J. Cryptology (1991) 3: 149–155

Journal of Cryptology

On the Chor–Rivest Knapsack Cryptosystem[1]

H. W. Lenstra, Jr.
Department of Mathematics, University of California, Berkeley, CA 94720, U.S.A.

Abstract. Among all public-key cryptosystems that depend on the knapsack problem, the system proposed by Chor and Rivest (*IEEE Trans. Inform. Theory* **34** (1988), 901–909) is one of the few that have not been broken. The main difficulty in implementing their system is the computation of discrete logarithms in large finite fields. In this note we describe the "powerline system," which is a modification of the Chor–Rivest system that does not have this shortcoming. The powerline system, which is not a knapsack system, is at least as secure as the original Chor–Rivest system.

Key words. Public-key cryptosystem, Finite field.

1. Introduction

Among all public-key cryptosystems that depend on the knapsack problem, the system proposed by Chor and Rivest [2], [3] is one of the few that have not been broken [1]. The Chor–Rivest system is based on arithmetic in finite fields. It has the curious feature that its security does not depend on the apparent hardness of any well-known computational problem, such as the discrete logarithm problem. Paradoxically, if the discrete logarithm problem in large finite fields would become tractable, then this would improve the system: it would make it easier to generate, but apparently not easier to break.

In this note we describe the *powerline system*, which is a modification of the Chor–Rivest system. The powerline system is not a knapsack system. It works directly in the multiplicative group of a finite field, without passing to discrete logarithms. The system depends on a collection of elements that all lie on the same line, and that are all raised to the same power. The powerline system achieves the same improvement in system generation that a solution of the discrete logarithm problem would bring about for the Chor–Rivest system.

The powerline system is at least as secure as the Chor–Rivest system, and if the discrete logarithm problem would become tractable then the two systems would be

[1] Date received: May 14, 1990. Date revised: January 7, 1991. The author was supported by NSF under Grant Nos. DMS 87-06176 and DMS 90-02939, and by NSA/MSP under Grant No. MDA90-H-4043.

equally secure. In fact, the fastest method for breaking the powerline system that we know is first applying a discrete logarithm algorithm to reduce it to the Chor–Rivest system, and next breaking the latter system by means of the attack of Brickell [3, Section VII].

Thus we see that the powerline system has a less paradoxical relation to the discrete logarithm problem than the Chor–Rivest system: the discrete logarithm problem does not enter into the system generation, but it does enter into algorithms for breaking the system.

The main advantage of the powerline system over the Chor–Rivest system is the greater freedom it allows in choosing the system parameters, since there is no need to restrict to finite fields for which the discrete logarithm problem is feasible. Using finite fields for which the discrete logarithm problem is *not* feasible might in fact add to the security of the powerline system. There is also a disadvantage: encryption in the powerline system is somewhat slower than in the Chor–Rivest system.

The reader is encouraged to examine the powerline system for possible weaknesses, and to find a feasible method for breaking it.

In Section 2 we describe the powerline system. In Section 3 the Chor–Rivest system, with a few inessential changes, is described. In Section 4 we prove that the powerline system is at least as secure as the Chor–Rivest system. We also compare the performance of the two versions. Section 5 contains the little we know about attacks on the powerline system.

2. Description of the Powerline System

(2.1) *System Generation.* (a) Choose a prime number p. Write $\mathbf{F}_p$ for the prime field of p elements. The elements of $\mathbf{F}_p$ can be represented by the integers $0, 1, \ldots, p-1$, the arithmetic operations being defined modulo p.

(b) Choose a positive integer n, and generate an irreducible polynomial $f \in \mathbf{F}_p[X]$ of degree n. This can be done as in [4]. Write q for p^n and $\mathbf{F}_q$ for the field $\mathbf{F}_p[X]/f\mathbf{F}_p[X]$. The elements of $\mathbf{F}_q$ can be represented as vectors $(x_i)_{i=0}^{n-1}$ over $\mathbf{F}_p$, with $(x_i)_{i=0}^{n-1}$ standing for the element $(\sum_{i=0}^{n-1} x_i X^i \bmod f)$ of $\mathbf{F}_q$. The arithmetic operations in $\mathbf{F}_q$ are performed modulo f.

(c) Choose a positive integer h, and generate an irreducible polynomial $g \in \mathbf{F}_q[Y]$ of degree h, as in [4]. Write $\mathbf{F}_{q^h}$ for the field $\mathbf{F}_q[Y]/g\mathbf{F}_q[Y]$. The elements of $\mathbf{F}_{q^h}$ can be represented as vectors $(y_i)_{i=0}^{h-1}$ over $\mathbf{F}_q$, with $(y_i)_{i=0}^{h-1}$ standing for the element $(\sum_{i=0}^{h-1} y_i Y^i \bmod g)$ of $\mathbf{F}_{q^h}$. The arithmetic operations in $\mathbf{F}_{q^h}$ are performed modulo g.

(d) Choose a random element $t \in \mathbf{F}_{q^h}$ satisfying $\mathbf{F}_{q^h} = \mathbf{F}_q(t)$. This can be done by selecting random elements $t \in \mathbf{F}_{q^h}$ until one is found for which $\mathbf{F}_{q^h} = \mathbf{F}_q(t)$. Notice that $\mathbf{F}_{q^h} = \mathbf{F}_q(t)$ if and only if the system $1, t, \ldots, t^{h-1}$ is linearly independent over $\mathbf{F}_q$, and if and only if $t^{q^{h/p'}} \neq t$ for each prime number p' dividing h.

(e) Choose a random element $u \in \mathbf{F}_{q^h}$ with $u \neq 0$.

(f) Choose a random integer k satisfying $1 \leq k \leq q^h - 1$, $\gcd(k, q^h - 1) = 1$.

(g) Choose a positive integer $s \leq q$. Write $S = \{1, 2, \ldots, s\}$.

(h) Choose a random injective map $\pi: S \to \mathbf{F}_q$.

(i) For each $i \in S$, calculate the element $v_i = (ut - u \cdot \pi(i))^k$ of $\mathbf{F}_{q^h}$.

(2.2) *Public Key.* The following information is to be made public: p, n, f, h, and g, so that the used models for $\mathbf{F}_q$ and $\mathbf{F}_{q^h}$ are publicly available; the integer s, and the s elements $v_1, v_2, \ldots, v_s$ of $\mathbf{F}_{q^h}$.

(2.3) *Private Key.* The following are kept secret: t, u, k, and π.

(2.4) *Message.* A message is by definition a sequence $m = (m_1, m_2, \ldots, m_s)$ of nonnegative integers satisfying $\sum_{i=1}^{s} m_i = h$. To transform conventional messages to this form we can apply an algorithm similar to that in Section IV.B of [3].

(2.5) *Encryption.* To encrypt a message $m = (m_1, m_2, \ldots, m_s)$, calculate the element $e(m) = \prod_{i=1}^{s} v_i^{m_i}$ of $\mathbf{F}_{q^h}$. This element is to be sent over the insecure channel.

(2.6) *Decryption.* Given $e(m)$, we calculate m as follows, using the secret information. Steps (j), (k), and (l) can be done once and for all at system generation.

(j) Express the elements (Y mod g) and t^h of $\mathbf{F}_{q^h}$ in the basis $1, t, \ldots, t^{h-1}$ of $\mathbf{F}_{q^h}$ over $\mathbf{F}_q$. This can be done by solving two linear systems over $\mathbf{F}_q$. Once it is done, we can use Horner's scheme to express any element ($\sum_{i=0}^{h-1} y_i Y^i$ mod g) of $\mathbf{F}_{q^h}$ in the basis $1, t, \ldots, t^{h-1}$ by performing $h-1$ additions and $h-1$ multiplications modulo the irreducible polynomial of t over $\mathbf{F}_q$:

$$\sum_{i=0}^{h-1} y_i Y^i = ((\cdots((y_{h-1} Y + y_{h-2}) Y + y_{h-3}) Y + \cdots + y_2) Y + y_1) Y + y_0 .$$

The irreducible polynomial of t is obtained from the expression of t^h in $1, t, \ldots, t^{h-1}$.

(k) Calculate the element u^{-h} of $\mathbf{F}_{q^h}$.

(l) Calculate a positive integer l satisfying $kl \equiv 1 \bmod (q^h - 1)$, using the extended Euclidean algorithm.

(m) Calculate $e(m)^l \cdot u^{-h} - t^h$, and express it in the basis $1, t, \ldots, t^{h-1}$ of $\mathbf{F}_{q^h}$ over $\mathbf{F}_q$, using the method described in (j):

$$e(m)^l \cdot u^{-h} - t^h = \sum_{i=0}^{h-1} w_i t^i, \qquad w_i \in \mathbf{F}_q .$$

(n) For each $i \in S$, the number m_i can now be computed as the multiplicity of $\pi(i)$ as a zero of the polynomial $Z^h + \sum_{i=0}^{h-1} w_i Z^i \in \mathbf{F}_q[Z]$. To prove this, it suffices to show that the elements $w_i' \in \mathbf{F}_q$ defined by

$$\prod_{i \in S} (Z - \pi(i))^{m_i} = Z^h + \sum_{i=0}^{h-1} w_i' Z^i$$

satisfy $w_i' = w_i$. Indeed, we have

$$e(m)^l \cdot u^{-h} - t^h = -t^h + u^{-h} \cdot \prod_{i \in S} v_i^{m_i l} = -t^h + u^{-h} \cdot \prod_{i \in S} (ut - u \cdot \pi(i))^{klm_i}$$

$$= -t^h + u^{-h} \cdot \prod_{i \in S} (ut - u \cdot \pi(i))^{m_i} = -t^h + \prod_{i \in S} (t - \pi(i))^{m_i} = \sum_{i=0}^{h-1} w_i' t^i ,$$

and $w_i' = w_i$ now follows from the linear independence of $1, t, \ldots, t^{h-1}$ over $\mathbf{F}_q$.

3. The Chor–Rivest System

For comparison, we present here the Chor–Rivest system.

(3.1) *System Generation.* (a) Perform steps (a), (b), (c) (with $h > 1$), and (d) of Section 2. This provides us with explicit models for the fields $\mathbf{F}_q$ and $\mathbf{F}_{q^h}$, and with an element $t \in \mathbf{F}_{q^h}$ for which $\mathbf{F}_{q^h} = \mathbf{F}_q(t)$.

(b) Perform steps (g) and (h) of Section 2. This provides us with a positive integer $s \leq q$ and an injective map $\pi: S = \{1, 2, \ldots, s\} \to \mathbf{F}_q$. (In [3] only the case $s = q$ was considered. See Section 5.)

(c) Determine a generator r of the multiplicative group of $\mathbf{F}_{q^h}$, and for each $i \in S$ calculate the integer $b_i \bmod (q^h - 1)$ for which $r^{b_i} = t - \pi(i)$. This amounts to the solution of s discrete logarithm problems, which is computationally feasible only for special choices of q and h, see the discussion in [3].

(d) Choose an integer $d \bmod (q^h - 1)$ at random, and calculate, for each $i \in S$, the integer c_i defined by $c_i \equiv b_i + d \bmod (q^h - 1)$, $0 \leq c_i < q^h - 1$.

(3.2) *Public Key.* The following information is to be made public: q, h, s, c_1, $c_2, \ldots, c_s$.

(3.3) *Private Key.* The following are kept secret: t, π, r, and d.

(3.4) *Message.* As in (2.4). (Following [1], we drop the requirement $m_i \in \{0, 1\}$ of [3].)

(3.5) *Encryption.* To encrypt a message $m = (m_1, m_2, \ldots, m_s)$, compute the integer $e'(m)$ defined by $e'(m) \equiv \sum_{i \in S} m_i c_i \bmod (q^h - 1)$, $0 \leq e'(m) < q^h - 1$. This number is to be sent over the insecure channel.

(3.6) *Decryption.* Given $e'(m)$, we calculate m as follows, using the secret information.

(e) Perform step (j) of Section 2. This step can be done once, at system generation. It enables us to express elements of $\mathbf{F}_{q^h}$ in the basis $1, t, \ldots, t^{h-1}$ of $\mathbf{F}_{q^h}$ over $\mathbf{F}_q$.

(f) Compute the element $r^{e'(m)-hd} - t^h$ of $\mathbf{F}_{q^h}$, and express it in the basis $1, t, \ldots, t^{h-1}$ of $\mathbf{F}_{q^h}$ over $\mathbf{F}_q$:

$$r^{e'(m)-hd} - t^h = \sum_{i=0}^{h-1} w_i t^i, \qquad w_i \in \mathbf{F}_q.$$

For each $i \in S$, the number m_i can now be computed as the multiplicity of $\pi(i)$ as a zero of the polynomial $Z^h + \sum_{i=0}^{h-1} w_i Z^i \in \mathbf{F}_q[Z]$. This follows from

$$r^{e'(m)-hd} = r^{(\sum_{i \in S} m_i c_i) - hd} = r^{\sum_{i \in S} m_i b_i}$$

$$= \prod_{i \in S} (r^{b_i})^{m_i} = \prod_{i \in S} (t - \pi(i))^{m_i},$$

as in (2.6) (n).

4. Comparison

(4.1) *Security.* The powerline system is at least as secure as the Chor–Rivest system. In other words, any algorithm that given the public information (2.2) and the encrypted message $e(m)$ of the powerline system finds m, can be transformed into an almost equally efficient algorithm that performs the same function for the Chor–Rivest system.

To prove this statement, suppose that the public key q, h, s, $c_1, c_2, \ldots, c_s$ from (3.2) and the encrypted form $e'(m)$ of a message m as in (3.5) are given, and that an algorithm for breaking the powerline system is available. Then m can be recovered as follows.

(a) Construct fields $\mathbf{F}_q \subset \mathbf{F}_{q^h}$ as in (2.1) (b) and (c).

(b) Determine a generator z of the multiplicative group of $\mathbf{F}_{q^h}$; since q, h are the parameters of an instance of the Chor–Rivest scheme, this is supposed to be feasible (see (3.1)(c)).

(c) Let $v_i = z^{c_i}$ for $1 \leq i \leq s$, and compute $z^{e'(m)}$. It is proved below that the models of $\mathbf{F}_q$ and $\mathbf{F}_{q^h}$ constructed in (a), together with the number s from (3.1) (b) and v_1, $v_2, \ldots, v_s$, constitute the public key for an instance of the powerline system (see (2.2)), with $e(m) = z^{e'(m)}$. Hence the algorithm for breaking that system that is supposed to be available can now be used to recover m.

To prove the assertion just made we may, by the uniqueness of finite fields, choose an identification of the finite fields $\mathbf{F}_q \subset \mathbf{F}_{q^h}$ used in (3.1) and the finite fields $\mathbf{F}_q \subset \mathbf{F}_{q^h}$ constructed in (4.1) (a). Modulo this identification, let t in (2.1) (d) be the same as the element t used in (3.1). Let u in (2.1) (e) be defined by $u = r^d$, with r, d as in (3.1) (c) and (d). Let k in (2.1) (f) be such that $z = r^k$, and let π in (2.1) (h) be the same as in (3.1) (b). Then the elements calculated in (2.1) (i) are found to be

$$(ut - u \cdot \pi(i))^k = u^k \cdot (t - \pi(i))^k = r^{dk} r^{b_i k} = r^{c_i k} = z^{c_i},$$

and the encrypted form of the message in (2.5) is

$$e(m) = \prod_{i \in S} (z^{c_i})^{m_i} = z^{e'(m)}.$$

This finishes the proof.

(4.2) *System Performance. System generation.* Once q, h, and s have been chosen, the finite fields that both systems need can be constructed by means of a random algorithm of which the expected running time is polynomial in $\log q$ and h. For the rest, the running time is dominated by step (2.1)(i) for the powerline system, and step (3.1)(c) for the Chor–Rivest system. Step (2.1)(i) can be done by performing $O(sh \log q)$ arithmetic operations in $\mathbf{F}_{q^h}$. With the standard algorithms this takes time $O(s(h \log q)^3)$, and with fast multiplication techniques $O(s(h \log q)^{2+\varepsilon})$ for any $\varepsilon > 0$. The time required by step (3.1)(c) will, even in favorable cases, be much more than this. How much more depends on how efficiently we can compute discrete logarithms in $\mathbf{F}_{q^h}$. The algorithm used in [3] runs in time $s(h \log q)^{O(1)}$ times the square root of the largest prime factor of $q^h - 1$. We conclude that generally the powerline system is easier to generate than the Chor–Rivest system.

Key size. The public key consists of about $sh(\log q)/\log 2$ bits for both systems. The same is true for the private key. This grows by only a constant factor if the precomputed information from steps (2.6)(j), (k), and (l) and (3.6)(e) is also taken into account.

Encryption. In the Chor–Rivest system, encryption amounts to adding h integers modulo $q^h - 1$, which can be done in time $O(h^2 \log q)$. In the powerline system, encryption amounts to multiplying h elements of $\mathbf{F}_{q^h}$. With the standard algorithms this can be done in time $O(h^3(\log q)^2)$. With fast multiplication techniques this can be reduced to $O(h^{2+\varepsilon}(\log q)^{1+\varepsilon})$ for any $\varepsilon > 0$. Hence the Chor–Rivest system is somewhat faster. Note that the model for $\mathbf{F}_{q^h}$ that we use is not secret. Therefore it may be chosen so as to optimize the speed of arithmetic operations.

Decryption. It is not difficult to see that essentially the same operations have to be performed in both systems. The main difference is that the powerline system must calculate $e(m)^l$ where the Chor–Rivest system calculates $r^{e'(m)}$. The latter computation can be made slightly faster if suitable powers of r have been precomputed.

Information rate. This is the same for both systems, namely

$$\left(\log\binom{s+h-1}{h}\right)\Big/(h \log q).$$

(4.3) *Choice of Parameters.* Chor and Rivest recommend taking $s = q \approx 200$ and $h \approx 25$ in their system [3]. Larger prameters make their system difficult to implement (see (4.2)), and smaller parameters affect the security. In the powerline system we do have the freedom to choose larger parameters, or perhaps even smaller ones, provided the discrete logarithm problem is infeasible. At the end of Section 5 we indicate why, for both systems, it may be wise to choose s somewhat smaller than q.

5. Breaking the System

We saw in the previous section that the powerline system is at least as secure as the Chor–Rivest system. From the proof of this assertion it is not difficult to see that the two versions of the system are actually equally secure if q, h are such that the discrete logarithm problem is tractable for $\mathbf{F}_{q^h}$. In fact, the fastest method for breaking the powerline system that we know is first applying a discrete logarithm algorithm to reduce it to the Chor–Rivest system, and next breaking the Chor–Rivest system by means of the attack of Brickell [3, Section VII].

The problem of breaking the powerline system can be formulated as follows. We are given s elements $v_1, v_2, \ldots, v_s$ of an explicitly given finite field $\mathbf{F}_{q^h}$, and in addition we are supplied with the information that there exists a positive integer l, coprime to $q^h - 1$, such that the lth powers of these elements lie on a straight $\mathbf{F}_q$-line, i.e., a set of the form $u\mathbf{F}_q + ut$, with $u, t \in \mathbf{F}_{q^h}$, $u \neq 0$. The problem is to determine such an integer l. (The additional restriction that $\mathbf{F}_{q^h} = \mathbf{F}_q(t)$ is equivalent to $\mathbf{F}_{q^h} = \mathbf{F}_q(v_i/v_j)$ for all $i \neq j$, which can be verified directly.)

Note that any solution l gives another solution when multiplied by $p \pmod{q^h - 1}$, so that there exist at least nh solutions $\pmod{q^h - 1}$; here $q = p^n$.

If we choose $s = q$ in (2.1)(g), then $\{v_i^l : 1 \leq i \leq q\}$ is *equal* to a straight line, which is equivalent to

$$\prod_{i=1}^{q} (Z - v_i^l) = Z^q - aZ - b$$

in the polynomial ring $\mathbf{F}_{q^h}[Z]$, for certain $a, b \in \mathbf{F}_{q^h}$. It is conceivable that this information represents a weakness, so that it would be advisable to choose s somewhat smaller than q.

References

[1] E. F. Brickell, A. M. Odlyzko, Cryptanalysis: a survey of recent results, *Proc. IEEE* **76** (1988), 578–593.

[2] B.-Z. Chor, *Two Issues in Public Key Cryptography, RSA Bit Security and a New Knapsack Type System*, MIT Press, Cambridge, Mass., 1986.

[3] B. Chor, R. L. Rivest, A knapsack-type public key cryptosystem based on arithmetic in finite fields, *IEEE Trans. Inform. Theory* **34** (1988), 901–909.

[4] A. K. Lenstra, Factorization of polynomials, in: H. W. Lenstra, Jr., and R. Tijdeman (eds), *Computational Methods in Number Theory*, pp. 169–198, Mathematical Centre Tracts 154/155, Mathematisch Centrum, Amsterdam, 1982.