

On Generic Groups and Related Bilinear Problems

David LUBICZ^{a,b} and Thomas SIRVENT^{a,b}

^a *DGA-CÉLAR, Bruz, France*

^b *IRMAR, Université de Rennes 1, France*

david.lubicz@univ-rennes1.fr thomas.sirvent@m4x.org

Abstract. Groups with pairing are now considered as standard building blocks for cryptographic primitives. The security of schemes based on such groups relies on hypotheses related to the discrete logarithm problem. As these hypotheses are not proved, one would like to have some positive security argument for them. It is usual to assess their security in the so called generic group model introduced by Nechaev and Shoup. Over the time, this model has been extended in different directions to cover new features.

The relevance of this model is nevertheless subject to criticisms: in particular, the fact that the answer to any fresh query is a random bit string is not what one expects from a usual group law.

In this paper, we develop a generic group model with pairing which generalizes all the models seen so far in the literature. We provide a general framework in order to prove difficulty assumptions in this setting. In order to improve the realism of this model, we introduce the notion of pseudo-random families of groups. We show how to reduce the security of a problem in such a family to the security of the same problem in the generic group model and to the security of an underlying strong pseudo-random family of permutations.

Keywords. Generic groups, Bilinear Diffie-Hellman assumptions, Pairings, Discrete logarithm, Pseudo-random permutations

Introduction

The discrete logarithm problem is a general way to build trapdoor functions for asymmetric cryptographic protocols. It can be used as long as one can find a family of cyclic groups satisfying certain properties:

- one would like to be able to compute quickly the group law and, as a consequence, the exponential map using for instance the square and multiply algorithm,
- it is also necessary for the discrete logarithm problem to be intractable in the considered family of groups.

There is no family of groups for which this last property is proved. Nevertheless, on the practical side, there is no known algorithm better than the generic ones (for instance Pollard's Rho [MOV01]) to compute the discrete logarithm problem in the family of rational points of elliptic curves defined over a finite field. These generic algorithms are of exponential time complexity. Some widely used cryptosystems, for instance Diffie-

*submitted to Identity-Based Cryptography, M. Joye and G. Neven Editors,
Cryptography and Information Security Series, IOS Press.*

D. Lubicz, T. Sirvent

Hellman key agreement and its derivative [DH76,MTI86], El Gamal encryption [ElG85a, ElG85b] or DSA and related signature schemes [FIPS186,ElG85a,ElG85b,Sch91], actually implement trapdoor functions such that their security relies on the intractability of a discrete logarithm related problem.

Generic group model. This situation is not really satisfying, and one would expect at least some positive argument for the discrete logarithm problem to be intractable. The generic group model designed by Nechaev and Shoup is supposed to fill that gap. More precisely, the papers [Nec94,Sho97] define the notion of “generic algorithms” which are, roughly speaking, automatons with memory which can only perform group operations. The authors managed to prove that the fastest algorithms solving the discrete logarithm problem and the Diffie-Hellman problems are in exponential time in this model of computation. In a modern formulation, in the generic group model, adversaries are Turing machines dealing with bit strings instead of group elements and are unable to compute group operations. These group operations are provided by oracles defined in the following way:

- all queries are stored in a list;
- when a new query is similar to a former one, the oracles return the same answer;
- when a new query is fresh, the oracles return a randomly chosen bit string.

At the end of the game, the challenger verifies that the simulation provided by these oracles is coherent with discrete logarithm values drawn randomly for each initial bit string submitted to the adversary and by checking that no collision occurs.

This general framework has subsequently been improved in order to take into account new group properties used in protocols such as pairings [BB04b,YW05]. As a matter of fact, the generic group model is now a standard tool for the proof of cryptographic protocols. It is used either directly to compute running time lower bounds for breaking a given protocol [Bro05a,Bro05b,LS08], or more generally in order to assess the security of a new computational or decisional hypothesis upon which is based the security of a protocol [Jou00,BB04a,BB04b].

Limitations and criticisms. Because of the widespread use of the generic group model, a natural and important question is the relevance of this model. As pointed out in [MF07], this model has some evident limitations which make it more restricted than the case of a full-fledged Turing Machine with access to group and pairing oracles. For instance, when receiving a bit string from an oracle, an attacker may flip a certain bit and use the resulting bit string to submit a new query. Intuitively, this type of query will not improve the probability of success of the attacker but still should be taken into account, in an extension of the model.

Actually the generic group model has been the target of numerous criticisms [Den02, SPMS02,NSS04] some of which have been turned down [KM07]. For instance, because the oracles answer with random bit strings to fresh queries, the generic group model is then often compared with the random oracle model. Such random behavior of the generic group model is not what one expects to modelize real groups, like it was pointed out in [KM07].

Our contribution. In this paper, we first revisit the generic group model. We present a complete mathematical background on groups and pairings in order to define precisely

On Generic Groups and Related Bilinear Problems

the families of groups corresponding to a generic model. We include extensions like the possibility for an adversary to use bit strings which have not been given as an input of the considered problem, nor as an answer to a query from an oracle. We follow moreover a more natural approach in the analysis of a given problem. In the usual generic group model, the answer of the group law oracles is just a random sequence of bits for any fresh query and it does not take into account any underlying algebraic structure. The group structure is only used at the end of the game in order to check the presence of a collision. In contrast, in our model, the group law is randomly chosen at the beginning of the game. We study the set of group laws that remain indistinguishable from this specific law during the whole game, and we compute a probability based on this set. This subtle difference will become very clear in the discussion at the end of Part 4.

In this way, the answers to oracle queries are formally not randomly chosen during the game, but really computed from a pre-determined group law. However, this is still not satisfying: the group law used in the computations is initially randomly chosen amongst all possible group laws, which is far from being realistic.

In order to improve the realism of the model, we introduce the notion of pseudo-random groups: a pseudo-random group is a group where the addition law is provided by the way of a strong pseudo-random permutation. As the family of random permutations is a particular case of strong pseudo-random permutations, it is easy to see that our model is a generalization of the generic group model. The main result of this paper shows that every result that can be proved in the generic group model can also be proved in the pseudo-random group model. In other words, we can consider families of groups where the group laws, provided by oracles, are drawn following an arbitrary distribution. The security of an hypothesis over such a family can be reduced to the robustness of this family as a pseudo-random family of groups, through the security of this same hypothesis in the generic model.

Organization of the paper. In Parts 1 and 2, we give a definition for the generic families of cyclic groups and generic families of cyclic groups with pairings. In Part 3, we develop the technical tools to assess the security of a problem in the generic group model. In Part 4, we define the pseudo-random families of cyclic groups and explain how to prove the difficulty of a problem in a pseudo-random family of cyclic groups by reduction upon the generic group model and the security of the underlying pseudo-random permutation family.

1. Cyclic groups and their representations

1.1. A unique cyclic group of order n

The cyclic groups are classified up to isomorphisms by their order, i.e. two cyclic groups are isomorphic if and only if they share the same order. In particular, a cyclic group G is of order $n \in \mathbb{N}^*$ if and only if it is isomorphic to the additive group $\mathbb{Z}/n\mathbb{Z}$. Of course, there can be several isomorphisms between G and $\mathbb{Z}/n\mathbb{Z}$, but one can consider the unique isomorphism l , from G with a generator g , into $\mathbb{Z}/n\mathbb{Z}$, such that $l(g) = 1$.

It should be remarked that for all $x \in G$, $l(x)$ is nothing but $\log_g(x)$, and then computing the isomorphism l is the same thing as computing the discrete logarithm map of G in base g .

D. Lubicz, T. Sirvent

1.2. Inherited structure of group

Let A be a set of n elements. Any bijective application f , from $\mathbb{Z}/n\mathbb{Z}$ into A , provides A with a structure of group, where the group law $(+_f)$ and the inverse law $(-_f)$ are given by the following rules:

$$\forall(x, y) \in A^2, x +_f y = f(f^{-1}(x) + f^{-1}(y)) \text{ and } -_f x = f(-f^{-1}(x)). \quad (1)$$

We denote by A_f the set A together with the group structure given by f . Then $f(0)$ and $f(1)$ are respectively the neutral element and a generator of A_f .

1.3. Generic family of cyclic groups

Let $\mathcal{F}(A)$ be the set of all bijective applications f , from $\mathbb{Z}/n\mathbb{Z}$ into A . For any subset $S \subset \mathcal{F}(A)$, one can consider the family $A_S = \{A_f, f \in S\}$. This family is a set of representations of the same group $\mathbb{Z}/n\mathbb{Z}$ over the set A . In the following, we use the generic family of cyclic groups, corresponding to the definition given in [Sho97] and defined by

Definition 1 (Generic family of cyclic groups). Let $B(n)$ be the set of binary representations of integers in $\{0, \dots, n-1\}$. The family $B(n)_{\mathcal{F}(B(n))}$ is called generic family of cyclic groups of order n . The union over $n \in \mathbb{N}^*$ of the generic families of cyclic groups of order n is called the generic family of cyclic groups.

Using the square and multiply algorithm, the application $f : \mathbb{Z}/n\mathbb{Z} \rightarrow B(n)_f$ can be computed in at most $2 \log(n)$ group operations from the knowledge of $f(1)$. As $\log(n)$ is the complexity parameter, the data of $+_f$ is stronger than the data of f in the class of polynomial time algorithms.

In the opposite direction, the application f^{-1} is the canonical isomorphism (called l in Section 1.1) associated to the group $B(n)_f$ together with the generator $f(1)$. We have seen before that this application f^{-1} is exactly the discrete logarithm map in $B(n)_f$. Thus, when the computation of the discrete logarithm is supposed to be hard in this group, an algorithm can not efficiently compute the group law using the rule (1).

1.4. Representations of cyclic groups

Definition 2 (Family of representations of cyclic groups). Let L be a language over $\{0, 1\}$, i.e. a subset of $\{0, 1\}^*$. A family of representations of cyclic groups over this language L is the data of:

- a denumerable system of parameters, which can also be represented by an infinite set of binary strings Ω , together with a function $c : \Omega \rightarrow \mathbb{N}^*$ computable in polynomial time, such that $\forall N \in \mathbb{N}, \exists \alpha \in \Omega / c(\alpha) \geq N$,
- for each $\alpha \in \Omega$, a finite subset L_α of L , of size $c(\alpha)$, together with two laws, $+\alpha$ and $-\alpha$, computable in polynomial time, and two elements in L_α : 0_α and g_α . We require $(L_\alpha, +_\alpha)$ to be a cyclic group of order $c(\alpha)$ with group law $+\alpha$, inverse law $-\alpha$, and such that 0_α and g_α are respectively the zero element and a generator of this group. We suppose moreover that $\max\{|x|, x \in L_\alpha\} = O(\log_2(c(\alpha)))$.

On Generic Groups and Related Bilinear Problems

In the previous definition, we require that $(c(\alpha))_{\alpha \in \Omega}$ is not upper-bounded. This means that the family contains groups of arbitrarily large orders so that we can obtain an asymptotic complexity for an adversary using this family. We require moreover that the elements of a group in the family have short representations in the language L .

Example 1. Consider the family of all elliptic curves of prime order over fields \mathbb{F}_p where p is a prime number larger than 3. An elliptic curve E in this family can be represented by a 7-uple $(p, a_1, a_2, a_3, a_4, a_6, g)$ defining its Weierstrass equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ over \mathbb{F}_p and a generator g of $E(\mathbb{F}_p)$.

The function c computing $\#E(\mathbb{F}_p)$ from a parameter $\alpha = (p, a_1, a_2, a_3, a_4, a_6, g)$ is based on the polynomial-time algorithm due to Schoof. From the Hasse theorem, $|\#E(\mathbb{F}_p) - p - 1| \leq 2\sqrt{p}$. There are thus curves of arbitrary large sizes in this family. An element of $E(\mathbb{F}_p)$ can be coded as a pair $(x, y) \in (\mathbb{F}_p)^2$ verifying its Weierstrass equation. The group laws, the generator g and the neutral of $E(\mathbb{F}_p)$ are directly deduced from a parameter $\alpha = (p, a_1, a_2, a_3, a_4, a_6, g)$. From these properties, we deduce that the set of parameters $(p, a_1, a_2, a_3, a_4, a_6, g)$, together with the cyclic groups $\{(x, y) = \lambda g / \lambda \in \mathbb{N}\}$, is a family of representations of cyclic groups.

Example 2. According to Section 1.3, an element of the generic family of cyclic groups is defined by an order n and two laws, $+_f$ and $-_f$. To connect this family with the definition of family of representations of cyclic groups, we use the following presentation:

- the parameter associated to an element of the generic family of cyclic groups is an integer n , together with two elements of $B(n)$ corresponding to 0_f and g_f ; we have an obvious function $c : (n, 0_f, g_f) \mapsto n$, computable in polynomial time,
- the set of group elements associated to the parameter $(n, 0_f, g_f)$ is $B(n)$,
- two oracles, $+_f$ and $-_f$, are built (see Section 1.2) from a function f randomly chosen in $\mathcal{F}(B(n))$ such that $f(0) = 0_f$ and $f(1) = g_f$.

The difference with the definition of a family of representations of cyclic groups is that the group laws can not be immediately deduced from the parameter. These group laws are only given through oracles.

1.5. Some standard problems

We state now some standard problems for families of representations of cyclic groups:

Definition 3. Let $(\Omega, (L_\alpha)_{\alpha \in \Omega})$ be a family of representations of cyclic groups over a language L . In this family,

- an algorithm solving the discrete logarithm problem computes $\log_{g_\alpha}(x)$ in the group L_α from the inputs $\alpha \in \Omega$, $x \in L_\alpha$;
- an algorithm solving the Diffie-Hellman problem computes $\log_{g_\alpha}(x) \cdot y$ in the group L_α from the inputs $\alpha \in \Omega$, $(x, y) \in (L_\alpha)^2$;
- an algorithm solving the decisional Diffie-Hellman problem decides if z is $\log_{g_\alpha}(x) \cdot y$ in the group L_α from the inputs $\alpha \in \Omega$, $(x, y, z) \in (L_\alpha)^3$.

Following Example 2, we can extend these definitions to the generic family of cyclic groups. An algorithm solving one of these problems in the generic family of cyclic groups has the same input and output. The only restriction is that it must use oracles to access the group laws.

D. Lubicz, T. Sirvent

2. Case of cyclic groups with pairing

2.1. Perfect pairings

Let G be a cyclic group of order n . It comes with a canonical structure of \mathbb{Z} -module: $\forall(k, x) \in \mathbb{N} \times G, k.x = x + x + \dots + x$ and $(-k).x = -(k.x)$. We denote by \hat{G} the dual of G , i.e. the group of $\mathbb{Z}/n\mathbb{Z}$ -linear forms on G . A pairing is a $\mathbb{Z}/n\mathbb{Z}$ -bilinear map from a pair of cyclic groups into another cyclic group, where n is the common order of the three groups involved. There exists a pairing from $G \times \hat{G}$ into $\mathbb{Z}/n\mathbb{Z}$, called canonical pairing of G and defined by: $\forall(x, v) \in G \times \hat{G}, e_c(x, v) = v(x)$.

Let G, G' and G'' be three cyclic groups of order n . A pairing $e : G \times G' \rightarrow G''$ is said to be perfect if it is isomorphic as a pairing to the canonical pairing of G , i.e. there exists three isomorphisms $m : G \rightarrow G, m' : G' \rightarrow \hat{G}$ and $m'' : \mathbb{Z}/n\mathbb{Z} \rightarrow G''$ such that $\forall(x, y) \in G \times G', e(x, y) = m''(e_c(m(x), m'(y)))$.

$$\begin{array}{ccc} G \times G' & \xrightarrow{e} & G'' \\ m \downarrow & & \downarrow m' \\ G \times \hat{G} & \xrightarrow{e_c} & \mathbb{Z}/n\mathbb{Z} \\ & & \uparrow m'' \end{array}$$

Let A be a set of n elements. Let $(f, h) \in \mathcal{F}(A)^2$, we would like to describe all possible perfect pairings from $A_f \times A_f$ into A_h . By definition, such a pairing is deduced from three isomorphisms: m, m' and m'' .

$$\begin{array}{ccc} A_f \times A_f & \xrightarrow{e} & A_h \\ m \downarrow & & \downarrow m' \\ A_f \times \widehat{A_f} & \xrightarrow{e_c} & \mathbb{Z}/n\mathbb{Z} \\ & & \uparrow m'' \end{array}$$

Without loss of generality, we can fix $m = id, m'' = h$ and $m' = \hat{f}^{-1} \circ i \circ f^{-1}$, where i is any isomorphism from $\mathbb{Z}/n\mathbb{Z}$ into $\widehat{\mathbb{Z}/n\mathbb{Z}}$ is any isomorphism, and where \hat{f} is the dual isomorphism of f , i.e. $\hat{f} : v \in \widehat{G'} \mapsto v \circ f \in \hat{G}$. The description of a perfect pairing is then equivalent to the data of an isomorphism i .

Remark 1. The perfect pairing e is uniquely determined by the value of $e(f(1), f(1))$. The isomorphism i is then determined as well when $e(f(1), f(1))$ is fixed. Thus, for all $(f, h) \in \mathcal{F}(A)^2$, there exists a unique perfect pairing e such that $e(f(1), f(1)) = h(1)$.

In the following, we will only consider cyclic groups with perfect pairing e such that $e(f(1), f(1)) = h(1)$. The pairing defined by $(x, y) \in A_f^2 \mapsto h(f^{-1}(x) \cdot f^{-1}(y)) \in A_h$ verifies this property. Since it is a perfect pairing, it is the unique perfect pairing mentioned in the previous remark.

2.2. Generic family of cyclic groups with pairing

Like in Section 1.3, we consider a set A of n elements, and the set $\mathcal{F}(A)$ of all bijective applications from $\mathbb{Z}/n\mathbb{Z}$ into A . For any subset $S \subset \mathcal{F}(A)$, let $\mathcal{P}(A, S)$ be a family

On Generic Groups and Related Bilinear Problems

of two representations of groups and a pairing parametrized by the set $\{(f, h) \in S^2\}$. From a pair $(f, h) \in S^2$, we deduce the groups A_f and A_h following Equation (1). Moreover, we consider the perfect pairing e from $A_f \times A_f$ into A_h defined by $\forall (x, y) \in (A_f)^2$, $e(x, y) = h(f^{-1}(x) \cdot f^{-1}(y))$.

Definition 4 (Generic family of cyclic groups with pairing). Let $B(n)$ be the set of binary representations of integers in $\{0, \dots, n-1\}$. The family $\mathcal{P}(B(n), \mathcal{F}(B(n)))$ is called generic family of cyclic groups of order n with pairing. The union over $n \in \mathbb{N}^*$ of the generic families of cyclic groups of order n with pairing is called the generic family of cyclic groups with pairing.

Like previously, the morphisms f and h can be computed from the group laws in $B(n)_f$ and $B(n)_h$. In the opposite direction, if the discrete logarithm problem is assumed to be hard in these groups, the inverse maps f^{-1} and h^{-1} are not efficiently computable.

2.3. Representations of cyclic groups with pairing

Definition 5. Let L and M be two languages over $\{0, 1\}$. A family of representations of cyclic groups with pairing, over these languages L and M , is the data of: two families of representations of cyclic groups, $(\Gamma, (L_\gamma)_{\gamma \in \Gamma})$ over L and $(\Delta, (M_\delta)_{\delta \in \Delta})$ over M , a parameter space $\Omega \subset \Gamma \times \Delta$, and for all $\alpha = (\gamma, \delta) \in \Omega$, a perfect pairing e_α from $L_\gamma \times L_\gamma$ into M_δ , computable in polynomial time, such that $e_\alpha(g_\gamma, g_\gamma) = g_\delta$. When Ω_1 (respectively Ω_2) denotes the set of left (respectively right) parts of elements of Ω , such family is denoted by: $(\Omega, (L_\gamma)_{\gamma \in \Omega_1}, (M_\delta)_{\delta \in \Omega_2}, (e_\alpha)_{\alpha \in \Omega})$.

Example 3. We go on with the example of elliptic curves. Now, we consider an elliptic curve E defined over a finite field \mathbb{F}_q of characteristic p . Let $l \geq 2$ be an integer prime to p . Let k be the smallest integer such that l divides $q^k - 1$. Then \mathbb{F}_{q^k} is the smallest extension of \mathbb{F}_q which contains the l^{th} roots of unity. One can consider the Weil pairing $e_W : E[l] \times E[l] \rightarrow \mathbb{F}_{q^k}$ (see [Sil86]).

As the Weil pairing is a skew symmetric form on $E[l]$, one has to consider a maximal isotropic subgroup $G(E[l])$ of $E[l]$ and an isomorphism with its dual to obtain a modified bilinear non trivial pairing $\tilde{e} : G(E[l]) \times G(E[l]) \rightarrow \mathbb{F}_{q^k}$, such that if P is a generator of $G(E[l])$ then $\tilde{e}(P, P)$ is a generator of the group of l^{th} roots of units in \mathbb{F}_{q^k} .

This perfect pairing can be computed in probabilistic polynomial time using Miller's algorithm [CF05]. It is easy to see that one can obtain from this construction a family of representations of cyclic groups with pairing.

Example 4. We already saw in example 2 that the generic family of cyclic groups is close to a family of representations of cyclic groups over $\{0, 1\}^*$. In the same way, the generic family of cyclic groups with pairing can be seen as a family of representations of cyclic groups with pairing over the languages $\{0, 1\}^*$ and $\{0, 1\}^*$.

The generic family of cyclic groups is used twice as a family of representations of cyclic groups. The system of parameters is defined by the set of pairs $((n, 0_f, g_f), (n, 0_h, g_h))$ where n is in \mathbb{N}^* . Like the group laws, the pairing is given through an oracle, computed from the functions f and h used to build the cyclic groups, as explained in Section 2.2: $(x, y) \mapsto h(f^{-1}(x) \cdot f^{-1}(y))$.

D. Lubicz, T. Sirvent

2.4. Bilinear Diffie-Hellman problems

We can now state the bilinear Diffie-Hellman problems in this formalism:

Definition 6. Let $(\Omega, (L_\gamma)_{\gamma \in \Omega_1}, (M_\delta)_{\delta \in \Omega_2}, (e_\alpha)_{\alpha \in \Omega})$ be a family of representations of cyclic groups with pairing over two languages L and M . In this family,

- an algorithm solving the bilinear Diffie-Hellman problem computes the element $\log_{g_\gamma}(w) \cdot e_{(\gamma, \delta)}(x, y)$ in the group M_δ from the inputs $(\gamma, \delta) \in \Omega$, $(w, x, y) \in (L_\gamma)^3$;
- an algorithm solving the decisional bilinear Diffie-Hellman problem decides if z is $\log_{g_\gamma}(w) \cdot e_{(\gamma, \delta)}(x, y)$ in the group M_δ from the inputs $(\gamma, \delta) \in \Omega$, $(w, x, y) \in (L_\gamma)^3$, $z \in M_\delta$.

Like previously, an algorithm solving one of these problems in the generic family of cyclic groups with pairing has the same input and output. It must only use oracles to access the group laws and the pairing, as explained in Example 4.

From now on, in order to express time and space complexity, we choose the computational model of Turing Machines that have access to some oracles, like in [Pap94] pp. 36. We define the time cost of a call to an oracle as one unit of time.

We remark that there exists a straightforward polynomial reduction of the bilinear Diffie-Hellman problem over the discrete logarithm problem in L_γ , provided by the Square and Multiply algorithm. As a consequence, using the algorithm given by Pollard in [Pol78] and the method proposed by Pohlig and Hellman in [PH78], the bilinear Diffie-Hellman problem can be solved with a complexity in the order of \sqrt{p} where p is the largest prime divisor of $|L_\gamma|$. The following part states that there does not exist a better algorithm to solve the bilinear Diffie-Hellman problem in the generic family of representations of cyclic groups with pairing.

3. Complexity analysis

3.1. A general framework

In this section, we present a general framework in order to assess the difficulty of a problem in the generic family of cyclic groups with pairing. An algorithm \mathcal{A} solving a problem in this family is provided at the beginning with the following inputs: $n \in \mathbb{N}^*$, $(0_f, g_f, 0_h, g_h) \in (B(n))^4$, a r_0 -uple $(x_1, \dots, x_{r_0}) \in (B(n))^{r_0}$ and a s_0 -uple $(y_1, \dots, y_{s_0}) \in (B(n))^{s_0}$, where $r_0 + s_0$ is the number of parameters of the given problem and $(x_1, \dots, x_{r_0}, y_1, \dots, y_{s_0})$ defines the instance of the problem.

Moreover, \mathcal{A} has access to oracles computing the group laws $+_f$ and $+_h$, the inverse laws $-_f$ and $-_h$, and the pairing e . All these oracles are built using two bijections f and h randomly chosen in $\mathcal{F}(B(n))$, but not given to \mathcal{A} . The bijections f and h must be compatible with $(0_f, g_f, 0_h, g_h)$: $f(0) = 0_f$, $f(1) = g_f$, $g(0) = 0_h$, $g(1) = g_h$.

We suppose that the algorithm \mathcal{A} is a probabilistic Turing machine. We assess its running time by the number of calls to the group and pairing oracles. We want to measure the asymptotic behavior of the average success probability of \mathcal{A} as n goes to infinity, the probability being taken for a fixed n over the set of pairs $(f, h) \in \mathcal{F}(B(n))^2$.

On Generic Groups and Related Bilinear Problems

In order to analyze the algorithm \mathcal{A} , we maintain two series of lists, R and S , with values in $B(n) \times (\mathbb{Z}/n\mathbb{Z})(X_1, \dots, X_n, Y_1, \dots, Y_n)$ where $(\mathbb{Z}/n\mathbb{Z})(X_1, \dots, Y_n)$ is the field of rational functions in the variables $X_1, \dots, X_n, Y_1, \dots, Y_n$. The lists R_k and S_k represent the “knowledge” of \mathcal{A} after k queries to the oracles. The integers r_k and s_k index the number of variables used in the lists R_k and S_k . We set ρ_k and σ_k to the cardinalities of R_k and S_k . When \mathcal{A} makes a new call to an oracle, r_{k+1} , s_{k+1} , ρ_{k+1} , σ_{k+1} , R_{k+1} and S_{k+1} are initialized with the values of r_k , s_k , ρ_k , σ_k , R_k and S_k and updated as follows:

- in the case of a call $a +_f b$, $-_f a$ or $e(a, b)$, if the element a (resp. b) is not the second member of a pair in R_k , we increase r_{k+1} and ρ_{k+1} by one, we define $x_{r_{k+1}} = a$ (resp. b) and add $(x_{r_{k+1}}, X_{r_{k+1}})$ to R_{k+1} ,
- in the case of a call $a +_h b$ or $-_h a$, if the element a (resp. b) is not the second member of a pair in S_k , we increase s_{k+1} and σ_{k+1} by one, we define $y_{s_{k+1}} = a$ (resp. b) and add $(y_{s_{k+1}}, Y_{s_{k+1}})$ to S_{k+1} ,
- when c is a fresh answer to the call $a +_f b$ (resp. $-_f a$), the pair $(c, P_a + P_b)$ (resp. the pair $(c, -P_a)$) is added in R_{k+1} , where (a, P_a) and (b, P_b) are in R_k , and we increase ρ_{k+1} by one,
- when c is a fresh answer to the call $a +_h b$ (resp. $-_h a$), the pair $(c, P_a + P_b)$ (resp. the pair $(c, -P_a)$) is added in S_{k+1} , where (a, P_a) and (b, P_b) are in S_k , and we increase σ_{k+1} by one,
- when c is a fresh answer to the call $e(a, b)$, the pair $(c, P_a \cdot P_b)$ is added in S_{k+1} , where (a, P_a) and (b, P_b) are in R_k , and we increase σ_{k+1} by one.

We remark that the previous rules imply that $\rho_{k+1} + \sigma_{k+1} \leq \rho_k + \sigma_k + 3$.

Definition 7 (Compatibility). A pair of bijections $(f, h) \in \mathcal{F}(B(n))^2$ is said to be compatible with (R_k, S_k) if:

- $\forall (v_R, P_R) \in R_k$, $P_R(f^{-1}(x_1), \dots, f^{-1}(x_{r_k}), h^{-1}(y_1), \dots, h^{-1}(y_{s_k}))$ is defined and equal to $f^{-1}(v_R)$,
- $\forall (v_S, P_S) \in S_k$, $P_S(f^{-1}(x_1), \dots, f^{-1}(x_{r_k}), h^{-1}(y_1), \dots, h^{-1}(y_{s_k}))$ is defined and equal to $h^{-1}(v_S)$.

We consider the algorithm \mathcal{A} after k calls to oracles. By definition 7, the group and pairing oracles initialized with any pair $(f, h) \in \mathcal{F}(B(n))^2$ compatible with (R_k, S_k) would have produced the same answers to the calls of the algorithm \mathcal{A} . The basic idea behind definition 7 is that the algorithm \mathcal{A} , after k calls to the oracles, has no mean to distinguish the problems defined by two different pairs of bijections compatible with (R_k, S_k) .

In order to state and to prove our main lemma, we need to define the notions of collision and of coherence:

Definition 8 (Collision). We say that there is a collision in (R_k, S_k) if there exists some pair $((v_1, P_1), (v_2, P_2))$ in $(R_k)^2$ or in $(S_k)^2$ such that: $v_1 = v_2$ and $P_1 \neq P_2$.

Definition 9 (Coherence). A $(r+s)$ -uple $(\alpha_1, \dots, \alpha_r, \beta_1, \dots, \beta_s) \in (\mathbb{Z}/n\mathbb{Z})^{r+s}$ is said to be coherent with a set Π of rational functions in $(\mathbb{Z}/n\mathbb{Z})(X_1, \dots, X_r, Y_1, \dots, Y_s)$ if: $\forall P \in \Pi$, $P(\alpha_1, \dots, \alpha_r, \beta_1, \dots, \beta_s)$ is defined and $\forall (P_1, P_2) \in \Pi^2$, $(P_1 \neq P_2 \implies P_1(\alpha_1, \dots, \alpha_r, \beta_1, \dots, \beta_s) \neq P_2(\alpha_1, \dots, \alpha_r, \beta_1, \dots, \beta_s))$.

D. Lubicz, T. Sirvent

Lemma 1. *Suppose that $n = p^\lambda$ for p a prime number. For a fixed k , we consider a collision-free pair of lists (R_k, S_k) , and some $(k+1)^{\text{th}}$ call to an oracle. Writing all the elements of R_k and S_k in reduced form, let d_1 be the maximum degree of numerators of R_k and S_k , d_2 be the maximum degree of denominators of R_k and S_k , and let $d = d_1 + d_2$. When $\rho_k + \sigma_k < \sqrt{2p/3d}$, the probability over all pairs of bijections compatible with (R_k, S_k) that a new call leads to a pair of lists (R_{k+1}, S_{k+1}) with collision is bounded by*

$$\frac{6d(\rho_k + \sigma_k)}{2p - 3d(\rho_k + \sigma_k)^2}.$$

Proof. We consider the set \mathcal{C} of $(r_{k+1} + s_{k+1})$ -uples which are simultaneously coherent (see Definition 9) with the set of rational functions in R_k and the set of rational functions in S_k .

For all pair of bijections (f, h) compatible with (R_k, S_k) , the $(r_{k+1} + s_{k+1})$ -uple $(f^{-1}(x_1), \dots, f^{-1}(x_{r_{k+1}}), h^{-1}(y_1), \dots, h^{-1}(y_{s_{k+1}}))$ is in \mathcal{C} , since there is no collision in (R_k, S_k) . Reciprocally, for any $(\alpha_1, \dots, \alpha_{r_{k+1}}, \beta_1, \dots, \beta_{s_{k+1}})$ in \mathcal{C} , we can build compatible pairs of bijections (f, h) such that $f^{-1}(x_i) = \alpha_i$ and $h^{-1}(y_j) = \beta_j$. To these r_{k+1} fixed values for f and s_{k+1} fixed values for h , the compatibility condition adds $\rho_k - r_k$ other fixed values for f and $\sigma_k - s_k$ other fixed values for h . Then, we have exactly $(\#\mathcal{C})(n + r_k - r_{k+1} - \rho_k)!(n + s_k - s_{k+1} - \sigma_k)!$ pairs of bijections compatible with (R_k, S_k) .

Even if the sets R_{k+1} and S_{k+1} are not fully defined since the answer to the $(k+1)^{\text{th}}$ call is unknown, the rational functions in R_{k+1} and in S_{k+1} are already given by the new call. We can thus define the set \mathcal{C}' of $(r_{k+1} + s_{k+1})$ -uples which are simultaneously coherent with the set of rational functions in R_{k+1} and the set of rational functions in S_{k+1} . Following the same enumeration as previously, we have exactly $(\#\mathcal{C}')(n + r_k - r_{k+1} - \rho_k)!(n + s_k - s_{k+1} - \sigma_k)!$ pairs of bijections compatible with (R_k, S_k) and leading to some collision-free (R_{k+1}, S_{k+1}) .

The probability that the $(k+1)^{\text{th}}$ call to an oracle leads to a pair of lists (R_{k+1}, S_{k+1}) with collision, is then $(\#\mathcal{C} - \#\mathcal{C}')/\#\mathcal{C}$. As previously mentioned, the $(k+1)^{\text{th}}$ call to an oracle corresponds to at most three new pairs in R_{k+1} and in S_{k+1} with only one of them, whose polynomial is called P , resulting in a possible collision. If there is such a new pair in R_{k+1} , then $(\alpha_1, \dots, \alpha_{r_{k+1}}, \beta_1, \dots, \beta_{s_{k+1}})$ is simultaneously coherent with the set of rational functions in R_k , and not coherent with the set of rational functions in R_{k+1} : there is some $(v_R, P_R) \in R_k$ such that $(P - P_R)(\alpha_1, \dots, \alpha_{r_{k+1}}, \beta_1, \dots, \beta_{s_{k+1}}) = 0$. We can consider at most ρ_k differences $P - P_R$ of rational functions. Each one of these differences, in the reduced form, has a numerator with at most $(d \cdot n^{r_{k+1} + s_{k+1}}/p)$ roots, using [Sch80] Lemma 1. The number of $(r_{k+1} + s_{k+1})$ -uples coherent with R_k , and not coherent with R_{k+1} is then bounded by $(\rho_k \cdot d \cdot n^{r_{k+1} + s_{k+1}}/p)$. In the same way, the number of $(r_{k+1} + s_{k+1})$ -uples coherent with S_k , and not coherent with S_{k+1} is bounded by $(\sigma_k \cdot d \cdot n^{r_{k+1} + s_{k+1}}/p)$. The probability that the $(k+1)^{\text{th}}$ call to an oracle leads to a pair of lists (R_{k+1}, S_{k+1}) with collision, is then bounded by $d \cdot n^{r_{k+1} + s_{k+1}}(\rho_k + \sigma_k)/(p \cdot \#\mathcal{C})$.

Using again [Sch80] Lemma 1 over the $(\rho_k(\rho_k - 1) + \sigma_k(\sigma_k - 1))/2$ numerators of differences of rational functions in R_k or in S_k , we obtain:

On Generic Groups and Related Bilinear Problems

$$\#\mathcal{C} \geq \frac{n!}{(n - r_{k+1} - s_{k+1})!} - (\rho_k(\rho_k - 1) + \sigma_k(\sigma_k - 1)) \cdot \frac{d \cdot n^{r_{k+1} + s_{k+1}}}{2p}.$$

Since $r_{k+1} + s_{k+1} \leq \sqrt{n}$, using a straightforward computation, we get

$$\frac{n!}{(n - r_{k+1} - s_{k+1})!} \geq \frac{n^{r_{k+1} + s_{k+1}}}{3}.$$

We obtain then $p \cdot \#\mathcal{C}_k \geq n^{r_{k+1} + s_{k+1}} \cdot (p/3 - d(\rho_k + \sigma_k)^2/2)$. The probability that the $(k+1)$ th call to an oracle leads to a pair of lists (R_{k+1}, S_{k+1}) with collision is then bounded by:

$$\frac{6d(\rho_k + \sigma_k)}{2p - 3d(\rho_k + \sigma_k)^2}.$$

□

3.2. Illustration with the bilinear Diffie-Hellman problem

We now consider the bilinear Diffie-Hellman problem over the generic family of cyclic groups with pairing, described as a family of representations of cyclic groups with pairing in example 4.

In the two following corollaries, we assume that n is a prime power: $n = p^\lambda$. The case of a general composite order is considered later, in Theorem 1. Let \mathcal{A} be an algorithm solving the bilinear Diffie-Hellman problem. Its input is $n \in \mathbb{N}^*$, together with $(0_f, g_f, 0_h, g_h) \in B(n)^4$ and $(x_1, x_2, x_3) \in B(n)^3$. At the beginning, we set $R_0 = \{(0_f, 0), (g_f, 1), (x_1, X_1), (x_2, X_2), (x_3, X_3)\}$ and $S_0 = \{(0_h, 0), (g_h, 1)\}$.

Corollary 1. *For a fixed $n = p^\lambda$, let $k \leq (\sqrt{p/3} - 5)/3$. When R_k and S_k contain polynomials of degree at most equal to 2, the probability that the pair of lists (R_k, S_k) is with collision after k calls to the oracles is bounded by $2(3k+4)^2/(p-3(3k+4)^2)$, where the probability is computed over all pairs of bijections compatible with (R_k, S_k) .*

Proof. In each call to an oracle, at most 3 new pairs are added in $R_k \cup S_k$. We obtain then that for all i , $\rho_i + \sigma_i \leq 3i + 7$. We deduce that $6(\rho_i + \sigma_i)/(p - 3(\rho_i + \sigma_i)^2)$ is upper-bounded by $6(3i + 7)/(p - 3(3i + 7)^2)$.

The probability that the pair of lists (R_k, S_k) is collision-free after k calls to the oracles is equal to the product of probabilities that the pair of lists (R_{i+1}, S_{i+1}) is collision-free, knowing that (R_i, S_i) is collision-free, where $i \in \{0, \dots, k-1\}$. Since $0 < 6(3k+4) < p - 3(3k+4)^2$, we can give a lower bound for this probability of no-collision after k calls using Lemma 1:

$$\prod_{i=0}^{k-1} \left(1 - \frac{6(3i+7)}{p-3(3i+7)^2}\right) \geq \left(1 - \frac{6(3k+4)}{p-3(3k+4)^2}\right)^k \geq 1 - \frac{2(3k+4)^2}{p-3(3k+4)^2}.$$

The probability that the pair of lists (R_k, S_k) is with collision after k calls to the oracles is then bounded by: $2(3k+4)^2/(p-3(3k+4)^2)$. □

D. Lubicz, T. Sirvent

Corollary 2. *Let \mathcal{A} be an algorithm solving the bilinear Diffie-Hellman problem in the generic family of cyclic groups with pairing. When $k < (\sqrt{2p/9} - 7)/3$, the probability of success for \mathcal{A} after k calls to the oracles, over groups of size p^λ , when (R_k, S_k) is collision-free, is less than $18(3k + 7)/(2p - 9(3k + 7)^2)$.*

Proof. Let z be the answer given by the algorithm \mathcal{A} , after k calls to oracles leading to a collision-free pair of lists (R_k, S_k) .

If there is no P_z such that $(z, P_z) \in S_k$, then z is indistinguishable from any other element of $B(n)$ with the same property (i.e. not in a pair in S_k). Thus, the answer z given by the algorithm \mathcal{A} is valid with a probability less than $1/(p - \sigma_k) \leq 1/(p - 3k - 2)$.

Else, if the answer z is correct, a “virtual” call to an oracle corresponding to the polynomial $X_1.X_2.X_3$ would lead to a collision in S_{k+1} . We can then reuse Lemma 1 to obtain the following bound for the probability of such an event:

$$\frac{18(\rho_k + \sigma_k)}{2p - 9(\rho_k + \sigma_k)^2} \leq \frac{18(3k + 7)}{2p - 9(3k + 7)^2}.$$

The probability that \mathcal{A} solves the problem is then bounded by the maximum of the two previously given probabilities, which is obviously the second one. \square

We obtain then the following theorem:

Theorem 1. *Let \mathcal{A} be an algorithm solving the bilinear Diffie-Hellman problem in the generic family of representations of cyclic groups with pairing. \mathcal{A} is supposed to have unbound computational power, and be able to call the groups and pairing oracles in a probabilistic manner. After k calls to the oracles, over groups of order divisible by a prime number p , when $k < (\sqrt{2p/9} - 7)/3$, the probability that \mathcal{A} succeeds is bounded by:*

$$\frac{2(3k + 4)^2}{p - 3(3k + 4)^2} + \frac{18(3k + 7)}{2p - 9(3k + 7)^2}.$$

Proof. Let n be the common order of the groups. First suppose that $n = p^\lambda$ with p a prime number. An algorithm \mathcal{A} may output a valid answer after k calls to oracles in two cases: (R_k, S_k) is with collision, or collision-free. The first case is considered in Corollary 1, the second one is considered in Corollary 2.

We consider now the case of a general composite order: $n = p^\lambda q$, where p does not divide q . From any algorithm \mathcal{A}_n solving the bilinear Diffie-Hellman problem in groups of order n , we build an algorithm \mathcal{A}_{p^λ} solving the bilinear Diffie-Hellman problem in groups of order p^λ , with at least the same probability of success:

1. \mathcal{A}_{p^λ} randomly chooses two bijections $\phi_1, \phi_2 : B(p^\lambda) \times \mathbb{Z}/q\mathbb{Z} \rightarrow B(n)$, and 3 elements $\alpha_1, \alpha_2, \alpha_3$ in $\mathbb{Z}/q\mathbb{Z}$,
2. \mathcal{A}_{p^λ} obtains $(0_f, g_f, 0_h, g_h) \in B(p^\lambda)^4$, and $(x_1, x_2, x_3) \in B(p^\lambda)^3$,
3. \mathcal{A}_{p^λ} computes $0'_f = \phi_1(0_f, 0)$, $0'_h = \phi_2(0_h, 0)$, $g'_f = \phi_1(g_f, 1)$, $g'_h = \phi_2(g_h, 1)$, $x'_1 = \phi_1(x_1, \alpha_1)$, $x'_2 = \phi_1(x_2, \alpha_2)$, and $x'_3 = \phi_1(x_3, \alpha_3)$,
4. \mathcal{A}_{p^λ} uses \mathcal{A}_n with the following input: $(0'_f, g'_f, 0'_h, g'_h)$ and (x'_1, x'_2, x'_3) ,
5. when \mathcal{A}_n outputs $z' = \phi_2(z, \alpha)$, \mathcal{A}_{p^λ} outputs z .

On Generic Groups and Related Bilinear Problems

For step 1, we remark that there is a natural bijection $\phi : \mathbb{Z}/p^\lambda\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z} \rightarrow \mathbb{Z}/n\mathbb{Z}$ given by $\phi(a, b) = a + p^\lambda b$, so that this step boils down to choosing random permutations over $B(n)$.

\mathcal{A}_{p^λ} uses the oracles $+_f, -_f, +_h, -_h$, and e to build the following oracles:

- $+'_f : (\phi_1(a_1, b_1), \phi_1(a_2, b_2)) \in B(n)^2 \mapsto \phi_1(a_1 +_f a_2, b_1 + b_2) \in B(n)$,
- $-'_f : \phi_1(a, b) \in B(n) \mapsto \phi_1(-_f a, -b) \in B(n)$,
- $+'_h : (\phi_2(a_1, b_1), \phi_2(a_2, b_2)) \in B(n)^2 \mapsto \phi_2(a_1 +_h a_2, b_1 + b_2) \in B(n)$,
- $-'_h : \phi_2(a, b) \in B(n) \mapsto \phi_2(-_h a, -b) \in B(n)$,
- $e' : (\phi_1(a_1, b_1), \phi_1(a_2, b_2)) \in B(n)^2 \mapsto \phi_2(e(a_1, a_2), b_1 \cdot b_2) \in B(n)$.

The set $B(n)$ has a structure of group with the laws $+'_f$ and $-'_f$, where $0'_f$ is the neutral element, and g'_f is a generator. The set $B(n)$ has another structure of group with the laws $+'_h$ and $-'_h$, where $0'_h$ is the neutral element, and g'_h is a generator. e' is a perfect pairing between these groups such that: $e'(g'_f, g'_f) = g'_h$.

If $z' = \phi_2(z, \alpha)$ is the solution of the bilinear Diffie-Hellman problem given to \mathcal{A}_n , then z is the solution of the bilinear Diffie-Hellman problem given to \mathcal{A}_{p^λ} , and $\alpha_1 \alpha_2 \alpha_3 = \alpha$ in $\mathbb{Z}/q\mathbb{Z}$. The probability of success of \mathcal{A}_{p^λ} is then at least equal to the one of \mathcal{A}_n . \square

Remark 2. As an immediate consequence of this last theorem, if the number k of calls to oracles is very small against p , then the probability that \mathcal{A} succeeds is bounded by $O(k^2/p)$.

Remark 3. Using the polynomial reduction of the bilinear Diffie-Hellman problem over the discrete logarithm problem given in Section 2.3, the Theorem 1 is immediately transposed with the same success probability to the usual discrete logarithm problem in the generic family of cyclic groups with pairing. In a way, the pairing does not help to solve the discrete logarithm problem.

Remark 4. Since there is also a polynomial reduction of the bilinear Diffie-Hellman problem over the usual Diffie-Hellman problem, the Theorem 1 immediately implies the difficulty of the computational Diffie-Hellman problem in the generic family of cyclic groups with pairing even if it is proved in [Jou02] that the decisional Diffie-Hellman problem is easy in this family.

3.3. Other problems

In fact, the technique described in Section 3.1 is quite general. We illustrate its ubiquity by proving the difficulty of the q -BDHI problem, introduced in [BB04a].

Definition 10 (q -BDHI problem). Let $(\Omega, (L_\gamma)_{\gamma \in \Omega_1}, (M_\delta)_{\delta \in \Omega_2}, (e_\alpha)_{\alpha \in \Omega})$ be a family of representations of cyclic groups with pairing over two languages L and M , restricted to prime order groups. In this family, an algorithm solving the q -bilinear Diffie-Hellman inversion (q -BDHI) problem computes the element $(1/\nu) g_\delta$ in the group M_δ from the inputs $(\gamma, \delta) \in \Omega$, $(\nu g_\gamma, \nu^2 g_\gamma, \dots, \nu^q g_\gamma) \in (L_\gamma)^q$.

Let \mathcal{A} be an algorithm solving the q -BDHI problem in the generic family of cyclic groups with pairing. Its input is a prime number p , together with $(0_f, g_f, 0_h, g_h) \in B(p)^4$ and $(\nu g_f, \dots, \nu^q g_f) \in B(p)^q$ for $\nu \in (\mathbb{Z}/p\mathbb{Z})^*$. We initialize the lists $R_0 = \{(0_f, 0), (g_f, 1), (\nu g_f, X), (\nu^2 g_f, X^2), \dots, (\nu^q g_f, X^q)\}$ and $S_0 = \{(0_h, 0), (g_h, 1)\}$. This pair of lists is updated as described in Section 3.1.

D. Lubicz, T. Sirvent

Corollary 3. *For a fixed prime $n = p$, let $k \leq (\sqrt{p/3q} - q - 5)/3$. When R_k and S_k contain polynomials of degree at most equal to $2q$, the probability that the pair of lists (R_k, S_k) is with collision, after k calls to the oracles, is bounded by*

$$2q \cdot (3k + q + 1)^2 / (p - 3q(3k + q + 1)^2).$$

Proof. The proof is exactly the same as the one of corollary 1. Applying lemma 1, we obtain the following lower bound for the probability of collision

$$\begin{aligned} \prod_{i=0}^{k-1} \left(1 - \frac{6q(3i + q + 4)}{p - 3q(3i + q + 4)^2}\right) &\geq \left(1 - \frac{6q(3k + q + 1)}{p - 3q(3k + q + 1)^2}\right)^k \\ &\geq 1 - \frac{2q(3k + q + 1)^2}{p - 3q(3k + q + 1)^2}. \end{aligned}$$

□

Corollary 4. *Let \mathcal{A} be an algorithm solving the q -BDHI problem in the generic family of cyclic groups with pairing. When $k < (\sqrt{2p/(6q + 3)} - q - 4)/3$, the probability of success for \mathcal{A} after k calls to the oracles, over groups of prime size p , when (R_k, S_k) is collision-free, is less than*

$$\frac{6(2q + 1)(3k + q + 4)}{2p - 3(2q + 1)(3k + q + 4)^2}.$$

Proof. Let z be the answer given by the algorithm \mathcal{A} , after k calls to oracles leading to a collision-free pair of lists (R_k, S_k) . As in the proof of corollary 2, if there is no P_z such that $(z, P_z) \in S_k$ then the answer z is valid with a probability less than $1/(p - 3k - 2)$. Else, if z is a correct guess, a “virtual” call to the oracle corresponding to the rational function $1/X$ would produce a collision in S_{k+1} . The Lemma 1 gives the following bound for the probability of such an event

$$\frac{6(2q + 1)(\rho_k + \sigma_k)}{2p - 3(2q + 1)(\rho_k + \sigma_k)^2} \leq \frac{6(2q + 1)(3k + q + 4)}{2p - 3(2q + 1)(3k + q + 4)^2}.$$

This second bound is larger than the first one, which implies the result. □

From Corollaries 3 and 4, one can immediately deduce the following theorem.

Theorem 2. *Let \mathcal{A} be an algorithm solving the q -BDHI problem in the generic family of cyclic groups with pairing. \mathcal{A} is supposed to have unbound computational power, and able to call the groups and pairing oracles in a probabilistic manner. After k calls to the oracles, over groups of prime order p , when $k < (\sqrt{2p/(6q + 3)} - q - 4)/3$, the probability that \mathcal{A} succeeds is bounded by:*

$$\frac{2q(3k + q + 1)^2}{p - 3q(3k + q + 1)^2} + \frac{6(2q + 1)(3k + q + 4)}{2p - 3(2q + 1)(3k + q + 4)^2}.$$

4. Pseudo-random family of cyclic groups

In this part, we introduce the notion of pseudo-random family of cyclic groups. Naively speaking a pseudo-random family of groups is the same thing as a generic family of groups except that the group law is not drawn at random in the set of all possible group laws: the group law follows a specific distribution which is computationally indistinguishable from a uniform distribution. We build a pseudo-random family of cyclic groups from a strong pseudo-random family of permutations.

Let \mathfrak{P} be a set of permutations over $B(n)$. The notation $f \leftarrow \mathfrak{P}$ means that f is randomly and uniformly drawn in \mathfrak{P} . A distinguisher \mathcal{D} is a Turing machine which has access to permutations over $B(n)$ through oracles and outputs a single bit. In the context of strong indistinguishability between two families of permutations \mathfrak{P}_1 and \mathfrak{P}_2 (see [LR88] for more details), a distinguisher \mathcal{D} has access to f and f^{-1} , where $f \leftarrow \mathfrak{P}_1$ or $f \leftarrow \mathfrak{P}_2$. When \mathcal{D} runs in time t and makes q oracles queries, its advantage is defined by the following formula:

$$\text{Adv}_{\mathfrak{P}_1, \mathfrak{P}_2}^{\text{s-ppp}}(\mathcal{D}, t, q) = \left| \Pr_{f \leftarrow \mathfrak{P}_1} [\mathcal{D}_{t,q}^{f, f^{-1}} = 1] - \Pr_{f \leftarrow \mathfrak{P}_2} [\mathcal{D}_{t,q}^{f, f^{-1}} = 1] \right|.$$

We say that \mathfrak{P}_1 and \mathfrak{P}_2 are (ϵ, t, q) -strongly indistinguishable if for all distinguisher \mathcal{D} , the advantage $\text{Adv}_{\mathfrak{P}_1, \mathfrak{P}_2}^{\text{s-ppp}}(\mathcal{D}, t, q)$ is upper-bounded by ϵ . We say that \mathfrak{P} is a (ϵ, t, q) -strong pseudo-random family of permutations if it is (ϵ, t, q) -strongly indistinguishable from the set \mathfrak{S}_n of all permutations over $B(n)$.

Definition 11. Let \mathfrak{P} be a (ϵ, t, q) -strong pseudo-random family of permutations over $B(n)$. The pseudo-random family of cyclic groups associated to \mathfrak{P} is the set of groups defined by the permutations f in \mathfrak{P} , with neutral element $f(0)$, generator $f(1)$ and laws $+_f$ and $-_f$, as defined in Section 1.3.

Like in the generic family of cyclic groups, the group laws are given only through oracles. In this way, it is clear that a generic family of cyclic groups is a pseudo-random family of groups. As a consequence the notion of pseudo-random family of groups constitutes a generalization of the notion of generic family of cyclic groups.

We can now define the advantage of an adversary against a discrete-logarithm based problem in a pseudo-random family of cyclic groups: let \mathcal{A} be an adversary, which has access to group laws over $B(n)$ through oracles $+_f$ and $-_f$. When \mathcal{A} runs in time t and makes q oracle queries, its advantages over the discrete logarithm, Diffie-Hellman and decisional Diffie-Hellman problems are defined by:

$$\begin{aligned} \text{Adv}_{\mathfrak{P}}^{\text{DL}}(\mathcal{A}, t, q) &= \Pr_{f \leftarrow \mathfrak{P}, x \in B(n)} [\mathcal{A}_{t,q}^{+_f, -_f}(f(0), f(1), x) = \log_{f(1)}(x)], \\ \text{Adv}_{\mathfrak{P}}^{\text{DH}}(\mathcal{A}, t, q) &= \Pr_{f \leftarrow \mathfrak{P}, (x,y) \in B(n)^2} [\mathcal{A}_{t,q}^{+_f, -_f}(f(0), f(1), x, y) = \log_{f(1)}(x) \cdot y], \\ \text{Adv}_{\mathfrak{P}}^{\text{DDH}}(\mathcal{A}, t, q) &= \left| \Pr_{f \leftarrow \mathfrak{P}, (x,y) \in B(n)^2} [\mathcal{A}_{t,q}^{+_f, -_f}(f(0), f(1), x, y, \log_{f(1)}(x) \cdot y) = 1] \right. \\ &\quad \left. - \Pr_{f \leftarrow \mathfrak{P}, (x,y,z) \in B(n)^3} [\mathcal{A}_{t,q}^{+_f, -_f}(f(0), f(1), x, y, z) = 1] \right|. \end{aligned}$$

D. Lubicz, T. Sirvent

The maximum of these advantages over all adversaries \mathcal{A} are respectively denoted by $\text{Adv}_{\mathfrak{P}}^{\text{DL}}(t, q)$, $\text{Adv}_{\mathfrak{P}}^{\text{DH}}(t, q)$, and $\text{Adv}_{\mathfrak{P}}^{\text{DDH}}(t, q)$. Theorems 3 and 4 give bounds of these probabilities, when \mathfrak{P} is a (ϵ, t, q) -strong pseudo-random family of permutations over $B(n)$.

Theorem 3. *Let \mathfrak{P} be a (ϵ, t, q) -strong pseudo-random family of permutations over $B(n)$. Then,*

$$\begin{aligned} \text{Adv}_{\mathfrak{P}}^{\text{DL}}(t, q/3 - 1) &\leq \text{Adv}_{\mathfrak{S}_n}^{\text{DL}}(t, q/3 - 1) + \epsilon, \\ \text{Adv}_{\mathfrak{P}}^{\text{DH}}(t, q/3 - 2) &\leq \text{Adv}_{\mathfrak{S}_n}^{\text{DH}}(t, q/3 - 2) + \epsilon. \end{aligned}$$

Proof. Let \mathcal{A} be an adversary of the discrete logarithm problem over groups, with laws given by the way of oracles built from a permutation f drawn randomly in \mathfrak{P} . From this adversary \mathcal{A} , we deduce a distinguisher \mathcal{D} on the strong pseudo-random permutation family \mathfrak{P} . This distinguisher \mathcal{D} takes as input a permutation f over $B(n)$ and its inverse f^{-1} . This permutation f has been randomly chosen in the strong pseudo-random permutation family \mathfrak{P} or in the set \mathfrak{S}_n of all permutations.

From this permutation, the distinguisher \mathcal{D} builds an instance of the discrete logarithm problem for the adversary \mathcal{A} : it randomly chooses $r \in \mathbb{Z}/n\mathbb{Z}$ and gives $f(0)$, $f(1)$ and $f(r)$ to the adversary \mathcal{A} . The distinguisher \mathcal{D} builds the oracles corresponding to the group laws: $x +_f y = f(f^{-1}(x) + f^{-1}(y))$, $-_f x = f(-f^{-1}(x))$.

The adversary \mathcal{A} eventually outputs an answer to the discrete logarithm problem. If its answer is correct i.e. is equal to r , the distinguisher \mathcal{D} outputs 1, else it outputs 0.

If the permutation f has been chosen randomly in the strong pseudo-random permutation family \mathfrak{P} , the probability that the adversary \mathcal{A} outputs a correct answer is exactly its advantage for the discrete logarithm problem in the family of groups defined by \mathfrak{P} .

In the other case, if the permutation f has been chosen randomly in the set \mathfrak{S}_n of all permutations over $B(n)$, the probability that the adversary \mathcal{A} outputs a correct answer is exactly its advantage for the discrete logarithm problem in the generic family of groups.

The advantage of the distinguisher \mathcal{D} is then exactly the difference of advantages of \mathcal{A} in the two cases. Thus, if \mathcal{A} runs in time t and can issue at most $(q/3 - 1)$ group oracle queries, this advantage is bounded by ϵ , when \mathfrak{P} is a (ϵ, t, q) -strong pseudo-random permutation family. Thus, the advantage of an adversary \mathcal{A} against a discrete logarithm problem in the family of groups defined by \mathfrak{P} is bounded by the advantage of solving this problem in the generic family of groups, plus ϵ .

This proof can moreover be directly translated for the Diffie-Hellman problem. \square

Theorem 4. *Let \mathfrak{P} be a (ϵ, t, q) -strong pseudo-random family of permutations over $B(n)$. Then,*

$$\text{Adv}_{\mathfrak{P}}^{\text{DDH}}(t, q/3 - 2) \leq \text{Adv}_{\mathfrak{S}_n}^{\text{DDH}}(t, q/3 - 2) + 2\epsilon.$$

Proof. Let \mathcal{A} be an adversary against the decisional Diffie-Hellman problem in a family of groups where the law is provided by oracles built from a permutation randomly drawn either from the random family of permutations \mathfrak{S}_n or from a strong pseudo-random family of permutations \mathfrak{P} . Let b be the bit in the DDH problem which must be guessed by the adversary. We have

On Generic Groups and Related Bilinear Problems

$$\text{Adv}_{\mathfrak{P}}^{\text{DDH}}(\mathcal{A}, t, q) = \left| \Pr_{f \leftarrow \mathfrak{P}} [\mathcal{A}_{t,q}^{+f,-f} = 1 / b = 1] - \Pr_{f \leftarrow \mathfrak{P}} [\mathcal{A}_{t,q}^{+f,-f} = 1 / b = 0] \right|.$$

$$\text{Adv}_{\mathfrak{S}_n}^{\text{DDH}}(\mathcal{A}, t, q) = \left| \Pr_{f \leftarrow \mathfrak{S}_n} [\mathcal{A}_{t,q}^{+f,-f} = 1 / b = 1] - \Pr_{f \leftarrow \mathfrak{S}_n} [\mathcal{A}_{t,q}^{+f,-f} = 1 / b = 0] \right|.$$

From \mathcal{A} , we build a distinguisher \mathcal{D} against the strong pseudo-random family of permutations. This distinguisher has access to an oracle permutation f and its inverse f^{-1} drawn either from \mathfrak{P} or \mathfrak{S}_n . Using the oracles f and f^{-1} , \mathcal{D} can reply to the group law queries issued by \mathcal{A} . It draws at random a bit b and two elements x, y in $B(n)$. If $b = 1$ it submits to \mathcal{A} the input $(f(0), f(1), x, y, f[f^{-1}(x) \cdot f^{-1}(y)])$ and if $b = 0$, it draws a random $z \in B(n)$ and submits to \mathcal{A} the input $(f(0), f(1), x, y, z)$.

At the end of the game \mathcal{A} returns a bit b' . The distinguisher \mathcal{D} returns 1 if $b' = b$ and 0 if $b' \neq b$. We have by definition

$$\text{Adv}_{\mathfrak{P}, \mathfrak{S}_n}^{\text{s-PRP}}(\mathcal{D}, t, q) = \left| \Pr_{f \leftarrow \mathfrak{P}} [\mathcal{D}_{t,q}^{f,f^{-1}} = 1] - \Pr_{f \leftarrow \mathfrak{S}_n} [\mathcal{D}_{t,q}^{f,f^{-1}} = 1] \right|.$$

With this distinguisher, this means

$$\begin{aligned} \text{Adv}_{\mathfrak{P}, \mathfrak{S}_n}^{\text{s-PRP}}(\mathcal{D}, t, q) &= \left| \Pr_{f \leftarrow \mathfrak{P}} [\mathcal{A}_{t,q/3-2}^{+f,-f} = 1 / b = 1] + \Pr_{f \leftarrow \mathfrak{P}} [\mathcal{A}_{t,q/3-2}^{+f,-f} = 0 / b = 0] \right. \\ &\quad \left. - \Pr_{f \leftarrow \mathfrak{S}_n} [\mathcal{A}_{t,q/3-2}^{+f,-f} = 1 / b = 1] - \Pr_{f \leftarrow \mathfrak{S}_n} [\mathcal{A}_{t,q/3-2}^{+f,-f} = 0 / b = 0] \right| / 2. \end{aligned}$$

As a consequence,

$$\text{Adv}_{\mathfrak{P}, \mathfrak{S}_n}^{\text{s-PRP}}(\mathcal{D}, t, q) \geq \left| \text{Adv}_{\mathfrak{P}}^{\text{DDH}}(\mathcal{A}, t, q/3 - 2) - \text{Adv}_{\mathfrak{S}_n}^{\text{DDH}}(\mathcal{A}, t, q/3 - 2) \right| / 2.$$

And thus, $\text{Adv}_{\mathfrak{P}}^{\text{DDH}}(\mathcal{A}, t, q/3 - 2) \leq \text{Adv}_{\mathfrak{S}_n}^{\text{DDH}}(\mathcal{A}, t, q/3 - 2) + 2 \text{Adv}_{\mathfrak{P}, \mathfrak{S}_n}^{\text{s-PRP}}(\mathcal{D}, t, q)$. \square

Remark 5. Theorems 3 and 4 and their proofs should serve as an illustration. Using the same kind of methods, one should be able to prove any reasonable assumption in a pseudo-random family of cyclic groups. It is moreover possible to define pseudo-random families of cyclic groups with pairing from two strong pseudo-random families of permutations: bilinear assumptions can then be proved in this pseudo-random context.

In the two preceding proofs, the permutation is fixed at the beginning of the game in the reduction. This is something essential if the permutation is drawn from the pseudo-random family. Actually in that case it is not possible to build this pseudo-random permutation during the game by returning a random bit string to any fresh query. This fact contrasts with the usual generic group model and illustrates an important feature of our presentation of the generic group model where the group law is fixed at the beginning of the game.

There is some other subtle differences between the generic group model and the pseudo-random group model. For instance, it was mentioned in [MF07] that the generic model of groups should be able to take into account the following behavior of the adver-

D. Lubicz, T. Sirvent

sary \mathcal{A} . Suppose that \mathcal{A} receives a bit string $x \in B(n)$ from a group oracle query then \mathcal{A} flips some bit, for instance the least significant bit, and use the resulting bit string in order to submit a new query. This kind of queries is not covered in the original model [Sho97] but it is easy to be convinced that these queries using fresh bit strings do not help to solve a given problem in the generic group model and this is what we prove in Part 3. In contrast in the pseudo-random group model this kind of queries may be used to attack the underlying pseudo-random permutation family and it is an important point to take them into account.

It should also be stressed that the pseudo-random groups problem depends on the computational power of the adversary. This situation contrasts with the usual generic group model where all the results are of information theoretic nature.

5. Conclusion

We have revisited the notion of generic group in order to describe a model which contains all the features already seen in the literature: the ability to submit fresh bit strings and queries which correspond to rational functions of the exponents are worth mentioning. We have proved a bound on the problem of finding a collision in this model (Lemma 1). From this first bound, it is easy to derive precise bounds for all the usual discrete logarithm related problems: we presented some examples (Theorems 1 and 2) to explain how to use our framework in a systematic manner.

From this model, it is possible to derive the notion of pseudo-random groups and to prove a reduction of some usual problems in the pseudo-random group model to their security in the generic group model and to the strong pseudo-random permutation hypothesis. As a matter of fact, because of the reduction that we prove, the pseudo-random group model does not bring new security insight. Still it constitutes an improvement because it is more realistic than the usual generic group model. Nevertheless it is still not really satisfying. Actually, in the family of groups of interest in cryptography, the permutation on the underlying sets induced by the different group laws is far from being pseudo-random. For instance, if the family of groups is the family of multiplicative elements of finite field, for a given cardinality n there is only $\phi(n)$ possible permutations, where ϕ is Euler's totient function. It should be interesting to generalize further the notion of pseudo-random groups in order to make it more realistic.

References

- [BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 223–238. Springer, Berlin, 2004.
- [BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 56–73. Springer, Berlin, 2004.
- [Bro05a] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. In *Designs, Codes and Cryptography*, volume 35, pages 119–152. 2005.
- [Bro05b] Daniel R. L. Brown. On the provable security of ECDSA. In I. Blake and G. Seroussi, editors, *Advances in Elliptic Curve Cryptography*, pages 21–40. Cambridge University Press, 2005.

On Generic Groups and Related Bilinear Problems

- [CF05] Henri Cohen and Gerhard Frey. *Handbook of elliptic and hyperelliptic curve cryptography*. Chapman and Hall/CRC, 2005.
- [Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In *Advances in cryptology—ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Comput. Sci.*, pages 100–109. Springer, Berlin, 2002.
- [DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, IT-22(6):644–654, 1976.
- [ElG85a] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology—CRYPTO 1984*, volume 196 of *Lecture Notes in Comput. Sci.*, pages 10–18. Springer, Berlin, 1985.
- [ElG85b] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, IT-31(4):469–472, 1985.
- [Jou00] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. In *Algorithmic number theory*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 385–393. Springer, Berlin, 2000.
- [Jou02] Antoine Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *Algorithmic number theory*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 20–32. Springer, Berlin, 2002.
- [KM07] Neal Koblitz and Alfred Menezes. Another look at generic groups. *Adv. Math. Commun.*, 1(1):13–28, 2007.
- [LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
- [LS08] David Lubicz and Thomas Sirvent. Attribute-based broadcast encryption scheme made efficient. In *Advances in cryptology—AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Comput. Sci.*, to appear. Springer, Berlin, 2008.
- [MF07] Abe Masayuki and Serge Fehr. Perfect NIZK with adaptive soundness. In *Theory of Cryptography Conference, TCC 2007*, volume 4392 of *Lecture Notes in Comput. Sci.*, pages 118–136. Springer, Berlin, 2007.
- [MTI86] Tsutomu Matsumoto, Youichi Takashima, and Hideki Imai. On seeking smart public-key-distribution systems. *The Transactions of the IECE of Japan*, E69:99–106, 1986.
- [MOV01] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. Chapman and Hall/CRC, 2001.
- [Nec94] V. I. Nechaev. On the complexity of a deterministic algorithm for a discrete logarithm. *Mat. Zametki*, 55(2):91–101, 189, 1994.
- [NSS04] David Naccache, Nigel Smart, and Jacques Stern. Projective coordinates leak. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 257–267. Springer, Berlin, 2004.
- [FIPS186] FIPS 186. Digital signature standard. Federal Information Processing Standards Publication 186, U.S. Dept. of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.
- [Pap94] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.
- [PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. Information Theory*, IT-24(1):106–110, 1978.
- [Pol78] John M. Pollard. Monte Carlo methods for index computation (mod p). *Math. Comp.*, 32(143):918–924, 1978.
- [Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.*, 27(4):701–717, 1980.
- [Sch91] Claus P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 256–266. Springer, Berlin, 1997.
- [Sil86] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.
- [SPMS02] Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel Smart. Flaws in applying proof

D. Lubicz, T. Sirvent

methodologies to signature schemes. In *Advances in cryptology—CRYPTO 2002*, volume 2442 of *Lecture Notes in Comput. Sci.*, pages 93–110. Springer, Berlin, 2002.

- [YW05] Tsz Hon Yuen and Victor K. Wei. Fast and proven secure blind identity-based signcryption from pairings. In *Topics in cryptology—CT-RSA 2005*, volume 3376 of *Lecture Notes in Comput. Sci.*, pages 305–322. Springer, Berlin, 2005.