Can one design a signature scheme based on error-correcting codes?

Jacques Stern

Laboratoire d'Informatique, École Normale Supérieure

Abstract. In this note, we show that the signature scheme based on error-correcting codes which has been proposed during the ASIACRYPT'94 conference and appears in this volume (see [1]) is not secure. The attack involves gathering a few hundred signatures. From then on, only elementary linear algebra is used.

1 The Use of Error-Correcting Codes in Cryptography.

Since the appearance of public key cryptography, there has been a continuous line of research aiming at the discovery of alternatives to the standard techniques based on arithmetical properties of large numbers. In order to achieve such a program, error-correcting codes seemed quite attractive. As soon as 1978, a public key cryptosystem based on the use of Goppa codes was proposed by Mc Eliece in [3]. This cryptosystem is still standing. Concerning the problem of identification, a scheme was proposed by the author in [4]. This time, random codes with no specific structure were used. It is of course possible to turn our authentication scheme into a signature scheme by the standard technique that replaces queries from the verifier by values suitably obtained from the message to sign together with the initial commitments of the prover (see [2]). Unfortunately, this yields rather long and therefore a bit unpractical signatures. All attempts to design a signature scheme based on error-correcting codes and having reasonable signature length have failed (see the references of [1]). We will show that the same is true with the scheme proposed in the present volume ([1]).

2 The Attack.

The scheme appearing in [1] is a bit intricate. Actually, it is not really necessary to understand the technical details of key generation, signature generation and signature check in order to follow our attack. We will focus on the weak part of the scheme which lies in the following equation, litterally taken from [1].

$$\underline{s}_j = \{\underline{e}_j \oplus f(\underline{m}_j, \underline{e}_j)W_A \oplus \underline{x}_j W_A^{-R} G_A\} P_A$$

In the above, W_A and G_A are secret $(k \times n)$ matrices, and P_A is a secret $(n \times n)$ matrix. W_A^{-R} is a known $(n \times k)$ matrix. Also, $\underline{s}_j$, $\underline{e}_j$, $\underline{x}_j$ and $f(\underline{m}_j, \underline{e}_j)$ are known row vectors, the first three of dimension n and the last one of dimension k. $\underline{s}_j$,

$\underline{x}_j$ are part of the signature and $\underline{e}_j$ is computed during signature check, as well as $f(\underline{m}_j, \underline{e}_j)$.

We write $\underline{y}_j$ in place of $\underline{x}_j W_A^{-R}$. Of course all matrices and vectors are over the two element field.

Now, the above equation provides a known linear equation between the coefficients of the unknown matrices P_A, $W_A P_A$ and $G_A P_A$. More precisely, each signature gives n scalar equations since both sides of the equation are n-bit vectors. Since there are $n(2n+k)$ unknowns, it should take something like $2k+n$ signatures to solve the system. Once this is done, P_A, $W_A P_A$ and $G_A P_A$ are known and new signatures can be easily manufactured.

It turns out that the above argument is not completely correct. In order to get the requested conclusion, we need to have a sample of vectors $z_j = \{\underline{e}_j, f(\underline{m}_j, \underline{e}_j), \underline{y}_j\}$ that generate the whole $2k+n$-dimensional space. There is a subtle point here: although the row vectors $\underline{y}_j$ and $f(\underline{m}_j, \underline{e}_j)$ are basically random, the vectors $\underline{e}_j$, are randomly chosen among n-bit vectors with a prescribed number t of ones (error patterns). Thus, if t is even, there is no way the z_j can generate the whole space since they are all included in the hyperplane consisting of vectors with an even number of ones among the first n bits.

Let F_N be the space generated by the z_j after N signatures have been collected. If F_N is not the entire space, let H_N be a hyperplane containing F_N and let $h=0$ be a cartesian equation of this hyperplane. Now if the linear functional h has some non zero coefficient among the last $2k$ ones, then, it is easily seen that with probability $1/2$, the next signatures takes you outside H_N. This shows that we will very quickly come up with a situation where h has its $2k$ trailing coefficients equal to zero. We call such a situation *favourable*. At this point, the z_j generate a subspace of the form $E \times F_2^{2k}$ and the linear application corresponding to the matrix obtained by appending the blocks P_A, $W_A P_A$ and $G_A P_A$ is known on this subspace. Especially P_A is known on E and $W_A P_A$ and $G_A P_A$ are completely determined.

Note that, in the above situation, it is already possible to sign whenever a vector of weight t can be found in E: this can be easily checked by going into the construction of [1]. Also note that, if no such vector can be found in E the next signature will provide one. This enables us to estimate an upper bound on the numbers of signatures required in order to issue a new unauthorized signature. After N signatures, the probability that a favourable situation has not been reached is bounded above by the probability of getting less than $2k+n$ heads after tossing a fair coin N times. Using classical estimates on binomial coefficients, this is equivalent to $2^{N(H_2(\lambda)-1)}$, where $\lambda = \frac{2k+n}{N}$ and H_2 is the entropy function defined by $H_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$. This upper bound quickly decreases when N increases. In [1] the following dimensions are suggested

- $n = 128$ and $k = 65$ in which case$2k + n = 258$. The above probability is bounded by 0.0027 for $N = 600$.
- $n = 256$ and $k = 152$ in which case$2k + n = 560$. The above probability is

bounded by 0.069 for $N = 1200$.

Thus, as announced, a few hundred signatures are enough to break the scheme. Furthermore, our analysis is overly pessimistic and a favourable situation is presumably reached much earlier than what is suggested by the bounds.

3 Can the Scheme be Rescued?

We do not think the scheme can be rescued. Of course, suggestions can be made in order to make the attack slightly more difficult: for example the error vector $\underline{e}_j$ could be computed from say $\underline{m}_j$ by deterministic hashing. Note that this suggestion does not appear in [1] (the same way the discussion about odd and even weights was not present in the paper included in the preproceedings). For this reason, we will only offer a qualitative analysis.

As the numbers of known signatures increases, the vector space denoted by E in our above analysis grows. We distinguish two cases

1. if t (the weight of error vectors) is odd, then E ultimately covers the whole space
2. if t is even, then E ultimately covers the whole hyperplane consisting of vectors of even weight

In both cases, any message can ultimately be signed, even for the tentatively "repaired" version of the scheme. Note that, in some unlucky cases, the convergence can be a bit slow, for example if we get "stuck" in some hyperplane whose equation is a linear functional with a single variable. Then, it will take an average of n/t signatures to get out. Again, a quantitative worst-case analysis based on these unlucky cases would be overly pessimistic and the generic case should converge fast enough.

References

1. M. Alabbadi and S. B. Wicker, A digital signature scheme based on linear error-correcting block codes, this volume.
2. A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Proceedings of Crypto 86*, Lecture Notes in Computer Science 263, 181-187.
3. R. J. McEliece, Public key cryptosystem based on algebraic coding theory, JPLDSN Progress Report 42-44, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, January and February 1978, 114-116.
4. J. Stern, A new identification scheme based on syndrome decoding. *Proceedings of Crypto 93*, Lecture Notes in Computer Science 773, 13-21.

This article was processed using the LaTeX macro package with LLNCS style