A Survey of Group-based Cryptography

S. D. Hasapis, D. Panagopoulos, E. Raptis

Contents

Abstract

In this report, we introduce some techniques of the cryptography since 1976. In the beginning, the basics of public key cryptography is discussed for the reason that this is the basic idea used in contemporary cryptography. After introducing the fundamental algorithms of cryptography, we mention the analogs in non-commutative algebraic cryptography. Using decision and search problems in Group theory as basis key agreement protocols introduced. Next attacks on group-based systems are discussed. While, choosing the platform group and the group problem is the key for the security of the system, a secret sharing scheme is introduced in behalf of the writer. Cayley Hash functions and attacks in such systems is introduced next, closing the survey with random strings in group theory and the generalization of some classical problems to the non-commutative case.

Mathematics Subject Classification : 20, 94A60
Keywords: group-based cryptography, secret sharing, Dehn's problems

1 Introduction

Group-based cryptography is, as the name suggests, about the application of group theory to cryptography. The subject has a rich history with many points of origin. In this survey article the authors try to present as many interconnections of the two subjects as possible. There exists a large corpus of texts on the subject but to the best of our knowledge this is the first that tries to give a birds eye view instead of focusing on only one cryptographic application. The following Section is about applications of group theory to public key cryptography. Section 3 presents secret sharing schemes. Hash functions based on Cayley graphs are discussed in Section 4. Section 5 discusses the creation of random strings using group theory. Finally, in Section 6 the group-theoretic analogs of classical problems like the Subset Sum and the Knapsack problems are studied.

2 Public key cryptography

In 1975 Diffie, Hellman and Merkle introduced public key cryptography. The basic idea is to use for encryption a so-called one way function, a function such that it is easy to compute $f(x)$ but difficult, in general, to compute $f^{-1}(y)$.

In 1976 Diffie and Hellman presented the Discrete Logarithm Key agreement protocol [12]. This protocol uses a finite cyclic group with generator g (the original implementation used the multiplicative group of integers modulo p as the basis group and a primitive root for g). A brief description of the scheme is given bellow.

Discrete Logarithm scheme

1. Alice and Bob agree, publicly, on a finite cyclic group G and a generating element g of G.
2. Alice randomly chooses an integer α and sends g^α to Bob.
3. Bob randomly chooses an integer β and sends g^β to Alice.
4. Bob computes $g^{\alpha\beta} = (g^\alpha)^\beta$.
5. Alice computes $g^{\beta\alpha} = (g^\beta)^\alpha$.

Since $K = g^{\alpha\beta} = g^{\beta\alpha}$, K may serve as a common key. The Discrete Logarithm Key agreement protocol is considered secure because it is, supposedly, difficult for an eavesdropper to compute $g^{\alpha\beta}$ from g^α, g^β. A problem that is connected (although not equivalent) to the discrete logarithm problem, i.e. the problem of recovering α from g and g^α [**?**, p. 6-7]. Koblitz [16] and Miller [19] independently suggested the use of the group of rational points of elliptic curves as a platform in 1985.

The most famous public key encryption protocol is RSA which was proposed by Rivest, Shamir and Adleman in 1977 [27].

RSA scheme

1. Alice chooses two primes p, q, calculates $n = pq$ and selects an integer $1 < e < \phi(n) = (p-1)(q-1)$ with $g.c.d(e, \phi(n)) = 1$. She publishes n, e. Her secret key is an integer d such that $ed = 1 \mod \phi(n)$.
2. Bob encodes his message using integers $0 \leq m \leq n-1$. For every integer m, Bob sends $m^e \mod n$ to Alice.
3. Alice computes $m = m^{ed} = (m^e)^d \mod n$.

The basic mathematical tool behind RSA is Euler's theorem which states that for every integers n and a with $0 \leq a \leq n-1$, $a^{\phi(n)} = 1 \mod n$. RSA is considered secure because it is related to the integer factorization problem since the only known way to recover $m \mod n$ from $m^e \mod n$ is to calculate $m^{ed} = (m^e)^d \mod n$. And finding d, supposedly, requires knowledge of $\phi(n) = (p-1)(q-1)$ and hence the factorization of n.

Most common public key cryptosystems in use, such as presented above, are based on abelian groups. However, since computing power expands every day and new innovative lines of attack are invented, the security of many of them is questioned. For example, there exists a wide bibliography concerning attacks to the RSA cryptosystem [8]. And it is not wise to place all of one's eggs in the same basket. The wide use of only a few cryptosystems means that should a line of attack proved successful, the consequences would reach a huge number of people and certainly, the news would attract a great deal of media attention. This was the case for an announcement made by A. Shamir back in 1999 which proposed a way to break RSA [17, 32].

Therefore, research in new cryptographic methods is in demand. One such method uses the foundations of group theory, especially non-commutative structures as platforms. In section 2, after a brief overview of some basic definitions of group theory, we present a few cryptographic methods based on non-commutative groups. In section 3, we present a secret sharing scheme which was proposed by the second author [24] and it is based on group presentations and the word problem.

2.1 Non-commutative Algebraic Cryptography

2.1.1 Group presentations and normal forms

Some prerequisites for the use of a group as a platform, are the presentation of a group and the possibility to obtain a normal form for an element of the group with a specific presentation.

A **generator** g of a group G is any element of a subset $S \subset G$, called **generating set**, such that any other element of the group can be expressed in terms of S. The generators of a group G could be connected by some **relations**. For example, the cyclic group of order 2 C_2 could be defined by an element g, as the generator of C_2, which is related to the identity element of G by the relation $g^2 = e$. A **presentation of a group** G consists of a generating set and some relations between those generators. Of course, for any group there is not a unique presentation. For example the next two presentations define the cyclic group of order 6:

$$(a : a^6 = 1)$$

$$(a, b : a^2 = 1, b^3 = 1, aba^{-1}b^{-1} = 1)$$

Although there is a way to pass from one presentation to another for isomorphic groups, using the Tietze transformations for example, this is not always trivial. A useful property of a group is the possibility of writing

any element in a standard way that becomes easy for us to choose and manipulate elements. This is the notion of **normal form**. The existence of a normal form is a characteristic of free groups for example [28]. There are two necessary principles that need to be held by a normal form. The first one is uniqueness; every object must have exactly one normal form of a given type, as the second principle states that two objects of the same normal form have to be equal. For instance, in the additive group of integers it is known that every integer has a unique decomposition as a product of prime numbers, not taking in consideration the order of the product. This is a good way for representing an integer, with all the advantages described above.

Another extra, useful property of a group is the capability of effectively rewriting an element in normal form. In Thompson's Group, for example, there is such a rewriting system, that converts a given word to its normal form. Another such example is the braid group where there is not only one type of normal form, but different types of normal forms, each one useful for another reason. The early use of braid groups in cryptography is partly due to the development of different types of normal forms. Thus, the existence of a normal form is crucial for a platform group in cryptosystems. But what are other useful properties for a group to be chosen as the base of a cryptosystem? We will discuss this question in section 2.3. Let us see first the underlying problems in group-based cryptography.

2.1.2 Decision and Search problems in Group Theory

Fundamental decision problems formulated by Max Dehn [10] in 1911, being used for implementing a one-way function are presented below [18]. Let G be a group which presentation is given. Then:

- **The Word Problem**: for a word W given in terms of generators of G, find in a finite number of steps whether $W = 1$ or not.
- **The Conjugacy Problem**: for two given words W_1, W_2 on the generators of G, decide in a finite number of steps whether

$$W_1 = g^{-1} W_2 g, \ for \ g \in G.$$

- **The Isomorphism Problem**: for two presentations G, G', decide in a finite number of steps whether the groups G, G' are isomorphic or not.

These problems are not solvable in general. The word problem is solvable in the following classes of groups: finite, polycyclic, one relator negative curvature, Coxeter, Braid, residually finite, finitely generated groups and others. The conjugacy problem expands the word problem, as the latter comes from the first one, just by substituting the word $W_1 = 1$. The isomorphism problem is believed to be the most difficult of Dehn's problems.

Although some of these have been solved for some special types of groups, as mentioned above, their solution is sometimes infeasible to compute; so a one-way function could be implemented based on this fact. Even though such a problem could be solved in a finite number of steps, it could be a difficult problem from a cryptographic point of view. So, one should know whether the problem could be solved in polynomial or subexponential time. In this way **search problems** emerge: Given a property P, known that there are objects with this property, try to find such an object.

2.1.3 Key Agreement Protocols based on non commutative groups

An analogue of Discrete Logarithm Problem (DLP) in group theory is the Conjugacy Search Problem (CSP): If G is a non-abelian group and $g, h \in G$ such that g and h are conjugate, find an element $y \in G$ so that the next equality occurs: $h = y^{-1}gy$. The CSP seems to be a really hard problem given an appropriate choice of a platform group. This choice is going to be discussed later on.

2.1.4 Ko et al. Key Agreement Protocol [15]

Suppose G is a non-abelian group chosen and $A, B \leq G$ commuting subgroups and let $g \in G$ be an element, all the above publicly known. A secret common key is developed by Alice and Bob, procceding as follows:

1. Alice selects $a \in A$, calculates $g^a = a^{-1}ga$ and sends g^a to Bob.
2. Bob selects $b \in B$, calculates $g^b = b^{-1}gb$ and sends g^b to Alice.
3. Each one computes $K_a = (g^b)^a$, $K_b = (g^a)^b = K$ which stands for the common secret key, as $ab = ba$ and hence $K_a = g^{ab} = g^{ba} = K_b$.

The platform group G chosen in this scheme is a very critical point. The authors Ko et al. used for instance the Braid group B_n. The real reason about that is that a good normal form for the elements of Braid groups exists.

2.1.5 Anshel et al. Key Agreement Protocol [1]

A non-abelian group G is used for this protocol too, but the need of any commuting subgroups is overtaken. So, except the group G, also elements $a_1, \ldots, a_k, b_1, \ldots, b_m \in G$ are made publicly known. The key establishment comes as follows:

1. Alice computes a private word $x = x(a_1, \ldots, a_k)$ on $a_1, \ldots, a_k$ and sends Bob $b_1^x, \ldots, b_m^x$, where b_i^x stands for the conjugate $x^{-1}b_i x$.

2. Bob computes a private word $y = y(b_1, \ldots, b_m)$ on $b_1, \ldots, b_m$ and sends Alice $a_1^y, \ldots, a_k^y$.

3. Then Alice computes $x(a_1^y, \ldots, a_k^y) = x^y = y^{-1}xy$ and Bob computes $y(b_1^x, \ldots, b_m^x) = y^x = x^{-1}yx$. The commutator $[x, y]$ stands for the common secret key K, as: $[x, y] = x^{-1}x^y$ and $[x, y] = \left(y^{-1}y^x\right)^{-1} = \left(y^{-1}x^{-1}yx\right)^{-1} = x^{-1}y^{-1}xy$ respectively.

An interesting implementation involves the braid groups. In particular, the n-braid group B_n is defined by the following group presentation.

$$B_n = \left\langle \sigma_1, \ldots, \sigma_{n-1} \,\middle|\, \begin{matrix} \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j, & if\ |i-j| = 1 \\ \sigma_i\sigma_j = \sigma_j\sigma_i, & if\ |i-j| \geq 2 \end{matrix} \right\rangle$$

Each element of B_n is called an n- braid. A n-braid can be displayed as a set of disjoint n strands all of which are attached to two horizontal bars at the top and at the bottom, so as each strand always heads downwards as one "walks" along the strand from the top to the bottom. The braid index is the number of strings. The multiplication ab of two braids σ_1 and σ_2 is the braid obtained by positioning σ_1 on top of σ_2. The identity e is the braid consisting of n straight vertical strands and the inverse of a is the reflection of a with respect to a horizontal line.

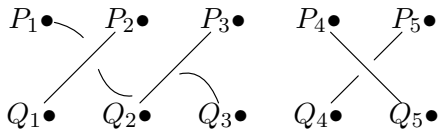

Figure 1: *5 - braid*

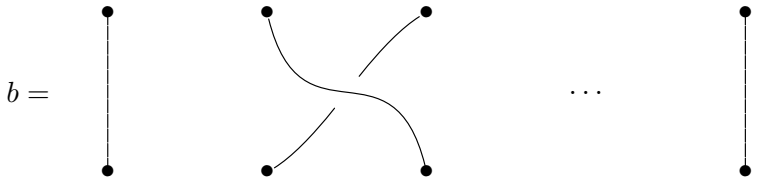

Figure 2: *A n - braid b*

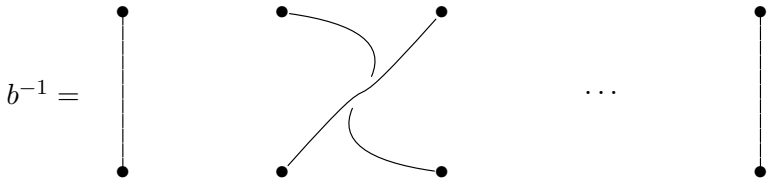

Figure 3: *The inverse n - braid* b^{-1}

Let us see why braid groups could be a good candidate as a platform group. First of all it has solvable word problem and there is a canonical form (in fact more [4]) for it's elements such that braids are easily compared. Also the best known algorithm to solve the conjugacy problem is of exponential time. Finally, the membership decision problem in a braid group $B_n,\ n > 6$ is algorithmically unsolvable, because such a group contains subgroups isomorphic to the free group product $F_2 \times F_2$, where the membership decision problem (determining whether or not a given $x \in G$ belongs to a subgroup of G generated by given $a_1, a_2, \ldots a_n$) is algorithmically unsolvable [**?**, p.48]. This is important because an adversary would have to know the x, y elements above not simply as a word in G, but as a word in public elements used $a_1, a_2, \ldots a_n$ or $b_1, b_2, \ldots b_m$ in order to reveal the secret key K ([22]).

2.1.6 Other protocols

There is an easy way to implement other protocols not based on the conjugacy search problem. Besides, the first naive scheme on group based cryptography by Wagner and Magyarik (1985) [33], was depended on the word problem, although it was not really a cryptosystem [3]. The basic idea is that we could have a defining relation anything like:

$$g^a = f(a)ga, \qquad f : G \to G$$

where f is a function in group G. In the case of conjugacy search problem f is defined as:

$$\begin{aligned} f : G &\rightarrow G \\ x &\mapsto f(x) = x^{-1} \end{aligned}$$

On the other hand f could be also the identity map, inducing the **decomposition problem protocols**. An extended reference on such themes can be found in [22].

2.2 Attacks on group-based systems

Several cryptanalytic methods try to exploit the group structure used in group-based cryptographic schemes. For example, if the basis group G that is used is linear and there exists a group isomorphism $\phi : G \rightarrow GL(V)$ which can be efficiently computed, then the word problem and the conjugacy problems can be solved with simple matrix multiplication and linear algebra respectively [6]. Below, we present two methods. More information can be found in [22].

2.2.1 Length based attacks

Given a group G where we can define a normal form we can define a length function l. $l(g)$ is the number of generators counted with multiplicity that appear in the normal form of g. In 2003 Hughes and Tannenbaum [13] proposed a method for solving the Conjugacy Search Problem using the length function.

We remind that the problem is given two elements $g, h = y^{-1}gy$ where $g, y \in G$ to find the element y. If $S = \{s_1, \ldots, s_n\}$ is a generating set of G then y can be written as a product of the generators i.e. $y = s_{i_1} \cdots s_{i_k}$, $s_j \in \{1, \ldots, n\}$. Hughes and Tannenbaum observed that in general the following inequality holds:

$$l(x) < l(s^{-1}xs),\ s \in S, x \in G$$

Hence given $h = y^{-1}gy = s_{i_k}^{-1} \cdots s_{i_1}^{-1} g s_{i_1} \cdots s_{i_k}$ we can form the products $s_1 h s_1^{-1}, \ldots, s_n h s_n^{-1}$ and calculate their lengths. If $l(s_m h s_m^{-1}) < l(h)$ for some element s_m then, hopefully, $s_{i_k} = s_m$. In that case we conjugate with s_m and repeat the process for the element $h' = s_{i_k}^{-1} \cdots s_{i_2}^{-1} g s_{i_2} \cdots s_{i_k}$.

In [29] Ruinsky et. al. remark that since in several protocols the element y in the Conjugacy Search Problem belongs to a subgroup H of G instead of using a length function we could use a subgroup distance function d i.e. a function $d : G \rightarrow \mathbb{R}^+$ that satisfies the following two axioms:

1. $d(h) = 0$ for all $h \in H$,

2. $d(g) > 0$ for all $g \in G \setminus H$.

2.2.2 Quotient attacks

Another general method for attacking group-based cryptosystems is the so called quotient attacks. The idea behind these attacks is for a given group G to find a "suitable" quotient G/N and a surjective group homomorphism $\phi : G \to G/N$. Then instead of solving a problem (ex. Conjugacy Search Problem, Membership Problem etc.) in G solve it in G/N. The solution then might be "lifted" in G.

For example, let G be a group and $H =< a_1, \ldots, a_m >$ a finitely generated subgroup. The membership problem is for a given element g in G to determine whether g is an element of H. If $\phi : G \to G/N$ is a surjective group homomorphism one may try to determine whether $\phi(g)$ belongs to $\phi(H) =< \phi(a_1), \ldots, \phi(a_m) >$. If $\phi(g) \in \phi(H)$ and $\phi(g) = w(\phi(a_1), \ldots, \phi(a_m))$ is the expression of $\phi(g)$ in the generating set $\{\phi(a_1), \ldots, \phi(a_m)\}$ then one can check if $g = w(a_1, \ldots, a_m)$. In case the last equation holds, we have $g \in H$.

2.3 Choosing the group and the problem

As noted above the construction of a group-based cryptosystem has two parts. The first one is choosing the platform group and the second one is the underlying problem. Concerning the group-based underlying problem it is not really clear, at the present, which is the best choice. For example, the conjugacy search problem may not provide a sufficient security level in braid groups, although it could be adopted in special cases [22].

On the other hand the choice of the platform group is believed that has to meet specific standards, such as mentioned below. The first one is the existence of an effective normal form, so that the word problem is solvable in real time. The normal form is also useful in hiding the message parts that could be obvious to recover; i.e. the part elements $x, y \in G$ in the product xy. Another, high priority, requirement is the size of the group. It is needed to be of super-polynomial growth, that means the elements of length n, growing faster than any polynomial in n so that a direct attack could not be implemented. Finally, all the above contribute to the choice of a well known group. Among groups meeting the above criteria and used so far are: braid groups, Thompson's group, Artin groups, solvable groups and others. Research in this context is open and has many potential.

3 Secret sharing

3.1 Secret sharing in general

A secret sharing scheme answers to the problem of distributing a secret among a group of n persons in such a way that it can be reconstructed only if at least t of them combine their shares. Such a scheme is called a (n, t) threshold scheme. If $n = t$ then, the most simple solution is to encode the secret as a vector $u = (u_1, \ldots, u_m)$ and distribute one vector $v_i = (v_{i1}, \ldots, v_{im})$ to each participant such that $u = v_1 + \cdots + v_n$. We call this method the XOR method. The problem for $n \neq t$ was first solved independently by A. Shamir [30] and G. Blakley [5] in 1979. In what follows we, briefly, present Shamir's scheme for creating a (n, t) threshold scheme.

Shamir's secret sharing scheme

Suppose that the secret is encoded by a number D.

1. Choose at random $t-1$ coefficients $a_1, \ldots, a_{t-1}$ and calculate the values $D_i = p(i)$ for $i = 1, \ldots, n$ of the polynomial $p(x) = D + a_1 x + \cdots + a_{t-1} x^{t-1}$. D_i are the distributed pieces of the key.

2. Given any subset of t of these D_i values we can find the coefficients of $p(x)$ by interpolation, and then evaluate $D = p(0)$. Knowledge of $t - 1$ (or fewer) of these values, on the other hand, does not suffice to calculate D. Knowing $t - 1$ (or fewer) shares provide no advantage over knowing no pieces (i.e. this is a perfect threshold scheme).

Shamir's solution to the secret sharing problem (as well as several other solutions) has some interesting properties. For example:

- its security is theoretical, it is not based on the hardness of a specific problem. Not knowing at least $t - 1$ pieces makes it not hard but impossible to find D,
- the keys are easy to change without changing the original secret information D,
- we can have a hierarchial scheme in which important persons have more pieces of the key,
- if t is fixed, then any of the pieces D_i can be dynamically added or deleted without this affecting the other pieces.

The use of group theory to secret sharing was introduced by the second author in [24] and it is based on group presentations and the word problem. A brief presentation of the scheme is given in the next subsection.

3.2 A secret sharing scheme based on the word problem

3.2.1 Description of the scheme

Suppose that a binary sequence must be distributed among n persons in such a way that at least t of them must cooperate in order to obtain the whole sequence. The secret sharing scheme consists of the following steps:

1. A group G with finite presentation $G =< x_1, x_2, \ldots, x_k / r_1, \ldots, r_m >$ and soluble word problem is chosen. We require that $m = \begin{pmatrix} n \\ t-1 \end{pmatrix}$.

2. Let $A_1, \ldots, A_m$ be an enumeration of the subsets of $\{1, \ldots, n\}$ with t-1 elements. Let $R_1, \ldots, R_n$ be n subsets of the relators set $\{r_1, \ldots, r_m\}$ where $r_j \in R_i$ if and only if $i \notin A_j$, $j = 1, \ldots, m$, $i = 1, \ldots, n$.

   Another way of viewing the sets $R_1, \ldots, R_n$ is the following: each set R_i is created from the relators set $\{r_1, \ldots, r_m\}$ after deleting the relations r_k for those k for which i belongs to A_k.

   Thus, for every $j = 1, \ldots, m$, r_j is not contained in exactly t-1 of the subsets $R_1, \ldots, R_n$. It follows that r_j is contained in any union of t of them whereas if we take any t-1 of the $R_1, \ldots, R_n$ there exists a j such that r_j is not contained in their union.

3. Distribute to each of the n persons one of the sets $R_1, \ldots, R_n$. The set $\{x_1, \ldots, x_k\}$ is known to all of them.

4. If the binary sequence to be distributed is $a_1 \cdots a_l$ construct and distribute a sequence of elements $w_1, \ldots, w_l$ of G such that $w_i =_G 1$ if and only if $a_i = 1$, $i = 1, \ldots, l$. The word w_i must involve most of the relations $r_1, \ldots, r_m$ if $w_i = 1$. Furthermore, all of the relations must be used at some point in the construction of some element.

Any t of the n persons can obtain the sequence $a_1 \cdots a_l$ by taking the union of the subsets of the relations of G that they possess and thus obtaining the presentation $G =< x_1, x_2, \ldots, x_k / r_1, r_2, \ldots, r_m >$ and solving the word problem $w_i =_G 1$ in G for $i = 1, \ldots, l$.

A coalition of fewer than t persons cannot decode correctly the message since the union of fewer than t of the sets $R_1, \ldots, R_n$ contains some but not all of the relations $r_1, \ldots, r_m$. Thus, such a coalition can only obtain a group presentation $G' =< x_1, \ldots, x_k / r_{j_1}, \ldots, r_{j_p} >$ with $p < m$ and $G \neq G'$, where $w_i =_G 1$ is not equivalent to $w_i =_{G'} 1$ in general.

In [24] it was proposed that Coxeter or polycyclic groups could be used for the implementation of the scheme. In the same article some more remarks were made on possible attacks to the scheme and ways to protect from them.

3.2.2 Some interesting remarks

The aforementioned scheme has several interesting properties:

- If in step 4 all the relations $r_1, \ldots, r_m$ are used f or the generation of a word w representing the digit 1, then the scheme is perfect (i.e. no information can be gained by a subset with fewer than t secret holders).
- Contrary to other schemes (e.g. Shamir's, Blakley's scheme), the secret sequence to be shared can be transmitted through open channels.
- The secret is not needed until the final step. Hence, it is possible for someone to distribute the sets $R_1, \ldots, R_n$ and decide at a later time what the sequence will be.
- Since the secret is not needed until the final step several different secrets can be shared without updating the long-term private information.
- Since the secret is not needed until the final step the scheme can also be used so that t of the n persons can verify the authenticity of a message. In particular the binary sequence in step 4 could contain a predetermined subsequence (signature) along with the normal message. Then t persons may check whether this predetermined sequence is contained in the encoded message thus validating it.
- Moreover, the signature mentioned above can be created in such a way that only a certain subset of n persons can verify it. This can be done by using only the relations that appear in the pieces of this particular subset of key holders. In general it can be arranged that only a specific subset of the key holders will be able to correctly decrypt a message. It can even be arranged that two different subsets of key holders will end up with entirely different messages after decryption.
- Like Shamir's scheme the security is theoretical. Any of the pieces can be added or deleted without this affecting the others and we can have a hierarchial scheme in which important persons have more pieces of the

key. On the other hand, contrary to Shamir's scheme, it is not obvious how to add a share. It is not clear, also, if a key can be changed.

3.2.3 Further developments

The major drawback of the previous (n, t) secret sharing scheme is that the number of the relationships $r_1, \ldots, r_m$ in the group's presentations grows very fast for increasing n. Habeeb et. al. [14] and Cavallo et. al. [9] try to remedy the problem by proposing variations of the classical secret sharing schemes that use group theory.

More specifically, they alter the XOR method by distributing groups $G_i =< x_1, \ldots, x_m / R_i >$ to each participant and then use them to encode and distribute the, binary in this case, vectors v_i. This is done for every vector $v_i = (v_{i1}, \ldots, v_{im})$ by a sequence of words $w_{i1}, \cdots, w_{im}$ where $w_{ij} = 1$ in G_i if and only if $v_{ij} = 1$. As for Shamir's scheme they encode the values D_i first into binary sequences and then use the same method to transmit them to the participants [9, 14]. In addition to this method in [9] it is proposed to use groups $G_i =< x_1, \ldots, x_m / R_i >$ where a normal form can be efficiently computed and which we can use to order the elements of the group. The encoding of each value D_i is a word in G_i whose number is D_i in the resulting ordering.

4 Hash functions

4.1 Hash functions in general

This section contains a few generalities on hash functions. We beginwith some definitions.

Definition 4.1. *A hash function h is a function that takes as input an arbitrarily long string of letters on an alphabet X, usually called a message, and outputs a bit string of fixed length n.*

$$h : X^* \to X^n,\ m \mapsto h(m)$$

Remark 4.2. *In the literature the alphabet X is $\{0, 1\}$. The reason for the generalization will become clear in the next subsection. (Our apologies to the cryptographic community.)*

A classical application of hash functions is in password storage [11]. The passwords of the users of an online service (ex. e-banking) are not stored

verbatim but for every password m the value $h(m)$ is calculated and stored. Every time a user tries to use the service he provides his password and its hash image is computed and compared with the stored value. Hash functions are designed such that:

1. it is highly improbable to have to strings m, m' such that $h(m) = h(m')$, hence a user passes the verification only if he provides the right password and
2. it is computationally impossible to find a pre-image m of a hashed value $h(m)$, hence even if a malicious agent manages to obtain the stored hashed values he will not be able to retrieve the users passwords.

The second property is the reason for saying that hash functions are one-way functions. Of course, from a mathematicians point of view a hash function is obviously not injective. The application above justifies the following definition:

Definition 4.3. *Let $h : X^* \longrightarrow X^n$ be a hash function. h is said to be:*

- *preimage resistant if given $h \in X^n$ it is computationally infeasible to find $m \in X^*$ such that $h = h(m)$.*
- *second preimage resistant if given $m \in \{0,1\}^*$ it is computationally infeasible to find $m' \in X^*$ such that $h(m) = h(m')$.*
- *collision resistant if it is computationally infeasible to find $m, m' \in X^*$, $m \neq m'$ such that $h(m) = h(m')$.*

Some simple arguments [11] show that if a hash function is collision resistant then, it is second preimage resistant. Furthermore, if a hash function is second preimage resistant then, it most probable is preimage resistant.

4.2 Cayley hash functions

At Eurocrypt'91 Zémor [34] introduced a hash function based on a Cayley graph of the group $SL_2(\mathbb{F}_p)$ where p is a large prime. More specifically, Zémor proposed to encode a string as a word on $\{A, B\}$ where

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

and then evaluate the word in $SL_2(\mathbb{F}_p)$, the resulting element being the hashed value of the word. Based on this example we can define hash functions based on finite groups and their generating sets.

Definition 4.4. *Let G be a finite group and S a generating set of the group, a Cayley hash function $h : S^* \to G$ is the function that sends a word on S to the corresponding element of G.*

When using Cayley graphs for hashing the following three group-theoretic problems are of paramount importance.

Definition 4.5. *[25] Let G be a group and let $S = \{s_1, \ldots, s_k\}$ be a set of generators of G. If $N \in \mathbb{N}$ then, we define the following problems*

- ***The Balance Problem**: find two words $x_1 \ldots x_m$ and $y_1 \ldots y_l$ in the alphabet S with $m, l < N$ such that $x_1 \ldots x_m = y_1 \ldots y_l$ in G.*
- ***The Representation Problem**: find a (non trivial) word $x_1 \ldots x_m$ in the alphabet S with $m < N$ such that $x_1 \ldots x_m = l$ in G.*
- ***The Factorization Problem**: given an element $g \in G$ find a word $x_1 \ldots x_m$ in the alphabet S with $m < N$ such that $x_1 \ldots x_m = g$ in G.*

It is obvious that the Balance, Representation and Factorization problems correspond to the notions of collision resistance, second preimage resistance and preimage resistance for a hash function respectively. We should note here the importance of the bound N in the above problems. For example, if the bounding condition on the length of the word is removed, then a simple answer to the Representation problem would be s^k, where $s_1 S$ is a generator and $k = |G|$ is the order of G.

As noted by Zémor [34, 35] when creating Cayley hash functions we require that the corresponding Cayley graph of the group G and its generator set S (i.e. the graph whose vertices are the elements of G and where (g, w) is an edge if and only if $gw^{-1} \in S \cup S^{-1}$) has large girth and small diameter. These questions are closely related to the Babai's conjecture:

Conjecture 4.6. *[2, conjecture 1.7] If G is a non-abelian finite simple group of order N and $diam(G)$ its diameter , then $diam(G) < (logN)^C$ for some constant C.*

As Petit and Quisquater noted "the factorization problem can be seen as providing a constructive proof of Babai's conjecture" [25]. But, while the conjecture has been proved for many spacial cases the proof is non constructive. Only for certain generating sets are constructive proofs are known. The interested reader may consult [25] for further references.

4.3 Attacks on Cayley hash functions

Soon after the introduction of Cayley hash functions by Zémor the initial scheme based on $SL_2(\mathbb{F}_p)$ was found to be insecure by Tillich and Zémor [31] (in the same article the authors presented a Cayley hash function based on $SL_2(\mathbb{F}_{2^n})$ to overcome the vulnerability of the initial protocol). In this section we give a brief description of the attack that was used.

The attack is based on the fact that every matrix of $SL_2(\mathbb{Z})$ with non negative entries is a product of $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and the fact that there is a factorization algorithm for the generating set

$$\left\{ T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \right\}.$$

Given a matrix M in $SL_2(\mathbb{F}_p)$ someone finds a matrix $M' \in SL_2(\mathbb{Z})$ that reduces to M modulo p and then expresses M' as a product of A and B. Finally, he reduces the expression modulo p and derives a factorization of M. The idea behind this "lifting" attack was used by Petit and Quisquater for breaking the hash function based in the group $SL_2(\mathbb{F}_{2^n})$ too [25].

We should note here that Cayley hash functions are also vulnerable to quotient attacks. More information can be found in [23] and [25] and the articles mentioned in them. The existence of these attacks has made Cayley hash functions out of fashion for awhile. But a closer look shows that the effectiveness of these attacks is based on the specific group of generators S that is used. A good choice of the base group G and the generator set S will keep the system safe from the known attacks.

5 Random strings and group theory

It would be a serious omission not mention the existence of a pseudo-random bit generator based on the discrete logarithm problem [7]. The following exposition is based on [11].

Definition 5.1. *Let $I = (I_k)_{k\in\mathbb{N}}$ be a key set with security parameter k. Let $f = (f_i : D_i \to R_i)_{i\in I}$ be a family of one-way functions with key generator K, and let $B = (B_i : D_i \to \{0,1\})_{i\in I}$ be a family of boolean predicates. Then, B is called a family of hard-core predicates if and only if:*

1. *B can be computed by a Monte Carlo algorithm,*
2. *$B(x)$ is not computable from $f(x)$ by an efficient algorithm*

Given a $I = (I_k)_{k\in\mathbb{N}}$ be a key set with security parameter k, a polynomial $Q \in \mathbb{Z}[x]$ such that $Q(x) > 0$ for all $x \in \mathbb{R}$, $x > 0$ and a family of one-way functions $f = (f_i : D_i \to D_i)_{i\in I}$ with hard-core predicate $B = (B_i : D_i \to \{0,1\})_{i\in I}$ where each f_i is bijective we can construct the following pseudo-random generator:

$$G := G(f, B, Q) = (G_i : D_i \to \{0,1\})_{k\in\mathbb{N}, i\in I}$$

$$x \in D_i \to (B_i(x), B_i(f_i(x)), B_i(f_i^2(x)), \ldots, B_i(f_i^{Q(k)-1}(x)))$$

The generator described above creates a bit sequence that is virtually indistinguishable from a truly random one [11, Th. 8.4]. In [7] it was proved that if

$$f = (exp_{p,g} : Z_{p-1} \to \mathbb{Z}_p^* \simeq \mathbb{Z}_{p-1})_{(p,g)\in I}$$

with $I = \{(p, g) : p \text{ prime}, g \in \mathbb{Z}_p^* \text{ a primitive root}\}$ and $B = (B_{p,g})_{(p,g)\in I}$ where $B_{p,g}(x) = 1$ if and only if $x \in \mathbb{Z}_p^*$ is the principal square root of x^2 then, B is a family of hard-core predicates. That is given the assumption that the discrete logarithm problem in $\mathbb{Z}_p^*$ is computationally hard.

The natural question is:

Problem 5.2. *Does a non-commutative version of the above construction exist?.*

6 Generalization of classical problems

We would like to end our survey by mentioning a research program initiated by Myasnikov, Nikolaev and Ushakov in a series of two articles [20, 21]. They suggest that the complexity of generalizations of various discrete optimization problems (ex. the subset sum problem, the knapsack problem, the Post correspondence problem) should be studied and that a corpus of results in this area would serve a variety of goals. In particular such a search would:

- deepen our understanding of the classical discrete optimization problems,
- some of the problems are interesting on their own,

Let G be a group. Examples of the generalized versions of the problems are [20]:

- **The Subset Sum Problem**: given $g_1, \ldots, g_m, g \in G$ decide if $g = g_1^{\epsilon_1} \cdots g_m^{\epsilon_m}$ for $\epsilon - 1, \ldots, \epsilon_m \in \{0, 1\}$.
- **The Knapsack Problem**: given $g_1, \ldots, g_m, g \in G$ decide if $g = g_1^{\epsilon_1} \cdots g_m^{\epsilon_m}$ for non-negative integers $\epsilon - 1, \ldots, \epsilon_m$.
- **The Submonoid Membership Problem**: given $g_1, \ldots, g_m, g \in G$ decide if g belongs to the submonoid generated by $g_1, \ldots g_m$ i.e. if g is equal to a product of the form $g_{i_1} \cdots g_{i_n}$ where $g_{i_1}, \ldots, g_{i_n} \in \{g_1, \ldots, g_m\}$.
- **The Bounded Submonoid Membership Problem**: given $g_1, \ldots, g_m$,$g \in G$ and $1^k \in \mathbb{N}$ (in unary) decide if g is equal to a product of the form $g_{i_1} \cdots g_{i_n}$ where $g_{i_1}, \ldots, g_{i_n} \in \{g_1, \ldots, g_m\}$ and $n \leq k$.

In case the set $\{g_1, \ldots, g_m\}$ is closed under inversions then, the Submonoid Membership Problem is actually the well-known generalized word problem of group theory. In case the set $\{g_1, \ldots, g_m\}$ is a generating set then, we have the Factorization problem cf.(Def. 4.5).

Already there are some interesting results. One of them is the following.

Theorem 6.1. *[20, Th. 5.11] Let G be a hyperbolic group. Then, the Bounded Submonoid Membership Problem is in **P**.*

Which should be contrasted with:

Theorem 6.2. *[26] There are hyperbolic groups with undecidable subgroup membership problem.*

7 Conclusion

This ends our brief encounter of group-based cryptography. We would like to stress that the use of group theory to cryptography is a very active multidisciplinary research area. Group-based cryptography is a relatively new field with many interesting, far reaching open problems. The authors hope that the article gives a glimpse in this exciting field.

References

[1] I.Anshel, M.Anshel, D.Goldfeld, An algebraic method for public-key cryptography, *Math.Res.Lett.*, **6**, (1999), 287-291.

[2] L. Babai, A. Seress, On the diameter of permutation groups, *European Journal of Combinatorics*, **13 4**, (1992), 231243.

[3] J.C.Birget, S.S.Magliveras, M.Sramka, On public key cryptosystems based on combinatorial group theory, *Tatra Mt. Publ.*, **33**, (2006), 137-148.

[4] J.S. Birman, Braids, links and mapping class groups, *Annals of Math. Study*, **82**, (1974).

[5] G.R. Blakley, Safeguarding cryptographic keys, *Proceedings of AFIPS*, **48**, (1979), 313-317.

[6] S. R. Blackburn, C. Cid, C. Mullan, Group Theory in Cryptography, *Groups St Andrews 2009 in Bath, London Mathematical Society Lecture Note Series,Cambridge University Press*, **387**(1), (2011),133149. Also on Arxiv: http://arxiv.org/abs/0906.5545v2.

[7] M. Blum, S. Mikali, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM Journal of Computation*, **13 4**, (1984), 850-864.

[8] D. Boneh, Twenty Years of Attacks on the RSA Cryptosystem, *Notices of the AMS*, **46**(2), (1999), 203-213.

[9] B. Cavallo, D. Kahrobaei, Secret sharing using non-commutative groups and the shortlex order, *Arxiv*, http://arxiv.org/abs/1311.7117, (2013).

[10] B. Chandler, W. Magnus, *The History of Combinatorial Group Theory: A Case Study in the History of Ideas*, Springer-Vedag New York, 1982.

[11] H. Delfs, H. Knebl, Introduction to Cryptography, *Springer*, (2007).

[12] W. Diffie, M. Hellman, New directions in cryptography, *IEEE Transaction on Information Theory*, **22**, (1976), 644-654.

[13] J. Hughes, A. Tannenbaum, Length-Based Attacks for Certain Group Based Encryption Rewriting Systems, , *Workshop SECI02 SEcurite de la Communication sur Intenet*, (2002). Also on Arxiv: http://arxiv.org/abs/cs/0306032.

[14] M. Habeeb, D. Kahrobaei, V. Shpilrain, A secret sharing scheme based on group theory and the word problem, em Computational and Combinatorial Group Theory and Cryptography, Contemporary Mathematics, **582**, (2012), 143-150.

[15] K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.s. Kang, C. Park, New public-key cryptosystem using braid group, *Advances in Cryptology - CRYPTO 2000, Lecture Notes in Computer Science*, Springer, Berlin, (2000), 166-183.

[16] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, **48**(177), (1987), 203-209.

[17] J. Markoff, Israeli Scientist Reports Discovery of Advance in Code Breaking, *The NY Times*, May 02, 1999.

[18] W. Magnus, A. Karrass, D. Solitar, *Combinatorial Group Theory*, Dover Publications Inc, 1976.

[19] V. Miller, Use of elliptic curves in cryptography, *Advances in Cryptology Proceedings of CRYPTO '85*, Springer, Berlin, (1985), 417-426.

[20] A. Myasnikov, A. Nikolaev, A. Ushakov, Knapsack Problems in Groups, *Arxiv*, http://arxiv.org/abs/1302.5671, (2013).

[21] A. Myasnikov, A. Nikolaev, A. Ushakov, The post correspondence problem in Groups, *Arxiv*, http://arxiv.org/abs/1310.5246, (2013).

[22] A. G. Myasnikov, V. Shpilrain, A. Ushakov, *Group-based cryptography*, Birkhäuser Verlag, 2008.

[23] C. Mullan, Some results in group-based cryptography, *Techical Report*, http://www.ma.rhul.ac.uk/static/techrep/2012/MA-2012-01.pdf, (2012).

[24] D. Panagopoulos, A secret sharing scheme using groups, *Arxiv*, http://arxiv.org/abs/1009.0026, (2010).

[25] C. Petit, J-J Quisquater, Rubik's for cryptographers, *Cryptology ePrint Archive, Report 2011/638*, http://eprint.iacr.org/2011/638, (2011). (A short version appeared in The Notices of AMS, June/July 2013).

[26] E. Rips, Subgroups of small cancellation groups, *Bulletin of the London Mathematical Society*, **14**, (1982), 45-47.

[27] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and pulic-key cryptosystems, *Communications of the ACM*, **21**, (1978), 120-126.

[28] D.J.S. Robinson, *A Course in the Theory of Groups*, Springer Verlag, 2nd edition, 1996.

[29] D. Ruinskiy, A. Shamir, B. Tsaban, Cryptanalysis of Group-Based Key Agreement Protocols Using Subgroup Distance Functions, *Public Key Cryptography PKC 2007 Lecture Notes in Computer Science*, Springer, **4450**, (2007), pp 61-75

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

[31] J. P. Tillich, G. Zémor, Hashing with SL_2, *Advances in Cryptology - CRYPTO'94, Lecture Notes in Computer Science*, Springer, **839**, (1994), 40-49.

[32] Wire magazine, Cryptography workshop makes news *The Wire online*, **13**(1), (1999).

[33] N.R. Wagner, M.R. Magyarik, A public key cryptosystem based on the word problem, *Advances in Cryptology - CRYPTO '84, Lecture notes in Computer Science*, Springer, **196**, (1985), 19-36.

[34] G. Zémor, Hash functions and graphs with large girths, *Advances in Cryptology EUROCRYPT 91, Lecture Notes in Computer Science*, Springer, **547**, (1991), 508-511.

[35] G. Zémor, Hash functions and Cayley graphs,*Designs, Codes and Cryptography*, **4**, (1994), 381-394.