

Algebraic generalization of Diffie-Hellman key exchange

Juha Partala

Physiological Signal Analysis Team
Center for Machine Vision and Signal Analysis,
University of Oulu
Oulu, Finland

This is a preprint version. Original article: Partala J. (2018), Algebraic Generalization of Diffie-Hellman Key Exchange, *Journal of Mathematical Cryptology*, 12(1), pp. 1-21, doi:10.1515/jmc-2017-0015

Abstract

The Diffie-Hellman key exchange scheme is one of the earliest and most widely used public-key primitives. Its underlying algebraic structure is a cyclic group and its security is based on the discrete logarithm problem (DLP). The DLP can be solved in polynomial time for any cyclic group in the quantum computation model. Therefore, new key exchange schemes have been sought to prepare for the time when quantum computing becomes a reality. Algebraically, these schemes need to provide some sort of commutativity to enable Alice and Bob to derive a common key on a public channel while keeping it computationally difficult for the adversary to deduce the derived key. We present a brief survey on the algebraic properties of existing key exchange schemes and identify the source of commutativity and the family of underlying algebraic structures for each scheme. We also present a universal algebraic view on the Diffie-Hellman protocol and formulate an algebraically generalized Diffie-Hellman scheme (AGDH) that, in general, enables the application of any algebra as the platform. We also formulate the underlying computational problems in the framework of average-case complexity. The scheme is secure if the problem of computing images under an unknown homomorphism is infeasible. Finally, we show that a symmetric encryption scheme possessing homomorphic properties over some algebraic operation can be turned into a public-key primitive with the AGDH provided that the operation is complex enough.

Keywords: cryptography, key exchange, homomorphic image problem, universal algebra

1 Introduction

Cryptographic key exchange is an essential part of modern communication. Such schemes enable two parties to derive a common secret key using a public channel. The Diffie-Hellman key exchange scheme [24], conceptualized by Merkle [55], is one of the most utilized public-key protocols and an integral part of many communication standards. Its underlying mathematical structure is a cyclic group $G = \langle g \rangle$, where g is a known generator. Alice and Bob choose secret elements $a, b \in \{1, 2, \dots, |G|\}$, exchange g^a, g^b and establish a common group element $g^{ab} = (g^a)^b = (g^b)^a$. The scheme works because exponentiation commutes and it is hard to compute the common element g^{ab} from g^a and g^b .

The original Diffie-Hellman scheme applies the multiplicative group of integers modulo p , where p is a prime. However, the discrete logarithm problem (DLP) on this group can be solved in sub-exponential time in the standard model [18]. Therefore, alternative versions of the original scheme were sought by replacing the cyclic group with another. In particular, the group $E(\mathbb{F}_q)$ of rational points on an elliptic curve E defined over a finite field \mathbb{F}_q turned out to yield instantiations with of an order of magnitude greater security [56, 47]. However, all discrete logarithm based schemes can be broken in polynomial time in the quantum computation model using Shor’s algorithm [71]. This means that in order to achieve quantum secure key exchange, it is necessary to consider other algebraic structures. A step towards this direction was taken, for example, in the supersingular isogeny Diffie–Hellman key exchange (SIDH) scheme [41] where exponentiation is combined with the application isogenies of the curve.

Our paper is an exploration of the idea that the less richness we need for the underlying algebraic structure, the harder the computational problems become. For example, elliptic curve isogenies can be constructed in sub-exponential time in the quantum computation model for ordinary elliptic curves. However, the non-commutativity of the endomorphism ring for the supersingular case foils these algorithms and the isogeny reconstruction problem remains exponential time. Therefore, it makes sense to study the algebraic properties of the Diffie-Hellman and other key exchange protocols suggested in the literature and to find the most general, applicable structures in order to minimize the number of tools available for the breaking of the underlying problems.

In this paper, we formulate an *algebraically generalized Diffie-Hellman scheme* (AGDH) that permits any type of algebra as its platform structure. We also formulate the computational problems associated to its security in the framework

¹Infotech Oulu Graduate School, Finnish Foundation of Technology Promotion, Nokia Foundation, Tauno Tönning Foundation, Walter Ahlström Foundation and The Finnish Foundation for Economic and Technology Sciences – KAUTE are gratefully acknowledged for the financial support. Work related to this manuscript has first appeared in the author’s doctoral thesis [62].

of average-case complexity. We start by presenting a brief survey on the algebraic properties of existing cryptographic key exchange schemes. Our emphasis is on the commutativity that results in the common key (for Diffie-Hellman it is the commutativity of exponentiation $(g^a)^b = (g^b)^a$), as well as on the most general algebraic platform structures possible for the scheme. We also give a characterization of the Diffie-Hellman scheme in the framework of universal algebra. Typically, the scheme is viewed as symmetric for Alice and Bob. Both compute an exponentiation map $g \mapsto g^x$, for some $x \in \{1, 2, \dots, |G|\}$. However, such an exponentiation map is both an endomorphism of G and a term function of the algebra. By introducing an asymmetry into the scheme by considering Alice to compute endomorphisms and Bob to compute term functions, we are able to freely choose the underlying algebraic structure provided that a sufficient amount of endomorphisms and term functions are found.

The AGDH is based on computing homomorphic images. To study its security, we define a *homomorphic image problem* (HIP) that asks to compute the image of a given element under an unknown homomorphism as an analogue to the Diffie-Hellman problem (DHP). Similarly to the DHP, we formulate both computational and decision versions of this problem and the common established element is indistinguishable from a random element of the algebra if the decision version is infeasible. Finally, we consider the homomorphic image problem induced by decryption functions of a homomorphic symmetric encryption scheme. We **do not** consider fully homomorphic schemes but schemes that have homomorphic properties over some operation. We devise a condition which ensures that the induced decision HIP is infeasible, essentially turning the encryption scheme into a public-key primitive using the AGDH.

The paper is organized as follows. In Section 2, we lay out the preliminaries for rest of the paper. Section 3 presents a brief survey on the algebraic properties of existing key exchange schemes. In Section 4, we present our main contribution by formulating the algebraically generalized Diffie-Hellman scheme AGDH and the computational and decision versions of the homomorphic image problem HIP. In Section 5, we study the problem of enabling key exchange with a homomorphic symmetric encryption scheme using the AGDH. Finally, Section 6 provides the conclusions.

2 Preliminaries

2.1 Computation

We follow the standard model of probabilistic polynomial time computation. A search problem is a binary relation $R = \{0, 1\}^* \times (\{0, 1\}^* \cup \{\perp\})$. For every $(x, y) \in R$, we call x an *instance* of the problem and y the *solution* to the instance x . If $y = \perp$, then we say that x has no solution. The set of solutions of an instance x is denoted by $R(x)$. A probability ensemble $X = \{X_k\}_{k \in \mathbb{N}}$ consists of random variables X_k indexed by the natural numbers. Our problems will be *distributional* meaning that a computational problem $P = (R, X)$ always

comes with a probability ensemble $X = \{X_k\}_{k \in \mathbb{N}}$ from which its instances are drawn. Here, the index $k \in \mathbb{N}$ determines the binary length of the instance. The notation $y \leftarrow A(x; r)$ means that a probabilistic algorithm A on input x and randomness r outputs y .

Given a distributional search problem $P = (R, X)$ and a probabilistic polynomial time (PPT) algorithm A , we are interested in the probability of A solving a typical instance, called the *advantage*,

$$\mathbf{Adv}_A^P(k) = \Pr[A(X_n) \in R(X_n)].$$

A function ϵ is negligible if for every $n \in \mathbb{N}$ there is $k' \in \mathbb{N}$ such that $\epsilon(k) \leq 1/n^k$ for every $k \geq k'$. A problem P is infeasible if $\mathbf{Adv}_A^P(k)$ is negligible for every PPT algorithm A . The problem of distinguishing probability ensembles $X = \{X_k\}_{k \in \mathbb{N}}$ and $Y = \{Y_k\}_{k \in \mathbb{N}}$ is denoted by $D(X, Y)$ and

$$\mathbf{Adv}_D^{D(X, Y)}(k) = |\Pr[1 \leftarrow D(X_k)] - \Pr[1 \leftarrow D(Y_k)]|$$

for every PPT algorithm D .

2.2 Diffie-Hellman key exchange

The Diffie-Hellman scheme [24] is defined as follows. Let us assume that S is an algorithm that on input the security parameter 1^s , where $s \in \mathbb{N}$, samples a cyclic group G of a suitably large order and a generator g of G . Depending on the representation of the group, the order of the group should be chosen so that the Diffie-Hellman problem (see Defs. 2.2 and 2.3) is infeasible.

Definition 2.1 (Diffie-Hellman key exchange (DH)). *Let the participants be Alice and Bob.*

$$\begin{array}{ccc}
 \begin{array}{l} \text{Alice} \\ \text{Sample } (G, g) \leftarrow S(1^s) \\ \text{Sample } a \leftarrow U(\mathbb{Z}_{|G|}) \end{array} & & \begin{array}{l} \text{Bob} \\ \\ \\ \end{array} \\
 & (G, g, g^a) \longrightarrow & \\
 & \longleftarrow g^b & \begin{array}{l} \text{Sample } b \leftarrow U(\mathbb{Z}_{|G|}) \\ \\ \end{array} \\
 k \leftarrow (g^b)^a = g^{ab} & & k \leftarrow (g^a)^b = g^{ab}
 \end{array}$$

The security of the scheme depends on the infeasibility of the *Diffie-Hellman problem*.

Definition 2.2 (Computational Diffie-Hellman problem (CDHP)). *Let $(G, g) \leftarrow S(1^s)$, where G is a cyclic group and g is a generator of G . Let $a, b \leftarrow U(\mathbb{Z}_{|G|})$. Given $(g, g^a, g^b) \in G^3$, find $y \in G$ such that $y = g^{ab}$.*

The infeasibility of the computational version is often insufficient. We want the adversary to be unable to determine any information about g^{ab} . This is formalized by the decision version of the problem.

Definition 2.3 (Decision Diffie-Hellman problem (DDHP)). *Let G be a cyclic group and let g be a generator of G sampled by $(G, g) \leftarrow \mathcal{S}(1^k)$. Let $B \leftarrow U(\{0, 1\})$ and $a, b, c \leftarrow U(\mathbb{Z}_{|G|})$. Given*

$$\begin{aligned} (g, g^a, g^b, g^{ab}) &\in G^4, \text{ when } B = 0, \\ (g, g^a, g^b, g^c) &\in G^4, \text{ when } B = 1, \end{aligned}$$

determine B .

DDHP is the problem of distinguishing the probability ensembles determined by (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) .

2.3 Universal algebra

Universal algebra encompasses general concepts underlying different algebraic structures such as groups, semigroups, modules and quasigroups. Let A be a non-empty set and let $n \in \mathbb{N}$. A (*finitary*) *operation* on A of *arity* n is a function $f : A^n \rightarrow A$. We define $A^0 = \{\emptyset\}$. A *type* of algebras is a function $\tau : \Omega \rightarrow \mathbb{Z}_{\geq 0}$, where the elements of Ω are the *basic operators* of the type. The type τ assigns an arity for each basic operator $f \in \Omega$.

An *algebra* (or an *algebraic structure*) of type τ is an ordered pair $\mathbf{A} = (A, F)$, where A is a non-empty set and F is a set of operations on A such that for every n -ary basic operator f of the type there is an n -ary operation $f^{\mathbf{A}}$ on A . By the notation $x \in \mathbf{A}$, we mean $x \in A$, where $\mathbf{A} = (A, F)$. We often write f for $f^{\mathbf{A}}$ when it is clear that we mean an operation and not an operator. If $\Omega = \{f_1, f_2, \dots, f_n\}$ for the type, we write $\mathbf{A} = (A, f_1, f_2, \dots, f_n)$ or $\mathbf{A} = (A, f_1^{\mathbf{A}}, f_2^{\mathbf{A}}, \dots, f_n^{\mathbf{A}})$ for $\mathbf{A} = (A, \{f_1^{\mathbf{A}}, f_2^{\mathbf{A}}, \dots, f_n^{\mathbf{A}}\})$ and often $\tau(f_1) \geq \tau(f_2) \geq \dots \geq \tau(f_n)$. The set A of an algebra $\mathbf{A} = (A, F)$ is called the *underlying set* (or the *universe*) of \mathbf{A} . An algebra $\mathbf{A} = (A, F)$ is finite, if A is a finite set.

Let $\mathbf{A} = (A, F_A)$ and $\mathbf{B} = (B, F_B)$ be algebras of the same type. If $B \subseteq A$ and for every basic operator f of the type, $f^{\mathbf{A}}|_B = f^{\mathbf{B}}$, then \mathbf{B} is a *subalgebra* of \mathbf{A} . In such a case, we write $\mathbf{B} \leq \mathbf{A}$. The set of subalgebras of an algebra \mathbf{A} is closed under intersections. Therefore, every $X \subseteq A$ determines the smallest subalgebra $\langle X \rangle \leq \mathbf{A}$ that contains X , *the subalgebra generated by X* .

Let $\mathbf{A} = (A, F_A)$ and $\mathbf{B} = (B, F_B)$ be algebras of the same type τ . A mapping $\alpha : A \rightarrow B$ is a homomorphism from \mathbf{A} to \mathbf{B} if

$$\alpha(f^{\mathbf{A}}(a_1, a_2, \dots, a_n)) = f^{\mathbf{B}}(\alpha(a_1), \alpha(a_2), \dots, \alpha(a_n))$$

for every n -ary basic operator f of the type and every ordered n -tuple $(a_1, a_2, \dots, a_n) \in A^n$. The set of homomorphisms from \mathbf{A} to \mathbf{B} is denoted by $\text{Hom}(\mathbf{A}, \mathbf{B})$. If $\mathbf{A} = \mathbf{B}$, then α is an *endomorphism*. The set of all endomorphisms of \mathbf{A} constitutes a semigroup and it is denoted by $\text{End}(\mathbf{A})$. If $\alpha : \mathbf{A} \rightarrow \mathbf{B}$ is a surjective homomorphism, then \mathbf{B} is a *homomorphic image* of \mathbf{A} .

Let τ be a type of algebras and let Ω be the set of basic operators of the type. Let X be a set of distinct objects called *variables*. The set of *terms* of type τ with variables X is the smallest set $T(X)$ such that $X \subseteq T(X)$ and for

every $p_1, p_2, \dots, p_n \in T(X)$ and every n -ary basic operator $f \in \Omega$ the string $f(p_1, p_2, \dots, p_n) \in T(X)$.

We often consider n -ary polynomials over a field \mathbb{F} as polynomial functions $\mathbb{F}^n \rightarrow \mathbb{F}$. Such a consideration can be also applied to terms. Let $p(x_1, x_2, \dots, x_n)$ be a term of type τ over a set of variables X . Given an algebra $\mathbf{A} = (A, F)$ of type τ , the *term function on \mathbf{A} corresponding to p* is $p^{\mathbf{A}} : \mathbf{A}^n \rightarrow \mathbf{A}$ defined by

1. if p is a variable x_i , then $p^{\mathbf{A}}(a_1, a_2, \dots, a_n) = a_i$ for $a_1, a_2, \dots, a_n \in A$,
2. if p is of the form $f(p_1(x_1, \dots, x_n), \dots, p_k(x_1, \dots, x_n))$, where f is an k -ary basic operator, then

$$p^{\mathbf{A}}(a_1, a_2, \dots, a_n) = f^{\mathbf{A}}(p_1^{\mathbf{A}}(a_1, \dots, a_n), \dots, p_k^{\mathbf{A}}(a_1, \dots, a_n)).$$

For our considerations, the term functions are useful since they behave like the finitary operations with respect to congruences and homomorphisms [14]. In particular, for every homomorphism $\alpha : \mathbf{A} \rightarrow \mathbf{B}$ and every n -ary term p we have

$$\alpha(p^{\mathbf{A}}(a_1, a_2, \dots, a_n)) = p^{\mathbf{B}}(\alpha(a_1), \alpha(a_2), \dots, \alpha(a_n))$$

for every $a_1, a_2, \dots, a_n \in \mathbf{A}$.

3 On the algebraic properties of key exchange schemes

In the algebraic point of view, we can easily identify a fundamental requirement for successful key exchange: something must commute. For the DH, we have $(g^a)^b = (g^b)^a$ for every $a, b \in \mathbb{N}$. In this section, we present a brief survey on the algebraic properties of two party key exchange schemes suggested in the literature. In particular, for each scheme we identify the source of commutativity and the most general suitable algebraic platform structure.

3.1 Cyclic group based schemes

Different versions of the DH have been obtained by replacing \mathbb{Z}_p^* with another cyclic group [70, 49]. There are suggestions based on finite extension fields [12] and groups based on elliptic curves over finite fields [56, 47]. A common element is obtained by the commutativity of exponentiation or multiplication. In particular, the elliptic curve groups $E(\mathbb{F}_p)$, for p prime, have yielded very successful variants of the DH. Other groups over Abelian varieties have been also suggested [48]. Another variant is the XTR [50] and its predecessors [51, 58, 59, 75, 49, 12]. Rubin and Silverberg suggested the group structure on an algebraic torus [69]. A common element is established by the commutativity of exponentiation in \mathbb{F}_q^m and mapping the result to the torus using a birational map. Buchmann and Williams suggested a generalization of the Diffie-Hellman scheme based on an "almost" cyclic group structure on a set of

reduced principal ideals of a real quadratic field [13]. A common reduced ideal is derived based on the commutativity of real number multiplication and addition. These methods are based on the idea of replacing the original group family. Therefore, algebraically such schemes can be considered in the framework of the original DH.

3.2 Diffie-Hellman based on pairings and multilinear maps

Pairings on elliptic curves have been used both in cryptanalytic investigations, as well as in many useful cryptographic constructions. Joux was the first to point out the cryptographic potential of such pairings and suggested a three-party generalization of the Diffie-Hellman scheme based on the Weil and Tate pairings [43]. The common key is established between three parties based on the homomorphic property of the pairing e . The security follows from the hardness of computing $e(P, P)^{abc}$ from (P, aP, bP, cP) , where P is a point on the curve and a, b, c are random integers. Based on Joux's scheme, Verheul suggested a variant with reduced exponentiations and half the number of exchanged bits, as well as a variant of the ElGamal encryption scheme [80]. Bilinearity was also used by Boneh and Franklin to construct a fully functional identity-based encryption scheme [7].

Boneh and Silverberg extended Joux's scheme to $n \geq 4$ parties using multilinear maps [8]. First practical schemes for n -party key exchange for any n were suggested by Garg et al [33] using ideal lattices (the GGH scheme) and Coron et al. [20] using the integers (the CLT scheme). However, these schemes have been shown to be insecure [39],[15]. The improved version of GGH [34] have been also shown to be insecure [19]. Obfuscation-based multilinear maps have been suggested in [83] [9] [1].

3.3 Schemes based on commuting functions

Several methods have been suggested to generalize the DH by replacing group exponentiations with other commuting functions. In principle, for such schemes, we are not interested in the underlying algebraic structure. However, the functions are often generated using algebraic methods. For example, Shpilrain and Zapata characterize discrete logarithm based primitives on groups of prime order as a group action $\text{Aut}(G) \times G \rightarrow G$ [73]. They suggest a generalization based on commuting semigroup actions on a set. To the best of our knowledge, semigroup actions were first suggested by Monico [57]. Similar suggestions can be found from [53] and [78]. There are also suggestions based on commuting chaotic maps [82].

3.4 Non-commutative structure based schemes

The field on non-commutative cryptography is often considered to have started with the work of Anshel et al. [4] and Ko et al. [46]. Ko et al. suggested a

Diffie-Hellman like scheme using the braid group and commuting inner automorphisms. According to [21], the same scheme has been independently suggested by Sidel'nikov [74] using a non-commutative semigroup. A polynomial time algorithm breaking the Ko et al. scheme on the braid group can be found from [16]. Baumslag et al. [5] suggested a scheme based on a finitely presented group G with two commuting subgroups $A, B \leq G$. A common key is derived using the identity $abgb'a' = bga'b'$ for every $g \in G, a, a' \in A$ and $b, b' \in B$. A semidirect product $A \rtimes B$ of two groups, where B is Abelian, was suggested by Habeeb et al. [37]. A common key is established based on two commuting embeddings $\varphi, \phi : A \rightarrow \text{Aut}(B)$.

In the supersingular isogeny key exchange (SIDH), Alice and Bob create distinct, non-commuting isogenies ϕ_A, ϕ_B of a known curve E . They generate point pairs (P_A, Q_A) and (P_B, Q_B) and share their images $(\phi_A(P_B), \phi_A(Q_B))$ and $(\phi_B(P_A), \phi_B(Q_A))$ under the secret isogenies. In the supersingular case, the endomorphism ring is non-commutative. However, based on the homomorphic properties of the two isogenies ϕ_A and ϕ_B , Alice and Bob are able to derive a shared curve E_{AB} that is isogenous to E . The established key is defined as the j -invariant of this curve [41].

For the Anshel-Anshel-Goldfeld (AAG) scheme [4], the common key follows from the homomorphic property $\beta(x, y_1 \cdot y_2) = \beta(x, y_1) \cdot \beta(x, y_2)$ together with $\gamma_1(x, \beta(y, x)) = \gamma_2(y, \beta(x, y))$. For the conjugation based AAG [4, 3] on a non-commutative group, the key is derived as the commutator $[a, b]$ of elements a and b contributed by Alice and Bob, respectively. Shpilrain and Ushakov [72] generalize the construction to use the centralizer instead of the commutator. For both of these schemes, the common key follows from the homomorphic property. Braid groups have been suggested as the platform. However, both schemes can be broken in polynomial time on the braid group [79].

Stickel [77] suggested the application of a non-commutative semigroup G for key exchange. Let $g_1, g_2 \in G$ be non-commuting elements. Alice and Bob exchange $g_1^{a_1} g_2^{a_2}$ and $g_1^{b_1} g_2^{b_2}$. A common key, $g_1^{a_1+b_1} g_2^{a_2+b_2}$, is derived by the commutativity $g^{a+b} = g^{b+a}$. The application of tropical algebras for the implementation of this scheme was suggested in [35]. Rabi and Sherman [67] suggested the use of associative one-way binary operations. In such a case, a common key is derived based on associativity.

3.5 Schemes based lattices

Due to strong security guarantees, lattice based schemes have become a strong alternative for post-quantum cryptography. Since the seminal work of Regev [68] on the learning with errors problem (LWE), it has attracted a lot of research on schemes implementing, for example, cryptographic hash functions, public-key cryptography, digital signature schemes, as well as fully homomorphic encryption [66].

In [42], Ding et al. introduced an extension of the Diffie-Hellman problem with errors based on the LWE (and the corresponding problem in a cyclotomic ring, R-LWE). The common key is derived based on the associativity

of matrix multiplication by computing a bilinear form in two different ways: $(x^T A)y = x^T(Ay)$, where T denotes transpose. Peikert applied the R-LWE in the construction of a key encapsulation mechanism and an authenticated key exchange scheme [65]. In the scheme, a "randomized function" `dbl`, a reconciliation function `rec` and two modular rounding functions $\lfloor \cdot \rfloor_2, \langle \cdot \rangle_2$ are used to establish a common key μ in two different ways: $\mu = \lfloor \text{dbl}(v) \rfloor_2$ and $\mu = \text{rec}(w, \langle \text{dbl}(v) \rangle_2)$, where $w = g(e_0 a + e_1) s_1$ and $v = g e_0 (a s_1 + s_0) + e_2$ are noisy ring elements. A key encapsulation mechanism can be also implemented based on the NTRU cryptosystem [38][22], as well as on error correcting codes [23].

Based on Peikert's scheme, Bos et al. investigated the parameters for a practical implementation in [11] and the resulting scheme was later optimized by Alkim et al. into a scheme called `NewHope` [2]. Based on the work of Ding et al. [42], Bos et al. applied the generic LWE in a scheme called `Frodo` [10]. A provably secure authenticated key exchange protocol applying the R-LWE was presented by Zhang et al. [84] and a password authenticated key exchange scheme was presented by Ding et al. [25].

3.6 Non-associative structure based schemes

There are many suggested applications of non-associative algebra in cryptography. It has been applied, for example, to construct block ciphers, stream ciphers, hash functions and authentication schemes.

Some suggestions for key exchange exist. The implementation of commuting semigroup actions based on both exponentiation and conjugation in a Moufang loop was suggested by Maze [52]. A generalization of the conjugation based AAG for LCC loops was given by Partala and Seppänen [64]. The construction works for any LCC left quasigroup [63] and, similarly to the original AAG, the common key is derived as the commutator but this time on the left multiplication group; the permutation group generated by the bijections $L_a(x) = a * x$, where $*$ is the binary operation of the left quasigroup. A generalized Diffie-Hellman scheme was first described in [61] and refined in this paper. The common key is derived based on the homomorphic property. Wang et al. [81] suggested a scheme similar to DH by considering conjugacy search in a monoid. The scheme works in a non-associative left distributive (LD) structure Q , satisfying $a * (b * c) = (a * b) * (a * c)$ for every $a, b, c \in Q$, induced by conjugation and a common key is derived based on the property $a^{n+m} * b = a^n * (a^m * b)$, where $*$ is the binary operation of the LD structure and exponentiation is conducted in the original monoid. In this case, the common key is a result of both the homomorphic property of conjugation and the commutativity of exponentiation.

We have gathered the essentially different key exchange schemes and their algebraic properties into Table 1.

Table 1: Algebraic properties of key exchange schemes

Scheme	Underlying structure	Key derivation	Suggested platform
Cyclic group		$g^{ab} = g^{ba}$	
DH [24]	cyclic group		\mathbb{Z}_p^* $E(\mathbb{F}_p)$
ECDH [56, 47]			real quadratic field \mathbb{F}_p^* \mathbb{F}_p
XTR [50]			algebraic torus
[69]			$E(\mathbb{F}_p) \rightarrow \mathbb{F}_p^k$
Pairings and multilinear maps			
[43]	cyclic group	$e(mP, nQ) = e(P, Q)^{mn}$	
[80]	cyclic group	$e(mP, nD(P)) = e(P, D(P))^{mn}$	
[81]	group	multi-linearity	ideal lattice
GGH [33]		multi-linearity	
CLT [20]		multi-linearity	\mathbb{Z}
Commuting functions			
[57]	comm. semigroup	$ag^h = xhg$	matrix semigroups Artin groups $E(\mathbb{F}_p)$
[73]			Moufang loops
[78]			
[52]			
[67]	group	$a(bc) = (ab)c$	
Non-commutative structure			
AAG (general) [4]	monoid	$\gamma_1(x, \beta(y, x)) = \gamma_2(y, \beta(x, y))$	braid group
AAG (group) [5]	non-comm. group	commutator $[a, b]$	braid group
[74]	non-comm. semigroup	$a^{-1}b^{-1}gba = b^{-1}a^{-1}gab$	
[46]	non-comm. group	$a^{-1}b^{-1}gba = b^{-1}a^{-1}gab$	braid group
[5]	non-comm. group	$abg'b'a' = bag'a'b'$	matrix groups
[37]	$A \rtimes B$, A, B groups, B comm.	$\varphi, \phi : A \rightarrow \text{Aut}(B)$, $\varphi\phi = \phi\varphi$	\mathbb{F}_p^n
[77]	non-comm. semigroup	$g_1^{a_1+b_1}g_2^{a_2+b_2} = g_1^{b_1+a_1}g_2^{b_2+a_2}$	matrix groups $E(\mathbb{F}_{p^2})$
SIDH [41]	non-comm. ring	$\alpha(xy) = \alpha(x)\alpha(y)$	braid group
[72]	non-comm. group	$\alpha(xy) = \alpha(x)\alpha(y)$	matrix semigroup
[50]	$G \rtimes H$, $H \leq \text{Aut}(G)$, G group	$\alpha(xy) = \alpha(x)\alpha(y)$	
[42]	lattice		
Prodo [10]	generic lattice	$(x^T A)y = x^T (Ay)$	
[65]	ideal lattice		$\mathbb{Z}q, q$ integer
NewHope [2]	ideal lattice	$[\text{db}(v)]_2 = \text{ec}(w, \langle \text{db}(v) \rangle_2)$ $[\text{db}(v)]_2 = \text{ec}(w, \langle \text{db}(v) \rangle_2)$	$\mathbb{Z}q/(x^{2^n} + 1)$ $\mathbb{Z}q/(x^{2^n} + 1)$
Non-associative structure			
[64]	LCC loop	$[\alpha, \beta]$	LCC loop on \mathbb{Z}_p^2
[63]	LCC left quasigroup		left quasigr.: $a(bc) = (ab)(ac)$
[81]	LD str., monoid conjugation	$a^n + m * b = a^n * (a^m * b)$	matrix monoid
[61]	universal algebra	$\alpha(xy) = \alpha(x)\alpha(y)$	$(\mathbb{F}_p^n, +, \cdot)$

4 Algebraic generalization of the Diffie-Hellman scheme

Many key exchange schemes in our brief survey can be seen as generalizations or different versions of the DH. A straightforward generalization is to use other commuting functions. However, it is not straightforward to construct commuting functions with the needed infeasibility requirements. Here, our emphasis is on the algebraic properties of the exponentiation map. Many generalizations have observed that exponentiations in a cyclic group commute. However, exponentiation in a cyclic group is also an endomorphism of the group. Typically, generalizations concentrate on the commutativity property instead of the homomorphic property. Notable exceptions include, for example, pairing-based schemes, such as the tripartite Diffie-Hellman scheme of Joux [43], where a common key is derived based on bilinearity of the pairing. In this paper, we also concentrate on the homomorphic property.

Based on it, we formulate a generalization for the DHP. Our main motivation for such a generalization is the possibility of lifting the Diffie-Hellman from cyclic groups to more general algebraic structures. In particular, the escape from cyclic groups is necessary to ensure security in the quantum computation model. The removal of algebraic laws enables us to do that and facilitates the development of new, quantum resistant, key exchange schemes. Existing suggestions require the platform structure to satisfy special laws, such as the group axioms. Our formulation permits the application of any algebraic structure without special algebraic laws except the existence of homomorphisms. In addition, as a direct generalization of the DH, it aims to preserve the utility of the DH.

Another motivation follows from cryptographically useful properties of homomorphisms. In particular, in most cases a homomorphism f is *resamplable* [28]. That is, there is a PPT algorithm A that on input (x, b) produces a distribution (X, \mathcal{B}) such that the event " $b = f(x)$ if and only if $b' = f(x')$ " holds with probability one for every $(x', b') \leftarrow (X, \mathcal{B})$. Resamplability is a special form of *random self-reducibility* [29] that allows us to infer average-case hardness of certain problems based on their worst-case infeasibility. Resamplability also enables us to derive tighter bounds on advantage when invoking the hybrid argument [28]. Therefore, due to resamplability and worst-case to average-case reductions, we expect homomorphism based schemes to obtain stronger guarantees for their security similar to learning with errors based schemes [68, 65], where several worst-case to average-case reductions are known.

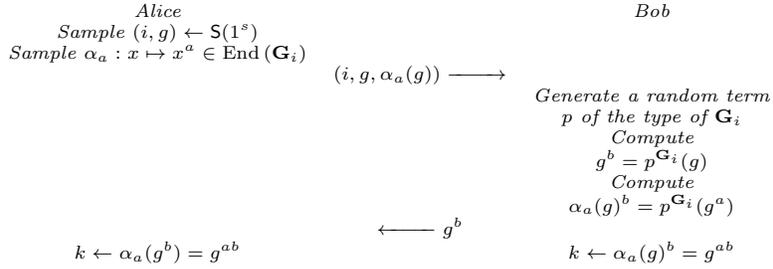
First, we give a universal algebraic view on the Diffie-Hellman scheme. We observe that the security of DH can be seen to be based on the infeasibility of computing a homomorphic image. Based on this observation, we formulate a *homomorphic image problem* (HIP) that asks to compute the image of a given element under an unknown homomorphism. We show that the required commutativity is induced by the homomorphic property and it is sufficient for key exchange. This consideration allows us to lift the DHP from a cyclic group to any pair of algebras \mathbf{A} and \mathbf{B} with a suitably large set of efficiently samplable

and computable homomorphisms from \mathbf{A} to \mathbf{B} . We define a notion that is analogous to a *group family* $\mathbb{G} = (\{\mathbf{G}_i : i \in I\}, \mathcal{S})$ that consists of a collection of cyclic groups $\{\mathbf{G}_i : i \in I\}$ and an algorithm \mathcal{S} to sample from that collection [6]. We define a similar *family of algebras* and use it to formulate a decision version of the HIP.

4.1 Universal algebraic view of the Diffie-Hellman scheme

Our construction is based on the following observation. Let us consider a cyclic group G_i as an algebra \mathbf{G}_i . Then, every exponentiation function $\alpha_a : x \mapsto x^a$ is both an endomorphism and a term function $\mathbf{G}_i \rightarrow \mathbf{G}_i$. Let us now consider the original Diffie-Hellman key agreement scheme in the following form that introduces an apparent asymmetry in the computational procedures of Alice and Bob.

Definition 4.1 (Diffie-Hellman key agreement). *Let the participants be Alice and Bob and let $\mathbb{G} = (\{\mathbf{G}_i : i \in I\}, \mathcal{S})$ be a group family.*



Alice first samples a private endomorphism $\alpha_a : x \mapsto x^a$, where $a \leftarrow U(\mathbb{Z}_{|\mathbf{G}_i|})$. Bob generates a random term p of the type of \mathbf{G}_i such that the term function $p^{\mathbf{G}_i}$ is polynomial time computable. He computes

$$g^b = p^{\mathbf{G}_i}(g) = \underbrace{gg \cdots g}_{b \text{ times}}.$$

The same term function is applied on $\alpha_a(g) = g^a$ to obtain a secret element

$$\alpha_a(g)^b = p^{\mathbf{G}_i}(g^a) = \underbrace{\alpha_a(g)\alpha_a(g) \cdots \alpha_a(g)}_{b \text{ times}} = \underbrace{g^a g^a \cdots g^a}_{b \text{ times}} = g^{ab}.$$

The binary operation is not actually applied $b - 1$ times. Rather, Bob chooses a term function such that the fast exponentiation algorithm can be applied to reach g^b and g^{ab} in a polynomial number of operations. Alice can compute $\alpha_a(g^b) = g^{ab}$ and the equality of the established key follows from the homomorphic property of α_a .

We can immediately see that it is possible to exchange the group family \mathbb{G} with a family of non-group algebras. That is, we can consider two algebras \mathbf{A}, \mathbf{B} of the same type and let $\alpha_a \in \text{Hom}(\mathbf{A}, \mathbf{B})$. There are three different algorithms implicit in the scheme:

1. The sampling algorithm S that can be considered to sample both (i, g) and α_a .
2. A probabilistic polynomial time *random composition algorithm* R that on input $i \in I$ and an element $x \in \mathbf{G}_i$ samples a term p and computes the term function on x .
3. A deterministic polynomial time *homomorphism computation algorithm* H that given $i \in I$, $a \in \mathbb{Z}$ and an element $x \in \mathbf{G}_i$ evaluates $\alpha_a(x)$.

For the generalization of the group family to a family of algebras, these algorithms need to be made explicit. For example, for the group family case both R and H compute x^a using the fast exponentiation algorithm.

4.2 The homomorphic image problem

In this section, we carefully construct a rigorous definition for the family of algebras, as well as for the homomorphic image problem. In order to be able to increase the security of the different constructions using a security parameter, the family has to consist of pairs $(\mathbf{A}_i, \mathbf{B}_i)$ indexed by a countably infinite index set I . We need a sampling algorithm S that samples such pairs, outputs the corresponding $i \in I$ and a set of generators a_1, a_2, \dots, a_n for \mathbf{A}_i . We also need the family to have a meaningful composition algorithm R , for term function generation, that can be randomized. For an algebra with n generators, potentially several such algorithms can be devised. In contrast, the only meaningful composition algorithm for a group family is the fast exponentiation algorithm with a randomized exponent. To see why this is the case, we observe that each element of a cyclic group \mathbf{G}_i is of the form g^x for $x \in \mathbb{N}$, where g is a generator of the group. Therefore, for every term p , there is $z \in \mathbb{N}$ such that $p^{\mathbf{G}_i}(g) = g^z$ and the fastest way to compute it is using the fast exponentiation algorithm.

Finally, we require participants to be able to efficiently compute homomorphisms $\varphi \in \text{Hom}(\mathbf{A}_i, \mathbf{B}_i)$ for every $i \in I$. Therefore, the family has to come with an explicitly stated set of efficiently computable homomorphisms and a deterministic homomorphism computation algorithm H .

We consider a *family of algebras* as a countably infinite set of triples $(\mathbf{A}_i, \mathbf{B}_i, \mathcal{H}_i)$, where \mathbf{A}_i and \mathbf{B}_i are algebras of the same type and $\mathcal{H}_i \subseteq \text{Hom}(\mathbf{A}_i, \mathbf{B}_i)$, together with the three algorithms explained above. Let us formulate these notions in a rigorous manner.

Definition 4.2. *An algebra $\mathbf{A} = (X_{\mathbf{A}}, F_{\mathbf{A}})$ is efficiently computable if for every $f^{\mathbf{A}} \in F_{\mathbf{A}}$ there exists a deterministic polynomial time algorithm A such that $f^{\mathbf{A}}(x_1, x_2, \dots, x_n) \leftarrow A(x_1, x_2, \dots, x_n)$ for every $x_1, x_2, \dots, x_n \in \mathbf{A}$, where n is the arity of $f^{\mathbf{A}}$.*

Definition 4.3. *Let \mathbf{A} and \mathbf{B} be algebras of the same type and let H be a countable index set. A set of homomorphisms $\mathcal{H} = \{\varphi_h : h \in H\} \subseteq \text{Hom}(\mathbf{A}, \mathbf{B})$ is efficiently computable if there is a deterministic polynomial time algorithm H such that $\varphi_h(x) \leftarrow H(h, x)$ for every $h \in H$ and $x \in \mathbf{A}$.*

Definition 4.4. Let I be a countably infinite index set. A collection of efficiently computable algebras is a countably infinite set of triples

$$\mathcal{C} = \{(\mathbf{A}_i, \mathbf{B}_i, \mathcal{H}_i) : i \in I\}$$

such that \mathbf{A}_i and \mathbf{B}_i are efficiently computable algebras and $\mathcal{H}_i \subseteq \text{Hom}(\mathbf{A}_i, \mathbf{B}_i)$ is a set of efficiently computable homomorphisms for every $i \in I$.

Definition 4.5. A family of algebras is a four-tuple

$$\mathbb{A} = (\mathcal{C}, \mathbf{S}, \mathbf{R}, \mathbf{H}),$$

where $\mathcal{C} = \{(\mathbf{A}_i, \mathbf{B}_i, \mathcal{H}_i) : i \in I\}$ is a collection of efficiently computable algebras, $\mathcal{H}_i = \{\varphi_h : h \in H_i\}$ and

1. $\mathbf{S}(1^s)$ is a PPT sampling algorithm such that given a security parameter 1^s outputs $(i, h, a_1, a_2, \dots, a_n) \leftarrow \mathbf{S}(1^s)$, where $i \in I, h \in H_i$ and $a_j \in \mathbf{A}_i$ for every $j \in \{1, 2, \dots, n\}$.
2. $\mathbf{R}(i, d, x_1, x_2, \dots, x_n)$ is a PPT random composition algorithm that given an index $i \in I$, a bit d determining whether we are composing elements of \mathbf{A}_i ($d = 0$) or \mathbf{B}_i ($d = 1$) and elements x_1, x_2, \dots, x_n of the corresponding algebra outputs a random element $x \leftarrow \mathbf{R}(i, d, x_1, x_2, \dots, x_n)$ such that

$$x \in \langle x_1, x_2, \dots, x_n \rangle$$

and

$$\varphi_h(\mathbf{R}(i, 0, z_1, z_2, \dots, z_n ; r)) = \mathbf{R}(i, 1, \varphi_h(z_1), \varphi_h(z_2), \dots, \varphi_h(z_n) ; r) \quad (1)$$

for every $i \in I, h \in H_i, z_1, z_2, \dots, z_n \in \mathbf{A}_i$ and every randomness r .

3. $\mathbf{H}(i, h, x)$ is a deterministic PT homomorphism computation algorithm that given $i \in I, h \in H_i$ and $x \in \mathbf{A}_i$, outputs $\varphi_h(x) \leftarrow \mathbf{H}(i, h, x)$.

The requirement (1) imposed on \mathbf{R} restricts it to respect the homomorphisms of the algebra. In general, it means that \mathbf{R} generates a random n -ary term p of the type such that \mathbf{R} can compute both term functions $p^{\mathbf{A}}$ and $p^{\mathbf{B}}$ in polynomial time. Then, depending on d , \mathbf{R} computes either $p^{\mathbf{A}}$ or $p^{\mathbf{B}}$.

Example 4.1. A group family is a family of algebras $\mathbb{G} = (\mathcal{C}, \mathbf{S}, \mathbf{R}, \mathbf{H})$, where $\mathcal{C} = (\mathbf{G}_i, \mathbf{G}_i, \text{End}(\mathbf{G}_i))$ is a collection of cyclic groups and

1. $(i, a, g) \leftarrow \mathbf{S}(1^s)$, where $i \in I$, g is a generator of \mathbf{G}_i and $a \leftarrow U(\mathbb{Z}_{|\mathbf{G}_i|})$,
2. $x^b \leftarrow \mathbf{R}(i, d, x)$, where $d \in \{0, 1\}, b \leftarrow U(\mathbb{Z}_{|\mathbf{G}_i|})$ and x^b is computed using the fast exponentiation algorithm,
3. $x^a \leftarrow \mathbf{H}(i, a, x)$, where x^a is computed using the fast exponentiation algorithm.

Let us consider the DH in the form of Def. 4.1. Alice obtains g^{ab} as the image under the endomorphism α . For an eavesdropper, the problem of computing g^{ab} from $(g, g^a, g^b) = (g, \alpha_a(g), g^b)$ can be seen as the problem of computing the image of g^b under an unknown endomorphism α_a . Therefore, we formulate an analogue for the computational DHP in the following manner: we give a set of elements and their homomorphic images under an unknown homomorphism. Then, we sample a random element x from the algebra and ask for its homomorphic image under the same homomorphism. We call this analogue of the DHP the homomorphic image problem (HIP).

Definition 4.6 (Computational HIP (CHIP)). *Let $\mathbb{A} = (\mathcal{C}, \mathcal{S}, \mathcal{R}, \mathcal{H})$ be a family of algebras and let $\mathcal{C} = \{(\mathbf{A}_i, \mathbf{B}_i, \mathcal{H}_i) : i \in I\}$. Suppose that $(i, h, a_1, a_2, \dots, a_n) \leftarrow \mathcal{S}(1^s)$ and $x \leftarrow \mathcal{R}(i, 0, a_1, a_2, \dots, a_n)$. Given*

$$i, (a_1, \varphi_h(a_1)), (a_2, \varphi_h(a_2)), \dots, (a_n, \varphi_h(a_n)), x,$$

compute $\varphi_h(x)$.

We can easily deduce a necessary condition for the infeasibility of the CHIP. Suppose that it is feasible to find a term p of the type such that the term function on a_1, a_2, \dots, a_n evaluates to x . Suppose also that the term function can be computed as a polynomial number of applications of the operations of \mathbf{A}_i . If such a factorization as a term is given, we can exchange each occurrence of a_j by $\varphi_h(a_j)$ and each occurrence of an operation of \mathbf{A}_i by the corresponding operation of \mathbf{B}_i . Since φ_h is a homomorphism, the image $\varphi_h(x)$ is then obtained by evaluating the obtained expression which can be done in polynomial time since \mathbf{B}_i is efficiently computable. Therefore, finding such a factorization as a term needs to be infeasible.

Definition 4.7 (Algebraic factorization problem (AFP)). *Let $\mathbb{A} = (\mathcal{C}, \mathcal{S}, \mathcal{R}, \mathcal{H})$ be a family of algebras of type τ . Let $(i, h, a_1, a_2, \dots, a_n) \leftarrow \mathcal{S}(1^s)$ and $y \leftarrow \mathcal{R}(i, 0, a_1, a_2, \dots, a_n)$. Find a term p of type τ such that the length of (the binary representation of) p is polynomial in i and*

$$y = p^{\mathbf{A}}(a_1, a_2, \dots, a_n).$$

The requirement for the polynomial length in i ensures that $p^{\mathbf{A}}$ can be evaluated in polynomial time. For the group family case, finding a factorization of g^a using the generator g is equivalent to the DLP.

Let us now formulate the decision version of the HIP.

Definition 4.8 (Decision HIP (DHIP)). *Let $\mathbb{A} = (\mathcal{C}, \mathcal{S}, \mathcal{R}, \mathcal{H})$ be a family of algebras and let $(i, h, a_1, a_2, \dots, a_n) \leftarrow \mathcal{S}(1^s)$, $x \leftarrow \mathcal{R}(i, 0, a_1, a_2, \dots, a_n)$ and $B \leftarrow U(\{0, 1\})$. Let the following be given:*

$$i, (a_1, \varphi_h(a_1)), (a_2, \varphi_h(a_2)), \dots, (a_n, \varphi_h(a_n)), (x, z),$$

where either

$$z = \varphi_h(x) \quad \text{if } B = 0,$$

or

$$z \leftarrow R(i, 1, \varphi_h(a_1), \varphi_h(a_2), \dots, \varphi_h(a_n)) \quad \text{if } B = 1.$$

Output B .

Note that when $B = 1$, R is run with fresh randomness. That is, we are either given the correct homomorphic image ($B = 0$) or a random element from $\langle \varphi_h(a_1), \varphi_h(a_2), \dots, \varphi_h(a_n) \rangle$ ($B = 1$) each with probability $1/2$. Let $S = \{S_s\}_{s \in \mathbb{N}}$ denote the probability ensemble corresponding to the choice of the string

$$(i, (a_1, \varphi_h(a_1)), (a_2, \varphi_h(a_2)), \dots, (a_n, \varphi_h(a_n)))$$

according to $(i, h, a_1, a_2, \dots, a_n) \leftarrow S(1^s)$ and let $X = \{X_s\}_{s \in \mathbb{N}}, Z = \{Z_s\}_{s \in \mathbb{N}}$ denote the probability ensembles corresponding to the choice of x and z according to $x \leftarrow R(i, 0, a_1, a_2, \dots, a_n)$ and $z \leftarrow R(i, 1, \varphi_h(a_1), \varphi_h(a_2), \dots, \varphi_h(a_n))$. If D is a probabilistic polynomial time algorithm, we define its DHIP-advantage on \mathbb{A} as

$$\text{Adv}_{D, \mathbb{A}}^{\text{DHIP}}(s) = |\Pr[1 \leftarrow D(1^s, S_s, (X_s, \varphi_h(X_s)))] - \Pr[1 \leftarrow D(1^s, S_s, (X_s, Z_s))]|.$$

Definition 4.9 (DHI-assumption). *A family of algebras \mathbb{A} satisfies the DHI-assumption if there is a negligible function ϵ such that*

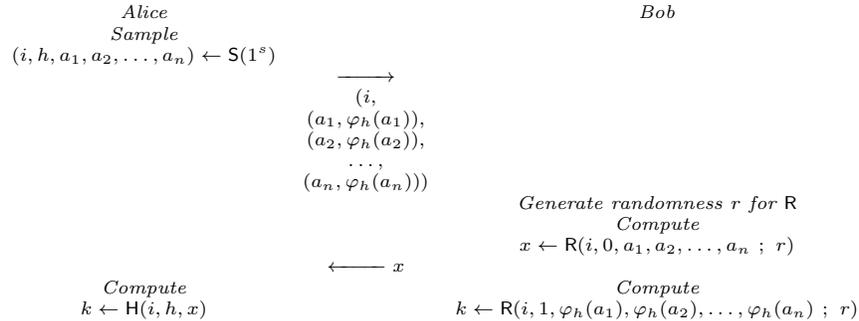
$$\text{Adv}_{\mathbb{A}}^{\text{DHIP}}(s) := \max_D \{ \text{Adv}_{D, \mathbb{A}}^{\text{DHIP}}(s) : D \text{ PPT} \} \leq \epsilon(s)$$

for every $s \in \mathbb{N}$.

For a group family \mathbb{G} , the DHI assumption is equivalent to the decision Diffie-Hellman assumption with the choice of S, R and H as in Example 4.1.

In the more general setting, the DH can be now written in the following form.

Definition 4.10 (Algebraically generalized Diffie-Hellman scheme (AGDH)). *Let the participants be Alice and Bob and let $\mathbb{A} = (\mathcal{C}, S, R, H)$ be a family of algebras.*



The secret randomness used by Alice is the index h of the homomorphism φ_h . For Bob, the secret randomness is the internal randomness r used by R .

Proposition 4.1. *AGDH is correct and the common element is indistinguishable from a randomly generated one under the DHI assumption.*

Proof. Correctness of the scheme follows from the homomorphic property of φ_h and the property (1) of R . If an eavesdropper observes the exchange of messages, she sees the index i and

$$(a_1, \varphi_h(a_1)), (a_2, \varphi_h(a_2)), \dots, (a_n, \varphi_h(a_n)) \text{ and } x,$$

which is an instance of the CHIP on \mathbb{A} . If \mathbb{A} satisfies the DHI-assumption, then an eavesdropper distinguishes $\varphi_h(x)$ from a random

$$y \leftarrow R(j, \varphi_h(a_1), \varphi_h(a_2), \dots, \varphi_h(a_n))$$

with only negligible probability. □ □

Comparing AGDH to DH we note that several properties of the platform algebra affect the performance of the scheme. For example, a large number of generators n results in a large number of transmitted elements from Alice to Bob. The optimal case is obtained with mono-generated algebras. In this regard, DH is optimal. On the other hand, contrary to DH, AGDH is not symmetric with respect to Alice and Bob. Asymmetry enables us to minimize the computational effort of Bob in a scenario where we want key exchange to be light-weight for one of the parties. Contrary to DH, where H and R essentially apply the same algorithm, in AGDH these can be different.

It is possible that, for some algebras, R can be made very efficient at the expense of S and H . For example, if the number of generators is large, then Alice needs to compute and communicate a large number homomorphic images. However, since the number of generators is large, Bob can reach a large number of different elements of the algebra with only a few applications of the finitary operations.

4.3 Potential instantiations

In this section, we offer some concrete examples of potential algebras for AGDH. To instantiate AGDH, the family of algebras has to support a large set of homomorphisms. We have identified four different approaches and described them below.

4.3.1 Homomorphic symmetric encryption schemes

For the AGDH, we need the computation of homomorphisms to be provably infeasible. If a symmetric encryption scheme is homomorphic in respect of some algebraic operation, its decryption algorithm induces a large set of functions that are homomorphisms from the ciphertext space to the plaintext space. Furthermore, if the scheme is provably secure, it is infeasible to compute these homomorphisms without a key. We will consider this approach more closely in Section 5.

4.3.2 Vector spaces

Vector spaces are a natural source for a large number of homomorphisms. If V is a finite dimensional vector space over a field \mathbb{F} , then $\text{End}(V)$ consists of all linear transformations $V \rightarrow V$ [40]. Linear transformations can be learned in polynomial time given uniformly random samples [30]. However, adding noise to the samples makes the problem infeasible. Noisy versions of problems based on linear transformations, such as learning parity with noise (LPN) and more generally learning with errors (LWE), have been utilized in several cryptographic constructions. Applying these problems in the instantiation of AGDH would lead to a scheme that bears similarities to lattice based key agreement schemes.

4.3.3 Left distributive groupoids

Let us consider the random composition algorithm R . Let $i \in I$, $h \in H_i$ and let the generators $a_1, a_2, \dots, a_n \in \mathbf{A}_i$ be fixed. For every randomness r used by R , let us define functions

$$R_r : \mathbf{A}_i \rightarrow \mathbf{A}_i, \quad R_r(x) \leftarrow R(i, 0, a_1, a_2, \dots, a_{n-1}, x ; r)$$

and

$$R'_r : \mathbf{B}_i \rightarrow \mathbf{B}_i, \quad R'_r(x) \leftarrow R(i, 1, \varphi_h(a_1), \varphi_h(a_2), \dots, \varphi_h(a_{n-1}), x ; r).$$

Then by (1),

$$a_n R_r \varphi_h = a_n \varphi_h R'_r \tag{2}$$

for every $h \in H_i$ and every randomness r . We saw that the hardness of solving the CHIP is based on the hardness of algebraically factoring $a_n R_r$ into a term p such that $p^{\mathbf{A}_i}$ and $p^{\mathbf{B}_i}$ are polynomial time computable without knowing r and the hardness of computing φ_h without h . Therefore, it seems useful to consider the case that both R_r and φ_h come from the same class of functions. This leads us naturally to the class of *left distributive (LD) groupoids* \mathbf{Q} that satisfy

$$a * (b * c) = (a * b) * (a * c)$$

for every $a, b, c \in \mathbf{Q}$.

Suppose that \mathbf{Q}_i is a LD groupoid and set $\mathbf{A}_i = \mathbf{B}_i = \mathbf{Q}_i$. Let $L_a(x) = a * x$ for every $a, x \in \mathbf{Q}_i$. The left distributivity property ensures that $L_a \in \text{End}(\mathbf{Q}_i)$ for every $a \in \mathbf{Q}_i$. Then, we can set both R and H to compute a series of such functions. The best known example of an LD structure arises from the conjugation operation $a * b = a^{-1}ba$ in a non-Abelian group G [76]. If we take for instance $\mathcal{H}_i \subseteq \langle L_a^* : a \in G_i \rangle$, then the hardness of the CHIP is closely related to the conjugacy problem on G . However, group conjugation is not the only possible source of left distributive groupoids. For example, such structures arise naturally in knot theory as a classifying invariant of a knot [44].

4.3.4 Medial groupoids

A groupoid \mathbf{Q} is *medial* (also called *entropic*) if

$$(a \cdot b) \cdot (c \cdot d) = (a \cdot c) \cdot (b \cdot d)$$

for every $a, b, c, d \in \mathbf{Q}$. For a medial groupoid the "squaring" function $e^2(x) = x \cdot x$ is an endomorphism of \mathbf{Q} . There is also a way of constructing new endomorphisms. For every $\alpha, \beta \in \text{End}(\mathbf{Q})$, let us define a function $\alpha + \beta$ by $(\alpha + \beta)(x) = \alpha(x) \cdot \beta(x)$. It follows from mediality that $\alpha + \beta \in \text{End}(\mathbf{Q})$ [31]. Therefore, there is a large set of efficiently computable endomorphisms of \mathbf{Q} whenever the binary operation of \mathbf{Q} is efficiently computable.

Medial operations can be induced by algebraic varieties and, in particular, algebraic plane curves that are good sources of a wide range of algebraic laws [54]. For example, the chord-tangent construction on a cubic plane curve defines a quasigroup operation that is medial [27]. The situation is depicted in Fig. 1. From this quasigroup operation the elliptic curve group law is also derived. However, mediality is not restricted to binary operations. It can be generalized to n-ary operations. Such algebras can be constructed, for example, by algebraic equations on fields [17].

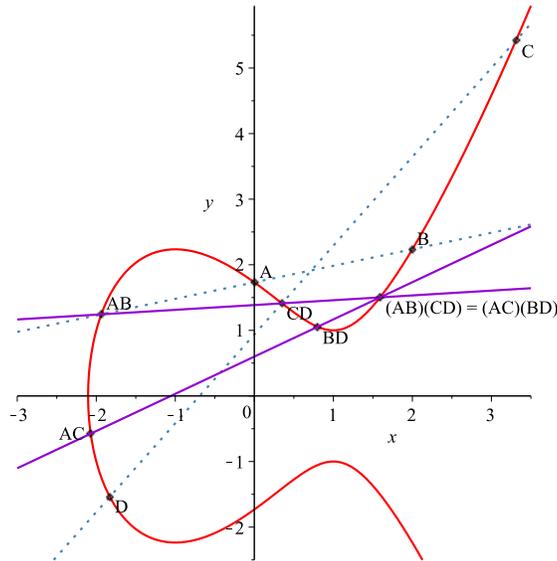


Figure 1: Medial quasigroup law on a cubic curve $y^2 = x^3 - 3x + 3$.

5 Symmetric homomorphic encryption and key exchange

In this section, we consider the question of turning a symmetric encryption scheme possessing homomorphic properties into a public-key primitive using the AGDH. If the encryption scheme is secure, then it is hard to compute images under the decryption functions without the key. Furthermore, if the decryption functions are homomorphisms with respect to some operation, then we have a natural candidate for the implementation of the AGDH. However, the hardness of decrypting is not sufficient for the induced key exchange to be secure. In addition, the underlying algebraic operation has to be sufficiently complex. In this section, we derive a condition such that there is an explicit construction for secure key exchange using the encryption scheme if the condition is satisfied. We call encryption schemes satisfying this condition *homomorphic key agreement capable*.

Let $\text{SE} = (\text{Gen}, \text{Enc}, \text{Dec})$ be an encryption scheme such that the decryption functions are homomorphic with respect to some operations on the ciphertext space C_s and the plaintext space M_s , where 1^s is the security parameter. In particular, suppose that SE is homomorphic from a finite algebra $\mathbf{C}_s = (C_s, F_{C_s})$ to a finite algebra $\mathbf{M}_s = (M_s, F_{M_s})$, where 1^s is the security parameter. Let the key space of SE be K_s . We stress that we do not require the scheme to be *fully homomorphic*. Instead, we only assume that there are non-trivial algebras of the same type on the ciphertext space C_s and on the plaintext space M_s such that the functions arising from decryption are homomorphisms $\mathbf{C}_s \rightarrow \mathbf{M}_s$. In addition, we do not require the scheme to be *strongly* homomorphic. An encryption scheme is called strongly homomorphic if it is possible to re-randomize ciphertexts without the secret key. Obviously, such schemes can be used for key transport.

An encryption scheme is malleable if, given a ciphertext, it is possible to generate a different ciphertext so that the two plaintexts are related [26]. A scheme that is homomorphic with respect to some operations is always malleable, since the homomorphic property enables us to derive related plaintexts. Due to malleability, it is impossible to achieve adaptive CCA-security (IND-CCA2), which is the standard notion of secure encryption, if we want to retain the homomorphic property [45]. It would be possible to achieve non-adaptive CCA-security (IND-CCA1), but, for our construction, CPA-security will be sufficient. It should be noted that standard transforms to convert CPA-secure schemes into CCA2-secure schemes, such as Naor-Yung double-encryption [60] or Fujisaki-Okamoto [32], can be applied when the scheme is used for encryption. However, our key exchange construction depends on homomorphic properties that will be destroyed by any such transform.

Let $\text{Adv}_{\text{A,SE}}^{\text{IND-CPA}}(s, n)$ denote the advantage of an adversary A in a CPA-experiment where A makes at most n queries to the encryption oracle. Since Dec is deterministic, each key $k \leftarrow \text{Gen}(1^s)$ determines a decryption homomorphism Dec_k from \mathbf{C}_s to \mathbf{M}_s . Let $\mathcal{D}_s = \{\text{Dec}_k : k \in K_s\}$ be the set of such functions

arising from Dec indexed by the keys $k \in K_s$. Let us consider a family of algebras $\mathbb{C} = (\mathcal{C}, \mathcal{S}, \mathcal{R}, \mathcal{H})$ such that

$$\mathbb{C} = \{(\mathbf{C}_s, \mathbf{M}_s, \mathcal{D}_s) : s \in \mathbb{N}\}.$$

Depending on the operations F_{C_s} and F_{M_s} , there could be many possible algorithms for randomly composing elements. Therefore, our results will be stated in terms of the choice of \mathcal{R} . Let us fix the other two required algorithms:

1. Sampling algorithm $\mathcal{S}(1^s)$: Sample $k_s \leftarrow \text{Gen}(1^s)$. Sample n_s distinct generators m_1, m_2, \dots, m_{n_s} from \mathbf{M}_s . Compute $a_t \leftarrow \text{Enc}(k_s, m_t)$ for every $t \in \{1, 2, \dots, n_s\}$. Output $(s, k_s, a_1, a_2, \dots, a_{n_s})$.
2. Homomorphism computation algorithm $\mathcal{H}(s, k_s, x)$: Output $z \leftarrow \text{Dec}(k_s, x)$.

In the following, we will be using probability ensembles on the key space and the plaintext space, as well as two ensembles on the ciphertext space. These have been defined below.

Definition 5.1. Let $(s, k_s, a_1, a_2, \dots, a_{n_s}) \leftarrow \mathcal{S}(1^s)$ and let $m_i \leftarrow \text{Dec}(k_s, a_i)$ for every $i \in \{1, 2, \dots, n_s\}$.

1. The key ensemble $K = \{K_s\}_{s \in \mathbb{N}}$ is the probability ensemble such that $K_s = \text{Gen}(1^s)$.
2. The random plaintext composition ensemble $Z = \{Z_s\}_{s \in \mathbb{N}}$ is the probability ensemble such that $Z_s = \mathcal{R}(s, 1, m_1, m_2, \dots, m_{n_s})$.
3. The random ciphertext composition ensemble $R = \{R_s\}_{s \in \mathbb{N}}$ is the probability ensemble such that $R_s = \mathcal{R}(s, 0, a_1, a_2, \dots, a_{n_s})$.
4. The encryption ensemble $E = \{E_s\}_{s \in \mathbb{N}}$ is the probability ensemble such that $E_s = \text{Enc}(K_s, Z_s)$.

If SE has indistinguishable encryptions there is a probability ensemble $X = \{X_s\}_{s \in \mathbb{N}}$ such that the probability ensemble $\text{Enc}(K_s, Y_s)$ is computationally indistinguishable from X for every efficiently samplable probability ensemble $Y = \{Y_s\}_{s \in \mathbb{N}}$ on M_s . Typically, X is the uniform probability ensemble U . However, we do not place such a restriction on X . We will be considering the random ciphertext composition ensemble R and show that the DHI-assumption holds whenever R is computationally indistinguishable from X . Let us first consider a modified version of the DHIP, which we denote by DHIP^Y , where R is replaced by Y . That is, for an instance of the DHIP^Y ,

$$s, (a_1, \text{Dec}(k_s, a_1)), (a_2, \text{Dec}(k_s, a_2)), \dots, (a_n, \text{Dec}(k_s, a_n)), (x, z),$$

we have $x \leftarrow Y_s$ instead of R_s .

Ultimately, our goal is to relate the hardness of the DHIP to the security of SE. We first bound the difference $|\text{Adv}_{\mathcal{A}, \mathcal{C}}^{\text{DHIP}}(s) - \text{Adv}_{\mathcal{A}, \mathcal{C}}^{\text{DHIP}^Y}(s)|$ based on the problem of distinguishing R and Y . It will help us later to achieve negligibility of $\text{Adv}_{\mathcal{A}, \mathcal{C}}^{\text{DHIP}}(s)$ with a proper choice of R and Y .

Proposition 5.1. *For every PPT algorithm A and every probability ensemble Y on the ciphertext space there is a PPT algorithm B such that*

$$\mathbf{Adv}_B^{D(R,Y)}(s) \geq \frac{1}{2} \left| \mathbf{Adv}_{A,C}^{\text{DHIP}}(s) - \mathbf{Adv}_{A,C}^{\text{DHIP}^Y}(s) \right|$$

for every $s \in \mathbb{N}$.

Proof. Let A be a PPT algorithm considered as a distinguisher for DHIP or DHIP^Y. We construct an algorithm B that applies A to distinguish between Y and R :

```

1: procedure  $B(1^s, x)$ 
2:    $(s, k_s, a_1, a_2, \dots, a_{n_s}) \leftarrow S(1^s)$ 
3:    $m_t \leftarrow H(s, k_s, a_t)$  for every  $t \in \{1, 2, \dots, n_s\}$ 
4:    $b \leftarrow U(\{0, 1\})$ 
5:   if  $b = 0$  then
6:      $z \leftarrow H(s, k_s, x)$ 
7:      $b' \leftarrow A(1^s, s, (a_1, m_1), (a_2, m_2), \dots, (a_{n_s}, m_{n_s}), (x, z))$ 
8:     output  $b'$ 
9:   else
10:     $z \leftarrow R(s, 1, m_1, m_2, \dots, m_{n_s})$ 
11:     $b' \leftarrow A(1^s, s, (a_1, m_1), (a_2, m_2), \dots, (a_{n_s}, m_{n_s}), (x, z))$ 
12:    output  $\bar{b}'$ 
13:   end if
14: end procedure

```

Let $S = \{S_s\}_{s \in \mathbb{N}}$ denote the probability ensemble corresponding to the choice of the string k , $(a_1, m_1), (a_2, m_2), \dots, (a_{n_s}, m_{n_s})$. By the description of B , the input to A is a valid instance of either DHIP ($x \leftarrow R_s$) or DHIP^Y ($x \leftarrow Y_s$). In addition, if $b = 0$, the homomorphic image of x is z , otherwise a random element Z_s . Both of these cases happen with probability $1/2$. Therefore,

$$\begin{aligned}
\Pr[1 \leftarrow B(1^s, R_s)] &= \frac{1}{2} (\Pr[1 \leftarrow B(1^s, R_s) | b = 0] \\
&\quad + \Pr[1 \leftarrow B(1^s, R_s) | b = 1]) \\
&= \frac{1}{2} (\Pr[1 \leftarrow A(1^s, S_s, (R_s, z))] \\
&\quad + \Pr[0 \leftarrow A(1^s, S_s, (R_s, Z_s))]) \\
&= \frac{1}{2} (1 + \Pr[1 \leftarrow A(1^s, S_s, (R_s, z))] \\
&\quad - \Pr[1 \leftarrow A(1^s, S_s, (R_s, Z_s))]) \\
&= \frac{1}{2} \left(1 + (-1)^e \mathbf{Adv}_{A,C}^{\text{DHIP}}(s) \right)
\end{aligned}$$

Note that both x_0 and x_1 are sampled according to Z_s . Since the challenge ciphertext $c_b \leftarrow \text{Enc}(k_s, Z_s)$, it is sampled according to E . If $b = 0$, then x_0 is the homomorphic image of c_b . If $b = 1$, x_0 is a random element sampled according to Z . Therefore, the input to \mathbf{A} is a valid instance of DHIP^E and \mathbf{A} succeeds with advantage $\mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s)$. Since \mathbf{B} outputs the same bit as \mathbf{A} ,

$$\mathbf{Adv}_{\mathbf{B}, \text{SE}}^{\text{IND-CPA}}(s, n_s) = \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s).$$

□

We are now ready to derive a bound on the DHIP. We achieve this by considering the indistinguishability of E and R . Intuitively DHIP^E and DHIP are both hard if E and R are indistinguishable. This is formalized in the following proposition.

Proposition 5.3. *For every PPT algorithm \mathbf{A} ,*

$$\mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) \leq 2 \cdot \mathbf{Adv}^{D(R, E)}(s) + \mathbf{Adv}_{\text{SE}}^{\text{IND-CPA}}(s, n_s)$$

Proof. Let \mathbf{A} be a PPT algorithm. Suppose that

$$\mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s) \geq \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s).$$

Then by Proposition 5.2 there is a PPT algorithm \mathbf{B} such that

$$\mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) \leq \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s) \leq \mathbf{Adv}_{\mathbf{B}, \text{SE}}^{\text{IND-CPA}}(s, n_s) \leq \mathbf{Adv}_{\text{SE}}^{\text{IND-CPA}}(s, n_s).$$

Therefore, we may assume that

$$\mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) \geq \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s).$$

By Proposition 5.1 there is a PPT algorithm \mathbf{B} such that

$$\begin{aligned} \left| \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) - \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s) \right| &= \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) - \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s) \\ &\leq 2 \cdot \mathbf{Adv}_{\mathbf{B}}^{D(R, E)}(s). \end{aligned}$$

But now, by Proposition 5.2 there is a PPT algorithm \mathbf{C} such that

$$\begin{aligned} \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}}(s) &\leq 2 \cdot \mathbf{Adv}_{\mathbf{B}}^{D(R, E)}(s) + \mathbf{Adv}_{\mathbf{A}, \mathbf{C}}^{\text{DHIP}^E}(s) \\ &\leq 2 \cdot \mathbf{Adv}_{\mathbf{B}}^{D(R, E)}(s) + \mathbf{Adv}_{\text{SE}, \mathbf{C}}^{\text{IND-CPA}}(k, n_s) \\ &\leq 2 \cdot \mathbf{Adv}^{D(R, E)}(s) + \mathbf{Adv}_{\text{SE}}^{\text{IND-CPA}}(k, n_s). \end{aligned}$$

□

As a corollary, we obtain the following result on the infeasibility of the DHIP.

Proposition 5.4. *If SE is IND-CPA secure and the random ciphertext composition ensemble R is computationally indistinguishable from the encryption ensemble E , then \mathbb{C} satisfies the DHI-assumption.*

Proposition 5.4 asserts that AGDH can be instantiated using a symmetric encryption scheme if the underlying algebra admits a suitably complex random composition algorithm R . This motivates the following definition for a symmetric encryption scheme SE.

Definition 5.2 (Homomorphic key agreement capable). *Let $SE = (\text{Gen}, \text{Enc}, \text{Dec})$ be an IND-CPA secure symmetric encryption scheme. If there exists a family of algebras $\mathbb{C} = (\mathcal{C}, \mathcal{S}, \mathcal{R}, \mathcal{H})$ such that $\mathcal{H}(s, k_s, x) = \text{Dec}(k_s, x)$ for every key k_s and every plaintext message x and the probability ensemble R induced by*

$$R(s, 0, a_1, a_2, \dots, a_{n_s})$$

with $(s, k_s, a_1, a_2, \dots, a_{n_s}) \leftarrow \mathcal{S}(1^s)$ is computationally indistinguishable from the probability ensemble E induced by $\text{Enc}(k_s, x)$ for

$$x \leftarrow R(s, 1, \text{Dec}(k_s, a_1), \text{Dec}(k_s, a_2), \dots, \text{Dec}(k_s, a_{n_s})),$$

then SE is called homomorphic key agreement capable.

In general, a key agreement capable symmetric encryption scheme can be always transformed into a public-key primitive using AGDH for key exchange. The resulting protocol is secure by Proposition 5.4.

6 Conclusions

We propose a universal algebraic generalization of the Diffie-Hellman scheme called AGDH. Its security is based on the hardness of a homomorphic image problem which requires the adversary to compute the image of a given element under an unknown homomorphism from an algebra \mathbf{A} to \mathbf{B} . We rigorously formulate computational and decision versions of this problem. AGDH provides a method of considering different algebraic structures for key exchange without placing structural restrictions on them. The study offers potential for the development of new algebraic key exchange schemes. We also identified four interesting approaches to instantiate the AGDH and pursued one of these options by considering the instantiation of AGDH using symmetric encryption schemes that are homomorphic over algebraic operations. We formulated a condition called homomorphic key agreement capability and showed that an IND-CPA secure scheme that satisfies this condition can be securely used for key exchange, essentially turning the symmetric scheme into a public-key primitive.

References

- [1] M. R. Albrecht, P. Farshim, D. Hofheinz, E. Larraia, and K. G. Paterson. Multilinear maps from obfuscation. In E. Kushilevitz and T. Malkin, editors, *Theory of Cryptography: 13th International Conference, TCC 2016-A*,

- Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 446–473, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- [2] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange—a new hope. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, Austin, TX, 2016. USENIX Association.
 - [3] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld. New key agreement protocols in braid group cryptography. In *Topics in cryptology—CT-RSA 2001*, volume 2020 of *Lecture Notes in Comput. Sci.*, pages 13–27. Springer, Berlin, 2001.
 - [4] I. Anshel, M. Anshel, and D. Goldfeld. An algebraic method for public-key cryptography. *Math. Res. Lett.*, 6(3-4):287–291, 1999.
 - [5] G. Baumslag, T. Camps, B. Fine, G. Rosenberger, and X. Xu. Designing key transport protocols using combinatorial group theory. *Contemporary mathematics*, 418:35, 2006.
 - [6] D. Boneh. The decision Diffie-Hellman problem. In J. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0054851.
 - [7] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *Advances in Cryptology — CRYPTO 2001: 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19–23, 2001 Proceedings*, pages 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
 - [8] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2002.
 - [9] D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2014, Proceedings, Part I*, pages 480–499, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
 - [10] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from lwe. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS ’16*, pages 1006–1018, New York, NY, USA, 2016. ACM.
 - [11] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570, May 2015.

- [12] A. Brouwer, R. Pellikaan, and E. Verheul. Doing more with fewer bits. In K.-Y. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – ASIACRYPT 1999*, volume 1716 of *Lecture Notes in Computer Science*, pages 321–332. Springer Berlin / Heidelberg, 1999. 10.1007/978-3-540-48000-6_26.
- [13] J. Buchmann and H. Williams. A key exchange system based on real quadratic fields. In G. Brassard, editor, *Advances in Cryptology – CRYPTO ’89*, volume 435 of *Lecture Notes in Computer Science*, pages 335–343. Springer New York, 1990.
- [14] S. Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.
- [15] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the multilinear map over the integers. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 3–12, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [16] J. H. Cheon and B. Jun. A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem. In *Advances in cryptology—CRYPTO 2003*, volume 2729 of *Lecture Notes in Comput. Sci.*, pages 212–225. Springer, Berlin, 2003.
- [17] J. R. Cho. Idempotent medialn-groupoids defined on fields. *Algebra Universalis*, 25(1):235–246, Dec 1988.
- [18] D. Coppersmith, A. Odlyzko, and R. Schroepfel. Discrete logarithms in $\text{GF}(p)$. *Algorithmica*, 1:1–15, 1986. 10.1007/BF01840433.
- [19] J.-S. Coron, M. S. Lee, T. Lepoint, and M. Tibouchi. Cryptanalysis of ggh15 multilinear maps. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 607–628, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- [20] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 476–493, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [21] P. Dehornoy. Braid-based cryptography. *Contemporary Mathematics*, 360:5–33, 2004.

- [22] R. del Pino, V. Lyubashevsky, and D. Pointcheval. The whole is less than the sum of its parts: Constructing more efficient lattice-based schemes. In V. Zikas and R. De Prisco, editors, *Security and Cryptography for Networks: 10th International Conference, SCN 2016, Amalfi, Italy, August 31 – September 2, 2016, Proceedings*, pages 273–291, Cham, 2016. Springer International Publishing.
- [23] J.-C. Deneuville, P. Gaborit, and G. Zémor. Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography : 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*, pages 18–34, Cham, 2017. Springer International Publishing.
- [24] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
- [25] J. Ding, S. Alsayigh, J. Lancrenon, S. RV, and M. Snook. Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In H. Handschuh, editor, *Topics in Cryptology – CT-RSA 2017: The Cryptographers’ Track at the RSA Conference 2017, San Francisco, CA, USA, February 14–17, 2017, Proceedings*, pages 183–204, Cham, 2017. Springer International Publishing.
- [26] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Review*, 45(4):727–784, 2003.
- [27] I. M. H. Etherington. Quasigroups and cubic curves. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 14:273–291, 12 1965.
- [28] B. Fefferman, R. Shaltiel, C. Umans, and E. Viola. On beating the hybrid argument. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS ’12*, pages 468–483, New York, NY, USA, 2012. ACM.
- [29] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.
- [30] A. Frieze. Learning linear transformations. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS ’96*, pages 359–359, Washington, DC, USA, 1996. IEEE Computer Society.
- [31] O. Frink. Symmetric and self-distributive systems. *The American Mathematical Monthly*, 62(10):pp. 697–707, 1955.
- [32] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, Jan 2013.
- [33] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology*

- *EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 1–17, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [34] C. Gentry, S. Gorbunov, and S. Halevi. Graph-induced multilinear maps from lattices. In Y. Dodis and J. B. Nielsen, editors, *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 498–527, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [35] D. Grigoriev and V. Shpilrain. Tropical cryptography. *Communications in Algebra*, 42(6):2624–2632, 2014.
- [36] M. Habeeb, D. Kahrobaei, C. Koupparis, and V. Shpilrain. Public key exchange using semidirect product of (semi)groups. In M. Jacobson, M. Lo-casto, P. Mohassel, and R. Safavi-Naini, editors, *Applied Cryptography and Network Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 475–486. Springer Berlin Heidelberg, 2013.
- [37] M. Habeeb, D. Kahrobaei, and V. Shpilrain. A public key exchange using semidirect products of groups. *Proceedings of the International Conference in Symbolic Computations and Cryptography*, pages 137–141, 2010.
- [38] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. P. Buhler, editor, *Algorithmic Number Theory: Third International Symposium, ANTS-III Portland, Oregon, USA, June 21–25, 1998 Proceedings*, pages 267–288, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- [39] Y. Hu and H. Jia. Cryptanalysis of ggh map. In M. Fischlin and J.-S. Coron, editors, *Advances in Cryptology – EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 537–565, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
- [40] J. E. Humphreys. *Introduction to Lie algebras and representation theory*, volume 9. Springer, 1972.
- [41] D. Jao and L. De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies Post-Quantum Cryptography. In B.-Y. Yang, editor, *Post-Quantum Cryptography*, volume 7071, pages 19–34. Springer Berlin / Heidelberg, 2011.
- [42] X. L. Jintai Ding, Xiang Xie. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. <http://eprint.iacr.org/2012/688>.

- [43] A. Joux. A one round protocol for tripartite diffie–hellman. In W. Bosma, editor, *Algorithmic Number Theory: 4th International Symposium, ANTS-IV Leiden, The Netherlands, July 2-7, 2000. Proceedings*, pages 385–393, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- [44] D. Joyce. A classifying invariant of knots, the knot quandle. *J. Pure Appl. Algebra*, 23(1):37–65, 1982.
- [45] J. Katz and M. Yung. Characterization of security notions for probabilistic private-key encryption. *Journal of Cryptology*, 19:67–95, 2006. 10.1007/s00145-005-0310-8.
- [46] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J.-s. Kang, and C. Park. New public-key cryptosystem using braid groups. In *Advances in cryptology—CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 166–183. Springer, Berlin, 2000.
- [47] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.
- [48] N. Koblitz. Hyperelliptic cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.
- [49] A. Lenstra. Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields. In V. Varadharajan, J. Pieprzyk, and Y. Mu, editors, *Information Security and Privacy*, volume 1270 of *Lecture Notes in Computer Science*, pages 126–138. Springer Berlin / Heidelberg, 1997. 10.1007/BFb0027920.
- [50] A. Lenstra and E. Verheul. The XTR public key system. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 1–19. Springer Berlin / Heidelberg, 2000. 10.1007/3-540-44598-6_1.
- [51] R. Lidl and W. Müller. Permutation polynomials in rsa-cryptosystems. In *Advances in Cryptology – CRYPTO 1984*, volume 83, pages 293–301, 1984.
- [52] G. Maze. *Algebraic Methods for Constructing One-way Trapdoor Functions*. PhD thesis, University of Notre Dame, 2003.
- [53] G. Maze, C. Monico, and J. Rosenthal. Public key cryptography based on semigroup actions. *Advances in Mathematics of Communications*, 1:489–507, 2007.
- [54] W. McCune and R. Padmanabhan. *Automated Deduction in Equational Logic and Cubic Curves*. Lecture Notes in Computer Science. Springer, 1996.
- [55] R. C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, Apr. 1978.

- [56] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in cryptology—CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, New York, NY, USA, 1986. Springer-Verlag New York, Inc.
- [57] C. Monico. *Semirings and semigroup actions in public-key cryptography*. PhD thesis, University of Notre Dame, 2002.
- [58] W. Müller. Polynomial functions in modern cryptology. *Contributions to General Algebra*, 3:7–32, 1985.
- [59] W. Müller and R. Nöbauer. Cryptanalysis of the Dickson-scheme. In F. Pichler, editor, *Advances in Cryptology – EUROCRYPT 1985*, volume 219 of *Lecture Notes in Computer Science*, pages 50–61. Springer Berlin / Heidelberg, 1986. 10.1007/3-540-39805-8_7.
- [60] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 427–437, New York, NY, USA, 1990. ACM.
- [61] J. Partala. Key agreement based on homomorphisms of algebraic structures. Cryptology ePrint Archive, Report 2011/203, 2011.
- [62] J. Partala. *Algebraic methods for cryptographic key exchange*. PhD thesis, University of Oulu, 2015.
- [63] J. Partala. Left conjugacy closed left quasigroups with pairwise distinct left translations. *JP Journal of Algebra, Number Theory and Applications*, 36:95–108, 2016.
- [64] J. Partala and T. Seppänen. On the conjugacy search problem and left conjugacy closed loops. *Appl. Algebra Engrg. Comm. Comput.*, 19(4):311–322, 2008.
- [65] C. Peikert. Lattice cryptography for the internet. In M. Mosca, editor, *Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, pages 197–219. Springer International Publishing, Cham, 2014.
- [66] C. Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, Mar. 2016.
- [67] M. Rabi and A. T. Sherman. Associative one-way functions: A new paradigm for secret-key agreement and digital signatures. Technical report, University of Maryland, College Park, MD, USA, 1993.
- [68] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

- [69] K. Rubin and A. Silverberg. Torus-based cryptography. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 349–365. Springer Berlin Heidelberg, 2003.
- [70] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991. 10.1007/BF00196725.
- [71] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, October 1997.
- [72] V. Shpilrain and A. Ushakov. A new key exchange protocol based on the decomposition problem. *Contemporary Mathematics*, 418:161–167, 2006.
- [73] V. Shpilrain and G. Zapata. Combinatorial group theory and public key cryptography. *Applicable Algebra in Engineering, Communication and Computing*, 17:291–302, 2006.
- [74] V. Sidel'nikov, M. Cherepnev, and V. Yashchenko. Systems of open distribution of keys on the basis of noncommutative semigroups. *Russian Academy of Sciences-Doklady Mathematics*, 48:384–386, 1994.
- [75] P. Smith and C. Skinner. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology – ASIACRYPT 1994*, volume 917 of *Lecture Notes in Computer Science*, pages 355–364. Springer Berlin / Heidelberg, 1995. 10.1007/BFb0000447.
- [76] D. Stanovský. *Left distributive left quasigroups*. PhD thesis, Charles University in Prague, 2004.
- [77] E. Stickel. A new method for exchanging secret keys. In *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, volume 2, pages 426–430, July 2005.
- [78] A. Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Advances in Mathematics of Communications*, 4:215–235, 2010.
- [79] B. Tsaban. Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography. *Journal of Cryptology*, pages 1–22, 2013.
- [80] E. R. Verheul. Evidence that xtr is more secure than supersingular elliptic curve cryptosystems. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings*, pages 195–210, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

- [81] L. Wang, L. Wang, Z. Cao, E. Okamoto, and J. Shao. New constructions of public-key encryption schemes from conjugacy search problems. In X. Lai, M. Yung, and D. Lin, editors, *Information Security and Cryptology*, volume 6584 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg, 2011.
- [82] D. Xiao, X. Liao, and K. Wong. An efficient entire chaos-based scheme for deniable authentication. *Chaos, Solitons & Fractals*, 23(4):1327 – 1331, 2005.
- [83] T. Yamakawa, S. Yamada, G. Hanaoka, and N. Kunihiro. Self-bilinear map on unknown order groups from indistinguishability obfuscation and its applications. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology – CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 90–107, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [84] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen. Authenticated key exchange from ideal lattices. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 719–751, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.