Modifications of the Rao-Nam Cryptosystem

Ángela I. Barbero[1] and Øyvind Ytrehus[2]

[1] University of Valladolid, Dept. of Mathematics Applied to Engineering, 47011 Valladolid, Spain
angbar@wmatem.eis.uva.es,
WWW home page: http://www.wmatem.eis.uva.es/~angbar
[2] University of Bergen, Dept. of Informatics, N-5020 Bergen, Norway
oyvind@ii.uib.no,
WWW home page: http://www.ii.uib.no/~oyvind

Abstract. Rao and Nam [7] proposed a secret-key cryptosystem based on error correcting codes. After breaking the original system by a chosen-plaintext attack, Struik and van Tilburg [8] improved the Rao-Nam cryptosystem. However, the size of the key remains a practical problem also for their improved scheme. We discuss several modifications of the improved Rao-Nam system. The goal of these modifications is to reduce the amount of secret key that needs to be exchanged, while maintaining the security of the system.[1]

1 Introduction

In 1978, McEliece [6] presented a public-key cryptosystem that was based on error-correcting codes. The private keys of the McEliece system are

- a generator matrix $\mathbf{G}$ of an $[n, k]$ t-error correcting binary Goppa code,
- a binary $k \times k$ invertible scrambler matrix $\mathbf{S}$, and
- a binary $n \times n$ permutation matrix $\mathbf{P}$.

The public key is the matrix product $\mathbf{G}' = \mathbf{SGP}$. A sender who wants to send a k-bit message $\mathbf{m}$ will transmit the cryptogram $\mathbf{c} = \mathbf{mG}' + \mathbf{e}$, where $\mathbf{e}$ is a random n-dimensional vector of Hamming weight at most t. Due to the error correcting capability of the Goppa code and to the existence of efficient decoding algorithms for the Goppa code, the legitimate receiver can successfully remove the random vector $\mathbf{e}$. On the other hand, an intruder without knowledge of the secret key faces the problem of decoding a general linear error-correcting code; a problem which is known to be NP-hard [1]. The best known attack on the McEliece system consists basically of guessing a subset of k error-free bits [5]. Lee and Brickell [5] devised a way to check whether such a subset indeed consists of error-free bits. The work factor (which is a rough estimate of the number of guesses, on average, before a message can be found in this way, times the number

[1] This work was supported by NFR, Grants 107542/410 and 107623/420, by Junta de Castilla y León under project VA 22/96, and by DGICYT, PB95-063-0002-02

J. Buchmann et al. (eds.), *Coding Theory, Cryptography and Related Areas*

of basic operations needed for each try) is approximately $2^{69.6} \approx 10^{21}$ for the case of $n = 1024$, k chosen ($= 644$) to maximize the work factor. (Recently, Berson [2] gave a much more efficient attack based on a weakness in the protocol in which the McEliece system is applied: In essence, if the intruder can get access to multiple cryptograms of the same message $\mathbf{m}$ and different error vectors $\mathbf{e}$, then for the code parameters given above the number of guesses are typically on the order of ten or less.)

Rao and Nam [7] proposed a secret-key cryptosystem which resembles the McEliece system. We will refer to this original scheme as the *RN scheme*, to distinguish it from the modified schemes described later. The McEliece system is public-key and the RN scheme is secret-key. In compensation for this difference, the RN scheme should be expected to offer better security with smaller keys and/or higher code rates. However, with smaller parameters, if the sender selects an error vector which is correctable (in the sense of an ordinary error-correcting code), then the scheme is vulnerable to an attack based on majority voting on each coordinate of several cryptograms corresponding to the same message. If the error vectors are random but with average weight different from $n(q-1)/q$, where q is the size of the field used, then the non-correct values in each coordinate will be outvoted by the correct ones. Thus for the RN scheme one should employ error vectors of weight approximately $n/2$, or $n(q-1)/q$ if a q-ary code is used. Such error vectors are not decodable in the ordinary sense, thus we need to represent the set of error vectors explicitly in the system.

A description of the RN scheme follows. Two parties, Alice and Bob, share a secret key consisting of

- a secret parity check matrix $\mathbf{H}$ of a (binary) $[n, k]$ error correcting code $\mathcal{C}$ (and, implicitly, a corresponding generator matrix $\mathbf{G}$ (which can be derived from $\mathbf{H}$ by some deterministic algorithm) of the code), and
- a predetermined set $\mathcal{E}$ of error vectors of length n, each lying in a unique coset of $\mathcal{C}$.

Alice will map a k-bit message $\mathbf{m}$ into a cryptogram $\mathbf{c}$ by calculating

$$\mathbf{c} = \mathbf{mG} + \mathbf{e}, \tag{1}$$

where $\mathbf{e}$ is a random vector from $\mathcal{E}$. In order to retrieve the message $\mathbf{m}$, Bob will

- calculate the syndrome $\mathbf{s} = \mathbf{cH}^T$,
- obtain $\mathbf{mG}$ by subtracting from $\mathbf{c}$ the error vector $\mathbf{e}$ which is identified by $\mathbf{s}$, and
- invert the encoding process.

This paper is organized as follows. In the next section, we explain why the original Rao-Nam scheme does not work in practice. Section 3 introduces this paper's modifications to the cryptosystem. In Section 4 we discuss possible attacks on the system. Section 5 contains a small toy example.