. Suppose $\mathcal{D}$ is an algorithm that breaks the decisional decoding hypothesis and distinguishes between a vector of the form $\mathbf{xG} + \mathbf{e}$, for a random codeword $\mathbf{xG}$ of a the random code generated by the random matrix $\mathbf{G}$. We will construct an algorithm $\mathcal{A}$ that contradicts Theorem 1. Algorithm $\mathcal{A}$ will be given as input

- a random matrix $\mathbf{G}$,
- a vector $\mathbf{y} = \mathbf{xG} + \mathbf{e}$, $\mathbf{e}$ of weight t,
- a random vector $\mathbf{r}$.

Algorithm $\mathcal{A}$ will then proceed to evaluate $\langle \mathbf{x}, \mathbf{r} \rangle$. Now let $\mathbf{r}$ and $\mathbf{s}$ be two uniform random vectors, in $\mathbb{F}_2^k$ and $\mathbb{F}_2^n$ respectively. Form the matrix $\mathbf{G}' = \mathbf{G} + \mathbf{r}^T\mathbf{s}$ and remark that $\mathbf{G}'$ is uniform random like $\mathbf{G}$. Remark also that

$$\mathbf{x}(\mathbf{r}^T\mathbf{s}) = \langle \mathbf{x}, \mathbf{r} \rangle \mathbf{s}$$

therefore

- either $\langle \mathbf{x}, \mathbf{r} \rangle = 0$ and $\mathbf{xG}' = \mathbf{xG}$,

- or $\langle \mathbf{x}, \mathbf{r} \rangle = 1$ and $\mathbf{x}\mathbf{G}' = \mathbf{x}\mathbf{G} + \mathbf{s}$.

We now define the code C to be the code generated *by the matrix* G'. Algorithm $\mathcal{A}$ simply gives the code C (defined by $\mathbf{G}'$) and the vector $\mathbf{y}$ to the distinguishing algorithm $\mathcal{D}$. If algorithm $\mathcal{D}$ says 'uniform', algorithm $\mathcal{A}$ declares $\langle \mathbf{x}, \mathbf{r} \rangle = 1$. If algorithm $\mathcal{D}$ says '$\mathbf{y} = \mathbf{c} + \mathbf{e}$', then algorithm $\mathcal{A}$ declares $\langle \mathbf{x}, \mathbf{r} \rangle = 0$. ∎

Proof of the Goldreich-Levin Theorem

Theorem 1 is a special case of the more general statement:

Theorem 3. *Let f be any one-way function from $\{0,1\}^n$ to $\{0,1\}^m$. There is no polynomial-time algorithm $\mathcal{A}$ that given $\mathbf{y} = f(\mathbf{x})$ for a uniform random input $\mathbf{x}$, and an independent uniformly random $\mathbf{r} \in \{0,1\}^n$, computes $\langle \mathbf{x}, \mathbf{r} \rangle$ with a non-negligible advantage, i.e. outputs $b \in \{0,1\}$ with $P(b = \langle \mathbf{x}, \mathbf{r} \rangle) \geqslant 1/2 + \varepsilon$, where ε is a polynomial function of $1/n$.*

Proof. We suppose $\mathcal{A}$ exists and use $\mathcal{A}$ to construct an algorithm that computes $\mathbf{x}$ from $f(\mathbf{x})$. First notice that for a fraction at least $\varepsilon/2$ of entries $\mathbf{x}$, algorithm $\mathcal{A}$ must predict $\langle \mathbf{x}, \mathbf{r} \rangle$ correctly from $f(\mathbf{x})$ for a proportion at least $1/2 + \varepsilon/2$ of choices of $\mathbf{r}$. We may therefore suppose that $\mathbf{x}$ is a fixed (but unknown) entry, for which algorithm $\mathcal{A}$ predicts $\langle \mathbf{x}, \mathbf{r} \rangle$ with an ϵ positive bias. From now on ε denotes this particular bias, rather than the average bias of Theorem 3. Since $\mathbf{x}$ is now fixed, we also denote by $\mathcal{A}(\mathbf{r})$ algorithm $\mathcal{A}$'s evaluation of $\langle \mathbf{x}, \mathbf{r} \rangle$.

Our goal is to compute the exact values of $\langle \mathbf{x}, e_i \rangle$, $i = 1, 2, \ldots, n$, for e_i the canonical basis of $\mathbb{F}_2^n$. This will give the individual coordinates of $\mathbf{x}$ and we will be done. If we were guaranteed that $\mathcal{A}$ always gave the right value $\langle \mathbf{x}, \mathbf{r} \rangle$ for every $\mathbf{r}$ (i.e. $\varepsilon = 1/2$), there would be nothing to prove. We have to deal however with an algorithm that is often wrong (though less often than it is right).

A key remark is that if $\mathcal{A}(\mathbf{r})$ and $\mathcal{A}(\mathbf{r} + e_i)$ are both correct (or both incorrect) we have $\langle \mathbf{x}, e_i \rangle = \mathcal{A}(\mathbf{r}) + \mathcal{A}(\mathbf{r} + e_i)$. To evaluate $\langle \mathbf{x}, e_i \rangle$, we are therefore tempted to take random values $\mathbf{r}$, compute $\mathcal{A}(\mathbf{r}) + \mathcal{A}(\mathbf{r} + e_i)$, and take a majority vote. Unfortunately, $\mathcal{A}(\mathbf{r})$ and $\mathcal{A}(\mathbf{r} + e_i)$, viewed as random variables (over the random choice of $\mathbf{r}$), need not be independent, and we can only guarantee that $\mathcal{A}(\mathbf{r}) + \mathcal{A}(\mathbf{r} + e_i)$ coincides with $\langle \mathbf{x}, e_i \rangle$ with probability $> 1/2$ for a bias $\varepsilon > 1/4$. We need to improve upon this strategy.

We shall use the following technical lemma:

Lemma 4. *Let V be a random subvector space V of $\mathbb{F}_2^n$ of dimension k. Let $S_i = e_i + V$, $i = 1, 2, \ldots, n$. Let $\eta > 0$ be a constant. Then if $k \geqslant \log_2(n^{1+\eta}/\varepsilon^2)$, we have that, for every i,*

$$\#\{\mathbf{r} \in S_i \,|\, \mathcal{A}(\mathbf{r}) = \langle \mathbf{x}, \mathbf{r} \rangle\} > |S_i|/2 \tag{1}$$

with probability (over the choice of V) at least $1 - 1/n^{1+\eta}$.

Lemma 4 implies that, when choosing a random subspace V, we have, with probability at least $1 - 1/n^{\eta}$, that (1) holds *for all* $S_i = e_i + V$, $i = 1, \ldots, n$ simultaneously.

Let $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$ be an arbitrary basis of the vector space V and let

$$\gamma = (\gamma_1, \ldots, \gamma_k) \in \mathbb{F}_2^k.$$

Consider the following function g_γ, defined on V:

$$\begin{array}{rcl} g_\gamma : V & \to & \mathbb{F}_2 \\ \mathbf{r} = \displaystyle\sum_{i=1}^{k} \lambda_i \mathbf{b}_i & \mapsto & \displaystyle\sum_{i=1}^{k} \lambda_i \gamma_i \end{array}$$

in words, g_γ guesses the values of $\langle \mathbf{x}, \mathbf{r} \rangle$ on basis elements $\mathbf{r} = \mathbf{b}_1 \ldots \mathbf{b}_k$, and uses linearity to extend this guess to the rest of the space V. The result is that, whenever $\mathbf{g}_\gamma$ is right on the whole of the basis $\mathbf{b}$, it is also always right on the whole space V, by linearity of the function $\mathbf{r} \mapsto \langle \mathbf{x}, \mathbf{r} \rangle$. Now we proceed as follows: for all 2^k possible values of γ we evaluate $\langle \mathbf{x}, e_i \rangle$ by computing $\mathcal{A}(\mathbf{r} + e_i) + g_\gamma(\mathbf{r})$ for all $\mathbf{r} \in V$ and by setting

$$\langle \mathbf{x}, e_i \rangle = \begin{cases} 0 & \text{if } \#\{\mathbf{r} \in V \mid \mathcal{A}(r + e_i) + g_\gamma(\mathbf{r}) = 0\} > |V|/2 \\ 1 & \text{if } \#\{\mathbf{r} \in V \mid \mathcal{A}(r + e_i) + g_\gamma(\mathbf{r}) = 1\} \geqslant |V|/2 \end{cases}$$

When $\gamma = (\gamma_1, \ldots, \gamma_k)$ coincides with $(\langle \mathbf{x}, \mathbf{b}_i \rangle)_{i=1..k}$ then $g_\gamma(\mathbf{r}) = \langle \mathbf{x}, \mathbf{r} \rangle$ for every $\mathbf{r} \in V$, and the value

$$\mathcal{A}(\mathbf{r} + e_i) + g_\gamma(\mathbf{r})$$

is a correct guess of $\langle \mathbf{x}, e_i \rangle$ if and only if $\mathcal{A}$ is correct on $\mathbf{r} + e_i$, which happens for a majority of $\mathbf{r}$ in V by (1). So for this particular choice of γ, we have all the coordinates $\langle \mathbf{x}, e_i \rangle$ with a probability $1 - 1/n^{\eta}$. Since we can check whether we have the right value of $\mathbf{x}$ by computing $f(\mathbf{x})$, we can stop when we have a satisfying answer for $\mathbf{x}$. ∎

Proof of Lemma 4. To randomly generate the subvector space V, choose k uniform random independent variables $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ in $\mathbb{F}_2^n$ and declare V to be generated by

$(\mathbf{v}_i)$. Note that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are linearly dependent with probability at most $1/2^{n-k}$. For every $\lambda = (\lambda_1, \ldots, \lambda_k)$, define $\mathbf{v}_\lambda = \sum_{i=1}^k \lambda_i \mathbf{v}_i$ and define the Bernoulli variable X_λ by

$$X_\lambda = \begin{cases} 1 & \text{if } \mathcal{A}(\mathbf{v}_\lambda) = \langle \mathbf{x}, \mathbf{v}_\lambda \rangle \\ 0 & \text{otherwise.} \end{cases}$$

The variables $\mathbf{v}_\lambda$ are pairwise independent, therefore so are the X_λ, and Chebychov's inequality implies therefore that

$$P\left(\frac{1}{2^k} \sum_{\lambda \in \mathbb{F}_2^k} X_\lambda \leqslant 1/2 \right) \leqslant \frac{1}{\varepsilon^2 2^k} = \frac{1}{n^{1+\eta}}.$$

∎

Modern variations

Variations on Alekhnovich's first cryptosystem

Variation 1. Let $\mathbf{A}$ be a random $k \times n$ matrix and let $\boldsymbol{\varepsilon}$ be a random vector of $\mathbb{F}_2^n$ of weight $t = o(n^{1/2})$. Let $\mathbf{s}$ be a uniform random vector in $\mathbb{F}_2^k$ and let

$$\mathbf{y} = \mathbf{s}\mathbf{A} + \boldsymbol{\varepsilon}.$$

The message (plaintext) space is $\mathcal{M} = \{0, 1\}$ and the matrix $\mathbf{A}$ and the vector $\mathbf{y}$ make up the public key.

Encryption. To encrypt $m \in \mathbb{F}_2$, output

$$\mathcal{C}(m) = (\mathbf{A}\mathbf{e}^T, m + \langle \mathbf{e}, \mathbf{y} \rangle)$$

where $\mathbf{e}$ is a random t-weight vector of $\mathbb{F}_2^n$.

– Exercice 1.

a) The secret key is $\mathbf{s}$*. Figure out how to decrypt by computing* $\langle \mathbf{e}\mathbf{A}^T, \mathbf{s} \rangle$*.*

b) Prove security.

Variation 2. Let $\mathbf{A}$ be a random $k \times n$ matrix and let $\mathbf{e}$ be a random t-weight vector of $\mathbb{F}_2^n$. The message (plaintext) space is $\mathcal{M} = \{0,1\}$ and the matrix $\mathbf{A}$ and the vector $\sigma = \mathbf{A}\mathbf{e}^T \in \mathbb{F}_2^k$ make up the public key. To encrypt $m \in \mathbb{F}_2$, output

$$\mathcal{C}(m) = (\mathbf{s}\mathbf{A} + \boldsymbol{\varepsilon}, m + \langle \mathbf{s}, \sigma \rangle)$$

where $\mathbf{s}$ is a uniform random vector of $\mathbb{F}_2^k$ and $\boldsymbol{\varepsilon}$ is a random t-weight vector of $\mathbb{F}_2^n$.

– Exercice 2.

a) Figure out how to decrypt with the secret key $\mathbf{e}$*.*

b) Prove security.

Variation on Alekhnovich's second cryptosystem

Let $\mathbf{A}$ be a random $k \times n$ matrix and let $\mathbf{S}$ be a uniform random $\ell \times k$ matrix. Let $\mathbf{E}$ be an $\ell \times n$ matrix such that all its rows are randomly and independently chosen among row vectors of weight t. Define the $\ell \times n$ matrix

$$\mathbf{Y} = \mathbf{S}\mathbf{A} + \mathbf{E}.$$

The message (plaintext) space is $\mathcal{M} = C \subset \{0,1\}^\ell$ where C is an error-correcting code that comes with a polynomial-time algorithm that almost always decodes correctly codewords that have been submitted to a binary symmetric channel of transition probability $p = t^2/n$. The matrices $\mathbf{A}$ and $\mathbf{Y}$ make up the public key. The parameter ℓ should be chosen so that $k + \ell < Rn$ for some constant $R < 1$.

Encryption. To encrypt $\mathbf{m} \in C$, output

$$\mathcal{C}(\mathbf{m}) = (\mathbf{A}\mathbf{e}^T, \mathbf{m} + \mathbf{Y}\mathbf{e}^T)$$

where $\mathbf{e}$ is a random t-weight vector of $\mathbb{F}_2^n$.

– Exercice 3.

a) Figure out how to decrypt with the secret key $\mathbf{S}$*.*

b) if we had $k + \ell \geqslant n$*, how could one decrypt without any secret key ?*

c) Prove security.

Argue first that a decryption algorithm that is only given $\mathbf{A}$ *and* $\mathbf{Y}$ *should also work with a random* $\mathbf{Y}$*. Then suppose that one is given a random* $u \times n$ *matrix*

$\mathbf{R}$ *and a vector* $\mathbf{z} \in \mathbb{F}_2^u$ *that is promised to be either uniform random or of the form* $\mathbf{Re}^T$ *for a random weight* t *vector* $\mathbf{e} \in \mathbb{F}_2^n$*. Set* $u = k + \ell$ *and let the first* k *rows of* $\mathbf{R}$ *make up a matrix* $\mathbf{A}$ *and the remaining* ℓ *rows of* $\mathbf{R}$ *make up a matrix* $\mathbf{Y}$*. Create a cryptosystem from* $\mathbf{A}$ *and* $\mathbf{Y}$*, split the vector* $\mathbf{z}$ *into two parts and feed the decryption algorithm the relevant cryptogram.*

References

[1] M. Alekhnovich, More on average case vs approximation complexity, *Comput. Complex.* 20 (2011), 755 – 786.

[2] B. Applebaum, Y. Ishai and E. Kushilevitz, Cryptography with constant input locality, *J. Cryptology* 22 (2009), 429–469.

[3] I Damgard and S. Park, Is Public-Key Encryption Based on LPN Practical ? `https://eprint.iacr.org/2012/699.pdf`

[4] L. Trevisan, Some Applications of Coding Theory in Computational Complexity, Electronic Colloquium on Computational Complexity, Report No. 43 (2004).