A generalization of DDH with applications to protocol analysis and computational soundness

Emmanuel Bresson[1], Yassine Lakhnech[2], Laurent Mazaré[3], Bogdan Warinschi[4]⋆

[1] DCSSI Crypto Lab, emmanuel@bresson.org
[2] VERIMAG Grenoble, yassine.lakhnech@imag.fr
[3] Amadeus SAS, laurent.mazare@m4x.org
[4] University of Bristol, bogdan@cs.bris.ac.uk

Abstract. In this paper we identify the (P, Q)-DDH assumption, as an extreme, powerful generalization of the Decisional Diffie-Hellman (DDH) assumption: virtually all previously proposed generalizations of DDH are instances of the (P, Q)-DDH problem. We prove that our generalization is no harder than DDH through a concrete reduction that we show to be rather tight in most practical cases. One important consequence of our result is that it yields significantly simpler security proofs for protocols that use extensions of DDH. We exemplify in the case of several group-key exchange protocols (among others we give an elementary, direct proof for the Burmester-Desmedt protocol). Finally, we use our generalization of DDH to extend the celebrated computational soundness result of Abadi and Rogaway [1] so that it can also handle exponentiation and Diffie-Hellman-like keys. The extension that we propose crucially relies on our generalization and seems hard to achieve through other means.

Keywords: Diffie-Hellman Assumptions, Protocol Security, Provable Security, Computational Soundness.

1 Introduction

The Decisional Diffie-Hellman (DDH) assumption postulates that, even if given g^x and g^y, it is difficult for any feasible computation to distinguish between g^{xy} and g^r, when x, y and r are selected at random. The simplicity of the statement and several other nice properties (for example random self-reducibility) make the DDH assumption a powerful building block for cryptographic primitives and protocols. Examples of its use include provably secure public-key cryptosystems [13,11], pseudo-random functions [19,9], and pseudo-random generators [4]. The assumption has been particularly successful in the design of efficient and provably secure protocols for key-exchange: two parties can exchange a key by

⋆ The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

sending to each other g^x and g^y (for randomly chosen x and y). Pseudorandomness of the established common key g^{xy} is ensured by the DDH assumption.

Several generalizations of the DDH assumption naturally appear in the context of extending the above scenario from the two-party case to group key-exchange protocols. Perhaps the best known such generalization is the Group Decisional Diffie-Hellman assumption proposed by Steiner et al. [23] and refined by Bresson et al. [7]. Here, the assumption is that given all values $g^{\prod_i x_i}$, for up to $n-1$ exponents, it is hard to distinguish $g^{x_1 \cdots x_n}$ from a random power g^r. The assumption is sufficient to prove secure a protocol where users that privately select powers $x_1, x_2, \ldots x_n$ agree on a common shared key $g^{x_1 \cdots x_n}$. Such generalizations serve two goals. On the one hand they provide simple solutions to the problem that inspired them. More importantly, whenever such general assumptions can be reduced to a more standard assumption (as is the case of many generalizations of DDH), security proofs for protocols can be made more modular: First, prove once and for all the equivalence between the general and the basic assumption. Then, use the more general assumption as a more convenient basic building block for protocols. In this paper we investigate the limits of extending the DDH assumption. Our results are as follows.

Generalization of DDH. Our generalization of DDH is as follows:

- The adversary receives elements of the form $g^{p(x_1,x_2,\ldots,x_n)}$; here p ranges over a fixed set of polynomials P. This setting generalizes significantly all of the previous work where only monomials were allowed in the exponents (*i.e.* the adversary was given only elements $g^{\prod_I x_i}$ for some subset I).
- The adversary receives several challenges of the form $g^{q(x_1,x_2,\ldots,x_n)}$; here q ranges over a fixed set of polynomials Q and the adversary has to determine if he is confronted with these challenges or random group elements. The adversary can see the challenges at any moment (*i.e.*, not necessarily at the end).

We call the problem associated to polynomial sets P and Q, the (P, Q)-DDH problem. In spite of its generality we show that the (P, Q)-DDH assumption reduces to the basic DDH assumption under several mild restrictions on the polynomials in P and Q. In particular, polynomials in Q have to be linearly independent from those in P since otherwise the problem becomes trivial. In general, the loss of security in the reduction that we provide from (P, Q)-DDH to DDH may be exponential. This is to be expected, and perhaps unavoidable, due to the general setting in which we work. Fortunately, we identify several situations where the security loss stays within practical bounds and note that all practical scenarios that we are aware of are instances of these situations. Furthermore, we show that the quality of the reduction can be often improved by using the random self reducibility property of DDH. We prove the equivalence with DDH via a hybrid argument which generalizes those used previously for other generalizations of DDH. We give the formal description of the (P, Q)-DDH problem and clarify its relation to basic DDH in Section 2.

Applications to protocol security. Next, we demonstrate the versatility of the (P, Q)-DDH assumption through several examples:

- we show that the multi-decisional Diffie-Hellman [6] and the Group Decisional Diffie-Hellman assumptions [7] are instances of the (P, Q)-DDH assumption for appropriately chosen P and Q. Interestingly, for the latter assumption our main theorem yields a better reduction to DDH than in previous works.
- we use the (P, Q)-DDH assumption to provide proofs for some DDH-based key-exchange protocols in the presence of *passive* adversaries. In particular, we supply a simple security proof for the Burmester-Desmedt protocol, and exemplify the use of our assumption for a simple protocol that we introduce.

Our examples show that the (P, Q)-DDH problem is an extremely convenient tool for proving the security of protocols in the presence of passive adversaries. In combination with generic results that map such protocols to protocols secure against active adversaries, our simple proofs form the basis of a powerful two-step methodology for the design of provably secure protocols. 1) Prove the protocol secure against passive adversaries using our flexible assumption 2) map the protocol to one secure against active adversaries using special purpose compilers such as the one developed by Katz and Yung for the case of group-key exchange protocols [14]. We develop the ideas sketched above in Section 3.

Application to computational soundness. Our final application is in the context of computational soundness framework. The general goal of this research direction is to allow symbolic, and thus mechanical reasoning about protocols at an abstract, symbolic level, in such a way that symbolically derived results imply security in the standard cryptographic sense. This would permit to prove the cryptographic security of protocols, but it would avoid the standard handmade, error-prone cryptographic proofs through the use of automated tools.

In all of the prior work in this direction, the translation of results from the symbolic world to the cryptographic world is done using so-called "soundness theorems". Notice that these theorems have to deal with *all arbitrary uses* of the primitives in *all possible protocols*! This explains perhaps why exponentiation and Diffie-Hellman like keys are conspicuously missing from all existing computational soundness results: one needs to identify precisely, and in a generic way which of all possible uses of exponentiation are secure and which not. The main result of this paper accomplishes precisely that.

Based on our result we incorporate Diffie-Hellman keys in the framework proposed by Abadi and Rogaway [1]. We extend appropriately the symbolic language introduced in [1] and show that it is possible to use the resulting language to symbolically prove indistinguishability of cryptographic distributions. In particular, this result yields a mechanical way of proving security of key-exchange protocols (in the presence of passive adversaries, with no corruption). The symbolic language and the soundness theorem are in Section 4.

Related work. A generalization of Diffie-Hellman to more general polynomials expressions was investigated by Kiltz in 2001 [15], where a (single) challenge of the form $g^{P(a,b)}$, with the adversary seeing g^a and g^b, is considered. We enlarge the setting in two distinct directions: first we allow many variables instead of just two (and thus, allow the adversary to "see" many polynomials in the exponent), second we allow multiple challenges. Moreover, we provide direct and concrete applications of our main results to the analysis of cryptographic protocols. We note that the work in [15] also studies the case of computational problems in generic groups [21]. Here we concentrate on the decisional case only, and use the standard cryptographic model. Essentially, all previous generalizations of DDH are particular case of our framework. This thus include the so-called "group Diffie-Hellman" assumptions [7], in which the challenge is $g^{x_1 \cdots x_n}$, but also the so-called "parallel Diffie-Hellman" assumption [6], in which the adversary sees $(g^{x_1}, \ldots, g^{x_n})$ and must distinguish tuples of the form $(g^{rx_1}, \ldots, g^{rx_n})$ from random ones $(g^{y_1}, \ldots, g^{y_n})$. Perhaps the closest assumption to the one that we study here is the General Diffie-Hellman Exponent (GDHE) introduced by Boneh et al. in the full version of [5]. We remark that GDHE has been designed to handle bilinear pairings, it has been designed with a single challenge, and its hardness has only been studied in the generic group model. Finally we notice that Square Exponent [16,10,22] and Inverse Exponent can [20] can be seen as instances of our setting.

2 A Generalization of the Decisional Diffie-Hellman problem

2.1 The DDH Problem.

A group family $\mathbb{G}$ is a set of finite cyclic groups $\mathbb{G} = \{G_\lambda\}$ where λ ranges over an infinite index set. We assume in the following that there exists a polynomial-time (in the bit-length of λ) algorithm that given λ and two elements in G_λ outputs their product. (We adopt the multiplicative notation for groups).

Let η be the security parameter. An Instance Generator IG for $\mathbb{G}$ is a probabilistic polynomial-time (in η) algorithm that outputs some index λ and a generator g of G_λ; therefore, IG induces a distribution on set of indexes λ. The Decisional Diffie-Hellman assumption states that for every probabilistic polynomial-time algorithm $\mathcal{A}$, every constant α and all sufficiently large η's, we have:

$$\left| \Pr\left[\mathcal{A}(\lambda, g, g^a, g^b, g^{ab}) = 1\right] - \Pr\left[\mathcal{A}(\lambda, g, g^a, g^b, g^c) = 1\right] \right| < \frac{1}{\eta^\alpha},$$

where the probabilities are taken over the random bits of $\mathcal{A}$, the choice of $\langle \lambda, g \rangle$ according to the distribution $IG(1^\eta)$ and the choice of a, b and c uniformly at random in $[1, |G_\lambda|]$.

In the remaining of the paper we will need to deal with concrete security results. We define the advantage of *any* algorithm $\mathcal{A}$ as the difference of probabilities above. We say that the DDH problem is (ϵ, t)-hard on $\mathbb{G}$ if the advantage of any algorithm running in time t is upper-bounded by ϵ. The (asymptotic) DDH assumption states it is the case for t polynomial and ϵ negligible (in η).

2.2 The (P,Q)-DDH Problem.

Here we introduce formally our generalization of the Decisional Diffie-Hellman problem. As discussed in the introduction we generalize the DDH problem in two crucial directions. First, the group elements that the adversary sees are powers of g that are polynomials (instead of monomials as in the original problem and prior generalizations). Second the adversary is confronted with multiple challenges simultaneously. That is, his goal is to distinguish a list of values obtained by raising g to various polynomials from a list of random powers of g.

Let P and Q be two sets of polynomials in $\mathbb{Z}_q[X_1, X_2, \dots, X_n]$. We assume that these sets are ordered, and write $p_1, p_2, \dots$ and $q_1, q_2, \dots$ for their elements, respectively. Informally, the (P,Q)-DDH-problem asks an adversary to distinguish the distributions:

$$\left(\{g^{p_i(x_1,x_2,\dots,x_n)}\}_{p_i \in P}, \{g^{q_j(x_1,x_2,\dots,x_n)}\}_{q_j \in Q}\right), \text{with } x_i \xleftarrow{\$} \mathbb{Z}_q \tag{1}$$

$$\text{and } \left(\{g^{p_i(x_1,x_2,\dots,x_n)}\}_{p_i \in P}, \{g^{r_j}\}_{j \in [|Q|]}\right), \text{with } x_i \xleftarrow{\$} \mathbb{Z}_q, r_j \xleftarrow{\$} \mathbb{Z}_q \tag{2}$$

Notice that our generalization is quite powerful. All previous generalizations of the DDH problem can be seen as instances of the (P,Q)-DDH problem for suitably chosen P and Q. For example:

- For sets $P = \{X_1, X_2\}$ and $Q = \{X_1 X_2\}$, the associated (P,Q)-DDH is the standard DDH problem.
- For sets $P = \{\prod_{i \in E} X_i \mid E \subsetneq [1,n]\}$ and $Q = \{X_1 X_2 \cdots X_n\}$ the associated (P,Q)-DDH problem corresponds to the group decisional Diffie-Hellman problem.
- For sets $P = \{X_1, X_2, \dots, X_n\}$ and $Q = \{X_1 X_{n+1}, X_2 X_{n+1}, \dots, X_n X_{n+1}\}$ the associated (P,Q)-DDH problem is the parallel Diffie-Hellman problem (see for instance [6]).

We call a pair of sets of polynomials (P,Q) a *challenge*. Our formalization of the (P,Q)-DDH problem departs from the more established formulations where an adversary is explicitly given as input samples from either distribution (1) or distribution (2) and has to decide which is the case. However here the size of sets P and Q may be exponential (for instance for the GDH problem the set P contains exponentially many polynomials), and yet we are typically interested in polynomial-time adversaries who may not have the time to read all the inputs. Therefore we provide the adversary with access to the two distributions via oracles.

Definition 1 ((P,Q)-DDH). *Let q be a prime number. Let $\mathbb{G}$ be a group of order q, g a generator of $\mathbb{G}$, and $P, Q \subseteq \mathbb{Z}_q[X_1, X_2, \dots, X_n]$ two sets of polynomials. We define the oracles $\mathsf{Real}_{(P,Q)}$ and $\mathsf{Fake}_{(P,Q)}$ as follows. Both oracles first select uniformly at random $x_i \xleftarrow{\$} \mathbb{Z}_q$, for $i \in [n]$. Then they answer two types of queries. On input $(\texttt{info}, i)$ for $1 \leq i \leq |P|$, both $\mathsf{Real}_{(P,Q)}$ and $\mathsf{Fake}_{(P,Q)}$ answer with $g^{p_i(x_1,x_2,\dots,x_n)}$. On each new input $(\texttt{chall}, j)$ for some $1 \leq j \leq |Q|$, oracle*

$\mathsf{Real}_{(P,Q)}$ answers with $g^{q_j(x_1,x_2,\ldots,x_n)}$ whereas oracle $\mathsf{Fake}_{(P,Q)}$ selects $r_j \xleftarrow{\$} \mathbb{Z}_q$ and answers with g^{r_j}. The adversary can intertwine `info` and `chall` queries. His goal is to distinguish between these two oracles.

We define the advantage of an adversary $\mathcal{A}$ to solve the (P,Q)-DDH problem by:

$$\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} = \left| \Pr\left[\mathcal{A}^{\mathsf{Real}_{(P,Q)}}(g) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{Fake}_{(P,Q)}}(g) = 1\right] \right|$$

where the probabilities are over the coins of the adversary and those used by the oracles. We say that the (P,Q)-DDH problem is (ϵ,t)-hard in $\mathbb{G}$, if for any $\mathcal{A}$ running within time t, $\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} \leq \epsilon$.

2.3 Our main result: DDH implies (P,Q)-DDH

Before giving our main theorem, we introduce some necessary notions and notations. For a polynomial p we write $mon(p)$ for the set of monomials occurring in p and write $var(p)$ for the set of variables that occur in p. The notation is naturally extended to sets of polynomials[1]. For a monomial m we denote by $ord(m)$ the order of m (i.e., the sum of the powers of its variables). We say p is *power-free* if any $X_i \in var(p)$ appears at power at most 1 (our results hold only for such polynomials). We write $\mathsf{PF}(\mathbb{Z}_q[X_1, X_2, \ldots, X_n])$ for the set of *power-free polynomials* with variables $\{X_1, \ldots, X_n\}$ and coefficients in $\mathbb{Z}_q$. Finally, we write $\mathsf{Span}(P)$ for the vector space over $\mathbb{Z}_q$ generated by P.

For some choice of the (P,Q) challenge, the (P,Q)-DDH problem is trivial (think of the case when $P = \{x_1, x_2\}$ and $Q = \{x_1 + x_2\}$). We therefore restrict the class of challenges only to the interesting cases where the polynomials in Q are linearly independent from those in P. Our main technical result will state that for all non-trivial challenges solving the (P,Q)-DDH problem reduces to solving DDH.

Definition 2 (Non-trivial challenge). *We say that challenge (P,Q) is non-trivial if $\mathsf{Span}(P) \cap \mathsf{Span}(Q) = \{0\}$ and polynomials in Q are linearly independent.*

First we identify a syntactic condition on the sets P and Q which ensures that the adversary has 0 advantage in breaking the (P,Q)-DDH problem. Our condition enforces that for these challenges, which we call *impossible challenges* the distribution of the $g^q(x_1, x_2, \ldots, x_n)$ (for all polynomials $q \in Q$) is statistically independent from the joint distribution $(g^p)_{p \in P}$. The definition is somewhat technical, and uses the graph $G_{(P,Q)}$ whose vertexes are $mon(P \cup Q)$, and in which there is an edge between monomials m_1 and m_2 if there exists $p \in P$ such that m_1, m_2 are in $mon(p)$. We denote by $mon_P^+(Q)$ the set of monomials reachable in this graph from $mon(Q)$ (that is, the strongly connected components of $G_{(P,Q)}$ containg $mon(Q)$). This set, informally, is the smallest superset of $mon(Q)$ that is stable through linear combinations with any polynomials of P containing a monomial of $mon_P^+(Q)$.

[1] For example, for set $P = \{X_1X_3 + X_1X_4, X_2 + X_1X_4\}$ it holds that $var(P) = \{X_1, X_2, X_3, X_4\}$, $mon(P) = \{X_2, X_1X_3, X_1X_4\}$.

Definition 3 (Impossible Challenge). *We say that a non-trivial challenge* (P, Q) *is* impossible *if the two following conditions hold:*

1. $\forall m \in mon_P^+(Q), ord(m) = 1$*: all monomials in* $mon_P^+(Q)$ *are variables,*
2. $\forall m \in mon_P^+(Q), \forall m' \in mon(P) \backslash mon_P^+(Q), m \notin var(m')$*: any monomial that occurs in* P *but not in* $mon_P^+(Q)$ *cannot contain an element of* $mon_P^+(Q)$ *as a variable.*

The first requirement asks that all polynomials in Q are actually sums of variables. The second requirement asks that all polynomials in P either do not use any variable linked to Q (i.e. from $mon_P^+(Q)$) or are sums of variables. The next lemma formally captures that for all challenges that satisfy these two requirements no adversary can win the associated (P, Q)-DDH problem.

Lemma 4. *If* (P, Q) *is an impossible challenge then* $\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} = 0$ *for all adversaries* $\mathcal{A}$*.*

Strategies. The proof of our main theorem is based on a hybrid argument: it uses a sequence of transformations from a non-trivial challenge (P, Q) into an impossible challenge, such that if an adversary succeeds in the original challenge with significantly better probability than in the transformed challenge, then DDH is easy. In our formalization we use power-free polynomials with 2^α variables, that is polynomials in $\mathsf{PF}(\mathbb{Z}_q[X_1, X_2, \ldots, X_{2^\alpha}])$, for some natural number α. It is convenient to identify the index of variables with subsets of $[\alpha]$, and by a slight abuse of notation we identify X_i and $X_{\{i\}}$ (for each $i \in [\alpha]$). Thus, we regard $\mathbb{Z}_q[X_1, X_2, \ldots, X_\alpha]$ as $\mathbb{Z}_q[X_{\{1\}}, X_{\{2\}}, \ldots, X_{\{\alpha\}}]$.

Given a non-trivial challenge (P, Q) with $P, Q \subseteq \mathsf{PF}(\mathbb{Z}_q[X_1, \ldots, X_\alpha])$ we show how to build a sequence of challenges $(P_0, Q_0), (P_1, Q_1), \ldots, (P_l, Q_l)$, with $P_i, Q_i \in \mathsf{PF}(\mathbb{Z}_q[X_1, X_2, \ldots, X_{2^\alpha}])$ such that:

(i). $(P, Q) = (P_0, Q_0)$

(ii). for each adversary $\mathcal{A}$ against the (P_i, Q_i)-DDH there exists an adversary $\mathcal{B}$ against DDH such that:

$$\mathbf{Adv}_{\mathcal{A}}^{(P_i,Q_i)\text{-}\mathsf{DDH}} \leq 2.\mathbf{Adv}_{\mathcal{B}}^{\mathsf{DDH}} + \mathbf{Adv}_{\mathcal{A}}^{(P_{i+1},Q_{i+1})\text{-}\mathsf{DDH}}$$

(iii). (P_l, Q_l) is an impossible challenge, so $\mathbf{Adv}_{\mathcal{A}}^{(P_l,Q_l)\text{-}\mathsf{DDH}} = 0$

Our main result follows by finding an appropriate bound on the length l of the sequence.

One possible way to construct a sequence as above is as follows. Set (P_0, Q_0) to be (P, Q). To obtain (P_{i+1}, Q_{i+1}) out of (P_i, Q_i) we select a pair of variables X_u and X_v that occur together in some monomial in $mon(P \cup Q)$, and *merge* them into a new variable $X_{u \cup v}$. More precisely, in each monomial $m \in mon(P \cup Q)$ where both X_u and X_v occur, we remove these two variables and replace them with $X_{u \cup v}$. (Recall that variables are indexed by subsets of $[\alpha]$.) We call one such transformation a DDH *step*. The procedure ends when we obtain an impossible challenge (P_l, Q_l) (condition (iii) above). We call a sequence of DDH reductions as above a *strategy*, and we represent strategies as lists of pairs of variables $(X_{u_1}, X_{v_1}), ..., (X_{u_l}, X_{v_l})$, with $u_i, v_i \subseteq [\alpha]$ for all i. The *length* of a strategy is the length of the associated list. A strategy σ is *successful* for challenge (P, Q), if the result of applying σ to (P, Q) is an impossible challenge.

Example 5. Take $P = \{X_1, X_2, X_3\}$ and $Q = \{X_1X_2X_3\}$. A successful strategy for (P, Q) is $(X_1, X_2), (X_{1,2}, X_3)$. That is, in the first step we replace X_1X_2 by $X_{1,2}$, and obtain $P_1 = P$ and $Q_1 = \{X_{1,2}X_3\}$. In the second step we replace $X_{1,2}X_3$ by $X_{1,2,3}$. The resulting challenge $(P, \{X_{1,2,3}\})$ is impossible.

The following lemma shows the obtained strategies satisfy condition (ii) above.

Lemma 6. *Let (P', Q') be a challenge obtained from challenge (P, Q) by a* DDH *step. Then for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{B}$ such that:*

$$\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} = 2.\mathbf{Adv}_{\mathcal{B}}^{\mathsf{DDH}} + \mathbf{Adv}_{\mathcal{A}}^{(P',Q')\text{-}\mathsf{DDH}}$$

Moreover, if $t_{\mathcal{A}}$ is the execution time of $\mathcal{A}$, $N_{\mathcal{A}}$ is a bound on the number of oracle queries made by $\mathcal{A}$, then the execution time $t_{\mathcal{B}}$ of $\mathcal{B}$ is bounded by $t_{\mathcal{A}} + N_{\mathcal{A}} t_{(P,Q)}$, where $t_{(P,Q)}$ is (a bound on) the execution time of the oracle related to challenge (P, Q). If (P, Q) is a non-trivial challenge then (P', Q') is also a non-trivial challenge.

The previous two lemmas yield the following concrete security relation between DDH and (P, Q)-DDH.

Proposition 7. *Let $P, Q \in \mathsf{PF}(\mathbb{Z}_q[X_1, X_2, \ldots, X_\alpha])$ form a non-trivial challenge. If (P, Q) has a successful strategy of length n and if the* DDH *problem is (ϵ, t)-hard, then the (P, Q)-DDH is (ϵ', t')-hard, for $\epsilon' = 2n \cdot \epsilon$ and $t' + N t_{(P,Q)} = t$ where N is a bound on the number of oracle queries and $t_{(P,Q)}$ a bound on the execution time of the oracle for challenge (P, Q).*

Generic Strategies. We now exhibit a class of strategies, that we call *generic strategies* that are successful for arbitrary challenges (P, Q). Recall that there are two conditions for a challenge (P, Q) to be impossible: all the monomials of $mon_P^+(Q)$ must be variables and these variables must not occur in any other monomial of P. The idea behind generic strategies is rather simple. First, we change monomials in $mon_P^+(Q)$ into monomials of order 1 by successively merging variables. This leads to an intermediate challenge (P', Q') for which all monomials of $mon_{P'}^+(Q')$ are variables. Next, we deal with the fact that some variables in $mon_{P'}^+(Q')$ may occur elsewhere in P'. Then, for any variable x in $mon_{P'}^+(Q')$, if x appears in a monomial m of P' whose order is greater than 2, then m is transformed using a DDH step so that x does not appear anymore in m. After applying these two steps, we obtain an impossible challenge.

Example 8. Consider the challenge (P, Q) where P has as single element the polynomial $p = X_1X_2X_3X_4$ and Q has as single element the polynomial $q =$

X_1X_2. The first step transforms q into a variable by using the DDH step (X_1, X_2). The resulting challenge is $(P', Q') = (\{X_{1,2}X_3X_4\}, \{X_{1,2}\})$. Notice that $X_{1,2}$ appears in P', so we apply the DDH step $(X_{1,2}, X_3)$ and obtain the challenge $(\{X_{1,2,3}X_4\}, \{X_{1,2}\})$. This challenge is impossible therefore we found a successful strategy whose length is 2.

Next, we provide a bound on the length of generic strategies, which in turn gives an upper bound on the length of successful strategies. Let (P, Q) be an arbitrary challenge. First, we define the *order* of Q within P which we denote by $ord^+_P(Q)$. This quantity is defined by $ord^+_P(Q) = \sum_{m \in mon^+_P(Q)} \big(ord(m) - 1\big)$.

The set $\mathsf{nm}(P, Q)$ of non-maximal elements of $mon^+_P(Q)$ is the set of monomials m which appear in $mon^+_P(Q)$ such that there exists a monomial m' that verifies the following two requirements:

1. m is a strict sub-monomial of m': all the variables of m appear in m' and m is different from m'.
2. m' is in $mon(P)$ but is not in $mon^+_P(Q)$.

Example 9. We still consider the challenge (P, Q) where P contains one element $p = X_1X_2X_3X_4$ and Q has one element $q = X_1X_2$. Then $mon^+_P(Q)$ contains only q. Moreover q is not maximal because $p = qX_3X_4$ hence the set of non-maximal elements $\mathsf{nm}(P, Q)$ is also equal to $\{q\}$.

We are able to show that for any non-trivial challenge there exist strategies whose length can be upper-bounded.

Proposition 10 (Bounded strategies). *For any non-trivial challenge (P, Q), there exists a successful strategy of length:*

$$ord^+_P(Q) + \big(2^{|\mathsf{nm}(P,Q)|} - 1\big).\big(\alpha + ord^+_P(Q)\big)$$

Combined with Proposition 7, we obtain our main theorem:

Theorem 11 (Relating (P, Q)-DDH to DDH). *Let (P, Q) be a non-trivial challenge on variables X_1 to X_α. If the DDH problem is (ϵ, t)-hard, then (P, Q)-DDH is (ϵ', t')-hard, for*

$$\epsilon' = 2\epsilon \left(ord^+_P(Q) + \big(2^{|\mathsf{nm}(P,Q)|} - 1\big).\big(\alpha + ord^+_P(Q)\big) \right)$$

and $t' + Nt_{(P,Q)} = t$ where N is a bound on the number of oracle queries.

Several remarks are in order. We restrict challenges to sets of power-free polynomials. Extending our result beyond this class, would require dealing with group elements of the form g^{X^2}. This seems to be a difficult problem since, for instance, the indistinguishability of (g^x, g^{x^2}) and (g^x, g^r) under the DDH assumption is an open problem [2]. On the other hand, we can easily lift the requirement that polynomials in Q are linearly independent, and modifying appropriately the behavior of the $\mathsf{Fake}_{(P,Q)}$ oracle. We choose to use the current formulation for simplicity.

The formulation of our theorem implies that in the worst case, the loss of security in our reduction may be exponential. We note however that in most, if not all, practical cases $\mathsf{nm}(P,Q)$ is empty, and in those cases the loss in security is only linear. Moreover, notice that in the case when P and Q contain only monomials the hypothesis of the theorem implies that $mon_P^+(Q) = mon(Q) = Q$ and $ord_P^+(Q) = \sum_{m \in Q} \big(ord(m) - 1\big)$. In the next section we consider a few examples where our theorem gives linear security reductions in several interesting applications. However for some applications (like the Burmester-Desmedt protocol), better reductions can be found using the random self-reducibility property of DDH.

Random Self-Reducibility. As said above, the DDH problems has the nice property to be *Random Self-Reducible* (RSR for short). Roughly, this property means that an efficient algorithm for the average case implies an efficient algorithm for the worst case. In the case of DDH, when randomizing an instance, one gets instances, which (1) are uniformly distributed, (2) have all the same solution as the original instance. Thus, being able to solve a single random instance implies that we can solve any instance. As an illustration, let $(X, Y, Z) = (g^x, g^y, [g^{xy}|g^z])$ be an instance of DDH (the notation $Z = [A|B]$ means that the problem is to decide whether Z equals A or B). It is easy to see that for α and β chosen at random, $(X^\alpha, Y^\beta, Z^{\alpha\beta})$ is a new, random instance with the same (decision) solution than the original one.

Here we use RSR as introduced in lemma 5.2 of [3]: from $(g^x, g^y, [g^{xy}|g^z])$ we generate two new instances of DDH: $(g^{\alpha x}, g^y, [g^{\alpha xy}|g^{\alpha z}])$, where α is randomly sampled in $\mathbb{Z}_q$ and $(g^{\alpha x}, g^{\beta y}, [g^{\alpha\beta xy}|g^{\alpha\beta z}])$ where α and β are sampled in $\mathbb{Z}_q$. Using this, we are able to lower the bound given in proposition 7 by giving a finer definition of the weight of a sequence. The idea is that multiple steps can be combined in a single step using RSR. A strategy $(X_{u_1}, X_{v_1}), \ldots, (X_{u_k}, X_{v_k})$ is said to be *randomly self-reducible* (RSR) for a challenge (P, Q) if:

- For step i, X_{u_i} and X_{v_i} have not been introduced in previous steps: for any $j < i$, u_i and v_i are different from $u_j \cup v_j$.
- For step i, X_{v_i} has to be fresh, i.e. this variable was never used in previous steps: for any $j < i$, v_i is different from u_j and v_j.
- Let X_u and X_v be two distinct variables from the strategy, if the product $X_u.X_v$ occurs in P or Q, then there exists a step i such that $u = u_i$ and $v = v_i$ (or $u = v_i$ and $v = u_i$).

Then the weight of such a sequence is 1 as it only counts as a single step and we can extend the result of lemma 6. The idea is that all the kind of "independence" of variables captured by the above conditions allows us to use a single DDH challenge to deal with all the steps (X_{u_i}, X_{v_i}) at once. Formally, we have the following:

Lemma 12. *Let (P', Q') be a challenge obtained from challenge (P, Q) by a RSR strategy. Then for any adversary $\mathcal{A}$ there exists an adversary $\mathcal{B}$ such that:*

$$\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} = 2.\mathbf{Adv}_{\mathcal{B}}^{\mathsf{DDH}} + \mathbf{Adv}_{\mathcal{A}}^{(P',Q')\text{-}\mathsf{DDH}}$$

Moreover, if $t_{\mathcal{A}}$ is the execution time of $\mathcal{A}$, $N_{\mathcal{A}}$ is a bound on the number of oracle queries made by $\mathcal{A}$, then the execution time $t_{\mathcal{B}}$ of $\mathcal{B}$ is bounded by $t_{\mathcal{A}} + N_{\mathcal{A}} t_{(P,Q)}$, where $t_{(P,Q)}$ is (a bound on) the execution time of the oracle related to challenge (P, Q).

We exemplify the use of RSR strategies in obtaining better reductions in Section 3 for the case of the Burmester-Desmedt protocol.

3 Applications: simple proofs for Diffie-Hellman-based protocols

In this section, we show the applicability of our main theorem in a few different contexts. First we apply it to reprove equivalence between the Group DDH problem and basic DDH. Our result yield a tighter security reduction than previous result. As explained in the introduction, our theorem can be used to easily obtain relation between the hardness of DDH and various of its extensions. To illustrate the simplicity associated to using the (P, Q)-DDH assumption we show how to use it to link the reverse DDH assumption (which we introduce) and basic DDH. Finally, we demonstrate that our theorem yields simpler proofs of security for group key-exchange protocols in the presence of passive adversaries, and we show how to obtain a proof for the Burmester-Desmedt protocol.

Throughout this section, we work in a group in which the DDH problem is (ϵ, t)-hard and work with polynomials with α variables $X_1, \ldots, X_\alpha$ (we assume α to be equal to the security parameter.)

GDDH. The Group Decisional Diffie-Hellman (GDDH) problem [23] can be formalized with the challenge (P, Q):

- $P = \{\prod_{i \in E} X_i \mid E \subsetneq [1, \alpha]\}$, that is, P contains all the monomials of order up to $\alpha - 1$.
- $Q = \{\prod_{1 \leq i \leq \alpha} X_i\}$.

Clearly, $\mathsf{Span}(P) \cap \mathsf{Span}(Q) = \{0\}$ and therefore we can apply Theorem 11. Notice that sets P and Q contain only monomials and since $X_1 X_2 \cdots X_\alpha$ is trivially maximal in P, it follows that the (P, Q)-DDH problem is (ϵ', t')-hard, with $\epsilon' = 2(\alpha - 1)\epsilon$ and $t' = t - N t_{(P,Q)} \geq t - t' t_{(P,Q)}$. Thus t' is greater than $t/(1 + t_{(P,Q)})$. Moreover, when calling the oracle, the worst case consists in generating all the X_i and multiplying them, which can be done time polynomial in α. Our results contrasts with that of [7] where the reduction is linear but requires an exponential time in α.

Reverse GDDH. We illustrate the use of non-maximal elements through an example that we call the Reverse GDDH problem. This problem is given by the challenge (P, Q):

- $P = \{\prod_{i \in E} X_i \mid E \subseteq [1, \alpha] \wedge E \neq \{1\}\}$, that is, P contains all the possible monomials except X_1.
- $Q = \{X_1\}$.

Since X_1 is not maximal in P we have that $|\mathsf{nm}(P,Q)| = 1$. By Theorem 11 we obtain that the loss of security is $\epsilon' = 2\alpha\epsilon$, which is linear in the security parameter.

The Burmester-Desmedt Protocol. Introduced in [8] and later analyzed in [14], this protocol is a two-round key exchange protocol between α parties. In the first round, each user U_i samples a random X_i and broadcasts g^{X_i}. In the second round, U_i broadcasts $g^{X_i X_{i+1} - X_{i-1} X_i}$ (with the convention that $X_0 = X_\alpha$ and $X_{\alpha+1} = X_1$). The common secret is $g^{X_1 X_2 + \cdots + X_\alpha X_1}$.

Recall that in the passive setting, security of such a group key-exchange protocol is roughly modeled as follows. First the (passive) adversary observes bitstrings for the different messages exchanged by the participants (using so-called $\mathsf{Execute}$ queries). At some point the adversary decides to challenge the shared secret by trying to distinguish that secret from a random element (the so-called Test query). The adversary is allowed to intertwine his queries. The model, actually corresponds to the (P,Q)-DDH assumption, where the polynomials that correspond to the messages sent by parties are placed in P and the polynomial that corresponds to the shared secret is in Q. Therefore the (P,Q)-DDH assumption that corresponds to polynomials:

- $P = \{X_i \mid 1 \leq i \leq \alpha\} \cup \{X_i X_{i+1} - X_{i-1} X_i \mid 1 \leq i \leq \alpha\}$ corresponds to the broadcast messages.
- $Q = \{\sum_{i=1}^{\alpha} X_i X_{i+1}\}$ corresponds to the shared secret.

is equivalent to the security of the Burmester Desmedt protocol against passive adversaries.

It is easy to check that $\mathsf{Span}(P) \cap \mathsf{Span}(Q) = \{0\}$ (see for instance [14]). Here again Q has only one element and this element is maximal in P. We get $ord_P^+(Q) = \alpha$ and after applying Theorem 11, $\epsilon' = 2\alpha\epsilon$, that is we obtain a linear reduction.

The reduction factor obtained through the use of Theorem 11 is based on generic strategies and is not optimal. Next we show that it is possible to use RSR strategies to obtain better reduction factors (essentially matching the ones that appear in [14]). For simplicity, we assume that α is a multiple of 3. The assumption does not change the asymptotic factors obtained through the reduction bellow. We proceed in two steps: First, we apply the RSR strategy:

$$(X_1, X_2)(X_4, X_5) \ldots (X_{3i+1}, X_{3i+2}) \ldots$$

Let (P', Q') be the resulting challenge. Finally, by applying the following RSR strategy

$$(X_2, X_3)(X_3, X_4)(X_5, X_6)(X_6, X_7) \ldots (X_{3i+2}, X_{3i+3}), (X_{3i+3}, X_{3i+4}) \ldots$$

we obtain an impossible challenge. Using Lemma 12 twice, we get that for any adversary $\mathcal{A}$ against (P,Q)-DDH there exists an adversary $\mathcal{B}$ (of similar time complexity) against DDH such that $\mathbf{Adv}_{\mathcal{A}}^{(P,Q)\text{-}\mathsf{DDH}} = 4\mathbf{Adv}_{\mathcal{B}}^{\mathsf{DDH}}$.

Centralized Diffie-Hellman. We introduce a toy group key exchange protocol in order to illustrate how our results can be used to easily prove such new protocols. This key distribution protocol considers $\alpha - 2$ users $U_1, \ldots, U_{\alpha-2}$ and a server S. Each user U_i randomly samples a group element X_i, while the server S samples two group elements $X_{\alpha-1}, X_\alpha$. Then each user U_i sends g^{X_i} to S and receives $g^{X_\alpha + X_i X_{\alpha-1}}$. The server also broadcasts $g^{X_{\alpha-1}}$. The shared secret is g^{X_α}. The security of the shared key is captured by the challenge (P, Q), where:

- $P = \{X_i \mid 1 \leq i \leq \alpha - 1\} \cup \{X_\alpha + X_i X_{\alpha-1} \mid 1 \leq i \leq \alpha - 2\}$ corresponds to the broadcast messages.
- $Q = \{X_\alpha\}$ corresponds to the shared secret.

Each monomial $X_i X_{\alpha-1}$ appears only once, thus $\mathsf{Span}(P) \cap \mathsf{Span}(Q) = \{0\}$. The set Q has only one element and this element which is maximal in P. Thus $mon(Q) = \{X_\alpha\}$ and $mon^+_P(Q) = \{X_\alpha, X_1 X_{\alpha-1}, \ldots, X_{\alpha-2} X_{\alpha-1}\}$ from which it follows that $ord^+_P(Q)$ is $\alpha - 2$. The loss of security in the reduction is thus only linear.

4 A symbolic logic for Diffie-Hellman exponentials and encryption

In this section we give a symbolic language for representing messages formed by using nonces, symmetric encryption and exponentiation. In some sense, the language that we give in this section is a formal "notation" for distributions. This notation has the crucial property that it can be used to *automatically* reason about the indistinguishability of distributions that arise in cryptographic protocols, without resorting to reduction proofs. For example, using this language, one can define and reason about the security of keys in multicast protocols (see for example [17]) in a way that is meaningful to standard cryptographic models. The main ingredient that enables for such results is a soundness theorem which explains how results at the abstract level of the notation that we introduce map to results about the indistinguishability of distributions.

Syntax. First we make precise the set of symbolic messages that we consider. Let **Keys**, **Nonce** and **Exponents** be three countable disjoint sets of symbols for keys, random nonces, and exponents. We let **Poly** be the set of power-free polynomials with variables in **Exponents** and coefficients in $\mathbb{Z}_q$. The set **Msg** of message expressions is defined by the following grammar:

$$\mathbf{Msg} ::= \mathbf{Keys} \mid \mathsf{g}^{\mathbf{Poly}} \mid \mathbf{Nonce} \mid (\mathbf{Msg}, \mathbf{Msg}) \mid \{\mathbf{Msg}\}_{\mathbf{Keys}} \mid \{\mathbf{Msg}\}_{h(\mathsf{g}^{\mathbf{Poly}})}$$

Equality for expressions is defined modulo polynomial equality. For example, let p and q be two polynomials from **Poly** such that $p = q$ (for classical polynomial equality, e.g. $p = X_1 + X_2 + X_1$ and $q = 2X_1 + X_2$), then $\mathsf{g}^p = \mathsf{g}^q$.

Computational Interpretation. One should think of the elements of **Msg** as symbolic representation for (ensembles of) distributions . For instance, elements of **Keys** represent (the distributions of) cryptographic keys obtained by running the key generation algorithm of some (fixed) encryption scheme. A term like g^X represents the distribution of g^x when exponent x is chosen at random, and $h(\mathsf{g}^{X_1X_2})$ represents the distribution of keys obtained by applying a hash function to $g^{x_1x_2}$ for random x_1 and x_2. A slightly more complex example is the expression: $(g^x, g^y, \{K\}_{h(g^{xy})})$ that represents the distribution of a conversation between two parties that first exchange a Diffie-Hellman key, and then use this key to encrypt a symmetric key.

Let us precise how symbolic expressions are mapped to distributions. Consider a symmetric encryption scheme $\Pi = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$, a family of groups $\mathbb{G} = (\mathbb{G}_\eta)_{\eta\in\mathbb{N}}$ which come with a publicly known generator $\mathfrak{g}$ for each security parameter, and an efficiently computable function $h : \mathbb{G}_\eta \to \{0,1\}^\eta$ to derive cryptographic keys out of exponentials.

We associate to each expression $E \in \mathbf{Msg}$ and security parameter $\eta \in \mathbb{N}$ a distribution $\widehat{E}$ (to avoid cluttered notation we omit to show the dependency on $\Pi, \mathbb{G}$ and η.) We define this distribution as the output of the following randomized algorithm: For each key symbol K that occurs in E we generate a value $\widehat{K} \stackrel{\$}{\leftarrow} \mathcal{KG}(\eta)$; for each variable $X_i \in \mathbf{Exponents}$ we select $\widehat{X} \stackrel{\$}{\leftarrow} \{1, \ldots, |\mathbb{G}_\eta|\}$; for every nonce $N \in \mathbf{Nonce}$ we select $\widehat{N} \stackrel{\$}{\leftarrow} \{0,1\}^\eta$. The output $\widehat{E}$ is computed inductively on the structure of E: $\widehat{(E_1, E_2)} = \widehat{E_1}.\widehat{E_2}$, $\widehat{\mathsf{g}^{p(X_1,\ldots,X_n)}} = \mathfrak{g}^{p(\widehat{X_1}\ldots,\widehat{X_n})}$, $\widehat{\{E\}_K} = \mathcal{E}(\widehat{E}, \widehat{K})$, and $\widehat{\{E\}_{h(\mathsf{g}^p)}} = \mathcal{E}\big(\widehat{E}, h(\widehat{\mathfrak{g}^p})\big)$.

The symbolic adversary. Now we explain how one can reason symbolically about secrecy of message in expressions. Security of encryption in symbolic messages is captured by an axiomatically defined deduction relation $\vdash$. The $\vdash$ relation defines precisely when an expression $E \in \mathbf{Msg}$ can be deduced from a finite set of expressions $S \subseteq \mathbf{Msg}$ (written $S \vdash E$) by a passive eavesdropper. The deduction relation $\vdash$ is an extension of the standard Dolev-Yao inference system [12] and is given by the following rules:

$$\frac{E \in S}{S \vdash E} \qquad \frac{S \vdash (E_1, E_2)}{S \vdash E_1} \qquad \frac{S \vdash (E_1, E_2)}{S \vdash E_2} \qquad \frac{S \vdash \{E\}_K \quad E \vdash k}{S \vdash E}$$

We only consider deduction rule in this axiomatisation. Indeed the $\vdash$ relation is only used to check that a key or an exponentiation can be deduced, thus composition rules are useless.

To the standard Dolev-Yao rules that capture security of encryption, we add several rules for dealing with exponentials, and keys derived from exponentials:

$$\frac{}{E \vdash \mathsf{g}^1} \qquad \frac{E \vdash \mathsf{g}^p \quad E \vdash \mathsf{g}^q}{E \vdash \mathsf{g}^{\lambda p+q}} \lambda \in \mathbb{Z}_q \qquad \frac{E \vdash \{m\}_{h(\mathsf{g}^p)} \quad E \vdash \mathsf{g}^p}{E \vdash m}$$

The first rule says that the adversary knows the generator g of the group; the second says that the adversary can multiply group elements that it knows, and raise group elements that it knows to arbitrary powers in $\mathbb{Z}_q$. The last rule allows the adversary to decrypt a ciphertext under a key derived from an exponential, provided that the adversary can compute that exponential.

Symbolic equivalence of expressions. In a symbolic expression, the information revealed via $\vdash$ can be characterized using *patterns* [1,17]. Intuitively, the pattern of expression $E \in \mathbf{Msg}$ is obtained by replacing all its unrecoverable sub-expressions (those sub-expressions that occur encrypted under keys that the adversary cannot derive from E) by the symbol $\Box$ (undecryptable). For an expression $E \in \mathbf{Msg}$ its pattern is formally defined by the following inductive rules:

$$\begin{array}{ll}
pattern\big((E_1, E_2)\big) = \big(pattern(E_1), pattern(E_2)\big) & \\
pattern\big(\{E'\}_K\big) = \{pattern(E')\}_K & \text{if } E \vdash K \\
pattern\big(\{E'\}_K\big) = \{\Box\}_K & \text{if } E \not\vdash K \\
pattern\big(\{E'\}_{h(\mathsf{g}^p)}\big) = \{pattern(E')\}_{h(\mathsf{g}^p)} & \text{if } E \vdash \mathsf{g}^p \\
pattern\big(\{E'\}_{h(\mathsf{g}^p)}\big) = \{\Box\}_{h(\mathsf{g}^p)} & \text{if } E \not\vdash \mathsf{g}^p \\
pattern(E') = E' & \text{if } E' \in \mathbf{Nonce} \cup \mathbf{Keys} \cup \mathsf{g}^{\mathbf{Poly}}
\end{array}$$

Two expressions $E_1, E_2 \in \mathbf{Msg}$ are deemed symbolically equivalent if they have the same pattern (an adversary can gather the same information out of both expressions): $E_1 \equiv E_2$ if and only if $pattern(E_1) = pattern(E_2)$.

We would like to claim that equivalent expressions have associated indistinguishable distributions. However, the equivalence defined above is too stringent: For example, expressions $(K_1, \{K_1\}_{K_2})$ and $(K_2, \{K_2\}_{K_3})$ are different, although they clearly have equal distributions. The solution is to relax the equivalence by allowing renaming of key and nonce symbols (and even renaming of polynomials). The above expressions become equivalent by renaming (in the first expression) K_1 and K_2 to K_2 and K_3, respectively.

Renaming the polynomials that occur in exponentials is more subtle. Notice that we would like to identify the expressions $E_1 = (\mathsf{g}^{X_1}, \mathsf{g}^{X_2}, \mathsf{g}^{X_1X_2})$ and $E_2 = (\mathsf{g}^{X_1}, \mathsf{g}^{X_2}, \mathsf{g}^{X_3})$ by renaming the polynomial X_1X_2 to the polynomial X_3 (this models the DDH assumption). However not all renamings of polynomials should be considered valid: the expression E_1 and $E_3 = (\mathsf{g}^{X_1}, \mathsf{g}^{X_2}, \mathsf{g}^{X_1+X_2})$ (which are distinguishable since the linear dependency that the adversary can observe in the second expression is absent in the first expression) should not be made indistinguishable by mapping for example X_1X_2 to $X_1 + X_2$. Based on the intuition that underlies our main theorem, we only consider *linear dependence preserving injective* renamings of polynomials (which are renamings that preserve all linear dependencies in the original expression).

Definition 13 (Linear dependance preserving renamings). *Let E be an expression and $\sigma : poly(E) \to \mathbf{Poly}$ be an injective renaming of the polynomials in E. Then σ is said to be linear dependence preserving (ldp) if:*

$$\forall p_1, p_2, \ldots, p_n \in poly(E),\ \forall a_1, ..., a_n, b \in \mathbb{Z},\ \sum_{i=1}^{n} a_i.p_i = b \Leftrightarrow \sum_{i=1}^{n} a_i.p_i\sigma = b$$

For the expression E_1 given above, it can be verified that σ defined by $\sigma(X_1) = X_1, \sigma(X_2) = X_2$ and $\sigma(X_1X_2) = X_3$ is ldp whereas if we set $\sigma(X_1X_2) = X_1+X_2$, σ the resulting renaming is not.

We say that two expressions E_1 and E_2 are *equivalent up to renaming*, and we write $E_1 \cong E_2$ if there exists a renaming σ that is injective on the sets of nonces, keys, and injective and dependence preserving on the set of polynomials, such that $\sigma(m) \equiv n$.

Soundness Theorem. We are now ready to state our soundness theorem. Similarly to the original paper of Abadi and Rogaway [1], we implement encryption using a scheme that besides being IND-CPA secure also hides the length of the plaintext. We write IND-CPA* for the resulting security notion. We emphasize that we use the additional requirement only for simplicity – this requirement can be easily lifted by refining the pattern definition as in [18,17]. The implementation that we consider uses a family of groups where the DDH problem is (asymptotically) hard. Finally, we require that the key derivation function h is such that $\mathcal{KG}(\eta)$ and $h(\mathfrak{g}^r)$ output equal distributions when r is selected at random. The soundness result holds for acyclic expressions, that is expressions where encryption cycles do not occur.

Definition 14 (Acyclic expression). *An expression E is* acyclic *if the two following conditions are satisfied:*

1. *If p is a polynomial such that $h(\mathsf{g}^p)$ occurs as an encryption key in E, then p is not a linear combination of the polynomials that occur in E (and are different from p).*
2. *There exists an order $\prec$ between keys and polynomials from poly(E): if u appears encrypted using v or g^v then $u \prec v$. This order must not have any cycles.*

The first condition is intended to avoid encryptions in which the plaintext and the encryption key are linearly dependent, as for example in $\{\mathsf{g}^{X_1}, \mathsf{g}^{X_1+X_2}\}_{h(\mathsf{g}^{X_2})}$. It can be easily shown that the occurrence of such a ciphertext can reveal the encryption key without contradicting IND-CPA-security of the encryption scheme.

The next theorem establishes the main result of this section: the distributions of equivalent expressions are computationally indistinguishable.

Theorem 15 (Symbolic equivalence implies indistinguishability). *Let E_1 and E_2 be two acyclic expressions, such that $E_1 \cong E_2$. Let Π be a symmetric encryption scheme that is* IND-CPA* *secure and $\mathbb{G}$ be a group such that the* DDH *assumption holds, then $\widehat{E_1} \approx \widehat{E_2}$.*

To appreciate the power that the above soundness theorem provides, consider the expression:

$$E(F) = \left(\mathsf{g}^{X_1}, \mathsf{g}^{X_2}, \mathsf{g}^{X_3}, \mathsf{g}^{X_1X_2}, \mathsf{g}^{X_1X_3}, \mathsf{g}^{X_2X_3}, \{K\}_{h(\mathsf{g}^{X_1X_2X_3})}, \{F\}_K\right)$$

where F is some arbitrary expression. Expression E represents the transcript of the executions of the following (toy) protocol: three parties with secret keys X_1, X_2 and X_3 first agree on a common secret key $h(\mathsf{g}^{X_1X_2X_3})$ (by broadcasting the first 6 messages in the expression). Then, one of the parties generates a new key K which it broadcasts to the other parties encrypted under $h(\mathsf{g}^{X_1X_2X_3})$. Finally, one of the parties, sends some secret expression F encrypted under K. To argue about the security of the secret expression (against a passive adversary) it is sufficient to show that the distributions associated to the expressions $E(F)$ and $E(0)$ are indistinguishable.

Although conceptually simple, a full cryptographic proof would require several reductions (to DDH and security of encryption), and most likely would involve at least one hybrid argument (for proving the security of encrypting K under $h(g^{X_1X_2X_3})$). The tedious details of such a proof can be entirely avoided by using our soundness theorem: it is straightforward to verify that $E(F) \cong E(0)$, and this procedure can be automated. Since $E(F)$ is acyclic, the desired result follows immediately by Theorem 15.

5 Conclusion

In this paper we propose a significant generalization of the DDH problem. We show that in most of the important cases our generalization is not harder than the classical two-parties DDH. As applications, we demonstrate that our generalization enables simple and tight security proofs for several existing key exchange protocols. Moreover, the generalization is instrumental in obtaining a computational soundness theorem that deals with exponentiation and Diffie-Hellman-like keys. We leave as an interesting open problem the question of how to extend this last result to the case of active adversaries.

Acknowledgments We would like to thank Mihir Bellare, Mathieu Baudet and anonymous reviewers for useful comments and suggestions. Some of the research was carried out while the fourth author was with LORIA, INRIA Lorraine in the CASSIS group. He was supported by ACI Jeunes Chercheurs JC9005 and ARA SSIA Formacrypt.

References

1. M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP TCS2000*, pp. 3–22.
2. F. Bao, R. Deng, and H. Zhu. Variations of Diffie-Hellman problem. In *ICICS 2003*, pp. 301–312.
3. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT '00*, pp. 259–274.
4. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. of Computing*, 13:850–864, 1984.
5. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT '05*, pp. 440–456.
6. E. Bresson, O. Chevassut, and D. Pointcheval. Group Diffie-Hellman key exchange secure against dictionary attacks. In *ASIACRYPT '02*, pp. 497–514.
7. E. Bresson, O. Chevassut, and D. Pointcheval. The group Diffie-Hellman problems. In *SAC 2002*, pp. 325–338.

8. M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system (extended abstract). In *EUROCRYPT '94*, pp. 275–286.
9. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO '97*, pp. 455–469.
10. D. Coppersmith and I. Shparlinski. On polynomial approximation of the discrete logarithm and the Diffie-Hellman mapping. *J. of Cryptology*, 13(2):339–360, 2000.
11. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO '98*, pp. 13–25.
12. D. Dolev and A. Yao. On the security of public key protocols. *IEEE IT*, 29(12):198–208, 1983.
13. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithm. *IEEE IT*, 31(4):469–472, 1985.
14. J. Katz and M. Yung. Scalable protocols for authenticated group key exchange. In *CRYPTO '03*, pp. 110–125.
15. E. Kiltz. A tool box of cryptographic functions related to the Diffie-Hellman function. In *Indocrypt '01*, pp. 339–350.
16. U. Maurer and S. Wolf. Diffie-Hellman oracles. In *CRYPTO '96*, pp. 268–282.
17. D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *TCC 2005*. pp. 245–263.
18. D. Micciancio and B. Warinschi. Completeness theorems for the Abadi-Rogaway logic of encrypted expressions. *J. of Computer Security*, 2004. Preliminary version in WITS 2002.
19. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS '97*, pp. 458–467.
20. A.-R. Sadeghi and M. Steiner. Assumptions related to discrete logarithms: Why subtleties make a real difference. In *EUROCRYPT '01*, pp. 244–261.
21. V. Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT '97*, pp. 256–266.
22. I. Shparlinski. Security of most significant bits of g^{x^2}. *IPL*, 83(2):109–113, 2002.
23. M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman key distribution extended to group communication. In *ACM CCS 96*, pp. 31–37.