# Soundness of Symbolic Equivalence for Modular Exponentiation

Yassine Lakhnech, Laurent Mazare, Bogdan Warinschi

# Soundness of Symbolic Equivalence for Modular Exponentiation

Yassine Lakhnech[1], Laurent Mazaré[1], and Bogdan Warinschi[2]

VERIMAG, Grenoble, France, {yassine.lakhnech,laurent.mazare}@imag.fr
LORIA, Nancy, France, bogdan@theory.stanford.edu

**Abstract.** In this paper, we study the Dynamic Decisional Diffie-Hellman (3DH) problem, a powerful generalization of the Decisional Diffie-Hellman (DDH) problem. Our main result is that DDH implies 3DH. This result leads to significantly simpler proofs for protocols by relying directly on the more general problem. Our second contribution is a computationally sound symbolic technique for reasoning about protocols that use symmetric encryption and modular exponentiation. We show how to apply our results in the case of the Burmester & Desmedt protocol.
**Keywords:** Diffie-Hellman Assumptions, Soundness of Formal Encryption, Provable Security.

*Context* Key agreement protocols are essential components of many practical applications (e.g. Kerberos, SSH, TLS, video conferencing), and good design of such protocols has been subject of much cryptographic research. Starting with the pioneering work for key agreement is the Diffie-Hellman protocol [DH76], a variety of protocols based on exponentiation have been proposed. Security of most of these protocols typically relies on the Decisional Diffie-Hellman (DDH). This assumption states that given $g^{x_1}, g^{x_2}$, it is difficult to distinguish between $g^{x_1 x_2}$ and $g^r$, so $g^{x_1 x_2}$ can be safely used as a shared key by parties with private keys $x_1$ and $x_2$.

The extensions of protocols from the 2-party setting to the many-party setting seemed to require stronger assumptions. For instance, a common extension of the DDH problem is the *Group DDH* problem (GDDH for short) [STW96,BCP02]. Here, the adversary gets to see several exponentials (for instance in a setting with three principles he observes $g^{x_1}, g^{x_2}, g^{x_3}, g^{x_1 x_2}, g^{x_1 x_3}, g^{x_2 x_3}$ and has to distinguish between $g^{x_1 x_2 x_3}$ and $g^r$). Nevertheless, most of these extensions were shown to be in fact equivalent to the standard DDH problem.

*Motivation* In most group key agreement protocols that use exponentiation, the messages that are sent by parties (and the key that is agreed upon) are of the form $g^{x_1 x_2 \ldots x_k}$, that is, $g$ raised to some monomial over the secret keys $x_1, x_2, \ldots, x_n$. However, in a few protocols, $g$ is raised to powers that are linear combinations of such monomials. For these protocols carrying out a reduction proof from the DDH assumption seems rather intricate. A good example of such a protocol is that of Burmester & Desmedt protocol [BD94] which has been proved secure only recently [KY03,BD05]. The goal of our work is to find a general form of the DDH problem that would enable simple reduction proofs for protocols even when exponents may be linear combinations of monomials.

*Dynamic Decisional Diffie-Hellman* We consider a group $\mathbb{G}$ of large prime order $q$ and a generator $g$ of $\mathbb{G}$. We introduce a powerful extension of the DDH problem on $\mathbb{G}$, that we call the Dynamic Decisional Diffie-Hellman (3DH) assumption. The generalization is three fold. 1) The exponents used may be polynomials (as opposed to simple monomials), 2) the adversary gets to select what exponents are used and 3) we allow more than one target exponential that the adversary has to distinguish from a random power.

Consider variables $x_1$ to $x_\alpha$ for some $\alpha \in N$. A *monomial* is a product of *distinct* variables and a *polynomial* is a linear combination of monomials. The 3DH assumption considers an adversary that can plays one of two different games, and as usual in decisional assumptions, the adversary has to guess against which game he is playing. In each game the adversary interacts with two oracles. Intuitively, the first oracle outputs the exponentials that the adversary is allowed to see (e.g. for the particular case that corresponds to DDH the first oracle outputs $g^{x_1}$, $g^{x_2}$). The second oracle is the challenge oracle which, depending on the game, outputs either exponentials consistent with the first oracle, or random powers of $g$ (in the case of DDH it outputs either $g^{x_1 x_2}$, or $g^r$).

A bit more formally, the oracles first sample uniformly and independently at random values (in $\mathbb{Z}_q$) for each of the variables $x_1$ to $x_\alpha$. Then they start to answer queries. The first oracle is the same in both games. It takes as argument a polynomial $P$ over $(x_i)_{i=1}^{\alpha}$ and returns $g^{P(x_1,\ldots,x_\alpha)}$. The second oracle is as follows. In the first game it takes as input polynomials $Q$ and returns $g^{Q(x_1,\ldots,x_\alpha)}$ in the first game, and $g^r$ for some randomly sampled $r$ in $\mathbb{Z}_q$ in the second game.

We impose several restrictions on the polynomials that the adversary can query. First, we require that in these polynomials variables do not occur at powers greater than 1. This restriction seems unavoidable since the indistinguishability of $g^x$ and $g^{x.x}$ under the DDH assumption is an open problem [BDZ03]. The second restriction takes care of trivial attacks. An adversary can first submit a polynomial to the first oracle then submit it to the second oracle. If the two results are the same, the adversary knows with high probability that he is in the first game. Otherwise, the adversary knows for sure that he is in the second game. We require that the polynomials submitted to the oracles are linearly independent.

Adversaries are polynomial time (in $\eta$) Turing machines. We say that the DDH assumption holds if the probability for any adversary to distinguish $(g^x, g^y, g^{x.y})$ from $(g^x, g^y, g^r)$ is negligible. In a similar way, the 3DH assumption holds if the probability for any adversary to distinguish the two previous games is negligible. Our main result is that if the DDH assumption holds, then the 3DH assumption also holds. The reciprocal is also true but rather easy to prove.

*Soundness of Symbolic Equivalence* Our second contribution is an extension of the celebrated result of Abadi and Rogaway [AR00]. This result states that symbolic equivalence implies computational indistinguishability for messages that use symmetric cryptography. We extend their result to the case when messages use symmetric encryption and modular exponentiation, and exponentials are used as symmetric keys.

As usual, message are represented as algebraic terms as in [DY83]. To the language in [AR00], we add new expressions of the form $\exp(p)$ for any polynomial $p$. Intuitively, this expression represents $g$ to the power of $p$, i.e. $g^p$. Exponentiations can be used as a

key or as a standard message, however polynomials can only occur in exponentiations and cannot be used as messages.

The deduction relation $\vdash$ is an extended version of the classical Dolev-Yao entailment relation [DY83]. Two new deductions are added in order to handle exponentiations:

$$\frac{E \vdash \exp(p) \quad E \vdash \exp(q)}{E \vdash \exp(\lambda p + q)} \lambda \in \mathbb{Z} \qquad \frac{}{E \vdash \exp(1)}$$

We are mainly interested in the soundness of symbolic equivalence as in [AR00,MP05]. Thus as in these papers, we use a pattern function which represents accessible information from a message $m$. Definition of our pattern function is classical except for exponentiation: $pattern\big(\exp(p)\big) = \exp(p)$. We say that two messages are *equivalent* if they have the same pattern. Two messages are *equivalent up to renaming* if they are equivalent up to some renaming of keys, nonces and polynomial.

$m \cong n$ if and only if $\exists \sigma_1$ a permutation of **Keys**
$\exists \sigma_2$ a permutation of **Nonces**
$\exists \sigma_3$ a linear dependence preserving bijection of polynomials
such that $m \equiv n\sigma_1\sigma_2\sigma_3$

Renaming of key was already used in [AR00]. Renaming of nonces works in the same way. However renaming of polynomials is more subtle: let us consider message $(\exp(x), \exp(y), \exp(x+y))$, if this message is simply transformed into $(\exp(x), \exp(y), \exp(z))$, then instantiations of these two messages are easy to distinguish in the computational setting. In the first case, the third element is the product of the two first ones where as in the second case, this is only the case with negligible probability. In order to fix this problem, we only consider *linear dependence preserving bijections* or ldp bijections. Such bijections have to preserve linear equations among polynomials which are exponentiated. Let us formalize this. Let $\sigma$ be a bijection from $poly(n)$ to $poly(m)$. Then $\sigma$ is said to be ldp if the exact same equations are satisfied after applying it:

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

Equivalence up to renaming is decidable. Let us give some examples of equivalent messages:

$$\big(\exp(x_1), \exp(x_2), \exp(x_1 + x_2)\big) \cong \big(\exp(x_1x_2 - x_2), \exp(x_2), \exp(x_1x_2)\big)$$

Linear relations between exponents are not hidden by modular exponentiation. In this case, the third element is linked to the two previous ones. Hence the same relation must hold on both side: the exponent of the third term is the sum of the exponents of the two first terms.

$$\big(\exp(x_1), \exp(x_2), \{0\}_{\exp(x_1x_2)}\big) \cong \big(\exp(x_1), \exp(x_2), \{1\}_{\exp(x_1x_2)}\big)$$

This example illustrates a passive adversary which observes a Diffie-Hellman key exchange protocol. Two exponentiations are exchanged that allow the two participants to build a shared secret key.

We prove computational soundness of this equivalence relation. Let $m$ and $n$ be two acyclic messages (for an adapted key-acyclicity notion) such that $m$ and $n$ are equivalent up to renaming. If the symmetric encryption used to concretize these messages is IND-CPA secure and the DDH assumption holds in $\mathbb{G}$, then the computational distributions of $m$ and $n$ are indistinguishable (i.e. the probability for any adversary to distinguish them is negligible).

*Some related work* Our contribution to sound symbolic analysis is part of a recent trend in bridging the gap which separates the symbolic and computational views of cryptography. There have been several extensions to the initial results of Abadi and Rogaway mostly concerned with adding different cryptographic primitives and supporting active adversaries. However only a few of the prior results consider modular exponentiation and none of them consider polynomials in exponents. Some of the relevant papers are as follows. In [GS05], a logic is used to verify protocols that use modular exponentiation and digital signature. However only two-party protocols are handled. J. Herzog presents in [Her04] an abstract model for DH key exchange protocols. While in this work the adversary is extended with the capability of applying arbitrary polytime functions, we stick to a more classical symbolic model in the style of [CKRT03,MS03].

*The Burmester-Desmedt Protocol* Our soundness result can be used to prove security for the Burmester-Desmedt protocol in the passive setting. This protocol aims at establishing a secret key between members $U_1$ to $U_n$ of a group and is only designed to be secure against passive adversaries as no authentication is provided.

The protocol uses two rounds: in the first one, each participant $U_i$ samples a random $x_i$ in $\mathbb{Z}_q$, and broadcasts $Z_i = g^{x_i}$. In the second round, each participant $U_i$ broadcasts $X_i = (Z_{i+1}/Z_{i-1})^{x_i}$, where $Z_{n+1}$ is defined as $Z_1$ and $Z_0$ as $Z_n$. Finally, the shared key is

$$K_i = Z_{i-1}^{nx_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} \cdots X_{i-2} = g^{\sum_{i=1}^{n} x_i x_{i+1}}.$$

Let us consider a message $M$ that represents the execution of the protocol and $N$ be the same message where the shared key is replaced by a random group element $g^r$:

$$M = \Big( \exp(x_1), \cdots, \exp(x_n), \exp(x_2x_1 - x_nx_1), \cdots, \exp(x_1x_n - x_{n-1}x_n), \exp(\sum_{i=1}^{n} x_i x_{i+1}) \Big)$$

$$N = \Big( \exp(x_1), \cdots, \exp(x_n), \exp(x_2x_1 - x_nx_1), \cdots, \exp(x_1x_n - x_{n-1}x_n), \exp(r) \Big)$$

Messages $M$ and $N$ are equivalent up to renaming and are acyclic. Hence using our soundness result, the two bit-string distributions are computationally indistinguishable and thus the shared key cannot be distinguished from a random key after execution of the protocol.

## References


[AR00] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)*, Sendai, Japan, 2000. Springer-Verlag.

[BCP02] E. Bresson, O. Chevassut, and D. Pointcheval. The group Diffie-Hellman problems. In *Proceedings of Selected Areas in Cryptography (SAC 2002)*. Springer-Verlag, 2002.

[BD94] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system (extended abstract). In *EUROCRYPT*, pages 275–286, 1994.

[BD05] M. Burmester and Y. Desmedt. A secure and scalable group key exchange system. *Inf. Process. Lett.*, 94(3):137–143, 2005.

[BDZ03] F. Bao, R. Deng, and H. Zhu. Variations of diffie-hellman problem. In *Proceedings of the Fifth Conference on Information and Communications Security (ICIS 2003)*, pages 301–312, 2003.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. In *Proceedings of the Twenty-Third Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2003)*. Springer-Verlag, 2003.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 1976.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 1983.

[GS05] P. Gupta and V. Shmatikov. Towards computationally sound symbolic analysis of key exchange protocols. In *Proceedings of the Third ACM Workshop on Formal Methods in Security Engineering: From Specifications to Code*, 2005.

[Her04] Jonathan Herzog. *Computational soundness for standard assumptions of formal cryptography*. PhD thesis, MIT, 2004.

[KY03] J. Katz and M. Yung. Scalable protocols for authenticated group key exchange. In *Proc. of Crypto '03*, volume 2729 of *Lecture Notes in Computer Science*, pages 110–125. Springer-Verlag, Berlin, 2003.

[MP05] D. Micciancio and S. Panjwani. Adaptive security of symbolic encryption. In *Proceedings of the Theory of cryptography conference (TCC 2005)*. Springer-Verlag, 2005.

[MS03] J. Millen and V. Shmatikov. Symbolic protocol analysis with products and diffie-hellman exponentiation. In *Proceedings of the Sixteenth Computer Security Foundations Workshop (CSFW 2003)*, 2003.

[STW96] M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS 1996)*, 1996.