Programming Techniques

R. Rivest
Editor

How to Share a Secret

Adi Shamir
Massachusetts Institute of Technology

In this paper we show how to divide data D into n pieces in such a way that D is easily reconstructable from any k pieces, but even complete knowledge of $k-1$ pieces reveals absolutely no information about D. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.

论文中，我们会展示如何将数据D分成n份的方法，这种方法可中选择任意K块就能很容易的重构数据，但是少于k块则不可能重构出D。

Key Words and Phrases: cryptography, key management, interpolation

CR Categories: 5:39, 5.6

1. Introduction

In [4], Liu considers the following problem:

> Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?

11名科学家在进行秘密项目。有一个放有机密文档的箱子，该箱子需要至少6名科学家才能够破解。求问至少需要多少锁，每名科学

It is not hard to show that the minimal solution uses 462 locks and 252 keys per scientist. These numbers are clearly impractical, and they become exponentially worse when the number of scientists increases.

In this paper we generalize the problem to one in which the secret is some data D (e.g., the safe combination) and in which nonmechanical solutions (which manipulate this data) are also allowed. Our goal is to divide D into n pieces $D_1, \ldots, D_n$ in such a way that:

(1) knowledge of any k or more D_i pieces makes D easily computable;

(2) knowledge of any $k-1$ or fewer D_i pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).

目标：将数据D分成n份，分别为D1,D2,...,Dn，并有以下规则：
1．任何k或更多的Di块，能够很容易的计算D。
2．任何k-1或更少的Di块，会使D完全不确定（不可能得到D）。
这就叫(k,n)门限方案。

Such a scheme is called a *(k, n) threshold scheme.*

Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt it, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location (a computer, a human brain, or a safe). This scheme is highly unreliable since a single misfortune (a computer breakdown, sudden death, or sabotage) can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches (computer penetration, betrayal, or human errors). By using a (k, n) threshold scheme with $n = 2k-1$ we get a very robust key management scheme: We can recover the original key even when $\lfloor n/2 \rfloor = k-1$ of the n pieces are destroyed, but our opponents cannot reconstruct the key even when security breaches expose $\lfloor n/2 \rfloor = k-1$ of the remaining k pieces.

为了保护数据，我们可以对数据进行加密，但是为了保护加密密钥，我们就需要一个截然不同的方法（更进一步的加密—对密钥加密，改变了问题而不是解决了问题。最安全的密钥管理模式使将密钥存放在一个单独的、有良好安保措施的环境（电脑、人脑、安全之地），但是这种模式使非常不可靠的，因为任意不幸事件发生都会使信息处于不可获取的情况。对于这种模式，一个显而易见的解决方式就是在不同的地方存储多个备份密钥，但是这样就增加了系统被统计的危险程度。
通过使用(k,n)门限方案（n为基数2k-1），我们可以得到一个很强壮的密钥管理模式:即便有floor(n/2)个密钥被摧毁，我们都能够将数据恢复，但是攻击者却不能通过这一般的密钥获得我们的数据。

门限方案不是在安全性和可靠性之间的妥协，而是在安全性和使用方便性。

In other applications the tradeoff is not between secrecy and reliability, but between safety and convenience of use. Consider, for example, a company that digitally signs all its checks (see RSA [5]). If each executive is given a copy of the company's secret signature key, the system is convenient but easy to misuse. If the cooperation of all the company's executives is necessary in order to sign each check, the system is safe but inconvenient. The standard solution requires at least three signatures per check, and it is easy to implement with a $(3, n)$ threshold scheme. Each executive is given a small magnetic card with one D_i piece, and the company's signature generating device accepts any three of them in order to generate (and later destroy) a temporary copy of the actual signature key D. The device does not contain any secret information and thus it need not be protected against inspection. An unfaithful executive must have at least two accomplices in order to forge the company's signature in this scheme.

不忠诚的执行官必须拥有至少两名共犯，才能伪造公司的签名。

Threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it.

门限方案非常理想的适用于存在利益冲突的互相可疑个体群组必须合作的场景。理想情况下，我们希望公司能够基于不变而相同的理念，但是这个机制所赋予每个成员的否决权可以使群体的活动麻痹。通过选择合适的k和n参数，我们能够给大多数人权利去采取行动，而给其余少数人权利去阻止它。

Author's present address: A. Shamir, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

This research was supported by the Office of Naval Research under contract no. N00014-76-C-0366.

2. A Simple (k, n) Threshold Scheme

门限方案使基于多项式插值法：给定二维平面(x_i,y_i)上k个x相异的点，这里有且仅存在唯一一个k-1次幂的多项式q(x)，使得对于所有的i，$q(x_i)=y_i$。

不失一般性，我们可以假设D是一个数字。为了将D分成块Di，我们随机选取k-1次幂多项式。

Our scheme is based on polynomial[1] interpolation: given k points in the 2-dimensional plane $(x_1, y_1), \ldots, (x_k, y_k)$ with distinct x_i's, there is one and only one polynomial $q(x)$ of degree $k-1$ such that $q(x_i) = y_i$ for all i. Without loss of generality, we can assume that the data D is (or can be made) a number. To divide it into pieces D_i, we pick a random $k-1$ degree polynomial $q(x) = a_0 + a_1 x + \ldots a_{k-1} x^{k-1}$ in which $a_0 = D$, and evaluate:

k-1次多项式：指的是一一映射的关系，即一个x，对应一个y，不存在一个x对应多个y的存在。

$$D_1 = q(1), \ldots, D_i = q(i), \ldots, D_n = q(n).$$

求解一个k-1次多项式，多项式有k个未知数$(a_0,a_1,a_2,\ldots,a_{k-1})$，求出a，需要至少k-1个D值。

Given any subset of k of these D_i values (together with their identifying indices), we can find the coefficients of $q(x)$ by interpolation, and then evaluate $D = q(0)$. Knowledge of just $k-1$ of these values, on the other hand, does not suffice in order to calculate D.

使用mod模运算代替实数运算。原理：整数集合对素数P求余会生成一个可以进行插值的域。

给定数据Data，随机选择一个比D和n都要大的素数。从[0,p)的正态分布中随机选取k-1个数，作为q(x)的系数a1,a2,...,ak-1，Di的值都需要经过mod P运算。

To make this claim more precise, we use modular arithmetic instead of real arithmetic. The set of integers modulo a prime number p forms a field in which interpolation is possible. Given an integer valued data D, we pick a prime p which is bigger than both D and n. The coefficients $a_1, \ldots, a_{k-1}$ in $q(x)$ are randomly chosen from a uniform distribution over the integers in $[0, p)$, and the values $D_1, \ldots, D_n$ are computed modulo p.

假设n块数据中的k-1块暴露给了攻击者。对于任意一个[0,p)上的可选值D'，他可以通过这k-1个参数构造一个唯一的k-1次多项式q'(x)，使得q'(0)=D',q'(i)=Di。通过构造，这些与p相关的可能多项式基本相似，所以攻击者绝对不可能知道关于D的任何信息。

Let us now assume that $k-1$ of these n pieces are revealed to an opponent. For each candidate value D' in $[0, p)$ he can construct one and only one polynomial $q'(x)$ of degree $k-1$ such that $q'(0) = D'$ and $q'(i) = D_i$ for the $k-1$ given arguments. By construction, these p possible polynomials are equally likely, and thus there is abolutely nothing the opponent can deduce about the real value of D.

二次多项式算法对于实际中的密钥管理已经是足够快的了。如果D太长了，建议将它分解成多个块以避免多精度算数运算问题。但是数据块block也不能太小，因为最小的可用值p的数量是n+1。

Efficient $O(n \log^2 n)$ algorithms for polynomial evaluation and interpolation are discussed in [1] and [3], but even the straightforward quadratic algorithms are fast enough for practical key management schemes. If the number D is long, it is advisable to break it into shorter blocks of bits (which are handled separately) in order to avoid multiprecision arithmetic operations. The blocks cannot be arbitrarily short, since the smallest usable value of p is $n+1$ (there must be at least $n+1$ distinct arguments in $[0, p)$ to evaluate $q(x)$ at). However, this is not a severe limitation since sixteen bit modulus (which can be handled by a cheap sixteen bit arithmetic unit) suffices for applications with up to 64,000 D_i pieces.

Some of the useful properties of this (k, n) threshold scheme (when compared to the mechanical locks and keys solutions) are:

(k,n)门限方案有用参数：
1. 子块的大小不能超过原数据块的大小。
2. 如果k保持固定不变，Di子块的数量在不影响其他子块的前提下，可以随意增减。
3. 在不改变原始数据D的前提下，可以很容易的改变子数据块Di，我们需要做的只是一个新的多项式罢了。定期更换多项式还要助于提高安全性。
4. 通过 多项式中的元组值，可以获得一个以数据块个数为标准的层次结构（数据块个数等价于重要性）。

(1) The size of each piece does not exceed the size of the original data.

(2) When k is kept fixed, D_i pieces can be dynamically added or deleted (e.g., when executives join or leave the company) without affecting the other D_i pieces. (A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.)

(3) It is easy to change the D_i pieces without changing the original data D—all we need is a new polynomial $q(x)$ with the same free term. A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the $q(x)$ polynomial.

of polynomial values as D_i pieces, we can get a hierarchical scheme in which the number of pieces needed to determine D depends on their importance. For example, if we give the company's president three values of $q(x)$, each vice-president two values of $q(x)$, and each executive one value of $q(x)$, then a $(3, n)$ threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone.

A different (and somewhat less efficient) threshold scheme was recently developed by G.R. Blakley [2].

Received April 1979; revised September 1979

References

1. Aho, A., Hopcroft, J., and Ullman, J. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, Mass., 1974.
2. Blakley, G.R. Safeguarding cryptographic keys. Proc. AFIPS 1979 NCC, Vol. 48, Arlington, Va., June 1979, pp. 313–317.
3. Knuth, D. *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., 1969.
4. Liu, C.L. *Introduction to Combinatorial Mathematics.* McGraw-Hill, New York, 1968.
5. Rivest, R., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM 21,* 2 (Feb. 1978), 120–126.

[1]The polynomials can be replaced by any other collection of functions which are easy to evaluate and to interpolate.