INFORMATICA, 2004, Vol. 15, No. 2, 251–270

New Digital Signature Scheme in Gaussian Monoid

Eligijus SAKALAUSKAS
Kaunas University of Technology, Department of Applied Mathematics
Studentų 50, 51368 Kaunas, Lithuania
e-mail: esakal@asi.lt

Received: September 2003

Abstract. A new digital signature scheme in non-commutative Gaussian monoid is presented. Two algebraic structures are employed: Gaussian monoid and a certain module being compatible with a monoid. For both monoid and module, presentation and action level attributes are defined. Monoid action level is defined as monoid element (word) action on module element as an operator. A module is a set of functions (elements) with special properties and could be treated as some generalization of vector space.

Signature scheme is based on the one-way functions (OWF) design using: three recognized hard problems in monoid presentation level, one postulated hard problem in monoid action level and one provable hard problem in module action level.

For signature creation and verification the word equivalence problem is solved in monoid action level thus avoiding solving it in monoid presentation level. Then the three recognized hard problems in monoid presentation level can be essentially as hard as possible to increase signature security. Thus they do not influence on the word problem complexity and, consequently, on the complexity of signature realization.

The investigation of signature scheme security against four kind of attacks is presented. It is shown that the signature has a provable security property with respect to the list of attacks presented here, which are postulated to be complete.

Key words: digital signature scheme, one-way function, Gaussian monoid, monoid action problem, conjugator search problem, square root problem, factors' search problem.

1. Introduction

We present a new digital signature scheme taking in mind two challenges on cryptographic information protection:

1. Cryptosystem implementation in limited environments like PDA's, mobile phones and smart cards. RSA or ElGamal type algorithms based on integer factorization and discrete logarithms are not well suited for that because they require large integer modular arithmetic and therefore costly specialized co-processors.
2. The most worrisome threat to integer factorization and discrete logarithm cryptosystems (including elliptic curve discrete logarithms) that cryptographic community can foresee right now comes from quantum computers. Shor (1997) showed that if such machines could be built, integer factorization and discrete logarithms

could be computed in polynomial time. The vulnerable ones are RSA and ElGamal cryptosystems.

A general concept for cryptographic primitive's design including digital signature is as follows: to define a suitable working area (an algebraic structure), to find hard problems in this structure, to design a one-way function(s) (OWF) referencing to these hard problems, and finally to build a requested cryptographic primitive.

Algebraic terms used in this study could be found in (Van der Waerden, 1967). We will consider an infinite, non-commutative, multiplicative algebraic structure named Gaussian monoid (Dehornoy and Paris, 1999). In addition we introduce a certain module that is compatible with a monoid in some sense defined below. Recall that monoid is a semigroup with unity element and module is an additive abelian group. The element of a monoid we will call a word. The first fundamental concept in infinite monoids is a word equivalence problem (word problem) determining the equivalence of words. Its solution must be feasible for cryptographic primitives' construction.

We consider the monoid and module having two levels of attributes: the presentation level and the action level. The monoid presentation level is defined by a finite set of generators, which we will call atoms, and a finite set of relations. The monoid action level can be defined as monoid action on our introduced module. This module is a relations-free abelian group (Magnus *et al.*, 1966). The module action level is defined as well.

The signature scheme is constructed using both the monoid and module presentation and action levels. The monoid action on module is defined as a particularly chosen operation that is distributive with respect to the module addition operation. The main tools for a signature scheme are the following five hard problems and corresponding OWFs defined: three recognized OWFs in the monoid presentation level, one OWF postulated in the monoid action level, and one provable OWF defined in module action level.

In our analysis we will consider some problems related both with group and monoid properties. We will do expansion of group properties to the monoid taking in mind that monoid has some sub-monoid as a group. Whether monoid or group problems are considered, will be clear from the context.

The methodological background for OWF design using non-commutative groups and semigroups is presented in (Sidelnikov *et al.*, 1993) and is employed for key agreement protocol realization. Two hard problems are formulated there which are the main tools for an OWF construction in abstract non-commutative groups and semigroups.

Rabi and Sherman (1993) introduced a concept of strong associative OWF on abelian semigroup and proposed a key agreement protocol and signature scheme using it. The authors left open the problem of finding a strong associative OWF. Among the other open problems presented there is a question: what other combinations of algebraic and security properties of cryptographic functions yield useful cryptographic objects.

This paper deals with these other combinations.

Some generalization of Sidelnikov *et al.* (1993) methodology is presented in (Anshel *et al.*, 1999).

Monico (2002) has presented an example of cryptosystem based on finite semigroup action problem (SAP). It is a multidimensional generalization of modular exponention

using finite semigroup of matrices or ring of matrix polynomials over finite vector field. As a consequence the proposed SAP is a multi-dimensional generalization of traditional (one-dimensional) discrete logarithm problem (DLP) and is more harder. This cryptosystem is used for session key agreement protocol and ElGamal-type encryption. According to the author, this cryptosystem requires further investigations and first of all secure key length needs to be determined.

In 2002 appeared a new signature scheme using conjugacy problem in infinite non-commutative groups (Ko *et al.*, 2002). This invention is based on a gap between the conjugacy decision problem (CDP) and conjugator search problem (CSP). It means that CSP is hard and CDP is feasible. The conjugation operation serves for signing and CDP provides a verification procedure. It is a pure signature scheme based on group theory mechanism (on group presentation level). The subtlety of scheme is to choose group parameters so that CSP is hard but CDP remains feasible.

Some realization of key agreement protocol using Sidelnikov *et al.* (1993) methodology with application to a semigroup action level could be found in (Sakalauskas and Burba, 2003). The concrete construction of required commutative sub-semigroups is presented there.

So far the main requirement for cryptographic primitive construction in the presentation level of infinite non-commutative groups is that the CSP must be hard but word problem must be feasible.

Our approach differs from the above in at least two items presented below:

1. By applying infinite non-commutative monoid belonging to the Gaussian monoid family.
2. Using monoid action level with defined action operation on some module satisfying distributivity condition.
3. Using and combining three OWF in monoid presentation level, one OWF in monoid action level and one OWF in module action level.

Gaussian monoid has a sufficiently complex structure and several essentially complicated problems for OWF design in presentation level. It includes Garside monoids as a sub-family and the latters in turn include Braid monoid family. These inclusions could be expressed by notation Braid $\subset$ Garside $\subset$ Gaussian. Until the paper (Dehornoy and Paris, 1999) appeared, there were only a few examples of Gaussian groups and Garside groups in the literature. But applying the tools presented by authors, one may construct more examples of Gaussian groups and infinite families of Garside groups.

The question of particular importance for any cryptographic primitive construction in infinite non-commutative groups is the feasibility of the word problem solution. One known solution for Braid groups is an application of left-weighted canonical (normal) form mechanism which is performed in $\mathrm{O}(p^2 n \log(n))$ time (Ko *et al.*, 2000), where p is canonical length and n is Braid group index. The other one is a Dehornoy normal form mechanism using word reversing process (Dehornoy and Paris, 1999) for Gaussian groups. For Garside groups, these normal forms have uniqueness property and $\mathrm{O}(l^2)$ transformation time, where l is a word length. For further we assume that word length is

the number of atoms in the word. In a general Gaussian group case there are situations when Dehornoy normal forms have exponential time or even exponentially growing time caused by generation process.

We replace the classical word problem in monoid presentation level by the word problem in monoid action level. Then it is not necessary to use normal forms created for this purpose and being time consuming procedures. As a consequence the CSP and other problems in group presentation level may be as hard as possible and they can (and usually do) influence the complexity of a word problem in this level since we have no doubt about this complexity. Using this approach the word problem solution in group can be extended to the monoid.

But nevertheless some kind of word transformation instead of normal forms is required because while executing cryptographic protocols, Alice and Bob are exchanging words through insecure communication channels. The initial words have factors required to be hidden, because these factors, as usual, bear information about the secret (private) keys. Since uniqueness of these transformations is not required (the word problem is solved in action level), we may choose the other non-unique normal form servicing for only one purpose – to hide the information about the factors making a word. Then the words must be rewritten in some special form using a set of relations defined in the group.

We propose to use a random rewriting protocol, mixing atoms in factors being hidden. This procedure we will call random mixing. After such random mixing only exponential algorithm of total scan area could be applied to find initial factors.

The created signature scheme belongs to the class of randomized schemes (Goldwasser *et al.*, 1988) and is invulnerable against an adaptively chosen message attack. The fundamental idea in the construction of such signatures is a signing procedure performed in two authentication steps: the first step authenticates a random value which is used in the second step to authenticate the message.

Our signature scheme has provable security with respect to the four attacks considered. It means that it is based on the following hard problems in Gaussian monoid and group:

1. Three recognized hard problems in Gaussian monoid and group presentation level.
2. One postulated hard problem in monoid action level.
3. One provable hard problem in module action level.

Each of these hard problems is linked with some OWFs. The four specially selected active attacks are considered and it is proved that these attacks fail due to the introduced OWFs. We think that these selected attacks are complete and are representing the main pool of other available attacks. On this assumption a provable security concept is based.

According to our knowledge this is a second proposal to use infinite non-commutative groups or monoids for signature scheme creation after (Ko *et al.*, 2002) result.

In Section 2 we present some basic concepts as preliminaries for signature scheme construction. We introduce a Gaussian monoid and group, a certain module compatible with a monoid and two action levels in monoid and module, respectively. Five hard prob-

lems are defined for construction of corresponding OWFs. Among them one OWF is postulated in monoid action level.

We present a signature scheme in Section 3.

Security analysis for four kinds of attacks is described in Section 4.

Section 5 is dedicated to some discussions on the presented signature scheme and to some comparative-qualitative analysis on implementation issue.

2. Preliminaries

The definitions and notations used in this study could be found in (Van der Waerden, 1967).

Let us consider some multiplicative Gaussian monoid, defined by the pair $(S, \cdot)$ (Dehornoy and Paris, 1999). The multiplicative monoid operation we denote as $\cdot$ and neutral element as 1. For convenience S is presented by a finite set of generators and relations (Magnus *et al.*, 1966). The generators we will call atoms. A monoid consists of elements named words and words consist of atoms. We use the term "atom", since this term better coincides with the procedure that we introduced for atoms' random mixing process in the word. So S is an infinite monoid finitely presented by the set of atoms and relations. The finite product of elements (words) of S will be another word in S. As usual the multiplication in Gaussian groups means concatenation of atoms as characters in a word. Then in our case $\cdot$ means concatenation (multiplication) operation. Assume that S has a structure where two mutually commutative subsets S_L, S_R could be defined in S. Then for any words $\alpha \in S_L$ and $\beta \in S_R$ the commutation property takes place

$$\alpha \cdot \beta = \beta \cdot \alpha.$$

Assume S has some subset $J \subset S$ having inverse elements where for all $\eta \in J$ there exists an unique $\eta^{-1} \in S$ such that

$$\eta \cdot \eta^{-1} = \eta^{-1} \cdot \eta = 1.$$

We do not claim J to be a group. The cases when J is a group will be considered below.

The subset of S having no finitely presented inverse elements we denote as $S \backslash J$.

Consider some module $(F, +)$ consisting of set of functions $\{f\}$ which is some generalization of vector space. Recall that F is an additive abelian group of elements (functions in our case). Assume that all functions f in F perform a mapping $f\colon \Omega \to \Omega$, where Ω is the domain of F and is a set of fixed length binary strings.

We consider S as a monoid of operators or multipliers acting on module M. Then for convenience M is called a module over the monoid S, or simply S-module in other legal notation. The action level of monoid S may be determined as an action of any its element σ on any function x in F as an operator, i.e., $\sigma\colon F \to F$. For this action restricted in $S \backslash J$

we introduce a new associative binary bijective operation (function) $\circ$: $S\backslash J \times M \to M$. This means that for all $\sigma \in S\backslash J$ and $f \in F$ there exists some $g \in F$, such that

$$g = \sigma \circ f. \tag{OWF}$$

Assume that two mutually commutative subsets S_L, S_R defined above are the subsets of $S\backslash J$. Let us introduce also a sufficiently large two mutually commutative subsets S_{L0} and S_{L1} in S_L for our construction.

So we defined three associative operations $\cdot, \circ$ and $+$. Assume for them the following order of execution takes place for any σ, $\rho \in S$ and $f, g \in F$, as illustrated by equation

$$\sigma \cdot \rho \circ f + g = \big((\sigma \cdot \rho) \circ f\big) + g = \big(\sigma \circ (\rho \circ f)\big) + g.$$

According to this and associativity of operations applied, the following expressions are equivalent

$$\sigma \cdot \rho \circ f = (\sigma \cdot \rho) \circ f = \sigma \circ (\rho \circ f) = \sigma \circ \rho \circ f.$$

Assume that the following distributive relation take place for all $\sigma \in S$ and $f, g \in F$

$$\sigma \circ (f + g) = \sigma \circ f + \sigma \circ g.$$

This means that $\circ$ is left distributive.

And finally let

$$1 \circ f = f.$$

We postulate that $\circ$ is an OWF associated with a monoid action on the module. The exact definition we will present below.

Determine now some properties required for functions f in F to define a module action level and corresponding OWF in a module action level.

1. Functions $f \in F$ are surjective. This means that that $f^{-1}(\omega)$ is not unique.
2. The Liapunov criteria λ of f is positive, i.e., $\lambda_f(\omega) > 0$, for all $\omega \in \Omega$. Then function f acts as an expansion mapping in Ω.

A similar concept of module action level is used in this study, taking in mind that f is acting on Ω. Then for any $\omega \in \Omega$ we have a simple action defined by the formula

$$\omega' = f(\omega), \quad \omega' \in \Omega.$$

Define recurrent calculations with a function f. Choose some integer $n > 1$ and binary string $\omega_0 \in \Omega$. Then by recursion we can compute the values $\omega_1, \omega_2, \ldots, \omega_n$,

$$\begin{aligned}
&\omega_1 = f(\omega_0),\\
&\omega_2 = f(\omega_1),\\
&\quad \ldots\\
&\omega_n = f(\omega_{n-1}).
\end{aligned}$$

The value ω_n could be expressed by the formula

$$f^n(\omega_0) = \omega_n.$$

It is clear that according to Property 1 and for considerable big n, there is impossible to determine ω_0 from the equation

$$\omega_0 = f^{-n}(\omega_n), \tag{RP1}$$

except using random guess in the total domain of preimages of f^n. This domain grows exponentially with increasing n.

From Property 2 it follows that recursion process does not converge to some constant point for all $f \in F$ and for any two different initial conditions $\omega_{01} \neq \omega_{02}$. As a consequence, for some natural n the probability of occurrence of the event

$$f^n(\omega_0) = \omega_0, \tag{RP2}$$

is negligible for any $\omega_0 \in \Omega$. This corresponds to the requirement that the probability for some element $\omega_0 \in \Omega$ to be involved in the closed orbit for a particular n is negligible.

Let us look at some problems related with infinite groups presented by finite set of atoms and relations assuming temporarily that J is group.

Dehn in 1911 has formulated three fundamental problems with increasing complexity related to these kinds of groups. Two of them are the following:

I: Word problem.

II: Conjugacy problem.

Word problem. It means word's equivalence problem.

Every word w in J has its equivalency class $[w]$ determined by the relations in J. In other words each word w_1 in $[w]$ could be obtained from any other word w_2 in $[w]$ using equivalent transformations defined by the set of relations in J.

It is said that two words $w_1, w_2 \in J$ are equivalent, i.e., $w_1 = w_2$, if there exists a finite algorithm using the set of relations which transform w_1 to w_2 or vice versa. Novikov in 1957 proved that the word's equivalence problem is very hard and can't be solved in a general case.

Conjugacy problem. Two words $w_1, w_2 \in J$ are said to be conjugate, i.e., $w_1 \sim w_2$, if there is an element $\eta \in J$ such that

$$w_1 = \eta \cdot w_2 \cdot \eta^{-1},$$

where the element η is called a conjugator. It is conveniently accepted that the conjugacy problem is harder than the word problem.

The solutions of problems mentioned above are known only for a few Garside groups. In most cases solutions are unknown or they do not exist at all. When the solution is

known in form of an algorithm involving finite steps, the problem may be treated as "easy" or "hard" to solve.

"Easy" and "hard" problems formalize the complexity theory (Menezes *et al.*, 1996). The "easy" solution means that the problem is solved by a polynomial-time algorithm whose on-average running time function is of the form $\mathrm{O}(n^k)$, where n is the input size, k is a constant and $\mathrm{O}(n^k)$ means asymptotic upper bound of running time less or equal to cn^k, for all $n \geqslant n_0$, where constants $c > 0$, $n_0 > 0$. The complexity class P is a set of all decision problems that are solvable in polynomial time.

When k is considerably big, it is said that we have super polynomial-time complexity. Any algorithm whose running time cannot be so bounded is called exponential-time algorithm. Both cases, super polynomial-time and exponential-time complexity, may be treated as "hard" problems.

The complexity class NP is a set of all problems for which a YES answer can be verified in polynomial time given some extra information called a certificate.

Main cryptographic primitives are using One Way Function (OWF) methodology (Rabi and Sherman, 1993; Menezes *et al.*, 1996). So far OWF design is based on the hypothesis that $P \neq NP$ and this means that OWF calculation corresponds to the complexity class P, but the inverse OWF algorithm is in the complexity class NP.

The useful scheme for OWF construction in some group is a feasible solution of word problem by the algorithm in complexity class P and infeasible solution of conjugacy problem by the algorithm in class NP. According to this, the conjugacy problem algorithm must have exponential or at least super polynomial-time complexity when the conjugator η is unknown.

The conjugacy problem itself is divided into two essential problems (Ko *et al.*, 2002): Conjugacy Decision Problem (CDP) and Conjugator Search Problem (CSP). The CDP means to determine whether $w_1 \sim w_2$ for given instances $w_1, w_2 \in J$. CSP is reckoned to be harder than the CDP. We do not consider now the complexity of word equivalence problem and CDP. We will return to the word problem later.

We would like to present now some reference problems assumed to be hard in non-commutative Gaussian group.

P1. *The Conjugator Search Problem* (CSP): to find $\eta \in J$, satisfying $w_1 = \eta \cdot w_2 \cdot \eta^{-1}$, for given $w_1, w_2 \in J$.

P2. *Square Root Problem* (SRP): to find $\alpha \in G$, for given α^2.

This problem is exponential in essence and is solved for Garside groups being members of Gaussian group family (Sibert, 2002). No other algorithm is known so far.

P3. *Factors Search Problem* (FSP). For a given word σ find factors η and μ in some equivalent unknown word $\sigma' = \eta \cdot \mu$, when $\sigma' = \sigma$.

For an arbitrary finite bounded length word σ in J there is an exponential amount of possible factors defining its equivalence class $[\sigma]$. Consider the initial word σ' in $[\sigma]$ consisting of concatenation of η and μ as is commonly done by performing a formal multiplication in non-commutative groups or monoids. The problem P3 becomes hard if we perform some equivalent transformation φ: $\sigma' \to \sigma$, using defined relations, when the atoms composing η and μ are mixing with each other. As it was mentioned above this

procedure applies when two words are verified for equivalence. Then φ is normal form transformation having a uniqueness property.

We propose to use random equivalence transformations φ providing random mixing of atom mechanism by randomly using relations defined in J. In the most known groups and monoids random mixing could be done in polynomial time. Moreover the sufficient random mixing could be achieved in time $\mathrm{O}(l)$, where l is the word length or the number of atoms. The unmixing procedure which determines η and μ will require an exponential time with respect to word length l.

The next problem we formulate for a monoid S and call it monoid action problem.

P4. *Monoid action problem* (MAP): for some given $\sigma \in S$, and known $a \in F$ find $x \in F$ from the equation $a = \sigma \circ x$.

We postulate that the MAP is hard and is associated with a correcponding OWF for our signature construction. The one-wayness of equation (OWF) from above is based on the postulate that σ^{-1} has no finite presentation. So the algorithm to find x using the relations

$$\sigma^{-1} \circ a = \sigma^{-1} \circ \sigma \circ x = 1 \circ x = x,$$

requires infinite amount of steps.

As a simple example, MAP could be transformed to the discrete logarithm problem (DLP) in the case if $S = F$ and both are finite cyclic groups of prime order p. Then we can construct the well known modular exponentiation function in the form

$$a = \sigma \circ x = \sigma^x \bmod p.$$

The generalization of this example for a multidimensional case could be found in (Monico, 2002) where the semigroup (ring) action problem is introduced for finite semigroup of matrices or for ring of matrix polynomials, both over finite vector field.

The last problem we formulate in module F action level and call it an inverse recurrency problem.

P5. *Inverse recurrency problem* (IRP): having f and some known nth iteration value $\omega_n \in \Omega$, find an initial value ω_0 from the equation $\omega_0 = f^{-n}(\omega_n)$.

This problem corresponds to the (RP1) and is based on the Properties 1 and 2 from above.

The problems P1–P3 are widely recognized as hard, especially in Gaussian groups.

The complexity of P4 in this study is postulated. We think it is a sensible postulate. The motivation could be based on some results presented in (Monico, 2002). Even in the case of finite semirings over finite fields and acting on certain set, the complexity of semigroup action problem noticeably exceeds the complexity of ordinary one-dimensional DLP. Recall that we are considering an infinite Gaussian monoid and so we are expecting much greater complexity.

The complexity of P5 is provable taking into account the Properties 1, 2 and (RP1), (RP2).

The purpose of this paper is to use hard problems P1–P5 as related with corresponding OWFs and to construct a signature scheme with provable security. The provable security has the following sense: several specially selected active attacks are considered and it is proved that these attacks fail because problems P1–P5 are infeasible. It is also based on the postulate that these selected attacks cover all the other possible attacks, i.e., these attacks are complete.

Recall our postulate that the MAP is hard and is an OWF. We use MAP and OWF as synonyms further, taking into account that there are the other four OWFs named by CSP, SRP, FSP and IRP.

We define now the monoid word problem solution in action level of the monoid S. Assume the probability of event $w_1 \circ f = w_2 \circ f$ is negligible if $w_1 \neq w_2$ for $w_1, w_2 \in S$. Then we can make a decision that if

$$w_1 \circ f = w_2 \circ f,$$

then w_1 and w_2 are equivalent with overwhelming probability, i.e., $w_1 = w_2$.

For effective verification of this condition we use a module action level defined above by providing recurrent calculations with any function $f \in F$.

PROPOSITION 1. The words $w_1, w_2 \in S$ are equivalent with a very high probability if for some integer $n \gg 1$, $f \in F$ and any $\omega \in \Omega$, the following relation is valid:

$$(w_1 \circ f)^n(\omega) = (w_2 \circ f)^n(\omega).$$

Proof. Assume $w_1 \neq w_2$ and $(w_1 \circ f)^n(\omega) = \omega_{n1}$. Let us formally apply the inverse recurrent function $(w_2 \circ f)^{-n}$ to both sides of the last equation, despite the infeasibility of inverse recursion operation, declared in (RP1). According to proposition $(w_2 \circ f)^{-n}(w_1 \circ f)^n(\omega) = \omega$. Let $(w_2 \circ f)^{-n}(w_1 \circ f)^n = g$, $g \in F$. Then functions' composition for a value ω can be expressed as,

$$g(\omega) = \omega.$$

But according to (RP2) the probability to satisfy the last equation is negligible. Then the latter equation is valid when g is identity function and then $(w_1 \circ f)^n = (w_2 \circ f)^n$. Taking in mind the bijectivity of $\circ$ we obtain that $w_1 = w_2$. This proves the proposition.

As it seems from the Proposition 1, the presented words' equivalence criterion does not depend on the word problem complexity in S presentation level.

Finally, define some notations for signature creation and verification.

The message space consisting of finite length binary strings we denote by T. Let a signer Alice intend to sign some message $T_A \in T$ and to send it to verifier Bob. As usual, Alice signs not a message T_A but some h-value m of original message. Assume

that there are three cryptographically secure h-functions (Menezes *et al.*, 1996) H, h and h', performing mappings

$$H\colon T \to F;$$
$$h\colon S \to F;$$
$$h'\colon F \to S_R.$$

Functions H and h are surjective and h' is injective. The data to be signed is expressed as $m = H(T_A)$.

Let the domain $\Omega \subset T$. Assume also that any function $m \in F$ could be represented in binary form as an element of Ω. Then if $f, m \in F$, we may represent m in binary form and having $m \in \Omega$, calculate the value $f(m) \in \Omega$. This convention helps to shorten the notations.

Alice creates a signature S on value m and sends it to verifier Bob. Bob has a publicly available verification function Φ to verify the signature S on m.

Alice and Bob communicate through insecure and open communication channels and all the data published and transmitted are available to the active adversary Eve. All parties share information about the structure of monoid S, module F, hash functions H and h, verification function Φ and public key of Alice. Eve can obtain, remove, forge and retransmit any message Alice sends to Bob.

3. Signature Creation and Verification

3.1. *Key Generation*

Alice chooses at random secret elements $\alpha \in S_L$, $\eta \in J$, $x \in M$ and non-secret element $a \in S_{L0}$. She calculates the elements $\alpha' \in S_L$, $\rho \in S \backslash J$ and $q \in M$:

$$\alpha' = a \cdot \alpha \cdot a^{-1};$$
$$\rho = \eta \cdot \alpha^2 \cdot \eta^{-1};$$
$$q = \eta \cdot \alpha^3 \circ x.$$

Then the Private Key (PrK) and Public Key (PuK) are as follows:

$$\text{PrK} = (\alpha, \alpha', \eta, x); \quad \text{PuK} = (a, \rho, q).$$

3.2. *Signature Creation*

Alice takes a message $T_A \in T$ to be signed, chooses at random $\xi \in S_{L1}$ and calculates the following elements:

$$m_A = H(T_A);$$

$$\mu = h'(\xi \circ m_A);$$
$$m = m_A + h(\xi \cdot \mu \cdot \alpha' \cdot \xi^{-1});$$
$$\zeta = \xi \cdot \mu \cdot \alpha \cdot \xi^{-1}.$$

The secret signature key is ξ.
The following signature parameters σ, s are calculated in addition

$$\sigma = \eta \cdot \mu \cdot \eta^{-1};$$
$$s = \sigma \circ (\eta \cdot \alpha \circ x + m_A).$$

Then she creates a signature S of the form

$$S = (m, \zeta, \sigma, s).$$

Alice sends S to Bob.

3.3. *Signature Verification*

Bob uses given m and ζ from signature S to find m_B by the formulas

$$\begin{aligned} m_B &= m - h(a \cdot \zeta \cdot a^{-1}) \\ &= m_A + h(\xi \cdot \mu \cdot \alpha' \cdot \xi^{-1}) - h(a \cdot \xi \cdot \mu \cdot \alpha \cdot \xi^{-1} \cdot a^{-1}) \\ &= m_A + h(\xi \cdot \mu \cdot \alpha' \cdot \xi^{-1}) - h(\xi \cdot a \cdot \mu \cdot \alpha \cdot a^{-1} \cdot \xi^{-1}) \\ &= m_A + h(\xi \cdot \mu \cdot \alpha' \cdot \xi^{-1}) - h(\xi \cdot \mu \cdot \alpha' \cdot \xi^{-1}) = m_A. \end{aligned}$$

Having signature's S components σ and s, the verification function $\Phi = \Phi(m, \sigma, s)$ is TRUE if

$$\rho \circ s = \sigma \circ q + \rho \circ \sigma \circ m_A. \tag{V}$$

The proof of verification condition (V) follows from the expressions

$$\begin{aligned} \rho \circ s &= \rho \circ (\sigma \circ (\eta \cdot \alpha \circ x + m_A)) \\ &= \rho \cdot \sigma \cdot \eta \cdot \alpha \circ x + \rho \circ \sigma \circ m_A \\ &= \eta \cdot \alpha^2 \cdot \eta^{-1} \cdot \eta \cdot \mu \cdot \eta^{-1} \cdot \eta \cdot \alpha \circ x + \rho \circ \sigma \circ m_A \\ &= \eta \cdot \alpha^2 \cdot \mu \cdot \alpha \circ x + \rho \circ \sigma \circ m_A \\ &= \eta \cdot \mu \cdot \alpha^3 \circ x + \rho \circ \sigma \circ m_A \\ &= \eta \cdot \mu \cdot \eta^{-1} \cdot \eta \cdot \alpha^3 \circ x + \rho \circ \sigma \circ m_A \\ &= \sigma \circ q + \rho \circ \sigma \circ m_A, \end{aligned}$$

taking in mind that $\alpha \cdot \mu = \mu \cdot \alpha$ and $\eta \cdot \eta^{-1} = 1$.

The verification condition (V) is sufficient and is realized in monoid S action level. The implementation of verification condition (V) we provide in module action level using Proposition 1 and recurrent calculations. Bob takes the binary representation of m_B, an integer $\nu \gg 1$ and verifies the equation

$$[\rho \circ s]^{\nu}(m_B) = [\sigma \circ q + \sigma \circ \rho \circ m_B]^{\nu}(m_B). \tag{VV}$$

If (VV) is valid, then Bob accepts a signature S on message m_B.

4. Security Analysis

Assume that the active eavesdropper Eve can obtain, remove, forge and retransmit any message Alice sends to Bob. Any forgered data d we denote as d^F.

We consider the four main and specially selected attacks and prove that these attacks fail, by referencing to the above introduced OWFs based on P1–P5: CSP, SRP, FSP, MAP and IRP. It is postulated that these selected attacks cover all other possible active attacks, i.e., these attacks are complete. The list of considered attacks is the following: PrK compromitation, Data+Signature forgering, Data implied forgering and Data implied forgering in Module action level.

4.1. PrK *Compromitation*

Instance: PuK$= (a, \rho, q)$.

Objective: find PrK$= (\alpha, \alpha', \eta, x)$.

Eve having PuK must sequentially solve three hard problems to find PrK. The question is to find a starting point. We could advise to begin from the expression determining the publicly known parameter ρ:

$$\rho = \eta \cdot \alpha^2 \cdot \eta^{-1}.$$

The problem is to find ρ factors η and α^2 by solving corresponding FSP. But this problem is postulated as hard. Even if factors η and α^2 are found, Eve must solve the next hard problem: to extract a square root and to find

$$\alpha = (\alpha^2)^{1/2}$$

by solving the SRP. But even this is not sufficient to achieve PrK compromitation. At a third step she must find unknown x from the equation

$$q = \eta \cdot \alpha^3 \circ x,$$

which corresponds to MAP.

So the three sequential hard problems FSP, SRP and MAP must be solved by the eavesdropper for PrK compromitation.

4.2. *Data+Signature Forgering*

According to Rivest, this kind of attack is called existential forgering (Goldwasser, 1988).

Assume Eve is trying to sign a forgered message T_A^F. Then being unable to find an actual PrK, she must forge it by replacing original PrK elements with forgered ones (α^F, η^F, x^F) and performing the following calculations after choosing some $\xi^F \in S_{L1}$:

$$
\begin{aligned}
&m_A^F = H(T_A^F),\\
&\mu^F = h'(\xi^F \circ m_A^F),\\
&m^F = m^F + h\big(\xi^F \cdot \mu^F \cdot \alpha'^F \cdot (\xi^F)^{-1}\big),\\
&\sigma^F = \eta^F \cdot \mu^F \cdot (\eta^F)^{-1}.
\end{aligned}
$$

In this attack the elements μ^F and σ^F are initial data for determination of s^F from the equation (V). If it is feasible to determine s^F in this way then the verification procedure will be successful and Eve can sign the forgered message T_A^F. So the forgered data must satisfy the following equation

$$\rho \circ s^F = \sigma^F \circ q + \rho \circ \sigma^F \circ \mu^F.$$

Formally Eve can write the last equation in the form

$$s^F = \rho^{-1} \circ (\sigma^F \circ q + \rho \circ \sigma^F \circ \mu^F),$$

taking in mind that $\rho^{-1} \circ \rho = 1$ and $1 \circ s^F = s^F$.

This is equivalent to find the inverse ρ^{-1} to the ρ. Then by definition

$$\rho^{-1} = \eta \cdot \alpha^{-1} \cdot \eta^{-1}.$$

But according to our assumption that $\alpha \in S_L \subset S \backslash J$. Then α^{-1} can not be found by the finite step algorithm so it can not found anyway. Then ρ^{-1} also can not. So Eve is not able to calculate the forgered s^F, having μ^F and σ^F.

The solution of forgered equation (V) with respect to σ^F with chosen s^F and taking in mind that $\rho \cdot \sigma^F = \sigma^F \cdot \rho$ is impossible, because the expression $(q + \rho \circ m^F)^{-1}$ has no sense.

4.3. *Data Implied Forgering*

Traditionally this kind of attack means to forge message T_A and to sign it with valid or partially forgered signature parameters. Eve creates her message T_A^F and then, using original signature parameter ζ and forgered $\xi^F \in S_{L1}$, calculates

$$
\begin{aligned}
&m_A^F = H(T_A^F),\\
&\mu^F = h''(\xi^F \circ m_A^F),\\
&m^F = m_A^F + h(a \cdot \zeta \cdot a^{-1}),
\end{aligned}
$$

where $h'' \neq h'$ and is specially constructed by Eve realizing a mapping h'': $F \to J$. By this mean Eve achieved that μ^F is invertible, i.e., $(\mu^F)^{-1}$ exists.

Then she must determine some η', having μ^F and choosing some σ', using equation

$$\sigma' = \eta' \cdot \mu^F \cdot \eta'^{-1}.$$

Assume Eve has obtained η' despite the declared infeasibility of solving CSP. But it is strongly unbelievable that she obtained $\eta' = \eta$. Moreover, for a valid verification she must find some $(\alpha \circ x)'$ from the equation

$$s = \sigma' \circ \left(\eta' \cdot (\alpha \circ x)' + m_A^F\right) = \sigma' \cdot \eta' \cdot (\alpha \circ x)' + \sigma' \circ m_A^F.$$

The term $(\alpha \cdot x)'$ could be expressed as follows

$$(\alpha \cdot x)' = \eta'^{-1} \cdot \sigma'^{-1} \circ s - \eta'^{-1} \circ m_A^F.$$

This is possible, because (as we mentioned above) if $(\mu^F)^{-1}$ exists, then σ'^{-1} also exists and satisfies equation $\sigma' = \eta' \cdot (\mu^F)^{-1} \cdot \eta'^{-1}$.

Eve forms the following signature S':

$$S' = (m_A^F, \zeta, \sigma', s),$$

with a forgered parameter m_A^F related to forgered data T_A^F.

But Bob discloses this attack due to verification condition (V) failure

$$\begin{aligned}
\rho \circ s &= \rho \circ \left(\sigma' \circ \left(\eta' \cdot (\alpha \circ x)' + m_A^F\right)\right) \\
&= \rho \cdot \sigma' \cdot \eta' \cdot (\alpha \circ x)' + \rho \circ \sigma' \circ m_A^F \\
&= \eta \cdot \alpha^2 \cdot \eta^{-1} \cdot \eta' \cdot \mu^F \cdot \eta'^{-1} \cdot \eta' \cdot (\alpha \circ x)' + \rho \circ \sigma' \circ m_A^F \\
&\neq \eta' \cdot \mu^F \cdot \eta'^{-1} \cdot \eta \cdot \alpha^3 \circ x + \rho \circ \sigma' \circ m_A^F = \sigma' \circ q + \rho \circ \sigma' \circ m_A^F.
\end{aligned}$$

Condition fails because $\eta^{-1} \cdot \eta' \neq 1$ and $\alpha^2 \cdot (\alpha \circ x)' \neq \alpha^3 \circ x$.

Comment. By achieving an invertability of μ^F, Eve changes a convenient structure of μ and σ. So instead the original σ, Eve can employ the modified σ' only. The difference between σ' and legal σ could be noticed by evidence. It could be also disclosed when Bob applies verification condition (VV) and his calculations were rejected by algorithm accepting invalid data.

4.4. *Data Implied Forgering in Module Action Level*

Assume Eve is trying to present such a forgered data that verification condition (VV) will not fail. Starting from the T_A^F she forms a signature

$$S^F = (\mu^F, \zeta^F, \sigma^F, s^F),$$

trying to choose the parameters satisfying (VV). For more clarity let us use the brackets [ ] instead of ( ) where possible. Then for a bit string m^F the validity condition holds if

$$[\rho \circ s^F]^\nu(m^F) = [\sigma^F \circ q + \sigma^F \circ \rho \circ m^F]^\nu(m^F).$$

Eve must find m^F satisfying formal equation

$$m^F = [\rho \circ s^F]^{-\nu}\big([\sigma^F \circ q + \sigma^F \circ \rho \circ m^F]^\nu(m^F)\big).$$

Refusing contradiction with (RP2), assume that m^F could be obtained. Then the verification condition (VV) succeeds, because applying the function [$\rho \circ s^F]^\nu$ to both sides of last equation we have

$$\begin{aligned}[\rho \circ s^F]^\nu(m^F) &= [\rho \circ s^F]^\nu\Big([\rho \circ s^F]^{-\nu}\big([\sigma^F \circ q + \sigma^F \circ \rho \circ m^F]^\nu(m^F)\big)\Big)\\ &= [\sigma^F \circ q + \sigma^F \circ \rho \circ m^F]^\nu(m^F).\end{aligned}$$

But according to IRP, it is infeasible to perform inverse recurrent calculations and thereby to determine m^F using (RP2) type equation.

Even if Eve could guess such a m^F she would not be able to choose some sensible message T_A^F for m^F because we assumed the H-function is cryptographically secure.

5. Discussions

5.1. *Theoretical Considerations*

We have presented a signature scheme in Gaussian monoid. This is a shortened name. In more detail the title of the scheme could be named as signature scheme in Gaussian monoid action level defined on module, compatible with respect to monoid action operation.

According to our knowledge this is a second proposal to use infinite non-commutative groups or monoids for signature scheme creation after (Ko *et al.*, 2002) result.

Except the three known problems existing in Gaussian groups named as CSP, SRP and FSP, the following main conditions (requirements) must take place for our signature scheme in addition:

1. The compatibility between monoid and module. Compatibility is determined by the distributive property of monoid action operation $\circ$ to module addition operation $+$.
2. The operation $\circ$ is an OWF and is based on the monoid action problem (MAP).
3. The monoid word problem solution in its action level.

The main advantage of our scheme is that there is no matter what complexity the word problem in monoid presentation level has. The listed above CSP, SRP, FSP could be (or could be required to be) as hard as possible to increase the security of our scheme.

Why Gaussian monoids? Formally, our scheme gives the opportunity to use Gaussian monoids having in mind the described complexity requirements and two level attributes: presentation and action levels. Gaussian monoids are considerably abstract and sufficiently complex monoids. So far only a few examples of Gaussian monoids were known. (Dehornoy and Paris, 1999) created tools for infinite families of Garside monoids as a special families of Gaussian monoids' construction. In general it is hard to expect that CSP, SRP and FSP algorithms for these monoids are feasible or will be constructed at all.

We are sure that at least for one family of Gaussian monoids known as Braid monoid, possessing a Braid group as sub-monoid, our scheme is suitable. However, we have no any knowledge about the impossibility to implement this scheme with the other Gaussian monoids so far. This sophism allows us to look ahead, and to expect implementation of this scheme on the other more complex Gaussian monoids providing more secure digital signatures.

The auxiliary results obtained in this study are:

1. We proposed a security proof based on the four specially selected attacks. These attacks fail due to the above introduced OWFs based on P1–P4: CSP, SRP, FSP, MAP and IRP. We postulate that these attacks cover all other possible attacks, i.e., that these attacks are complete.
2. We presented a probabilistically sufficient condition for word problem solution in a monoid using monoid and module action levels. Thus, we avoided a complex, in general, word problem solution in monoid presentation level.
3. We proposed a secure word factors' hiding protocol, using random mixing of atoms in the word procedure, and so providing a $O(l)$ complexity for it, where l is word length (number of atoms).

5.2. *Performance Analysis*

It is hard to estimate performance of proposed scheme because no practical implementation is realized yet. Nevertheless, some qualitative and indirect analysis could be done. Firstly it could be done by using some algebraic and cryptographic prototype and comparing it with a possible realization of our scheme. Secondly this cryptographic prototype could be compared with other traditional cryptosystems.

As an algebraic prototype we choose a Braid group, and as cryptographic prototype, a public-key Braid cryptosystem (BCS) published in (Ko *et al.*, 2000).

What similarities and differences could be noticed between this BCS and our signature scheme?

1. The number of key parameters in our scheme is greater than in the BSC, but nevertheless the key length could be comparable, taking in mind that our key pair is better protected.
   The PrK in BCS consists of 1 braid, in our system there are 3 braids plus one module element.
   The PuK in BCS consists of 2 braids, in our system there are 2 braids plus one module element.

Our key lengths could be shorter because they are protected not only by CSP as in BCS but by the SRP, FSP and MAP simultaneously.

2. Operation speed is also comparable. Assume both algorithms uses a left-weighted canonical form mechanism. The left-weighted canonical form is performed in time $\mathrm{O}(p^2 n \log n)$, where p is a number of canonical factors in the form and n is a braid index (Ko *et al.*, 2000).

BCS uses it for encryption and decryption.

Our scheme together with left-weighted canonical forms also uses monoid action, random mixing and recurrence calculation procedures. The latter procedures are executed in time $\mathrm{O}(l)$ and are independent of braid index n. There l is a braid length (number of atoms). The list of comparable operations is presented in the Table 1.

As a consequence the public key length and execution time in our scheme are comparable with those in Braid cryptosystem.

Having in mind qualitative comparison of our scheme with BCS as a prototype, we can illustrate comparative results of BCS with other traditional public key cryptosystems, obtained by (Karu and Loikkanen, 2001). The authors performed calculations on Pentium 500 MHz computer. Some selected figures are presented in the Table 2.

Having in mind that our signature scheme as in BCS does not use arithmetics with large integers it is more suitable for implementation in mobile phones and smart cards. The key length in our scheme is comparable with RSA cryptosystem, but it is processed piece by piece and hence this procedure could be realized in ordinary processors.

Table 1

The list of comparable operations

List of operations			
BCS		Our signature scheme	
Encryption	Decryption	Signing	Verification
1. Left-weighted canonical form $\times 2$ times	1. Left-weighted canonical form $\times 1$ time	1. Left-weighted canonical form $\times 2$ times	1. Left-weighted canonical form $\times 1$ time
		2. Monoid action $\times 2$ times	2. Monoid action $\times 3$ times
		3. Random mixing $\times 1$ time	3. Recurrence calculation $\times 2$ times

Table 2

Comparative results of BCS with other cryptosystems

Parameter	RSA1024	ECC168	NTRU263	BCS
Public key size (bits)	1024	169	1841	1000
Encryption speed (ms)	4.28	140	1.9	29.8
Decryption speed (ms)	48.50	67	3.5	14.9

And finally, so far we have no any knowledge about quantum information algorithms capable to break a cryptosystem based on BCS. The same, of course, is valid for our scheme as well. Therefore we think that the signature scheme presented here requires further investigations as possible alternative to the traditional RSA and ElGamal signature schemes because BCS is already reckoned as an alternative to RSA and ElGamal cryptosystems.

References

Anshel, I., M. Anshel and D. Goldfeld (1999). An algebraic method for public-key cryptography. *Mathematical Research Letters*, **6**, 1–5.

Dehornoy, P., and L. Paris (1999). Gaussian groups and Garside groups: two generalizations of Artin groups. *Proc. London Math. Soc.*, **79**(3), 569–604.

Goldwasser, S., S. Micali and R. Rivest (1988). A digital signature scheme secure against adaptive chosen message attacks. *SIAM J. Comput.*, **17**, 281–308.

Karu, P., and J. Loikkanen (2000). *Practical Comparison of Fast Public-Key Cryptosystems*. Manuscript, 2001. Available at: `http://www.tml.hut.fi/Opinnot/Tik-110.501/2000/papers.html`.

Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park (2000). New public-key cryptosystem using braid groups. Advances in cryptology. In *Proc. Crypto 2000, LNCS 1880*. Springer-Verlag. pp. 166–183.

Ki Hyoung Ko, Doo Ho Choi, Mi Sung Cho, Jang Won Lee (2002). *New Signature Scheme Using Conjugacy Problem*. Department of Mathematics, KAIST, Daejeon. `http://eprint/iacr.org`.

Magnus, V., A. Karrass and D. Solitar (1966). *Combinatorial Group Theory*. Interscience Publishers, NY.

Menezes, A., P. van Oorschot and S. Vanstone (1996). *Handbook of Applied Cryptography*. CRC Press.

Monico, C. (2002). *Semirings and Semigroup Actions in Public-Key Cryptography*. Phd. thesis, University of Notre Dame.

Rabi, M., and A. Sherman (1993). *Associative One-Way Functions: A New Paradigm for Secret Key Agreement and Digital Signatures*. University of Maryland, Computer Science Department.

Sakalauskas, E., and T. Burba (2003). Basic semigroup primitive for cryptographic Session Key exchange Protocol (SKEP). *Information Technology and Control*, **3**(28).

Sibert, H. (2002). Extraction of roots in Garside groups. *Comm. in Algebra*, **30**(6), 2915–2927.

Shor, P.W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, **26**, 1484–1509.

Sidelnikov, V., M. Cherepnev and V. Yaschenko (1993). Systems of open distribution of keys on the basis of noncommutative semigroups. *Russian Acad. Sci. Dokl. Math.*, **48**(2), 566–567.

Van der Waerden, B.L. (1967). *Algebra*. Springer-Verlag.

E. Sakalauskas received PhD degree from Kaunas Polytechnical Institute in 1983. Currently he is a head of Laboratory of Information and Energetic Systems and senior reseacher in Institute of Defence Technology in Kaunas University of Technology. The scope of scientific interests is a system theory, identification and cryptography. E. Sakalauskas has published over 26 scientific papers and 7 of them are published in journals included in ISI Master Journal List Catalog.

Nauja skaitmeninio parašo shema Gauso monoide

Eligijus SAKALAUSKAS

Pateikta nauja skaitmeninio parašo schema Gauso monoide. Naudojamos dvi tarpusavyje suderintos algebrinės sistemos: Gauso monoidas ir modulis. Monoidas sudaro operatorių aibę, veikiančią modulyje.

Parašo schema paremta trimis pripažintomis sunkiomis problemomis monoido atvaizdavimo lygmenyje; viena postuluota sunkia problema monoido veikimo lygmenyje ir viena sunkia problema modulio veikimo lygmenyje.

Pateikta schemos saugumo analizė keturiems klastotės atvejams, tuo įrodant, kad pateikta schema turi įrodomo saugumo savybę. Pateiktas palyginimas su kitomis parašo schemomis.