An Efficient Identification Scheme Based on Permuted Kernels (extended abstract)

Adi Shamir
Applied Mathematics Department
The Weizmann Institute of Science

Abstract. *In 1985 Goldwasser Micali and Rackoff proposed a new type of interactive proof system which reveals no knowledge whatsoever about the assertion except its validity. The practical significance of these proofs was demonstrated in 1986 by Fiat and Shamir, who showed how to use efficient zero knowledge proofs of quadratic residuosity to establish user identities and to digitally sign messages. In this paper we propose a new zero knowledge identification scheme, which is even faster than the Fiat-Shamir scheme, using a small number of communicated bits, simple 8-bit arithmetic operations, and compact public and private keys. The security of the new scheme depends on an NP-complete algebraic problem rather than on factoring, and thus it widens the basis of public key cryptography, which has become dangerously dependent on the difficulty of a single problem.*

1. The Basic Scheme

Notation:

Throughout this paper, we use upper case letters to denote vectors and matrices, and lower case letters to denote values. Greek letters denote permutations over $\{1,\ldots,n\}$, and their effect V_π on n-vectors V is defined as the vector W such that $w_j = v_{\pi(j)}$ for $1 \leq j \leq n$. The effect of permutations on matrices is defined as the column permutation $A_\pi = [a_{i\pi(j)}]$ so that for any matrix A and vector V, $A_\pi V_\pi = [\sum_{j=1}^{n} a_{i\pi(j)} v_{\pi(j)}] = [\sum_{j=1}^{n} a_{ij} v_j] = AV$. Permutations are composed as functions, and thus $V_{\pi\sigma}$ is defined as the vector W such that $w_j = v_{\pi(\sigma(j))}$ for $1 \leq j \leq n$. All the arithmetic operations in this paper are carried out modulo p, where p is a (small) prime. We define the kernel $K(A)$ of a rectangular $m \times n$ matrix A as the set of n-vectors W such that $AW = 0 \pmod p$, where 0 is the m-vector of zeroes. It is easy to see that $K(A)$ is a linear subspace of $\mathbf{Z}_p^n$ and that $K(A_\sigma) = (K(A))_\sigma$.

The Permuted Kernel Problem (PKP) is:

Given: a $m \times n$ matrix A, a n-vector V, and a prime p;
Find: a permutation π such that $V_\pi \in K(A)$.

The related problems of finding some, all, or randomly chosen vectors in $K(A)$ can be solved by straightforward techniques in linear algebra. The problem of

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 606-609, 1990.

finding good approximations in $K(A)$ to a given vector V (and in particular small non-zero vectors in $K(A)$) can be solved by more complicated (but polynomial) lattice reduction techniques. What makes the Permuted Kernel Problem difficult is that it forces us to choose a kernel vector with a particular set of entries. In fact, it is easy to see that the problem is NP-complete even for $m = 1$ and $V = (+1, +1, \ldots, +1, -1, -1, \ldots, -1)$ since this is just the partition problem for the weights in A. A slightly more complicated reduction from the problem of 3-partition (Garey and Johnson [1979], pp 224) shows that the PKP is NP-complete in the strong sense (i.e., its difficulty grows exponentially in p rather than in $\log(p)$, under appropriate assumptions). This makes it possible to use small numbers in the proposed identification scheme, which greatly enhances its simplicity and speed.

To use the permuted kernel problem as an identification scheme, the users agree on a universal matrix A and prime p, and then each user chooses a random permutation π (which serves as his secret key) and a random vector V such that $V_\pi \in K(A)$ (which serves as his public key). Users can now establish their identity by proving their knowledge of the secret permutation π. By using zero knowledge proofs, provers can guarantee that eavesdroppers and dishonest verifiers will not learn anything about π which will later enable them to misrepresent themselves as the prover to others.

The following protocol uses a hash function which commits the prover to his chosen values without revealing them prematurely to the verifier. Since the function is applied to highly redundant inputs with a large compression ratio in a non-invertible way, we believe that efficient DES-like functions will be sufficiently secure in practice.

Zero knowledge proofs for the Permuted Kernel Problem:

1. The prover chooses a random vector R and a random permutation σ, and sends the cryptographically hashed values of the pairs (σ, AR) and $(\pi\sigma, R_\sigma)$ to the verifier.
2. The verifier chooses a random value $0 \leq c < p$ and asks the prover to send $W = R_\sigma + cV_{\pi\sigma}$.
3. After receiving W, the verifier asks the prover to reveal either σ or $\pi\sigma$. In the first case the verifier checks that $(\sigma, A_\sigma W)$ hashes to the first given value, and in the second case the verifier checks that $(\pi\sigma, W - cV_{\pi\sigma})$ hashes to the second given value.

An honest prover who knows π will always pass this test, since $A_\sigma W = A_\sigma(R_\sigma + cV_{\pi\sigma}) = A(R + cV_\pi) = AR + cAV_\pi = AR$ and $W - cV_{\pi\sigma} = R_\sigma$ by definition. When a dishonest prover tries to choose the commited values in step 1, he should be prepared to answer $2p$ possible questions. If he can answer correctly $p + 2$ questions, then for the same committed (σ, X) and (τ, Y), there are at least two distinct values c' c'' whose response vectors W' W'' satisfy both conditions.

This leads to the following system of equations:

$$A_\sigma W' = X \qquad A_\sigma W'' = X \qquad W' - c'V_\tau = Y \qquad W'' - c''V_\tau = Y$$

This implies that $(W' - W'') \in K(A_\sigma)$ and $(W' - W'') = (c' - c'')V_\tau$. Since $c' - c'' \neq 0$, $V_{\tau\sigma^{-1}} \in K(A)$ and thus the secret permutation $\pi = \tau\sigma^{-1}$ can be extracted from any $p+2$ correct answers. Consequently, the probability of success when such an π is not known is at most $(p+1)/2p$. Since this value is essentially $1/2$, only 20 iterations are required to reduce the probability of cheating bellow the practical security threshold of 1/1,000,000 for each misrepresentation attempt.

The technical proof that this protocol is zero knowledge will be given in the full version of the paper, but the intuition behind it is very simple: The randomness of R makes the vectors W, AR and R_σ completely random, and the randomness of σ makes the permutation $\pi\sigma$ completely random. The individual messages sent by the prover convey no knowledge, and it is only the prover's willingness to answer both questions for all the possible $c's$ which convinces the verifier that the prover is genuine.

2. Implementation details

The minimum recommended size of n has not been determined so far, but we believe that it should be between 32 and 64. For these n the number of permutations π ranges between $32! = 2^{120}$ and $64! = 2^{296}$, while the fastest attacks we are aware of require between 2^{76} and 2^{184} steps. The prime p should not be too small (since multiple occurrences of values in V (mod p) reduce its number of distinct permutations), and should not be too large (since multiprecision arithmetic is slow). The best choice of p for 8 bit microprocessors seems to be $p = 251$. The choice of m should be based on the approximation $p^m \approx n!$, which describes the combination of parameters at which a randomly chosen instance of PKP is likely to have a unique solution ($p^m > n!$ implies that some of the m rows of A can be discarded without adding spurious PKP solutions, while $p^m < n!$ implies that some of the entries in π can be arbitrarily fixed without losing all the PKP solutions). For $p = 251$ and $n = 32$, m should be about 16, and for $p = 251$ and $n = 64$, m should be about 37.

The matrix A should be randomly chosen. Without loss of generality we can assume that A is given in the block form $A = [A' \mid I]$ where A' is a random $m \times (n-m)$ matrix and I is the $m \times m$ identity matrix, since both users and opponents can apply Gauss elimination to the published A without changing its kernel. Calculating AR (or $A_\sigma W$) is particularly easy in this representation. To demonstrate the actual time complexity of the new zero knowledge proofs, we consider the concrete case of a 16×32 matrix $A = [A' \mid I]$ and $p = 251$. The application of permutations and the addition of vectors of size 32 require negligible amounts of time. In addition, the prover performs one matrix-vector multiplication per iteration, and the verifier performs one matrix-vector multiplication every two iterations (on the average). The simplified 16×16 matrix-vector multiplications require 256 single-byte multiplications, which can be carried out in a few milliseconds

on today's microprocessors. This compares very favorably with number-theoretic schemes, in which the calculation of the product of two 512 bit numbers requires 4096 single-byte multiplications (in addition to the overhead caused by the carry propagation and the modular reduction in multiprecision arithmetic). Since two hashed values (64 bits each) one vector (256 bits) and one permutation (120 bits) are sent in each iteration, the total communication is about 500 bits per round.

Another advantage of the new scheme (which is particularly important in smart card applications) is that it needs very little memory: The public key V of each user can be stored in 256 bits, and the secret key π can be stored in 120 bits. The universal matrix A' can be stored as a pseudo random function of i and j, rather than as an explicit matrix. Since we believe that most A' are usable, fairly simple pseudo random functions can suffice in practice. The elements of A' can be generated upon demand (in the original or permuted order) by invoking this function with appropriate arguments, and thus the calculation of the matrix-vector product needs only a few bytes of working space.

3. Extensions

The basic scheme can be extended in a variety of obvious ways. The underlying field $\mathbf{Z}_p$ can be replaced by other ring structures, the homogeneous equations can be replaced by non-homogeneous equations, and the matrix-vector products can be replaced by higher order tensor products. By adding the message m to the list of hashed arguments the prover can authenticate the contents of the message in addition to proving his identity, and by using the general technique introduced in Fiat and Shamir [1986] this authentication scheme can be turned into a signature scheme. However, PKP-based signatures are much longer than Fiat-Shamir signatures, and their practical significance is unclear.

A detailed analysis of the security of the new identification scheme for various choices of the parameters is underway, and its results will be published in the full version of this paper. In the meantime, we encourage readers to attack the scheme and warn potential users not to adopt it prematurely.