Automated Analysis of Cryptographic Assumptions in Generic Group Models

Gilles Barthe[1], Edvard Fagerholm[1,2], Dario Fiore[1], John Mitchell[3], Andre Scedrov[2], and Benedikt Schmidt[1]

[1] IMDEA Software Institute, Madrid, Spain
{gilles.barthe, dario.fiore, benedikt.schmidt}@imdea.org
[2] University of Pennsylvania, USA
{edvardf,scedrov}@math.upenn.edu
[3] Stanford University, USA
mitchell@cs.stanford.edu

Abstract. We initiate the study of principled, automated, methods for analyzing hardness assumptions in generic group models, following the approach of symbolic cryptography. We start by defining a broad class of generic and symbolic group models for different settings—symmetric or asymmetric (leveled) k-linear groups—and by proving "computational soundness" theorems for the symbolic models. Based on this result, we formulate a very general master theorem that formally relates the hardness of a (possibly interactive) assumption in these models to solving problems in polynomial algebra. Then, we systematically analyze these problems. We identify different classes of assumptions and obtain decidability and undecidability results. Then, we develop and implement automated procedures for verifying the conditions of master theorems, and thus the validity of hardness assumptions in generic group models. The concrete outcome of this work is an automated tool which takes as input the statement of an assumption, and outputs either a proof of its generic hardness or shows an algebraic attack against the assumption.

1 Introduction

Sophisticated abstractions have often been instrumental in recent breakthroughs in the design of cryptographic schemes. Bilinear maps are perhaps the most striking instance of such an abstraction; over the last fifteen years, they have been used for building advanced and previously unknown cryptographic schemes. Now it is believed that multilinear maps will lead to similar breakthroughs. Compared to the "classical" algebraic settings based on the purported hardness of the Factoring/RSA or Discrete-log/Diffie-Hellman problems, bilinear and multilinear maps indeed provide richer and more versatile algebraic structures that are particularly suitable for new constructions. At the same time, one unsettling consequence of using such sophisticated abstractions is a significant growth in the number of hardness assumptions used in security proofs. Moreover, these assumptions are not as well studied as their classical and standard counterparts.

While it is widely acknowledged that this situation is far from ideal, relying on non-standard assumptions is sometimes the *only* known way to construct some new (or some efficient) cryptographic scheme, and hence it cannot be completely disregarded. A common view to resolving this dilemma is to develop principled, rigorous approaches for analyzing and comparing non-standard hardness assumptions.

This question has been previously considered in the literature, in which we identify at least two approaches. One approach is to devise assumptions that are general enough to be reused and allow for simple security proofs, and at the same time are shown to hold under more classical assumptions (e.g., [14,31]). A second approach is to develop idealized models, such as the Generic Group [30,32,27] and the Generic Bilinear Group [9] models, and to provide (in the form of so-called master theorems) necessary and sufficient conditions for the security of an assumption in these models. Proving the hardness of an assumption in these models is essentially a way to rule out the possibility of algebraic attacks against the underlying algorithmic problem, and it can be considered the minimal level of guarantee we need to gain confidence in an assumption. Two prominent examples along this direction are the "Uber assumption" (aka "Master theorem") of Boneh, Boyen and Goh [9,13] and the Matrix Decisional Diffie-Hellman assumption family recently proposed by Escala et al. [16].

However, although these results are quite general, they can be quite difficult to apply. Indeed, in order to argue the hardness of an assumption using the Uber assumption in [9,13] (resp. the Matrix-DDH assumption in [16]) one has to show the independence (resp. irreducibility) of certain polynomials contained in the statement of the assumption. A similar problem arises in the context of interactive assumptions such as [26,2], in which the hardness crucially relies on the restrictions posed on the queries performed by the adversary. In summary, applying these general results to verify the validity of a given assumption is far from being a trivial task, and may be error-prone, as witnessed by unfortunate failures [34,22].

In this paper, we initiate the study of principled, automated methods for analyzing hardness assumptions in generic group models. Our main contribution is essentially threefold. First, we reformulate master theorems in the style of the celebrated "computational soundness" theorem of Abadi and Rogaway [1], and formally show that the problem of analyzing assumptions in the generic group reduces to solving problems in polynomial algebra. Second, we systematically analyze these problems: while we show that the most general problem is undecidable, we distill a set of properties (capturing most interesting cases) for which the problem is decidable. Finally, by applying tools from linear algebra, we develop and implement automated procedures for verifying the conditions of master theorems, and thus the validity of hardness assumptions in generic group models. The concrete outcome of this work is an automated tool[4] which takes as input an assumption and outputs either a proof of its generic hardness (along with concrete bounds) or shows an algebraic attack against the assumption.

[4] The tool is available at `http://www.easycrypt.info/GGA`

1.1 An Overview of Our Contribution

The key contribution of our work is the development of automated decision procedures for testing the validity of hardness assumptions in generic group models. Towards this goal, we first settle a rigorous framework for carrying out this analysis. Basically, this framework consists of formalizing a class of generic group models and then stating a general master theorem. Finally, our decision procedures will be aimed at verifying the side conditions of our master theorem.

Generic Group Models. We formalize a broad class of generic group models capturing many interesting cases used in cryptography: symmetric and asymmetric k-linear groups, with both leveled and non-leveled maps, and with the possibility of modeling efficiently computable isomorphisms between the groups. For any experiment stated in these generic models, we generalize the commonly-used step of applying the Schwartz-Zippel Lemma, and obtain a generic transformation (cf. Theorem 1) for switching from the generic group model experiment, in which variables are uniformly sampled in the underlying field, to a completely deterministic experiment that works in a corresponding symbolic group model.

A General Master Theorem. We give a general version of the Master theorem in [9] which can be stated in any of the generic group models mentioned above. As in [9], we formulate an assumption as a list $\boldsymbol{L}$ of polynomials in $\mathbb{F}_p[X_1, \ldots, X_n]$ where $X_1, \ldots, X_n$ is a set of random variables. In particular, a decisional (aka left-or-right) assumption is defined by two lists of polynomials $\boldsymbol{L}$ and $\boldsymbol{L}'$ (one for the "left" and one for the "right" distribution), and the assumption is said to hold if the adversary cannot distinguish whether it receives polynomials from $\boldsymbol{L}$ or $\boldsymbol{L}'$. Very informally, our Master theorem states that viewing $\boldsymbol{L}$ and $\boldsymbol{L}'$ as the generating sets of two vector spaces[5], then the linear dependencies within $\boldsymbol{L}$ and within $\boldsymbol{L}'$ are the same. Previous master theorems [9,16] considered only decisional assumptions with the real-or-random formulation in which the adversary is given a list of polynomials $\boldsymbol{L}$ and either a "challenge" polynomial f or a fresh random variable Z. Beyond obtaining a theorem that works in (leveled) k-linear groups, our general formulation allows us to capture virtually all decisional assumptions, based on k-linear groups (for any $k \geq 1$), that are used in cryptography. To mention some examples, assumptions captured by our theorem include the Matrix-DDH assumption [16], the k-BDH assumption [4], and recently proposed assumptions such as (n, k)-MMDHE [21].

Automated Methods. Once we have settled the above framework, our goal is to develop a collection of automated methods to verify the side condition of the Master theorem for any given assumption stated in the framework. While the statement of the above side condition already suggests how to use linear algebra to make these checks, a crucial challenge is that in many important cases (e.g., ℓ-BDHI, k-Lin, etc.) the size of the lists $\boldsymbol{L}$ and $\boldsymbol{L}'$ is a variable parameter. That

[5] We are oversimplifying. More precisely, one has to consider lists C and C' containing all polynomials computable by doing multiplications over $\boldsymbol{L}$ and $\boldsymbol{L}'$ respectively, and then look at linear dependencies in C and C'.

Assumption Type	Algorithm	Examples
Non-parametric	D, C	DBDH [11], 2-lin, 3-lin, Freeman assm. 3&4 [17]
Parametric (real-or-random, monomials inputs)		
Fixed #vars, Par. linear degree and Par. arity	U, I	(ℓ, k)-MMDHE [21]
Fixed #vars, Par. linear degree, Fixed arity	D, C	ℓ-DHI [8], ℓ-DHE [12]
Parametric #vars, Par. arity, Fixed degree	I	(k)-BDH [4], k-Lin in k-linear groups
Interactive bounded	I,C	LRSW [26], CDDH 1&2 [2], M-LRSW [6], IBSAS-CDH [7]
Interactive unbounded	I	LRSW [26], Strong-LRSW [3], s-LRSW [19]

Fig. 1. Summary of our automated analysis methods. U=undecidable problem, D=decision procedure, I = incomplete procedure, C=find counterexample for invalid assumptions.

is, to check that the side condition holds, one would have to do computations on a vector space of variable dimension: a challenging problem for automation.

We study this problem for three main categories of hardness assumptions: (1) non-parametric, (2) parametric, and (3) interactive. *Non-parametric* assumptions are non-interactive assumptions in which the number of inputs is fixed, no input is quantified over a variable and the number of levels is fixed (examples include DDH, DBDH [11], as well as assumptions in k-linear groups for fixed k, e.g., 3-Lin in 3-linear groups). Conversely, an assumption is *parametric* if one or more of the above restrictions do not hold. Finally, *interactive* assumptions are those ones where the adversary is granted access to additional oracles (in addition to the oracles for the algebraic operations). By carefully analyzing each of these categories, we obtain the following results summarized in Fig. 1.

For non-parametric assumptions, we show how to reduce the check on the side condition to computing the kernels of certain matrices (of fixed dimension) that are derived from the lists of polynomials in the assumption's definition. Using computer algebra tools (SAGE [33]), we implement a decision procedure that shows a concrete hardness bound in the corresponding generic group model in the positive case, and an algebraic attack if the assumption does not hold.

Our methods for non-parametric assumptions offer a complete decision procedure to verify arbitrary instances of parametric assumptions where all the parameters have been fixed. This might be sufficient to test quickly a new assumption (and find attacks if any), but it is often desirable to obtain stronger guarantees that hold for *all* parameters. We show that, contrary to the non-parametric case, the side condition becomes undecidable in general. However, we identify classes of assumptions for which we develop automated methods. Interestingly, these classes still contain most cryptographic assumptions. Considering the class of real-or-random assumptions, we develop two different methods. The first method focuses on the case in which the number of random variables is fixed, and the input elements are monomials. Our method shows how to reduce the check of the side condition to an integer programming problem. Interestingly, we can show the following: if the degree of the monomials is *not* a linear polynomial, or the arity of the map is variable, then the problem is *undecidable*; otherwise (if the monomials have linear degree and the arity of the map is fixed) the problem is decidable. We implemented the translation procedure to integer programming problems and use SMT solvers to check satisfiability. For the decidable fragment

of assumptions mentioned above, we obtain a complete decision procedure that also shows an attack if the assumption is invalid. For the undecidable fragment, our procedure successfully analyzes all significant examples from the literature.

Our second method focuses on the case where the number of random variables is parametric. As in the previous case, our method provides a way to reduce the side condition to a system of equations. However, the same idea as before does not work since a parametric number of variables would lead to an infinite number of equations. Therefore, we focus on a restricted, but significant, class of assumptions (one restriction is that inputs are expressed as monomials). Our method is incomplete but successfully analyzes all relevant examples in this class.

Finally, we study interactive assumptions such as LRSW [26]. To analyze interactive assumptions, we first formulate an interactive version of our master theorem. Interestingly, once applying our general "computational soundness" theorem and switching to the symbolic model, our interactive master theorem essentially becomes a variant of the non-interactive master theorem for parametric computational assumptions. This allow us to apply similar techniques as for parametric assumptions. More specifically, we use SMT solvers and Gröbner bases computations as an incomplete method to show the validity of such assumptions and find attacks. For instance, our tool automatically proves the validity of LRSW [26] and exhibits attacks for m-LRSW [6] and CDDH [2].

Extensions and Additional Material. We extend our results to composite-order groups. Precisely, we formulate the generic group model and our master theorem in a general way that captures also composite-order groups, and we show how to extend our decision procedures for non-parametric assumptions to this setting. Another extension of our results is handling assumptions in which the adversary receives rational values in the exponent. These extensions, full detailed proofs and some running examples appear only in the full version.

Limitations. While our master theorem is very general, our automated methods require to specify the assumptions in a concrete language, essentially to describe the distribution of the polynomials defining the assumption. Such language cannot support the expression of very abstract properties, and thus rules out a few examples. For instance, the definition of the Decision Multilinear No-Exact-Cover Assumption [18] is parametrized by an instance (with no solution) of the Exact-Cover NP-complete problem. Although fixing a specific Exact-Cover instance yields lists of polynomials which can be analyzed using our methods, a definition *for any instance* is too general. For a similar reason, our tool cannot handle the Matrix-DDH assumption in its full generality, unless one fixes a specific distribution for the matrix (e.g., k-Lin).

Discussion. Although well-studied standard assumptions should always be preferred when designing cryptographic schemes, the use of non-standard ones is not likely to stop. In this sense, we believe the study and development of rigorous methods for analyzing cryptographic assumptions is relevant, and that automated analysis tools can support cryptographers in multiple directions. Mainly, they provide a rigorous, fast way to test the validity of candidate assumptions in generic models by delegating this task to a machine. This is especially relevant

in the recent setting of leveled multilinear maps, that have a rich algebraic structure and for which even simple assumptions may become difficult to analyze. We believe that the importance of such tools is motivated by the fact that proofs validating the hardness of an assumption in the generic group model fall exactly in the so-called "mundane part"[6] of cryptographic proofs mentioned by Halevi [20], and constitute a perfect candidate of a proof to be delegated to a machine.

Our work shows the feasibility and relevance of developing automated methods to analyze assumptions in generic group models. It can also be seen as the first step towards analyzing cryptographic protocols directly in the generic model; we expect that such analyses would allow to discover subtle flaws in protocols and supplant existing methods based on symbolic cryptography.

1.2 Related Work

The problem of analyzing and comparing hardness assumptions has been earlier considered in the literature, e.g., [29]. In particular, we identify two main approaches in previous work. The first approach aims to define generalized assumptions that reduce to standard ones. Examples of works in this direction include: the Square Diffie-Hellman assumption, shown to be equivalent to CDH by Maurer and Wolf [28]; the (P, Q)-Decisional Diffie-Hellman assumption of Bresson et al. [14] which is shown to reduce to DDH; and the decisional subspace problems of Okamoto-Takashima [31] that are reduced to DLin.

The other approach aims at directly analyzing assumptions by means of idealized models, such as the generic group model. This model was introduced by Nechaev [30] and further refined and generalized by Shoup [32], and Maurer [27]. Our work follows closely Maurer's model, in which the main difference compared to previous proposals is to model the adversary's access to group elements via handles instead of random bitstrings as in [30,32]. These two models have been proven equivalent in [24]. Worth mentioning in this context is the semi-generic group model of Jager and Rupp [23]. This is a weaker version of the bilinear generic group model, and its basic idea is to model the base groups of pairings as generic groups, whereas the target group is given in the standard model.

Two works that address the problem of devising general assumptions in the generic group are the Master theorem of Boneh, Boyen and Goh [9] (generalized by Boyen [13]), and the Matrix DDH assumption of Escala et al. [16]. Roughly speaking, the former provides a framework for arguing about the validity of several pairing-based assumptions in the generic group model, and it captures a significant fraction of assumptions in the literature. The latter is an assumption that subsumes classical problems like DDH or DLin and also introduces assumptions, such as k-Casc, that are proven hard in the generic k-linear group

[6] In [20], Halevi informally divides proofs in two categories (quoting): "*Most (or all) cryptographic proofs have a creative part (e.g., describing the simulator or the reduction) and a mundane part (e.g., checking that the reduction actually goes through). It often happens that the mundane parts are much harder to write and verify, and it is with these parts that we can hope to have automated help.*"

model. Also worth mentioning is the work of Freeman [17] which extends the BBG Master theorem to challenges in the source group and uses the computer algebra system Magma to verify the side conditions required to prove two of the assumptions. Our work is also close to the line of work on automation of cryptographic proofs in both the computational and symbolic models, see [5] for an overview.

1.3 Preliminaries

In our work, we denote by λ the security parameter. We use $\mathbb{G}_i$ to denote additive cyclic groups of prime order and P_i to denote a generator of $\mathbb{G}_i$. For any element $Q = xP_i$, we denote with $x = dlog(Q)$ its discrete logarithm. We use $\boldsymbol{a}$ or $\boldsymbol{v}$ to denote vectors, $\boldsymbol{a}\|\boldsymbol{b}$ for the concatenation of two vectors, and $\boldsymbol{a} \cdot \boldsymbol{b}$ to denote their inner product. We denote the power set of S with $\mathcal{P}(S)$, the i-th element of a list with $L[i]$, the range $\{n, \ldots, n+l\}$ with $[n, n+l]$, and $[1, n]$ with $[n]$.

A *symmetric k-linear group* is a pair of groups $\mathbb{G}_1$ and $\mathbb{G}_2$ together with an admissible k-linear map $e : \mathbb{G}_1^k \to \mathbb{G}_2$. An *asymmetric k-linear group* is a sequence of groups $\mathbb{G}_1, \ldots, \mathbb{G}_k, \mathbb{G}_{k+1}$ together with an admissible k-linear map $e : \mathbb{G}_1 \times \cdots \times \mathbb{G}_k \to \mathbb{G}_{k+1}$. For a k-linear map $e : \mathbb{G}_1 \times \cdots \times \mathbb{G}_k \to \mathbb{G}_{k+1}$, we call $\mathbb{G}_{k+1}$ the *target group* and other groups $\mathbb{G}_i$ *source groups*. We can further assume existence of isomorphisms $\mathbb{G}_i \to \mathbb{G}_j$ between source groups.

A *symmetric leveled k-linear group* is a sequence of groups $\mathbb{G}_1, \ldots, \mathbb{G}_k$ together with bilinear maps $e : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j}$ for $i, j \in [1, k]$ and $i + j \leq k$. We say that $\mathbb{G}_n$ is the group at level n and call $\mathbb{G}_k$ the target group. An *asymmetric leveled k-linear group* is a collection of groups $\{\mathbb{G}_S\}$ for $S \in \mathcal{P}([k])$ together with bilinear maps $e_{S,T} : \mathbb{G}_S \times \mathbb{G}_T \to \mathbb{G}_{S \cup T}$ for all $S \cap T = \emptyset$.

2 Generic Group Models and Symbolic Group Models

In this section, we define a class of generic group models that captures the previously described group settings. Afterwards, we define a symbolic group model where instead of computing with (randomly sampled) group elements, the challenger computes with (fixed) polynomials. We prove that this model is equivalent to the generic group model up to some usually small error.

Generic Group Models. A generic group model for a concrete group setting captures all operations that an adversary with black-box access can perform.

Definition 1. *A* group setting *is a tuple $\mathcal{GS} = (p, \mathcal{G}, \Phi, \mathcal{E})$ where $\mathcal{G} = \{\mathbb{G}_i\}_{i \in \mathcal{I}}$ is a set of cyclic groups of prime order p indexed by a totally ordered set $\mathcal{I}$, Φ is a set of isomorphisms $\phi : \mathbb{G}_i \to \mathbb{G}_j$, and $\mathcal{E}$ is a set of maps, where for each $e \in \mathcal{E}$, there is a k s.t. $e : \mathbb{G}_{i_1} \times \ldots \times \mathbb{G}_{i_k} \to \mathbb{G}_{i_{k+1}}$ is an admissible k-linear map.*

The generic model *for a group setting $(p, \mathcal{G}, \Phi, \mathcal{E})$ and a distribution $\mathcal{D}$ on indexed sets $\{L_i\}_{i \in \mathcal{I}}$ of lists of elements of $\mathbb{G}_i$ is defined as follows. The challenger maintains lists $\boldsymbol{L} = \{L_i\}_{i \in \mathcal{I}}$ where each list L_i contains elements from $\mathbb{G}_i$. The lists are initialized by sampling from $\mathcal{D}$ and the adversary can apply the group*

operations, isomorphisms, and k-linear maps to list elements by providing the indices of elements as handles. For an operation $o : \mathbb{G}_{i_1} \times \ldots \times \mathbb{G}_{i_k} \to \mathbb{G}_{i_{k+1}}$*, the corresponding oracle takes handles* $h_1, \ldots, h_k$*, computes* $a = o(a_1, \ldots, a_k)$ *for* $a_j = L_{i_j}[h_j]$*, appends* a *to* $L_{i_{k+1}}$ *and returns* a*'s handle* $h = |L_{i_{k+1}}|$*. Note that handles are not unique, but the challenger provides an equality oracle to check if two handles refer to the same group element. A formal definition of the game appears in the full version.*

Remark 1. As mentioned in Section 1.2, our generic group model closely follows Maurer's model [27]. We provide the adversary with access to the internal state variables of the challenger via handles, and we assume that the equality queries are "free", in the sense that they do not count when measuring the computational complexity of the adversary.

Example 1. To model a asymmetric leveled k-linear map, we use the index set $\mathcal{I} = \mathcal{P}([k])$, $\Phi = \emptyset$, and $\mathcal{E} = \{e_{T,R} : \mathbb{G}_T \times \mathbb{G}_R \to \mathbb{G}_{T \cup R} \mid T, R \in \mathcal{I} \wedge T \cap R = \emptyset\}$.

Definition 2. *For a list of lists* $\boldsymbol{L} = L_1, \ldots, L_k$ *of polynomials over* $\mathbb{F}_p[X_1, .., X_n]$*, we define the distribution* $\mathcal{D}_{\boldsymbol{L}}$ *by the following procedure. Uniformly sample a point* $\boldsymbol{x} \in \mathbb{F}_p^n$ *and return the list of lists* $\boldsymbol{L}' = L'_1, \ldots, L'_k$ *where* $L'_i = [f_1(\boldsymbol{x})P_i, \ldots, f_{|L_i|}(\boldsymbol{x})P_i]$ *for* $f_j = L_i[j]$*. A distribution* $\mathcal{D}$ *is* polynomially induced *if* $\mathcal{D} = \mathcal{D}_{\boldsymbol{L}}$ *for some* $\boldsymbol{L}$*.*

Most hardness assumptions in generic group models belong to the following classes of decisional, computational, or generalized extraction problems stated with respect to a group setting $\mathcal{GS}$:

- Decisional problem for $\mathcal{D}_{\boldsymbol{L}}$ and $\mathcal{D}_{\boldsymbol{L}'}$:
  Return $b \in \{0, 1\}$ to distinguish the corresponding generic group models.
- Computational problem for $\mathcal{D}_{\boldsymbol{L}}$, polynomial f, and group index i:
  Return handle to $f(\boldsymbol{x})P_i$, where $\boldsymbol{x}$ is the random point sampled by $\mathcal{D}_{\boldsymbol{L}}$.
- Generalized extraction problem for $\mathcal{D}_{\boldsymbol{L}}$, $n, m, i_1, \ldots, i_m, H$:
  Return $\boldsymbol{a} \in \mathbb{F}_p^n$ and handles $h_1, \ldots, h_m$ such that the random point $\boldsymbol{x}$ sampled by $\mathcal{D}_{\boldsymbol{L}}$ satisfies $H(\boldsymbol{x}, \boldsymbol{a}, dlog(L_{i_1}[h_1]), \ldots, dlog(L_{i_m}[h_m])) = 0$.

The above classification generalizes the one proposed by Maurer [27]. Precisely, in addition to decisional and computational assumptions, Maurer considered "straight" extraction problems (such as discrete logarithm) in which the adversary has to extract the random value x of a handle. Our class of *generalized extraction problems* captures extraction problems like discrete logarithm, but also captures problems like the Strong Diffie-Hellman Problem [8].[7] Moreover, note that our class of generalized extraction problems contains the class of computational problems.

From Generic to Symbolic Group Models. The *symbolic group model* for a group setting $(p, \mathcal{G}, \Phi, \mathcal{E})$ and a distribution $\mathcal{D}_{\boldsymbol{L}}$ provides the same adversary

[7] Set $n = 1$, $m = 0$, $H(X, a_1) = X - a_1$ for DLOG and $n = m = 1$,$H(X, a_1, Y) = (X - a_1)Y - 1$ for SDH.

interface as the corresponding generic group model. The difference is that, internally, the challenger now stores lists of polynomials in $\mathbb{F}_p[X_1, \ldots, X_n]$ where $X_1, \ldots, X_n$ are the variables occurring in $\boldsymbol{L}$. The oracles perform addition, negation, and equality checks in the polynomial ring. To define the polynomial operations corresponding to applications of isomorphisms and n-linear maps, observe that for all isomorphisms ϕ there is an $a \in \mathbb{F}_p^\times$ such that $\phi(g_i) = g_j^a$. We therefore define the oracle $\mathsf{isom}_\phi(h)$ such that it computes $a \cdot L_i[h]$. Similarly, we define the oracle $\mathsf{map}_e(h_1, \ldots, h_k)$ such that it computes $a \cdot (L_{i_1}[h_1] \cdots L_{i_k}[h_k])$. We also define a symbolic version $S(E)$ of a generic winning condition E. For decisional problems and computational problems, the symbolic event is equal to the generic event, i.e., $S(E) = E$. For generalized extraction problems, the event E is translated to checking whether $H(X_1, \ldots, X_n, \boldsymbol{a}, L_{i_1}[h_1], \ldots, L_{i_m}[h_m]) = 0$ holds in the polynomial ring. We denote the symbolic group model for a group setting $\mathcal{GS}$ and a distribution $\mathcal{D}_{\boldsymbol{L}}$ with $Sym_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}$ and the corresponding generic group model with $Gen_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}$.

Theorem 1. *Let $(p, \mathcal{G}, \Phi, \mathcal{E})$ denote a group setting, $\mathcal{D}_{\boldsymbol{L}}$ a distribution, $\mathcal{A}$ an adversary performing at most q queries, and E the winning event of a decisional, computational, or generalized extraction assumption. If d is an upper bound on the degrees of the polynomials occurring in the internal state of $Sym_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}(\mathcal{A})$ and $S(E)$, s is the sum of the sizes of the lists in $\boldsymbol{L}$, and the event $S(E)$ contains at most e equality tests, then*

$$|Pr[\, Gen_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}(\mathcal{A}) : E \,] - Pr[\, Sym_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}(\mathcal{A}) : S(E) \,]| \leq (s+q)^2 * d/2p + ed/p$$

where the probability is taken over the coins of $Gen_{\mathcal{GS}}^{\mathcal{D}_{\boldsymbol{L}}}$ and $\mathcal{A}$.

By applying this theorem, we can therefore analyze the hardness of assumptions in the simpler symbolic model. We note that existing master theorems usually include a similar step in their proofs. Here we explicitly prove the equivalence of the *Gen* and *Sym* experiments. This stronger result is required for our decidability results.

3 Master Theorem for Non-Interactive Assumptions

In this section we state our master theorem for decisional, non-interactive problems. In Section 5, we give a master theorem for interactive assumptions which cover generalized extraction problems (and computational ones per Section 2).

To state our theorem, we first define the completion $\mathcal{C}(\boldsymbol{L})$ of a list $\boldsymbol{L}$ with respect to the group setting $(p, \mathcal{G}, \Phi, \mathcal{E})$. This notion will be instrumental to define the side condition of our master theorem. Intuitively speaking, given a list $\boldsymbol{L}$, its completion $\mathcal{C}(\boldsymbol{L})$ is the list of all polynomials that can be computed by the adversary by applying isomorphisms and maps to polynomials in $\boldsymbol{L}$.

We compute the completion $\mathcal{C}(\boldsymbol{L})$ of $\boldsymbol{L}$ in two steps. In the first step, we compute the *recipe lists* $\{R_i\}_{i \in \mathcal{I}}$ using the algorithm given in Figure 2. The elements of the recipe lists are monomials over the variables $W_{i,j}$ for $(i,j) \in \mathcal{I} \times [|L_i|]$.

$\underline{\mathsf{foreach}\ i \in \mathcal{I}:}\ S'_i = \emptyset;\ S_i = \{W_{i,1}, \ldots, W_{i,|L_i|}\}$
$\underline{\mathsf{while}\ \boldsymbol{S} \neq \boldsymbol{S'}:}$
  $\boldsymbol{S'} := \boldsymbol{S}$
  $\underline{\mathsf{foreach}\ e : \mathbb{G}_{j_1} \times \ldots \times \mathbb{G}_{j_n} \to \mathbb{G}_{j_{n+1}} \in \mathcal{E}:}$
    $S_{j_{n+1}} := S_{j_{n+1}} \cup \{f_1 \cdots f_n \mid f_i \in S_{j_i},\ i \in [n]\}$
  $\underline{\mathsf{foreach}\ \phi : \mathbb{G}_i \to \mathbb{G}_j \in \Phi:}\ S_j := S_j \cup S_i$
$\underline{\mathsf{foreach}\ i \in \mathcal{I}:}\ R_i := setToList(S_i)$

Fig. 2. Computation of lists of recipes R_i for input lists L_i.

The monomials characterize which products of elements in $\boldsymbol{L}$ the adversary can compute by applying isomorpisms and maps. The result of the first step is independent of the elements in the lists $\boldsymbol{L}$ and only depends on the lengths of the lists. In the second step, we compute the actual polynomials from the recipes as

$$\mathcal{C}(\boldsymbol{L})_i = [m_1(\boldsymbol{L}), \ldots, m_{|R_i|}(\boldsymbol{L})] \text{ for } [m_1, \ldots, m_{|R_i|}] = R_i$$

where every m_i is a monomial over the variables $W_{i,j}$ and $m_i(\boldsymbol{L})$ denotes the result of evaluating the monomial m_i for the values $L_i[j_i]$.

To ensure that the computation of the recipes terminates, we restrict ourselves to group settings without cycles. We also assume that the group setting contains a target group. Formally, for a group setting $(p, \mathcal{G}, \Phi, \mathcal{E})$, we define the weighted directed graph $G = (V, E)$ with $V = \mathcal{G}$ and E defined as follows. For each isomorphism $\mathbb{G}_i \to \mathbb{G}_j \in \Phi$, there is an edge from $\mathbb{G}_i$ to $\mathbb{G}_j$ of weight 0. Similarly, given any $\mathbb{G}_{i_1} \times \cdots \times \mathbb{G}_{i_n} \to \mathbb{G}_{i_{n+1}} \in \mathcal{E}$, there are edges from $\mathbb{G}_{i_j}$ to $\mathbb{G}_{i_{n+1}}$ of weight 1 for $j \in [n]$. We assume that the graph G contains no loops of *positive* weight. Furthermore, we assume there is a unique $\mathbb{G}_t \in V$ called the *target group*, such that from any $\mathbb{G}_i \in V$ there is a path to $\mathbb{G}_t$ and $\mathbb{G}_t$ does not have any outgoing edges.

Theorem 2. *Let $\mathcal{GS} = (p, \{\mathbb{G}_i\}_{i \in \mathcal{I}}, \Phi, \mathcal{E})$ denote a group setting, and $\mathcal{D}_{\boldsymbol{L}}, \mathcal{D}_{\boldsymbol{L'}}$ be polynomially-induced distributions such that $|L_i| = |L'_i|$ for all $i \in \mathcal{I}$. Let t denote the index of the target group, $s = \sum_{i \in \mathcal{I}} |L_i|$, $r = |\mathcal{C}(\boldsymbol{L})_t|$, and let d denote an upper bound for the total degrees of the polynomials in the completions of the lists. If*

$$\{\boldsymbol{a} \in \mathbb{F}_p^r \mid \boldsymbol{a} \cdot \mathcal{C}(\boldsymbol{L})_t = 0\} = \{\boldsymbol{a} \in \mathbb{F}_p^r \mid \boldsymbol{a} \cdot \mathcal{C}(\boldsymbol{L'})_t = 0\},$$

then

$$|Pr[\ Gen^{\mathcal{GS}}_{\mathcal{D}_{\boldsymbol{L}}}(\mathcal{A}) = 1\ ] - Pr[\ Gen^{\mathcal{GS}}_{\mathcal{D}_{\boldsymbol{L'}}}(\mathcal{A}) = 1\ ]| \leq (s+q)^2 * d/p$$

for all adversaries $\mathcal{A}$ that perform at most q operations.

Note that deciding the side condition is sufficient for deciding the hardness of the corresponding decisional problem for a fixed group setting and fixed distributions. Either the side condition is satisfied or there exists an $\boldsymbol{a} \in \mathbb{F}_p^r$ that is

included in one of the sets, but not in the other one. In the first case, the distinguishing advantage is upper-bounded by the ϵ given above. In the second case, we can construct an adversary that distinguishes the two symbolic models with probability 1, which implies that it distinguishes the corresponding generic models with probability $1 - \epsilon$. Note that for real-or-random assumptions where the adversary is given $\hat{\boldsymbol{L}}$ and must distinguish f from a fresh variable Z in the target group $\mathbb{G}_t$, our side condition simplifies to $\sum_{j=1}^{r} a_j \mathcal{C}(\hat{\boldsymbol{L}})_t[j] \neq f$ for all $\boldsymbol{a} \in \mathbb{F}_p^r$. This is similar to the independence condition in the BBG master theorem [10].

4 Automated Analysis of Non-Interactive Assumptions

In this section, we present methods to automatically verify or falsify the hardness of decisional assumptions. As mentioned earlier, our master theorem is stated with respect to a fixed group setting and fixed distributions. To consider multiple group settings or distributions at once, we define a decisional assumption $\mathbb{A}$ as a possibly infinite set of triples $(\mathcal{GS}, \mathcal{D}_{\boldsymbol{L}}, \mathcal{D}_{\boldsymbol{L}'})$. $\mathbb{A}$ is *generically hard* if the distinguishing probability is upper-bounded by ϵ in Theorem 2 for all triples in $\mathbb{A}$.

We distinguish between *non-parametric* assumptions and *parametric* assumptions. An assumption is non-parametric if only the concrete groups, isomorphisms, and n-linear maps vary, but the structure of the group setting and the lists $\boldsymbol{L}$ and $\boldsymbol{L}'$ defining the distributions remain fixed. This captures assumptions such as "3-lin is hard in all groups with a symmetric 3-linear map". Conversely, an assumption is parametric if one or more of these restrictions do not hold.

4.1 Non-Parametric Assumptions

We perform the following computations over $\mathbb{Z}$ to decide the hardness of a decisional assumption defined by lists $\boldsymbol{L}$ and $\boldsymbol{L}'$ for all group settings $\mathcal{GS}$ with a given index set and types of isomorphisms and n-linear maps.

1. Initialize the set T of distinguishing tests and the set E of exceptional primes to $\emptyset$.
2. Compute the completions $\mathcal{C}(\boldsymbol{L})$ and $\mathcal{C}(\boldsymbol{L}')$ and set $\overline{L}_t := \mathcal{C}(\boldsymbol{L})_t$, $\overline{L'}_t := \mathcal{C}(\boldsymbol{L}')_t$
3. Compute a generating set K of the $\mathbb{Z}$-module $\{\boldsymbol{a} \in \mathbb{Z}^{|\overline{L}_t|} \mid \boldsymbol{a} \cdot \overline{L}_t = 0\}$ as follows:
   (a) Represent all polynomials $g \in \overline{L}_t$ as vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ and denote by M the matrix, where row i is $\boldsymbol{v_i}$ with respect to the basis $\mathit{monomials}(\overline{L}_t)$.
   (b) Compute the Hermite Normal Form N of M and read off a generating set K of the left kernel from N and the transformation matrix. Set $E := E \cup F$ where F is the set of factors of pivots of N.

   Perform the same steps for $\overline{L'}_t$ to obtain M' and K'.
4. Check for every $\boldsymbol{k} \in K$ if $\boldsymbol{k}M' = 0$. If $\boldsymbol{k}M' = c \neq 0$, then set $T := T \cup \boldsymbol{k}$ and $E := E \cup F$ where F denotes the set of common factors of c. Perform the same steps for K' and M.
5. Compute distinguishing probability ϵ from degrees in $\overline{L}_t$ and $\overline{L}'_t$.
6. If T is empty, return that distinguishing probability is upper-bounded by ϵ except (possibly) for primes in E. If T is nonempty, return that using the tests in T, an adversary can distinguish with probability $1 - \epsilon$ except (possibly) for primes in E.

Note that performing division-free computations over $\mathbb{Z}$ allows us to track the set of exceptional primes, which we return. We have implemented this algorithm in a tool that takes a group setting and two sequences of group elements as input and decides if the corresponding decisional assumption is hard returning ϵ, E, and the distinguishing tests T (if nonempty).

4.2 Parametric Assumptions

For parametric decisional assumptions, we restrict ourselves to the real-or-random case. The approach can also be adapted to handle computational assumptions. We distinguish parametricity in two dimensions. First, an assumption may be parameterized by *range limits* $l_1, \ldots, l_m$ (ranging over $\mathbb{N}$) that determine the size of the adversary input. We use *range expressions* $\forall r \in [\alpha, \beta]. h_r$, where α and β are polynomials over range limits, to express such assumptions. The polynomials h_r can use the *range index* r in the exponent or as the index of an indexed variable X_r. We will denote range expressions with capital letters R. Second, the group setting of an assumption may be parameterized by an *arity* k that captures the maximum number of multiplications that can be performed.

Parametricity in the input size allows us to analyze assumptions such as "l-DHE is hard for all l". Parametricity in the arity allows us to analyze assumptions such "2-BDH is hard for all k-linear groups". Combining both types of parametricity allows us to analyze assumptions such as "k-lin is hard in k-linear groups" or "(l, k)-MMDHE is hard for all l and $k \geq 3$". In the following, we will present two methods that deal with both parametricity in the input size and parametricity in the arity. The first method assumes a fixed number of random variables. The second method allows for indexed random variables, but assumes that the degree of adversary input and challenge is fixed.

Fixed Number of Variables. We assume a real-or-random decisional assumption in a (leveled) k-linear group where the challenge polynomial g is in the target group, and the adversary input is expressed using range expressions $R_1, \ldots, R_n$ on the levels $\lambda_1, \ldots, \lambda_n$. Here λ_i is either of the form c or of the form $k - c$ for a constant $c \in \mathbb{N}$. Furthermore, we assume that the assumption uses random variables $\boldsymbol{X}$ and range limits $\boldsymbol{l}$. To simplify the presentation, we will use the notation $\boldsymbol{X}^{\boldsymbol{f}} = X_1^{f_1} \cdots X_m^{f_m}$. Then the ranges are of the form

$$R_i = \forall r_{i,1} \in [\alpha_{i,1}, \beta_{i,1}], \ldots, r_{i,t_i} \in [\alpha_{i,t_i}, \beta_{i,t_i}]. \boldsymbol{X}^{\boldsymbol{f_i}}$$

where every $\alpha_{i,j}$ and $\beta_{i,j}$ is a polynomial over $\boldsymbol{l}$ and every $f \in \boldsymbol{f}_i$ is a polynomial over k, $\boldsymbol{l}$, and $r_{i,1}, \ldots, r_{i,t_i}$. The challenge polynomial is of the form $g = \sum_{i=1}^{w} c_i \boldsymbol{X}^{\boldsymbol{u_i}}$. Using the independence condition derived from Theorem 2, it follows that real distribution and the random distribution are indistinguishable iff there is a monomial $\boldsymbol{X}^{\boldsymbol{u_i}}$ that is not an element of the completion of the R_i.

To check this condition, we proceed in two steps. In the first step, we compute a single range expression $\overline{R}$ that denotes the completion of the R_i in the target group. In the second step, we check for each $\boldsymbol{X}^{\boldsymbol{u_i}}$ whether $\boldsymbol{X}^{\boldsymbol{u_i}} \in \overline{R}$, by encoding the required equalities of the exponent-polynomials into a set of diophantine

(in)equalities. We then show that satisfiability checking for such constraints is undecidable in general. Nevertheless, we identify two decidable fragments and demonstrate that SMT solvers can handle most instances derived from practical cryptographic assumptions, even those that are not in the decidable fragments.

If $R_1, \ldots, R_n$ denote the sets $S_1, \ldots, S_n$, then the completion $\overline{R}$ of $R_1, \ldots, R_n$ in the target group must denote the set

$$\bigcup_{\boldsymbol{\delta} \in \mathbb{N}^n \text{ s.t. } \sum_{i=1}^{n} \delta_i \cdot \lambda_i = k} S_1^{\delta_1} \cdots S_n^{\delta_n}$$

where $SS' = \{ss' \mid s \in S \wedge s' \in S'\}$ and $S^\delta = \{\prod_{i=1}^{\delta} s_i | s_1 \in S \wedge \ldots \wedge s_\delta \in S\}$. We therefore define multiplication of range expressions with distinct range indices as

$$\begin{aligned}&(\forall r_1 \in [\alpha_1, \beta_1], \ldots, r_t \in [\alpha_t, \beta_t].\, \boldsymbol{X}^{\boldsymbol{f}})(\forall r'_1 \in [\alpha'_1, \beta'_1], \ldots, r'_s \in [\alpha'_{t'}, \beta'_{t'}].\, \boldsymbol{X}^{\boldsymbol{f'}}) \\ &= \forall r_1 \in [\alpha_1, \beta_1], \ldots, r_t \in [\alpha_t, \beta_t], r'_1 \in [\alpha'_1, \beta'_1], \ldots, r'_s \in [\alpha'_{t'}, \beta'_{t'}].\, \boldsymbol{X}^{\boldsymbol{f}+\boldsymbol{f'}}.\end{aligned}$$

To define the δ-fold product of a range expression, we restrict ourselves to exponent-polynomials that can be expressed as $\hat{f}+\tilde{f}$ such that $\hat{f} = \sum_{j=1}^{t} r_j\, \phi_j(\boldsymbol{l}, k)$ for polynomials ϕ_j in $\mathbb{Z}[\boldsymbol{l}, k]$ and such that $\tilde{f}$ is a polynomial in $\mathbb{Z}[\boldsymbol{l}, k]$. The δ-fold product is then defined as

$$\begin{aligned}&(\forall r_1 \in [\alpha_1, \beta_1], \ldots, r_m \in [\alpha_t, \beta_t].\, \boldsymbol{X}^{\hat{\boldsymbol{f}}+\tilde{\boldsymbol{f}}})^\delta \\ &= \forall r_1 \in [\delta\alpha_1, \delta\beta_1], \ldots, r_m \in [\delta\alpha_t, \delta\beta_t].\, \boldsymbol{X}^{\hat{\boldsymbol{f}}+\delta\tilde{\boldsymbol{f}}}.\end{aligned}$$

Given range expressions $R_1, \ldots, R_n$, we can now compute $\overline{R}$ by introducing fresh variables $\delta_1, \ldots, \delta_n$, computing the range expressions $R_i^{\delta_i}$, and then computing the product of these range expressions.

The remaining task is now to check if

$$\boldsymbol{X}^{\boldsymbol{u}} \in (\forall r_1 \in [\alpha_1, \beta_1], \ldots, r_t \in [\alpha_t, \beta_t].\, \boldsymbol{X}^{\boldsymbol{f}}) = \overline{R}$$

where $\boldsymbol{u} \in \mathbb{Z}[\boldsymbol{l}, k]^m$, $\alpha_i, \beta_i \in \mathbb{Z}[\boldsymbol{\delta}, \boldsymbol{l}]$, $\boldsymbol{f} \in \mathbb{Z}[\boldsymbol{l}, k, r_1, \ldots, r_t]^m$, and $\sum_{i=1}^{n} \delta_i \cdot \lambda_i = k$. To achieve this, we compute the following set of integer constraints that is satisfiable iff $\boldsymbol{X}^{\boldsymbol{u}} \in \overline{R}$:

$$\begin{cases} 0 \leq \delta_i & \text{for } i \in [1, n] \\ \alpha_i \leq r_i \leq \beta_i & \text{for } i \in [1, t] \\ u_i = f_i, & \text{for } i \in [1, m] \\ \sum_{i=1}^{n} \delta_i \lambda_i = k \end{cases}$$

If we allow for both types of parametricity, it is possible to reduce Hilbert's 10th problem to the generic hardness of cryptographic assumptions expressed as previously described. This yields the following theorem.

Theorem 3. *Deciding hardness of parametric assumptions with a fixed number of variables in the generic group model is undecidable, even if all exponent-polynomials are linear in range limits, range indices, and the arity.*

However, for a restricted class of assumptions, the problem is decidable.

Theorem 4. *For all parametric assumptions with a fixed number of variables such that all exponent-polynomials $f_{i,j}$ and range bounds $\alpha_{i,j}$ and $\beta_{i,j}$ in the input are linear, and either (1) the arity k is fixed or (2) the assumption does not contain range limits l_i and the input exponent-polynomials do not use k, deciding hardness in the generic group model is decidable.*

Proof (Sketch). In both cases, we transform the constraint system into a system of linear constraints. Note that the first type of constraint is already linear. In the first case, the arity k is fixed and we can eliminate the variables δ_i by performing a case distinction since there are only finitely many possible values. Then, the constraints of the first and fourth type are constant and the constraints of the second and third type are linear. If there are no range limits, then the range bounds are constants and we can eliminate the range indices by expanding all range expressions into finite sets of monomials. Then the constraints of the second type are constant and we can linearize the constraints of the last type since λ_i is either a constant c or of the form $k - c$. For constraints of the third type, every u_i is a linear polynomial in $\mathbb{Z}[k]$ and every f_i is a linear polynomial in $\mathbb{Z}[\boldsymbol{\delta}, k]$.

We have implemented this method in our tool and use Z3 [15] to check the constraints. Our experiments confirm that Z3 can prove most assumptions taken from the literature, even those outside the decidable fragment.

Indexed Random Variables. For the case of indexed random variables, we have developed an (incomplete) constraint solving procedure that deals with assumptions parametric in the arity k and a range limit l. Let M denote monomials built from indexed variables and M' denote monomials built from non-indexed variables. Our procedure supports all assumptions where the challenge is of the form $\sum_{i\in[0,l]} MM'$ and the input consist of ranges $\forall i \in [0, l].\, MM'$ and non-indexed monomials M'.

5 Interactive Assumptions

In this section, we present our methods for the analysis of interactive assumptions such as LRSW [26]. To simplify the presentation, we focus on assumptions where exactly *one* additional oracle $\mathcal{O}$ is provided to the adversary and the problem is a generalized extraction problem. In the remainder, we fix a group setting $\mathcal{GS} = (p, \{\mathbb{G}\}_{i\in\mathcal{I}}, \Phi, \mathcal{E})$ and a distribution $\mathcal{D}_{\boldsymbol{L}}$. We use $\boldsymbol{X}$ to denote the variables occurring in $\boldsymbol{L}$ and $\boldsymbol{x}$ to denote the point sampled by $\mathcal{D}_{\boldsymbol{L}}$.

Generalizing *Gen* and *Sym*. Our first step is generalizing the generic group and symbolic group models to the interactive setting. Let q', n, m, l denote positive integers, let $\boldsymbol{i} \in \mathcal{I}^l$, and let $\boldsymbol{F}$ denote an l-dimensional vector of polynomials in $\mathbb{F}_p[\boldsymbol{X}, Y_1, \ldots, Y_m, A_1, \ldots, A_n]$. We say $\mathcal{O}$ is defined by $(q', n, m, l, \boldsymbol{i}, \boldsymbol{F})$ if $\mathcal{O}$ answers at most q' queries and answers queries for parameter $\boldsymbol{a} \in \mathbb{F}_p^n$ by sampling

a point $\boldsymbol{y} \in \mathbb{F}_p^m$ and returning handles to the group elements $F_j(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a})P_{i_j} \in \mathbb{G}_{i_j}$ for $j \in [l]$ where P_{i_j} is the generator of $\mathbb{G}_{i_j}$. Similarly, the symbolic version of $\mathcal{O}$ answers queries for $\boldsymbol{a} \in \mathbb{F}_p^n$ by choosing m fresh variables $\boldsymbol{Y}$, adding the polynomials $F_j(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{a})$ to the lists L_{i_j} for $j \in [l]$, and returning their handles. To formalize winning conditions of interactive assumptions, we extend the previously given definition of generalized extraction problem with inequalities. Concretely, the winning condition is formalized by polynomials $H_1, \ldots, H_{d_1}, G_1, \ldots, G_{d_2}$ that capture the required equalities and inequalities for the field elements $\boldsymbol{b}$ and the handles $\boldsymbol{h}$ returned by the adversary. These polynomials are elements of $\mathbb{F}_p[\boldsymbol{X}, (\boldsymbol{Y_i})_{i\in[q']}, (\boldsymbol{A_i})_{i\in[q']}, \boldsymbol{B}, \boldsymbol{Z}]$. Intuitively, $\boldsymbol{X}$ and $\boldsymbol{Y_i}$ model random variables sampled initially and by $\mathcal{O}$, $\boldsymbol{A_i}$ and $\boldsymbol{B}$ model parameters chosen by the adversary, and $\boldsymbol{Z}$ models group elements referenced by the handles $\boldsymbol{h}$. An adversary, that queries the oracle with $\boldsymbol{a_1}, \ldots, \boldsymbol{a_{q'}}$ and returns $\boldsymbol{b}$ and $\boldsymbol{h}$, wins if the following conditions are satisfied for $\boldsymbol{y_j}$ sampled in the j-th oracle call:

$$H_j(\boldsymbol{x}, \boldsymbol{y_1}, \ldots, \boldsymbol{y_{q'}}, \boldsymbol{a_1}, \ldots, \boldsymbol{a_{q'}}, \boldsymbol{b}, dlog(L_{i_1}[h_1]), \ldots, dlog(L_{i_m}[h_m])) = 0 \ , j \in [d_1]$$
$$G_j(\boldsymbol{x}, \boldsymbol{y_1}, \ldots, \boldsymbol{y_{q'}}, \boldsymbol{a_1}, \ldots, \boldsymbol{a_{q'}}, \boldsymbol{b}, dlog(L_{i_1}[h_1]), \ldots, dlog(L_{i_m}[h_m])) \neq 0 \ , j \in [d_2]$$

Since Theorem 1 captures generalized extraction problems (with inequalities) in such an interactive setting, we can analyze such assumptions in the symbolic group model. As mentioned earlier, the symbolic version of the winning event can be obtained by plugging in the polynomials $L_{i_j}[h_j]$ for the variables Z_j instead of using the discrete logarithm.

Interactive Master Theorem. To define the interactive master theorem, we introduce the notion of parametric completion. The *parametric completion* of $\boldsymbol{L}$ with respect to a group setting $\mathcal{GS}$ and an oracle $\mathcal{O}$ defined by $(q', n, m, l, \boldsymbol{i}, \boldsymbol{F})$ is a family L_i of lists of polynomials in $\mathbb{F}_p[\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{A}]$. Here, the variables $Y_{u,v}$ range over $u \in [m]$ and $v \in [q']$ and the variables $A_{u,v}$ range over $u \in [n]$ and $v \in [q']$. They model the random values sampled by $\mathcal{O}$ and the parameters given to $\mathcal{O}$. The parametric completion first extends the lists L_{i_j} with

$$\{F_j(\boldsymbol{X}, Y_{1,v}, \ldots, Y_{m,v}, A_{1,v}, \ldots, A_{n,v}) \mid v \in [q']\}$$

for $j \in [l]$. Then, it performs the previously defined completion with respect to the isomorphisms and n-linear maps in $\mathcal{GS}$. We denote the result with $\mathcal{C}^{\mathcal{O}}(\boldsymbol{L})$.

To state our interactive master theorem, we exploit that in the symbolic model, we can translate a generalized extraction problem to an equivalent generalized extraction problem where the adversary returns only elements in $\mathbb{F}_p$ and no handles. Let $\mathcal{C}^{\mathcal{O}}(\boldsymbol{L}) = \overline{L_{i_1}}, \ldots, \overline{L_{i_l}}$ denote the lists in the completion. Then, we can translate $H(\boldsymbol{X}, (\boldsymbol{Y_i})_{i\in[q']}, (\boldsymbol{A_i})_{i\in[q']}, \boldsymbol{B}, Z_1, \ldots, Z_l)$ to

$$H'(\boldsymbol{X}, \overrightarrow{\boldsymbol{Y}}, \overrightarrow{\boldsymbol{A}}, \boldsymbol{B}, \boldsymbol{C_1}, \ldots, \boldsymbol{C_l}) = H(\boldsymbol{X}, \overrightarrow{\boldsymbol{Y}}, \overrightarrow{\boldsymbol{A}}, \boldsymbol{V}, \boldsymbol{C_1} \cdot \overline{L_{i_1}}, \ldots, \boldsymbol{C_l} \cdot \overline{L_{i_l}}).$$

The two problems are equivalent since the adversary can return a handle to a polynomial f in L_{i_j} if and only if f is in the span of $\overline{L_{i_j}}$.

Theorem 5. *Let $\mathcal{GS}$ denote a group setting and let $\mathcal{D}_{\boldsymbol{L}}$ denote a polynomially-induced distribution. Consider the $(\hat{n}, \hat{m}, \boldsymbol{j}, \boldsymbol{H}, \boldsymbol{G})$-extraction problem in the generic and symbolic group models for $\mathcal{GS}$, $\mathcal{D}_{\boldsymbol{L}}$, and the oracle defined by $(q', n, m, l, \boldsymbol{i}, \boldsymbol{F})$. Let $\boldsymbol{H}'$ and $\boldsymbol{G}'$ denote the translations of $\boldsymbol{H}$ and $\boldsymbol{G}$ with respect to this model that do not use handles. Then the problem is symbolically hard if there exist no vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ in $\mathbb{F}_p$ such that*

$$\left(\bigwedge\nolimits_{j=1}^{|\boldsymbol{H}'|} H'_j(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) = 0\right) \wedge \left(\bigwedge\nolimits_{j=1}^{|\boldsymbol{G}'|} G'_j(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \neq 0\right).$$

In this case, the winning probability for the generic version is upper-bounded by $(s + q + q' l)^2 * d/2p + ed/p$ where p is the group order, s is the sum of the sizes of the lists in $\boldsymbol{L}$, q the number of queries to the group-oracles, q' the number of queries to $\mathcal{O}$, d an upper bound on the degrees (in $\boldsymbol{X}$ and $\boldsymbol{Y}$) stored by the corresponding symbolic model and occuring in $\boldsymbol{H'}$ and $\boldsymbol{G'}$, and $e = |\boldsymbol{H'}| + |\boldsymbol{G'}|$.

In the proof of this theorem, we use Theorem 1 to switch to the symbolic model. In the symbolic model, the winning condition is equivalent to our side condition.

Automated Analysis. We have developed two methods for the automated analysis of interactive assumptions. Our first method deals with the bounded case, i.e., where the number of oracle queries q' is fixed. Informally, we use Gröbner basis techniques and SMT solvers to prove that there is (1) no solution for all primes, (2) no solution for all primes except for some bad primes, (3) a solution over the rationals which can be converted into an attack for almost all primes, or (4) a solution over $\mathbb{C}$. Even though we only encountered cases (1-3) in practice, case (4) is the reason for the incompleteness of our algorithm since the existence of a solution over $\mathbb{C}$ does not imply the existence of solutions over $\mathbb{F}_p$. In the unbounded case, we perform most steps symbolically to obtain results that are valid for all possible values of q'. Concretely, we encode the hardness of the assumption into a formula in the theory of non-linear arithmetic over $\mathbb{C}$ with uninterpreted function symbols, which we use to encode parameters used in queries and returned by the adversary. We use Z3 to prove the unsatisfiability of these formulas exploiting the support for nonlinear arithmetic over the reals [25] by encoding complex numbers as pairs of reals. In our experiments, Z3 can prove the unsatisfiability of formulas obtained from most valid assumptions in seconds.

Acknowledgements. This work is supported in part by ONR grant N00014-12-1-0914, Madrid regional project S2009TIC-1465 PROMETIDOS, and Spanish projects TIN2009-14599 DESAFIOS 10 and TIN2012-39391-C04-01 Strongsoft. Additional support for Mitchell, Scedrov, and Fagerholm is from the AFOSR MURI "Science of Cyber Security: Modeling, Composition, and Measurement" and from NSF Grants CNS-0831199 (Mitchell) and CNS-0830949 (Scedrov and Fagerholm). The research of Fiore and Schmidt has received funds from the European Commission's Seventh Framework Programme Marie Curie Cofund Action AMAROUT II (grant no. 291803).

References

1. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 20(3):395, 2007.
2. M. Abdalla and D. Pointcheval. Interactive Diffie-Hellman assumptions with applications to password-based authentication. In A. Patrick and M. Yung, editors, *FC 2005*, volume 3570 of *LNCS*, pages 341–356. Springer, Feb. / Mar. 2005.
3. G. Ateniese, J. Camenisch, and B. de Medeiros. Untraceable RFID tags via insubvertible encryption. In V. Atluri, C. Meadows, and A. Juels, editors, *ACM CCS 05*, pages 92–101. ACM Press, Nov. 2005.
4. K. Benson, H. Shacham, and B. Waters. The k-BDH assumption family: Bilinear map cryptography from progressively weaker assumptions. In E. Dawson, editor, *CT-RSA 2013*, volume 7779 of *LNCS*, pages 310–325. Springer, Feb. / Mar. 2013.
5. B. Blanchet. Security protocol verification: Symbolic and computational models. In *POST 2012*, volume 7215 of *Lecture Notes in Computer Science*, pages 3–29, Heidelberg, 2012. Springer.
6. A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM CCS 07*, pages 276–285. ACM Press, Oct. 2007.
7. A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. Cryptology ePrint Archive, Report 2007/438, revised 21 Feb 2010, 2007.
8. D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, May 2004.
9. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, May 2005.
10. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. Cryptology ePrint Archive, Report 2005/015, 2005.
11. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Aug. 2001.
12. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Aug. 2005.
13. X. Boyen. The uber-assumption family (invited talk). In S. D. Galbraith and K. G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, Sept. 2008.
14. E. Bresson, Y. Lakhnech, L. Mazaré, and B. Warinschi. A generalization of DDH with applications to protocol analysis and computational soundness. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 482–499. Springer, Aug. 2007.
15. L. De Moura and N. Bjørner. Z3: An efficient smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.
16. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, 2013.

17. D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 44–61. Springer, May 2010.
18. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.
19. K. Gjøsteen and Ø. Thuen. Password-based signatures. In *Public Key Infrastructures, Services and Applications*, pages 17–33. Springer, 2012.
20. S. Halevi. A plausible approach to computer-aided cryptographic proofs. Cryptology ePrint Archive, Report 2005/181, 2005.
21. S. Hohenberger, A. Sahai, and B. Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 494–512. Springer, Aug. 2013.
22. J. Y. Hwang, D. H. Lee, and M. Yung. Universal forgery of the identity-based sequential aggregate signature scheme. In W. Li, W. Susilo, U. K. Tupakula, R. Safavi-Naini, and V. Varadharajan, editors, *ASIACCS 09*, pages 157–160. ACM Press, Mar. 2009.
23. T. Jager and A. Rupp. The semi-generic group model and applications to pairing-based cryptography. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 539–556. Springer, Dec. 2010.
24. T. Jager and J. Schwenk. On the equivalence of generic group models. In J. Baek, F. Bao, K. Chen, and X. Lai, editors, *ProvSec 2008*, volume 5324 of *LNCS*, pages 200–209. Springer, Oct. / Nov. 2008.
25. D. Jovanović and L. De Moura. Solving non-linear arithmetic. In *Automated Reasoning*, pages 339–354. Springer, 2012.
26. A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In H. M. Heys and C. M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, Aug. 1999.
27. U. M. Maurer. Abstract models of computation in cryptography (invited paper). In N. P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Dec. 2005.
28. U. M. Maurer and S. Wolf. Diffie-Hellman oracles. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 268–282. Springer, Aug. 1996.
29. M. Naor. On cryptographic assumptions and challenges (invited talk). In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Aug. 2003.
30. V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.
31. T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Aug. 2010.
32. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, 1997.
33. W. Stein et al. *Sage Mathematics Software (Version 5.12)*. The Sage Development Team, 2013. `http://www.sagemath.org`.
34. M. Szydlo. A note on chosen-basis decisional diffie-hellman assumptions. In *Financial Cryptography and Data Security*, pages 166–170. Springer, 2006.