PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2012, № 1, p. 49–52

Informatics

A PUBLIC-KEY ENCRYPTION SCHEME BASED ON SIMS' ALGORITHM

A. V. SOGHOYAN[*]

Chair of Discrete Mathematics and Theoretical Informatics YSU, Armenia

A public-key encryption scheme is proposed, which uses a special case of Sims' algorithm for construction of "strong" generator sets for permutation groups.

***Keywords*:** permutation group, "strong" generator set, Sims' algorithm, public-key encryption.

The basic ideas for handling permutation groups based on "strong" generator sets were introduced in [1, 2]. Since then the Sims methods are at the heart of the most of computational group theory algorithms.

It was shown that a "strong" set of generators for S_n – the symmetric group of all permutations of an n-element set (or for large enough subgroups in S_n) – can be used to develop a symmetric-key encryption scheme for a block cipher [3]. The aim of the present article is to extend the ideas from [3] to create public-key schemes (definitions for symmetric and public-key encryption schemes can be found in any standard handbook on cryptography like [4]).

Let $G \le S_n$ and $G_{1,2,\dots,i}$ be the subgroup of all permutations in G that point wise stabilize the set $\{1,2,\dots,i\}$, i.e. $G_{1,2,\dots,i} = \{\alpha \in G \mid \alpha(k) = k, k \in \{1,2,\dots,i\}\}$ (here and throughout the article $\alpha(k)$ stands for the image of k under the permutation α). Thus, $G \equiv G_0 \ge G_1 \ge \dots \ge G_{1,2,\dots,i} \ge \dots \ge G_{1,2,\dots,n-1} = \{e\}$. Let X_i be a system of distinct representatives of cosets in the factor-group $G_{1,\dots,i-1}/G_{1,\dots,i}$ (the representative for shall be the identity permutation e). The collection of permutations $X_1 \cup \dots \cup X_{n-1}$ forms a "*strong*" *generators set* (*SGS*) for *G*. This means that any permutation from *G* can be uniquely factored as a product of the form $\alpha_1\alpha_2\dots\alpha_{n-1}$, where $\alpha_i \in X_i$, $i = 1,\dots,n-1$.

Given any set of generators for *G* the Sims' algorithm produces a *SGS*, which usually is arranged as a $n \times n$ table with permutations placed in its cells as follows. Each cell can either contain a permutation or be vacant. The rows and columns of the table are numbered from 1 to *n*. All diagonal cells contain identity permutation and the cells below the diagonal are not used. The cell at the intersection of the *i*-th row and the *j*-th column with $i<j$ may contain only a permutation that maps i into j. Permutations that are contained in the cells of the *i*-th row form a system of distinct representatives for $G_{1,\dots,i-1}/G_{1,\dots,i}$, i.e. the set X_i (see above). A table with a *SGS* for S_n does not contain vacant cells.

[*] E-mail: armensog@gmail.com

Having a table with *SGS* for a group *G* (denoted as SGS_G) and a permutation $\alpha \in S_n$, it is easy to test whether $\alpha \in G$. This aim in view, one has to apply a standard procedure called "cascade" as follows: $\beta := \alpha$;

```
for $i$=1 to $n-1$
    do
     $j := \beta(i)$;
     if (the $j$-th cell in the $i$-th row is vacant) then return "$\alpha \notin G$";
     /*Let $\gamma_i$ is the permutation contained in the $j$-th cell of the $i$-th row;*/
     $\beta := \gamma_i^{-1}\beta$;
    end
return "$\alpha = \gamma_1 \dots \gamma_{n-1}$".
```

The "cascade" procedure applied to α either determines that $\alpha \notin G$ or returns the unique expression of α as $\alpha = \gamma_1 \dots \gamma_{n-1}$ through permutations of the *SGS* from the table, which obviously means that $\alpha \in G$. We denote the result of this procedure by cascade (α, SGS_G).

As described in [3], a table with a *SGS* for S_n may serve as a secret key for a symmetric block cipher. The general scheme is as follows. The set of blocks of the plaintext is one-to-one mapped into the set of vectors $(m_1, \dots, m_{n-1})$, where $m_i \in \{i+1, \dots, n\}$. To encrypt a block of the plaintext one calculates the product $\gamma_{m_1}\gamma_{m_2}\dots\gamma_{m_{n-1}}$, where γ_{m_i} is the permutation contained in the m_i-th cell of the i-th row in the table with *SGS*. So each block of the plaintext is enciphered by the permutation from S_n and the resulting ciphertext is a sequence of permutations. To decrypt the ciphertext one has to apply the "cascade" procedure to its permutations to find the vectors $(m_1, \dots, m_{n-1})$ and to recover the corresponding blocks of the plaintext.

The above scheme can be extended to a public-key encryption scheme. We slightly generalize the definition of the *SGS*.

Let $G \le S_n$ and $G \equiv G_0 \ge G_1 \ge \dots \ge G_i \ge \dots \ge G_m = \{e\}$ be a subgroup chain in *G*. As above, let X_i be a system of distinct representatives of cosets in the factor-group G_{i-1}/G_i (the representative for G_i shall be the identity permutation e). It can be readily verified that each permutation in *G* can be uniquely expressed as a product of the form $\alpha_1\alpha_2 \dots \alpha_m$, where $\alpha_i \in X_i$, i=1,...,m. The collection of permutations $X_1 \cup \dots \cup X_m$ forms a *free "strong" generators set* (*FSGS*) for *G*. As in the case with *SGS* we arrange a *FSGS* in the form of a table that has *m* rows and the number of columns is equal to $\max_i |X_i|$, where $|X_i|$ stands for the number of elements in X_i. All permutations from X_i are distributed in the cells of the *i*-th row contiguously starting from the leftmost cell in an arbitrary order. Note that some rightmost cells may stay vacant.

Let`s assume that a table with *FSGS* for *G* (with *m* rows and k_i non-vacant cells in the *i*-th row) and tables with *SGS* for $G_1, \dots, G_{m-1}$ are given. Now denote *SGS* table for G_i as SGS_i. To verify whether a given permutation α belongs to *G* one can use an extended "cascade" procedure as follows:

```
$\beta := \alpha$; $x := TRUE$;
  for $i = 1$ to $m$
    do
      if ($x$=$FALSE$) then return "$\alpha \notin G$";
```

```
        for $j = 1$ to $k_i$
         do
        $x := FALSE$;
        /*Let $\gamma_j$ is the $j$-th permutation from the $j$-th cell in the $i$-th row;*/
        cascade($\gamma_j^{-1}\beta$, $SGS_i$); /* testing whether $\gamma_j^{-1}\beta \in G_i$ */
        if ($\gamma_j^{-1}\beta \in G_i$) then
          do $\beta := \gamma_j^{-1}\beta$; $\delta_i := \gamma_j$; $x := TRUE$; break;
          end
        end
      end
  return "$\alpha = \delta_1 \dots \delta_m$".
```

Obviously, if the above procedure returns the expression of α as $\delta_1 \dots \delta_m$, then $\alpha \in G$.

Now we describe a general scheme for public-key encryption based on the *FSGS*. Let $FSGS_i$ be a table with *FSGS* for S_n, $S_n \equiv G_0 \geq G_1 \geq \dots \geq G_i \geq \dots \geq G_m = \{e\}$ be a subgroup chain and SGS_i be a *SGS* table for G_i, $i \geq 1$. Let also $\beta_1, \beta_2, \dots, \beta_m$ be a sequence or arbitrary chosen permutations from S_n. The table *FSGS* is transformed using permutations β_i in the following way: each permutation α from the i-th row for $i<m$ in *FSGS* is replaced by the permutation $\beta_i \alpha \beta_{i+1}^{-1}$ and for the last row – each permutation α is replaced by $\beta_m \alpha \beta_1$. Now denote this transformed table as $\overline{FSGS}$ and take that as a public key (as is the case above m stands for the number of rows and k_i for the number of permutations in the i-th row). The private key is the collection of SGS_i tables and the permutation β_1. The encryption is done in a manner similar to the symmetric scheme. The set of blocks of the plaintext is one-to-one mapped into the set of vectors $(s_1, \dots, s_m)$, where $s_i \in \{1, \dots, k_i\}$. To encrypt a block of the plaintext one calculates the product $\gamma_{s_1} \gamma_{s_2} \dots \gamma_{s_m}$, where γ_{s_i} is the permutation contained in the s_i-th cell of the i-th row of the $\overline{FSGS}$ table. So, each block of the plaintext is enciphered by a permutation from S_n and the resulting ciphertext is a sequence of permutations. To decrypt the ciphertext one has to apply the following procedure to each of the permutations of the ciphertext. Every permutation δ to be decrypted can be uniquely expressed as $\gamma_{s_1} \gamma_{s_2} \dots \gamma_{s_m}$ (as per encryption procedure). Each γ_{s_i} for $i<m$ is expressed as $\beta_i \alpha_{s_i} \beta_{i+1}^{-1}$, where α_{s_i} is the permutation contained in the s_i-th cell of the i-th row of the $\overline{FSGS}$ table, and $\gamma_{s_m} = \beta_m \alpha_m \beta_1$. Obviously, $\delta = \gamma_{s_1} \gamma_{s_2} \dots \gamma_{s_m} = (\beta_1 \alpha_1 \beta_2^{-1})(\beta_2 \alpha_2 \beta_3^{-1}) \dots (\beta_{m-1} \alpha_{m-1} \beta_m^{-1})(\beta_m \alpha_m \beta_1) = \beta_1 \alpha_1 \alpha_2 \dots \alpha_m \beta_1$. One has to calculate $\delta := \beta_1^{-1} \delta \beta_1^{-1}$, which is equal to $\alpha_1 \alpha_2 \dots \alpha_m$ and then perform the *extended "cascade"* procedure to δ using *FSGS* table and SGS_i tables. As a result, one gets an expression $\alpha_1 \alpha_2 \dots \alpha_m$ of δ through the permutations from the *FSGS* together with the corresponding vector $(s_1, \dots, s_m)$ that was used to encrypt the block from the plaintext. Then the block is recovered from $(s_1, \dots, s_m)$ vector.

It is worthy of note that the above scheme remains valid also, if a reduced *FSGS* table is used – the one obtained from the *FSGS* with some of the permutations removed from its cells.

To perform the extended "cascade" procedure on $\overline{FSGS}$ one needs *SGS* for the groups G_i and permutations β_i in order to determine the permutations in the selected rows and to retrieve any valuable information on the groups G_i from $\overline{FSGS}$. The reduced *FSGS* can be chosen to have only 3 permutations per row with large number of rows.

Obviously the number of permutations that can be generated by the *FSGS* (or the reduced one) is equal to $\prod_{i=1}^{m} k_i$. Therefore, the sizes of the *FSGS* table (the numbers $m, k_1, \ldots, k_m$) should be chosen to make $\prod_{i=1}^{m} k_i$ very large. This is achievable, as $\prod_{i=1}^{m} k_i$ has an exponential growth.

A person (possible attacker) that does not possess the private key is faced with a task of solving the following problem. Given a permutation $\alpha \in S_n$ and a sequence of sets of permutations $T_1, T_2, \ldots, T_m$ from S_n, such that $|T_i| = k \geq 3$ for all $i \in \{1, \ldots, m\}$, one must determine whether α can be expressed as a product of permutations $\beta_1 \beta_2 \ldots \beta_m$ with $\beta_i \in T_i, i = 1, \ldots, m$, and in the case of positive answer find those permutations. As the number of products $\beta_1 \beta_2 \ldots \beta_m$ to be tested is equal to k^m and the permutations in the sets T_i are arbitrary, it seems that this is a very hard computational problem, which is obviously in *NP* and is a good candidate for being *NP*-complete.

It is worth wile to mention that the resulting ciphertext has very good statistics, as the blocks of the plaintext are converted into permutations. Some ciphertexts generated by the symmetric-key scheme had been tested using standard data compression algorithms like PKZIP, LZW or RAR. None of attempts to compress the ciphertexts were successful.

The implementation of the above scheme requires some accuracy and attention. Parameters $n, m, k_1, \ldots, k_m$ as well as SGS_i and *FSGS* tables should be chosen properly avoiding some obviously "bad" choices and guaranteeing that $\overline{FSGS}$ (or reduced one) is completely random. However, these questions are out of the scope of the present article, which has to expose a new field of application of such a classical method of computational group theory as the Sims' algorithm.

Received 18.01.2012

REFERENCES

1. **Sims C.C.** Computational Methods in the Study of Permutation Groups. In Computational Problems in Abstract Algebra. Oxford: Pergamon Press, 1970, p. 169–183.
2. **Sims C.C.** Computation with Permutation Groups. In Proc. Second Symposium on Symbolic and Algebraic Manipulation. New York: ACM Press, 1971, p. 23–28.
3. **Alexanyan A., Aslanyan H. and Soghoyan A.** // Mathematical Problems of Computer Science, Transactions of IPIA NAS RA, 2010, v. 33, p. 127–134.
4. **Menezes A.J., van Oorschot P.C. and Vanstone S.A.** Handbook of Applied Cryptography, CRC Press. October 1996, 816 p.