J. Cryptology (1992) 5: 133–137

Journal of Cryptology

Some Remarks on the Security of the Identification Scheme Based on Permuted Kernels

Jean Georgiades
Siemens AG, ZFE ST SN 5, Otto-Hahn-Ring 6,
W-8000 München 83, Federal Republic of Germany

Communicated by Rainer A. Rueppel

Received 27 April 1990 and revised 11 October 1990

Abstract. We present in this paper an idea of how to reduce the number of possible permutations when trying to solve the permuted kernels problem. We refer to the identification scheme of Shamir [2] and we also show how a dishonest prover can maximize his prospects to pass the test.

Key words. Permuted kernels, Number of permutations, Probability of cheating.

1. Introduction

In the rump session of Crypto '89 Shamir presented a new identification scheme based on permuted kernels [2]. The users of this identification scheme agree on a prime p and on a universal $m \times n$ matrix A. The coefficients of A are elements of GF(p). Without loss of generality we can assume that A is given in the block form $A = [I|A']$ where I is the $m \times m$ identity matrix and A' is a random $m \times (n-m)$ matrix. Each user chooses as his secret key a random permutation π and as his public key an n-vector $V = (v_1, v_2, \ldots, v_n)^T \in (\mathrm{GF}(p))^n$ such that $V_\pi \in K(A)$, e.g., $AV_\pi = 0 \bmod p$ (V_π denotes the application of the secret permutation π on the components of the vector V). By proving knowledge of the secret permutation π, users can establish their identity.

In this paper we show how to reduce the number of permutations to be considered when trying to find the secret permutation π. We also present a strategy which maximizes the prospects of a dishonest prover to pass the test proposed in [2].

2. The Algorithm

The identification protocol presented by Shamir is as follows:

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.

It is obvious that an honest prover who knows π will always pass this test.

3. On the Security of the Proposed Scheme

Now consider the kernel $K(A)$ of A. Its size is $p^k, k = n - m$. $K(A)$ can be represented as linear combination of k n-vectors $V_1, V_2, \ldots, V_k$ ($V_i, i = 1, 2, \ldots, k$, being, in the following equation, then n-vector multiplied with λ_i, respectively):

$$K(A) = \lambda_1 \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ \vdots \\ a_{1,m} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} a_{2,1} \\ a_{2,2} \\ \vdots \\ a_{2,m} \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + \lambda_k \begin{bmatrix} a_{k,1} \\ a_{k,2} \\ \vdots \\ a_{k,m} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{1}$$

with constants $a_{1,1}, \ldots, a_{k,m} \in \mathrm{GF}(p)$ (depending on the components of A) and parameters $\lambda_1, \ldots, \lambda_k \in \mathrm{GF}(p)$. Consequently, $K(A)$ is the set of vectors of the form

$$(f_1, f_2, \ldots, f_m, \lambda_1, \lambda_2, \ldots, \lambda_{k-1}, \lambda_k)^{\mathrm{T}} \tag{2}$$

with $f_i = \lambda_1 a_{1,i} + \lambda_2 a_{2,i} + \lambda_3 a_{3,i} + \cdots + \lambda_k a_{k,i} \bmod p$, for $i = 1, 2, \ldots, m$.

It is obvious that when trying to find out the permutation π we do not have to place correctly *all* n but only *the last* $n - m$ coefficients of V_π. In case of success the first m coefficients will be calculated and placed automatically in the kernel equation (1) or (2), else the kernel equation will not provide acceptable coefficients. Therefore the number of the permutations to be considered is not $n!$ (that is the number of all possible permutations of a set of n elements) but is equal to the number of possibilities of picking $(n - m)$ elements out of a set of n elements and placing them correctly. That is,

$$n!/((n - m)! * m!) * (n - m)! = n!/m! \tag{3}$$

Furthermore, we may reduce the number $(n!/m!)$ of permutations to be considered if we succeed in forming equations including $\lambda_1, \lambda_2, \ldots, \lambda_k$. We can obtain such equations by considering the coefficients of the vectors V and V_π. Since $V_\pi \in K(A)$ we get, by using (2),

$$\sum_{i=1}^{n} v_i^r \mod p = \sum_{i=1}^{n} v_{i\pi}^r \mod p = \sum_{i=1}^{m} f_i^r \mod p + \sum_{i=1}^{k} \lambda_i^r \mod p, \tag{4}$$

r being a positive integer.

To compute (4) is easy for $r = 1$ or $r = 2$ but is very complicated for $r > 2$. Nevertheless, by using these equations for $r = 1$ and $r = 2$ we may reduce the number of permutations to be considered to $(n!/(m + 2)!)$ since (according to (3)) we will have to pick $(n - m - 2)$ elements out of a set of n elements and place them correctly.

By using (4) for $r = 1$ we will replace in (1) some λ_i by a linear combination of the other $(k - 1)$ parameters. By using (4) for $r = 2$ we will have to deal with a quadratic equation when replacing another λ_j by the remaining $(k - 2)$ parameters. That means that it will be necessary to compute square roots in GF(p). The effort to compute such a root requires $O(\log p)$ steps [1] but because of the small size of p (8 bits [2]) it seems more efficient to store all elements of GF(p) and their squares in a file which requires only *once* at most $(p - 1)/2$ squarings modulo p. The double effort needed in some cases where the solution of the quadratic equation will provide two different *valid* values for λ_j will be at least equalized by the cases where the equation mentioned above will provide no valid solution for λ_j. The effort needed to compute the remaining first m coefficients of the n-vector V_π is no more than the effort needed to check for a given n-vector T if $T \in K(A)$.

That means by using (4) for $r = 1$ and $r = 2$ we will have to consider, for $n = 32$ and $m = 16$, only $(32!/18!) \approx 2^{65}$ and, for $n = 64$ and $m = 37$, only $(64!/39!) \approx 2^{142}$ possible permutations. These numbers, although still large, are much smaller than the numbers presented in [2].

The following example serves only to demonstrate how (4) can be used. Consider $p = 13$, $m = 3$, $n = 6$, $V = (v_1, v_2, \ldots, v_6)^T = (11, 10, 6, 7, 12, 4)^T$, and

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 & 4 & 6 \\ 0 & 1 & 0 & 5 & 7 & 3 \\ 0 & 0 & 1 & 8 & 9 & 1 \end{bmatrix}.$$

By the following triples we denote the squares (first component) and their square roots (second and third component) modulo 13:

$$(0, 0, 0), (1, 1, 12), (3, 4, 9), (4, 2, 11), (9, 3, 10), (10, 6, 7), (12, 5, 8).$$

Further, it holds that

$$v_1 + v_2 + \cdots + v_6 = 11 \mod 13 \tag{5}$$

and

$$v_1^2 + v_2^2 + \cdots + v_6^2 = 11 \mod 13. \tag{6}$$

$K(A)$ has, according to (2), the following form:

$$K(A) = \begin{bmatrix} 11\lambda_1 + 9\lambda_2 + 7\lambda_3 \\ 8\lambda_1 + 6\lambda_2 + 10\lambda_3 \\ 5\lambda_1 + 4\lambda_2 + 12\lambda_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \mod 13. \tag{2a}$$

By using (4) we get

1. for $r = 1$ (with (5)),

$$11 = 12\lambda_1 + 7\lambda_2 + 4\lambda_3 \mod 13$$

which is equivalent to

$$\lambda_1 = 7\lambda_2 + 4\lambda_3 + 2 \mod 13 \tag{4a}$$

and

2. for $r = 2$ (with (6)),

$$11 = 3\lambda_1^2 + 4\lambda_2^2 + 8\lambda_3^2 + 9\lambda_1\lambda_2 + 5\lambda_1\lambda_3 + 4\lambda_2\lambda_3 \mod 13$$

which provides (with (4a))

$$\lambda_2 = 3\lambda_3 - 2 \pm (5\lambda_3^2 + 6\lambda_3 - 3)^{1/2} \mod 13. \tag{4b}$$

We now only need to find among the six components of V the one corresponding to λ_3.

If $\lambda_3 = 4$, then we get, for λ_2, the values 3 or 4, neither of which is acceptable since no remaining component of V has such a value.

If $\lambda_3 = 6$, 7, 10, or 11, then the value of the discriminant in (4b) is 5 or 11, neither of which is a quadratic residue modulo 13.

If $\lambda_3 = 12$, then we compute, for λ_2, the values 11 and 5. Only the first one is acceptable.

This solution provides, by using (4a) and (2a), the vector $V_\pi = (7, 6, 4, 10, 11, 12)$ and the secret permutation $\pi = (1, 5, 6, 3, 2, 4)$.

This way we reduced the possibilities from $n! = 6! = 720$ to $n!/(m+2)! = 6!/5! = 6$.

4. The Probability To Succeed with Cheating

In [2] it is shown that the probability of a dishonest prover to pass the test given in Section 2 above without knowing a suitable permutation π is at most $(p+1)/2p$. In this section we present a strategy which shows that this bound is tight.

Strategy. At step 1 of the protocol the dishonest prover chooses a random n-vector R and two random permutations σ and τ, and sends the cryptographically hashed values of the pairs (σ, AR) and (τ, R_σ) to the verifier. The prover computes the permutation $\mu = \tau\sigma^{-1}$ and checks if $V_\mu \in K(A)$. If this is the case the dishonest prover can use μ and follow Shamir's protocol with guaranteed success. If μ does not behave as wished, then the dishonest prover can maximize his prospects as follows:

1. If $c = 0$ (with probability $P(c = 0) = 1/p$), then the dishonest prover will pass the test anyway.
2. If $c \neq 0$ (with probability $P(c \neq 0) = (p-1)/p$), then the dishonest prover has to guess whether the verifier will ask him to reveal σ or $\pi\sigma$. The probability of the prover to guess correctly is 1/2.

2.1. In the first case the prover chooses an n-vector $S \in K(A)$ and sends to the verifier the vector $W = R_\sigma + cS_\sigma$. Since $A_\sigma W = A_\sigma(R_\sigma + cS_\sigma) = AR + cAS = AR$ the verifier cannot reveal the bluff.

2.2. In the second case the prover sends to the verifier the n-vector $W = R_\sigma + cV_\tau$. Since $W - cV_\tau = R_\sigma$ by definition the prover passes the test.

Consequently, the probability of success of a dishonest prover to pass the test without knowing such a π is $1/p + (p-1)/2p = (p+1)/2p$.

In addition we point out that if π is the secret key and even if $V_\mu \in K(A)$, the equality $\pi = \mu$ is not guaranteed. It is very easy to give examples for larger parameters but for space reasons let us consider the following example for $p = 13$, $m = 3$, $n = 6$, $V = (10, 11, 9, 8, 7, 0)^T$, $\pi = (1, 4, 3, 5, 2, 6)$, and

$$A = \begin{bmatrix} 1 & 0 & 0 & 3 & 12 & 4 \\ 0 & 1 & 0 & 8 & 3 & 5 \\ 0 & 0 & 1 & 11 & 6 & 8 \end{bmatrix}.$$

It is obvious to show that $V_\pi = (0, 7, 8, 10, 9, 11)^T \in K(A)$. However, by also using the permutation $\mu = (1, 2, 5, 6, 4, 3)$ we find out that $V_\mu = (9, 10, 8, 0, 11, 7)^T \in K(A)$. This example shows that there exist cases in which some lucky dishonest prover may always succeed in his activities without knowing the secret key π.

5. Summary

In this paper we made some remarks on the security of the identification scheme of Shamir based on permuted kernels. We showed a way to reduce the number of the permutations to be considered when trying to find the secret permutation π. We also showed that the bound of the probability of a dishonest prover to pass the test given in [2] is tight.

References

[1] D. H. Lehmer, Computer technology applied to the theory of numbers. In W. J. LeVeque, *Studies in Number Theory*, pp. 117–151, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[2] A. Shamir, An efficient identification scheme based on permuted kernels. In G. Brassard, editor, *Advances in Cryptology—Cryoto '89 Proceedings*, Santa Barbara, California, August 20–24. (Lecture Notes in Computer Sciences, Vol. 435), pp. 606–609, Springer-Verlag, Berlin, 1990.