Prace Naukowe Wyższej Szkoły Pedagogicznej w Częstochowie
Matematyka VIII, Częstochowa 2000 - 2001

Distributed Password for Quorum and Veto

Wojciech Dobrzański and Marian Srebrny

Abstract

A simple algorithm for sharing a secret with some nonmonotone access structures will be presented. This is achieved by splitting a secret password, represented by a large prime number, between n participants in such a way that only by pooling a qualified set of shares one can recover the original password. Executing the power of veto in addition to a qualified set provides a nonqualified (forbidden) superset. The cryptographic power of our algorithm is based on computational hardness of the integer factorization.

Introduction

In this paper we study some problems of enhanced protection of secret passwords/passphrases/cryptographic keys. In this introduction we give some motivation and terminology - the basic concepts of the area of secret sharing and their context. In section 1 we recall very briefly the algorithm introduced recently in [3]. In section 2 we present an expansion of that algorithm realizing a simple nonmonotone access structure for a scheme of sharing a secret with a vetoing share.

The problem of secret sharing was originally formulated and solved by G.R.Blakley [2] and A.Shamir [5] about twenty years ago. The interested reader is referred also to Doug Stinson's WWW page [6] containing a nearly up-to-date bibliography of the area. In this paper we use a recently proposed algorithm of [3], cryptographic power of which is based on intractability of integer factorization.

Dynamic progress in the area of information processing technology and fast integration of computer systems via local and global networks brings new types of challenges concerned with increased level of security. Often teleinformation systems take care of serious financial transactions and transfers and process very sensitive data. Access system protection has become one of the central issues. Modern cryptography provides specially designed

algorithms and techniques of encryption and mechanisms for controlling access to sensitive resources.

This paper is devoted to some methods of enhanced protection of private secret access passwords and cryptographic keys, based on secret sharing and their use in the protocols of electronic voting.

The passwords and keys that we are interested in will be represented as numbers or sequences of digits, although in general they could be any mathematical or physical objects. A bank PIN code can serve as a motivating example. A client gets his PIN code from his bank and uses it to identify himself to the bank's computer system. The bank's computer compares it with the pattern stored at the bank.

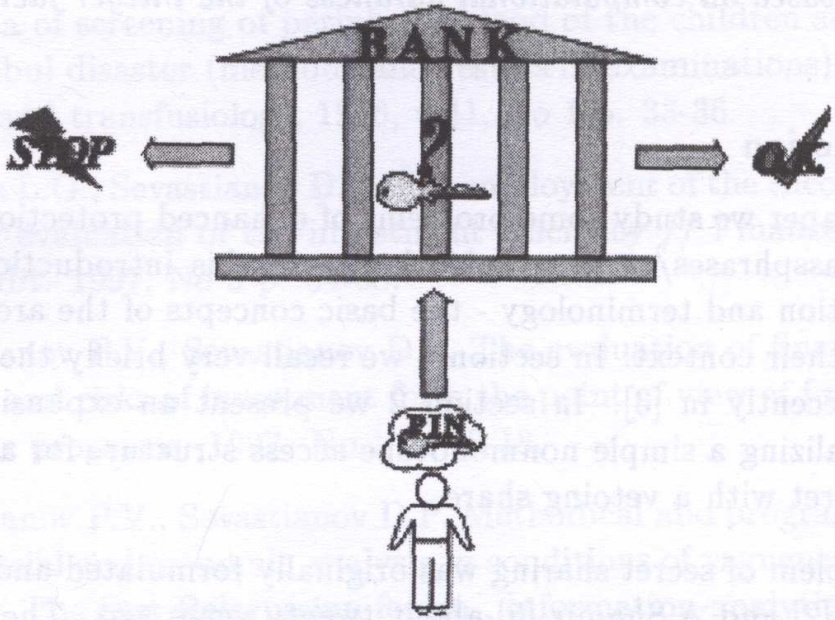

Figure 1. A scheme of identity verification.

So far the files of access passphrases are stored on specially guarded computers, usually in the enciphered form with one of the well known ciphering algorithms (e.g., DES, 3DES, SHA-1, MD5).

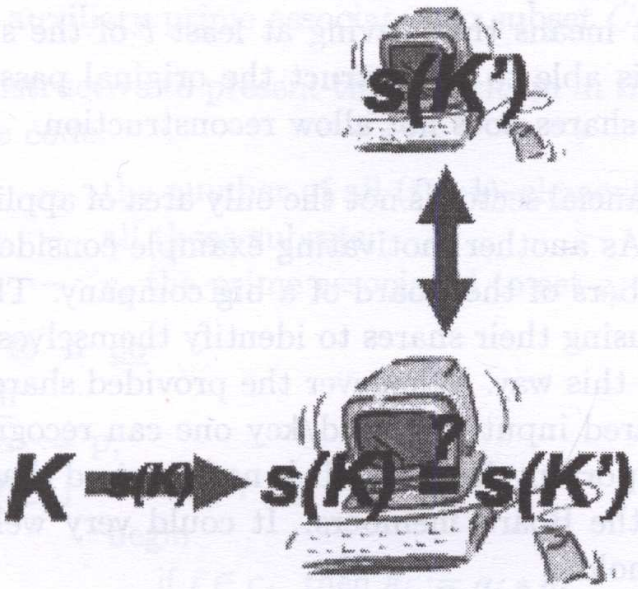

Figure 2. Password/key K stored in the system „in one enciphered piece".

Enhanced protection can be achieved by splitting a given password/key K into n portions, often called shadows or **shares**. These are not supposed to be just structural subsets but rather kinds of partial information about K. Storing such shares on distinct computers can contribute signicantly to the level of security of the whole system.

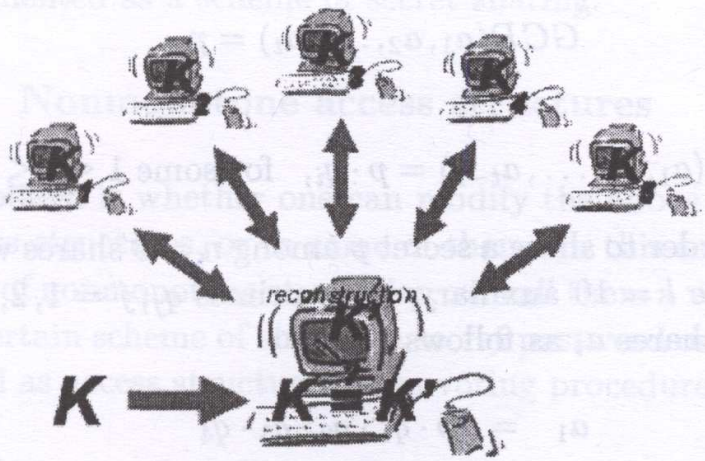

Figure 3. Password/key K stored in the system as 5 distributed shares: $K_1, K_2, \ldots, K_5$.

Potential intruder who wishes to get hold on the secret stored in this way will have to break into several computers, not just one, in order to steal the shares he needs to recover a given password. The information how many shares are necessary and sufficient is often called the threshold t of the distribution. It means that having at least t of the shares of a given password/key one is able to reconstruct the original password/key, while pooling less than t shares does not allow reconstruction.

Clearly, the financial sector is not the only area of applications of secret sharing protocols. As another motivating example consider distribution of shares among members of the Board of a big company. They can communicate by Internet using their shares to identify themselves. One can think of electronic voting this way. Whenever the provided shares are enough to reconstruct the shared input password/key one can recognize the quorum entitled to make important decisions. It is not required that quorum means more than half of the Board members. It could very well be formulated with another threshold.

Section 1. Monotone access structures

In this section we recall very briefly the algorithm introduced in [3] for secret sharing with monotone access structures. The algorithm takes a secret password/key as represented by a large prime integer p. It returns n shares with threshold t. It computes each share $a_i, i = 1, 2, \ldots, n$ as the product of p and several auxiliary primes q_j. A list of $l = \binom{n}{t-1}$ of those auxiliary primes is another input data. Notice that l is the number of the $(t-1)$–element subsets of $\{1, 2, \ldots, n\}$. We want the shares to satisfy the following conditions:

$$GCD(a_1, a_2, \ldots, a_t) = p$$

and

$$GCD(a_1, a_2, \ldots, a_{t-1}) = p \cdot q_i, \quad \text{for some } 1 \leq i \leq l.$$

For example, in order to share a secret p among $n = 5$ shares with threshold $t = 3$, we generate $l = 10$ auxiliary, large primes $q_j, j = 1, 2, \ldots, 10$. Then we calculate the shares a_i as follows:

$$\begin{aligned} a_1 &= p \cdot q_1 \cdot q_2 \cdot q_3 \cdot q_4 \\ a_2 &= p \cdot q_1 \cdot q_5 \cdot q_6 \cdot q_7 \\ a_3 &= p \cdot q_2 \cdot q_5 \cdot q_8 \cdot q_9 \\ a_4 &= p \cdot q_3 \cdot q_6 \cdot q_8 \cdot q_{10} \\ a_5 &= p \cdot q_4 \cdot q_7 \cdot q_9 \cdot q_{10} \end{aligned}$$

In general:

$$a_i = p \cdot \prod\{q_C : C \subset \{1, \ldots, n\}, |C| = t-1, i \in C\}$$

where q_c is an auxiliary prime associated to subset C.

It seems instructive to present this algorithm in the form of a fragment of a Pascal-like code:

l – the number of all $(t-1)$–element subsets of$\{1, \ldots, n\}$;
$\{c_j\}_{1 \le j \le l}$ – all those subsets;
$\{q_j\}_{1 \le j \le l}$ – q_j the prime associated to set c_j

```
for i := 1 to n do
    begin
        a_i := p;
        for j := 1 to l do
            begin
                if i ∈ c_j then a_i := a_i * q_j
            end;
    end;
```

In the context of the above example of electronic voting, each subset $C \subset \{1, 2, \ldots, n\}$ with $|C| \geq t$ can be considered as forming a quorum during a potential meeting. Those subsets can be called the qualified sets. These are exactly the subsets enabling reconstruction of the secret password/key p. In this example, clearly, the family of qualified sets, called also the access structure, is closed under supersets. Such families are called monotone. In finite combinatorics these are called filters. It was shown in [1] and [4] that every monotone family of qualified subsets (access structure) can be imlemented as a scheme of secret sharing.

Section 2. Nonmonotone access structures

The problem is whether one can modify the above scheme to nonmonotone access structures, or to some of them. In this section we introduce certain type of nonmonotone structures, we call them a veto problem, which determine certain scheme of sharing a secret password or key. These can be implemented as access structures or as voting procedures with presidential veto power.

In the above mentioned example of electronic voting we would like the President of the Board to have an extra option to veto any possible decision of the Board. Technically, it means that the Presidential vetoing share should block the password/key reconstruction procedures available to all of

the so far qualified sets. Clearly, in this scenario the access structure, the family of qualified sets, is not closed under supersets. It is not monotone.

In order to implement it we expand the algorithm of section 1 by providing an extra share a_v to be given to the President of the Board. We define it as the product of any two of the auxiliary large primes q_j; that is,

$$a_v = q_i \cdot q_j, \quad \text{for some} \quad 1 \leq i, j \leq l,$$

Assuming the auxiliary primes have been carefully securely generated (see [3]), it is easy to see that supplying his share a_v in the process of voting the President prevents reconstruction of the secret password/key:

$$GCD(a_{i_1}, a_{i_2}, \ldots, a_{i_s}, a_v) \neq p, \quad \text{for any} \quad 1 \leq i_s \leq n.$$

In the case of distribution between $n = 5$ shares with threshold $t = 3$ we get

$$\begin{aligned} a_1 &= p \cdot q_1 \cdot q_2 \cdot q_3 \cdot q_4 \\ a_2 &= p \cdot q_1 \cdot q_5 \cdot q_6 \cdot q_7 \\ a_3 &= p \cdot q_2 \cdot q_5 \cdot q_8 \cdot q_9 \\ a_4 &= p \cdot q_3 \cdot q_6 \cdot q_8 \cdot q_{10} \\ a_5 &= p \cdot q_4 \cdot q_7 \cdot q_9 \cdot q_{10} \end{aligned}$$

and the vetoing share:

$$a_v = q_1 \cdot q_2.$$

Concluding the paper, we would like to emphasize that the same idea gives realization of some other nonmonotone access structures. Definitely, however, it does not work in many other nonmonotone cases. It remains as an intriguing open question what nonmonotone families of sets can be efficiently implemented as secret sharing access structures.

REFERENCES

[1] Benaloh J. and Leichter J.: *Generalized secret sharing and monotone functions,* CRYPTO'88, Lecture Notes in Computer Science 403(1990), pp. 27-35.

[2] Blakley G.R.: *Safeguarding cryptographic keys,* AFIPS Conference Proceedings 48(1979), pp. 313-317.

[3] Dobrzański W., Soroczuk A., Srebrny M.A.Jr and Srebrny M.: *Distributed password,* Technical Report, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, December 1999.

[4] Lonc Z., Marek V.W. and Srebrny M.: *A Combinatorial Algorithm for Sharing a Key,* in preparation.

[5] Shamir A.: *How to share a secret,* Communications of the ACM 22(1979), pp.612-613.

[6] Stinson D.: http://cacr.math.uwaterloo.ca/ dstinson/

Wojciech Dobrzański
Institute of Mathematics and Computer Science
Technical University of Częstochowa
wojtekd@matinf.pcz.czest.pl

Marian Srebrny
Institute of Computer Science
Polish Academy of Sciences, Warsaw
marians@ipipan.waw.pl