A COMMENT ON NIEDERREITER'S PUBLIC KEY CRYPTOSYSTEM

Bernard Smeets
Department of Computer Engineering
University of Lund
P.O. Box 118, S-221 00 Lund, SWEDEN

Summary - In this comment we show that a recently proposed public key cryptosystem is not safe for most of the practical cases. Furthermore, it is shown that the security of this system is closely connected with the problem of computing logarithms over a finite field.

1 Introduction

At the EUROCRYPT 85 workshop H. Niederreiter proposed a public key distribution system based on shift register sequences. In this comment we show that the proposed system that works with sequences over a finite field of characteristic 2 will be unsecure for most of the practical cases.

Let us briefly summarize what is known publicly in one of the proposed systems. For a full description we refer to [1]. First a polynomial g(x) of degree n over some finite prime field K=GF(p), p prime, is chosen, i.e.

$$g(x)=x^n-b_{n-1}x^{n-1}-\ldots-b_1x^1-b_0 \in K[x].$$

We demand that

$$n\geq 2 \quad \text{and} \quad (b_0,b_1)\neq(0,0).$$

For the moment let p=2 and g(x) a prime polynomial over K(=GF(2)). Furthermore, let s be the sequence obtained from a linear feedback shift register with g(x) as its feedback polynomial and initial conditions $s_0=\ldots=s_{n-2}=0, s_{n-1}=1$. Besides g(x) two sequences are public; $s^{(h)}:=s_h,s_{2h},\ldots,s_{(2n-1)h}$ and $s^{(k)}:=s_k,s_{2k},\ldots,s_{(2n-1)k}$, where h and k are two integers such that $1\leq h,k<\text{ord}(g(x))=e$ and gcd(h,e)=gcd(k,e)=1. The task of the cryptanalyst is to determine h and k.

F. Pichler (Ed.): Advances in Cryptology - EUROCRYPT '85, LNCS 219, pp. 40-42, 1986.

2 A possible attack

Since $s^{(k)}$ is known the cryptanalyst can compute the minimal polynomial $g^{(k)}(x)$ of this decimated sequence. From gcd(k,e)=1, it follows that $g^{(k)}(x)$ is prime over K and deg $g^{(k)}(x)=n$, [2]. Hence the roots of g(x) and $g^{(k)}(x)$ lie in an extension field F of K of degree n. The cryptanalyst proceeds by writing

$$g(x) = \prod_{i=1}^{n} (x-r_i),\ r_i \in F(=GF(2^n)),$$

and

$$g^{(k)}(x) = \prod_{j=1}^{n} (x-t_j),\ t_i \in F.$$

Both polynomials are prime, hence the roots of each of them are distinct. In general, obtaining all the roots of g(x) (or $g^{(k)}(x)$) in F requires $O(n\ln n)^2 \ln\ln n \ln p)$ operations in F by one of the root finding algorithms, [3], see also [4]. The field F is realized as K[x]/p(x)K[x], where p(x) is a maximum length polynomial of degree n over K. The polynomial p(x) is chosen such that the choice has a profitable effect on the algorithm for computing discrete logarithms in F, [5]. Now there exist a j such that $t_1=r_j^k$. Let us assume that j=1. If j≠1 then the cryptanalyst has to try an other value of j. The latter implies that when our cryptanalyst is very unlucky he has to repeat n times the computations that remain at this point. Having expressed both r_1 and t_1 as elements in the field K[x]/p(x)K[x], the cryptanalyst can compute the logarithms of these two elements. Let l denote log r_1 and let l′ denote log t_1. From the relation $t_1=r_1^k$ follows the equation l′=kl mod 2^n-1. This equation has to be solved for k under the condition $1 \leq k < e$. Using the fact that $e=(2^n-1)/d$, where $d=\gcd(l,2^n-1)$, and recalling that gcd(k,e)=1 and thus $\gcd(k,2^n-1)=1$, the cryptanalyst obtains

$$k=l'l^{-1} \bmod (2^n-1)/d.$$

Now it is clear that the values of h and k can be computed. When ignoring the computations of the roots the amount of work is roughly O(nf(n)), where O(f(n)) is the required work to compute a logarithm in F when the necessary precomputations have been done. In [6] it is indicated that for n=460 these precomputations require about a year on a modern supercomputer. In the same paper it is shown that O(f(n)) can be done in much less time.

3 Conclusions

When g(x) is not a prime polynomial then mutatis mutandis the same approach can be used to obtain h and k. And since the logarithm algorithm also works when one would have taken $GF(2^m)$ instead of GF(2), the PKC is vulnerable even in those cases. Summerizing, for the most interesting practical situations, i.e. p=2 and n<500, the proposed PKC system is not secure. In general the system is at most n times as complex as the discrete logarithm problem after precomputations.

REFERENCES

[1] H. Niederreiter, "A public-key cryptosystem based on shift register sequences", Proceedings of EUROCRYPT 85 , (F. Pichler,T. Beth, ed), Springer Lecture Notes, to appear.

[2] N. Zierler, "Linear recurring sequences", SIAM Journ., vol. 7, (1959), pp. 31-48, Reprinted in Linear sequential switching circuits , (W.H. Kautz, ed), Holden-Day, San Fransisco, 1965.

[3] M. Ben-Or, "Probabilistic algorithms in finite fields", Proceedings of Foundations of Computer Science , 1981, pp.394-398.

[4] E.R. Berlekamp, "Factoring polynomials over large finite fields", Math. Comp., vol. 24, (1970), pp. 713-735.

[5] D. Coppersmith, "Fast evaluation of logarithms in finite fields of characteristic two, "IEEE Trans. on Inform. Theory, IT-30, (1984), pp. 587-594.

[6] A.M. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance", preliminary report, Bell Labs.