Key agreement based on homomorphisms of algebraic structures

Juha Partala
Computer Science and Engineering laboratory,
University of Oulu, Finland
juha.partala@ee.oulu.fi

Abstract

We give a generalization of the Diffie-Hellman key agreement scheme that is based on the hardness of computing homomorphic images from an algebra to another. We formulate computational and decision versions of the homomorphic image problem and devise a key agreement protocol that is secure in the Canetti-Krawczyk model under the decision homomorphic image assumption. We also give an instantiation of the protocol using an additively homomorphic symmetric encryption scheme of Armknecht and Sadeghi. We prove that the instantiation is secure under the assumption that the encryption scheme is IND-CPA secure.

Keywords: cryptography, key exchange, session key agreement, algebraic system, universal algebra

1 Introduction

Key agreement protocols allow two or more parties to derive a common secret key over an adversially controlled channel. The derived key can be used for example to establish a secure channel with a symmetric encryption scheme. Diffie-Hellman key agreement [12] is one of the most widely used key agreement protocols. It allows two parties to agree on a common session key without sharing any secrets in advance. The original Diffie-Hellman scheme is based on exponentiation of integers modulo a prime. Its security is based on the hardness of computing discrete logarithms. Since the seminal paper of Diffie and Hellman, sub-exponential time algorithms have been devised for the discrete logarithm problem [9]. To avert such methods, generalizations the original scheme have been studied. For example, in elliptic curve Diffie-Hellman key agreement [20, 17] the platform group has been changed into a cyclic group arising from the group structure over an elliptic curve.

Algebraic generalizations of the Diffie-Hellman scheme can be constructed based on the properties of the original scheme. For example, the commutativity of exponentiation enables us to derive common values between two parties in an obvious way. The commutativity is the principal property in many schemes that can be considered as generalizations of the Diffie-Hellman scheme such as pairing based key agreement over algebraic curves [14]. On the other hand, exponentiation in a cyclic group can be also seen as a group automorphism. Several discrete logarithm based primitives can be characterized by considering a group of automorphisms acting on a group [24]. Swapping exponentiation with conjugation yields schemes that work on non-commutative groups [16, 23]. Such generalizations typically concentrate on the commutativity rather than on the homomorphic property. In this paper, we show that if the homomorphic property is satisfied, then the commutativity is not needed. We formulate a generalization of the Diffie-Hellman scheme that can be considered, in the homomorphic sense, the most general possible. Instead of group automorphisms, we have a set of homomorphisms from a finitely generated algebra to another. Our construction is based on the presumed infeasibility of a *homomorphic image problem* (HIP) that asks for the image of a given element under an unknown homomorphism. The homomorphisms do not need to commute in our scheme and there are not any structural requirements for the algebras other than the infeasibility of the HIP.

We formulate both a computational and a decision version of the HIP and devise a key agreement protocol that is secure in the unauthenticated links model formalized by Bellare et al. [4] and later extended by Canetti and Krawczyk [7]. We prove the security under the decision homomorphic image assumption. We also give an example construction of our scheme based on a symmetric encryption scheme of Armknecht and Sadeghi [3] that is additively homomorphic over a vector space. We show that the protocol is secure in the Canetti-Krawczyk model whenever the IND-CPA security assumption of the encryption scheme holds.

1.1 Related work

Many key agreement schemes can be classified under the algebraic generalizations of the Diffie-Hellman scheme. Naturally, it is possible to replace $(\mathbb{Z}/p\mathbb{Z})^*$ with another cyclic group if computing discrete logarithms is infeasible for that group. An example of such a generalization is the elliptic curve Diffie-Hellman key agreement scheme [20, 17]. Other groups over Abelian varieties can be also used [18].

Conjugacy search problem is one of the possible generalizations of the discrete logarithm problem to non-commutative groups. Commuting automorphisms have both the commutativity and the homomorphic property of the exponentiation operation. In [16], Ko et al. suggest a Diffie-Hellman like scheme using the braid group and commuting inner automorphisms. According to Dehornoy [11], the same scheme has been independently suggested by Sidel'nikov et al. using a non-commutative semigroup [25].

In [24], the possibility of using commuting endomorphisms instead of automorphisms is suggested. In [24], Shpilrain and Zapata also algebraically classify exponentiation and conjugation based schemes using a group action. Shpilrain and Zapata also give a general definition for an algebraic public-key cryptographic system based on a one-way group action [24, Definition 1]. They suggest a generalization of the Diffie-Hellman scheme using commuting semigroup actions. To the best of our knowledge, semigroup actions were first suggested by Monico [21]. Similar schemes can be found in [19, 26].

We are not aware of any generalizations based on non-commuting homomorphisms. A related algebraic key agreement scheme is suggested by Anshel et al. in [2]. The scheme is based on three special mappings $\beta, \gamma_1, \gamma_2$ defined using two monoids. Of the three functions, β has a homomorphic property. In [2, 1], Anshel et al. suggest conjugation and the braid group as the basis for a key agreement protocol. Although the scheme of Anshel et al. is not a generalization of the Diffie-Hellman scheme, it bears many similarities to the scheme suggested in this paper. Namely, we choose generators for an algebra and compute their images under a secret homomorphism. The security of the scheme depends on finding a factorization of a given element in terms of the generators of the algebra. However, for the scheme of Anshel et al. the functions γ_1 and γ_2 need to satisfy special properties and the underlying algebra is a monoid. Such requirements are not needed for our scheme.

1.2 Organization

The organization of the paper is the following. Section 2 contains the necessary preliminaries for the rest of the paper. In Section 3, we give a basic representation of our scheme. We show that the scheme is secure against fully recovering the common element in the setting of eavesdroppers whenever the computational version of the homomorphic image problem (CHIP) is infeasible. We also show that the CHIP is reducible to the so called factorization problem that asks for a representation of a given element in terms of the generators and the operations of the algebra. We also show that the original Diffie-Hellman scheme is a special case of our construction.

In Section 4, we formulate a decision version of the HIP. Based on the infeasibility of this problem, we devise a key agreement protocol that is secure in the Canetti-Krawczyk model [7]. Finally, in Section 5 we give an instantiation of the protocol based on an IND-CPA secure additively homomorphic symmetric encryption scheme suggested by Armknecht and Sadeghi [3]. We use the encryption scheme to implement the set of homomorphisms. The infeasibility of the decision HIP follows from the IND-CPA security and a certain ciphertext re-randomization property of the Armknecht-Sadeghi scheme.

2 Preliminaries

2.1 Universal algebra

An *algebraic language* $L = (O, r)$ is a first order language that consists of a set of operation symbols O together with a function r, which assigns a non-negative integer $r(s)$, the arity of s, for each element $s \in O$. A finitary operation $f^{\mathbf{A}}$ is a function $X^m \to X$, where m is the arity of $f^{\mathbf{A}}$. A model of L,

$$\mathbf{A} = (X, f^{\mathbf{A}}\ (f \in O)),$$

is called an *algebra* (also an *algebraic system*). That is, an algebra is a system $\mathbf{A} = (X, F)$, where X is a non-empty set and F is a set of finitary operations on $\mathbf{A}$. The *type* of an algebra is its language.

Let $\mathbf{A}, \mathbf{B}$ be algebras of the same type. A mapping $h : \mathbf{A} \to \mathbf{B}$ is a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if for every operation symbol f of the type and $x_1, x_2, \ldots, x_{r(f)} \in \mathbf{A}$,

$$h(f^{\mathbf{A}}(x_1, x_2, \ldots, x_{r(f)})) = f^{\mathbf{B}}(h(x_1), h(x_2), \ldots, h(x_{r(f)})).$$

An *endomorphism* of $\mathbf{A}$ is a homomorphism from $\mathbf{A}$ to itself. The set of endomorphisms comprises a semigroup $\mathrm{End}\,(\mathbf{A})$. An *automorphism* is a bijective endomorphism and the set of automorphisms comprises a group $\mathrm{Aut}\,(\mathbf{A})$.

For a treatise on universal algebra, see for example [6].

2.2 Diffie-Hellman key agreement

The Diffie-Hellman scheme [12] is a two-party key agreement scheme that, in its general form, allows two parties (say Alice and Bob) to derive a common secret element in the following way. Let Alice and Bob agree on a finite cyclic group G and a generator $g \in G$. Both randomly choose $a, b \in \{1, 2, \ldots, |G|\}$ as their private keys, respectively, and exchange g^a and g^b. The common secret element is g^{ab}. The *computational Diffie-Hellman problem* (CDHP) asks to compute g^{ab} given g^a and g^b. An eavesdropper cannot compute g^{ab} whenever the CDHP is infeasible.

The function $g \mapsto g^a$ is an automorphism of G for every $a \in \{1, 2, \ldots, |G|\}$. We can therefore see the Diffie-Hellman scheme as a group of commuting automorphisms $K \leq \mathrm{Aut}\,(G)$ acting on G. Alice and Bob choose random automorphisms α and β as their private keys, respectively, and exchange $\alpha(g)$ and $\beta(g)$. The common key is established as $\alpha(\beta(g)) = \beta(\alpha(g))$. It has to be infeasible to deduce $\alpha(\beta(g))$ from $\alpha(g)$ and $\beta(g)$.

2.3 Formal security of key agreement protocols

A general framework for considering the security of session-based multiparty protocols has been formalized by Bellare et al. in [4]. We will follow an extended formalism that was introduced by Canetti and Krawczyk in [7]. In a two-party key agreement protocol, two principals communicate to establish a secret shared session key. Each party (denoted by P_i) is modeled as a probabilistic polynomial time algorithm. By efficient and feasible computations we refer to probabilistic polynomial time computation. A function ϵ is *negligible* if for every positive polynomial p there is an integer $n_p > 0$ such that $|\epsilon(k)| < 1/p(k)$ for every

$k \geq n_p$. For computation in a specific algebra, we naturally require that there is an efficient algorithm that for every n-ary operation $f^{\mathbf{A}}$ and any n elements of $\mathbf{A}$ outputs the application of $f^{\mathbf{A}}$ on these elements.

An input for each party P_i is of the form $(P_i, P_j, s, \text{role})$, where P_j is the identity of another party, s is a unique *session identifier* and role is either an *initiator* or a *responder*. Two sessions are *matching* if their inputs for P_i and P_j are $(P_i, P_j, s, \text{initiator})$ and $(P_j, P_i, s, \text{responder})$, respectively. After activation, P_i and P_j exchange messages and generate locally an output that contains the names of the principals of the session, the session identifier and a computed session key. When such an output is generated, the session is *completed* for that principal. A session can also *expire*, which means that the corresponding session key and state information is erased from the memory of the principal.

An *adversary* is modeled as a probabilistic polynomial time algorithm that has full control on the communication channel. An adversary can also control scheduling of the protocol events that include initiation and delivery of messages. Furthermore, the adversary is given access to secret information using the following special queries:

- *Party corruption*: The adversary learns all information in the memory of a principal.
- *Session key query*: The adversary learns the session key of a session.
- *Session state reveal*: The adversary learns the internal state of an incomplete session.

This model is called the *unauthenticated links model* (UM). We also define a restricted model called the *authenticated links model* (AM) in a similar way but assume that the adversary cannot modify the messages that have been generated by the principals. That is, in the AM we assume that all communication is completely authenticated.

At any point, an adversary $\mathcal{U}$ can run a *test session query* for a session that is completed, unexpired and has not been revealed. When such a query is invoked, we choose $b \in \{0, 1\}$ uniformly at random. If $b = 0$ we give the session key to $\mathcal{U}$. If $b = 1$ we give $\mathcal{U}$ a randomly chosen key from the distribution of session keys. The adversary now has to distinguish the real session key from a random one.

Definition 1 (Canetti-Krawczyk security)**.** *A key agreement protocol is secure if for every adversary $\mathcal{U}$ in the UM (similarly for AM)*

1. *whenever a matching session is completed for two uncorrupted principals, they both output the same key,*
2. *the probability that $\mathcal{U}$ correctly guesses b is at most $1/2 + \epsilon(k)$, where ϵ is a negligible function and k is the security parameter.*

Any protocol that is secure in the AM can be converted into a secure protocol in the UM using algorithms called *authenticators*. Such algorithms can be constructed based on different cryptographic mechanisms such as digital signatures or MACs. For details, see [4] and [7, 8].

3 Homomorphism based key agreement scheme

In the following, we describe a two-party key agreement scheme that is based on the hardness of computing the image of an element under an unknown homomorphism from an algebra $\mathbf{A}$ to an algebra $\mathbf{B}$ of the same type. The goal of the scheme is to establish a common element of $\mathbf{B}$ in a way that is infeasible for an eavesdropper to deduce. It is not guaranteed, however, that some information could not be gained about the element in a particular algebraic platform. The situation is analogous to the Diffie-Hellman protocol. For instance, there are many groups for which the computational Diffie-Hellman problem is generally considered infeasible, but there is an efficient algorithm for the decision version [5]. For clarity, we shall not in this section consider full indistinguishability of the session key. We call the construction "a scheme" in order to differentiate from a complete protocol. We also do not restrict ourselves to any particular model of computation. Instead, we only assume that certain problems are infeasible in the chosen model.

We start by defining two problems. Let $\mathbf{A} = (X_{\mathbf{A}}, F_{\mathbf{A}})$ and $\mathbf{B} = (X_{\mathbf{B}}, F_{\mathbf{B}})$ be (possibly non-associative) algebras of the same type.

Definition 2 (Factorization problem (FP))**.** *Let n be a non-negative integer and let*

$$A = \{a_1, a_2, \ldots, a_n\} \subset X_{\mathbf{A}}.$$

Given an element y in the subalgebra of $\mathbf{A}$ generated by A, find a representation of y using the generators A and the operations in $F_{\mathbf{A}}$.

Definition 3 (Computational homomorphic image problem (CHIP))**.** *Given a set of pairs of elements from $X_{\mathbf{A}} \times X_{\mathbf{B}}$,*

$$(a_1, \varphi(a_1)), (a_2, \varphi(a_2)), \ldots, (a_n, \varphi(a_n)),$$

where φ is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$ and an element x of the subalgebra of $\mathbf{A}$ generated by $A = \{a_1, a_2, \ldots, a_n\}$, compute $\varphi(x)$.

The corresponding assumptions are that these problems are infeasible in the chosen computational model. Let $\mathbf{A}_A$ denote the subalgebra of $\mathbf{A}$ generated by A. It is easy to see that these two problems are connected.

Proposition 1. *Let $A = \{a_1, a_2, \ldots, a_n\}$. If the FP is feasible on $\mathbf{A}_A$, then an instance*

$$(a_1, \varphi(a_1)), (a_2, \varphi(a_2)), \ldots, (a_n, \varphi(a_n)), x$$

of the CHIP can be feasibly solved for any homomorphism $\varphi : \mathbf{A} \to \mathbf{B}$ and any element $x \in \mathbf{A}_A$.

Proof. Since the FP is feasible on $\mathbf{A}_A$, it is feasible to find a representation of x using the generators A and the operations $F_{\mathbf{A}}$. By exchanging each occurrence of a_i in the representation of x by $\varphi(a_i)$ and each occurrence of an operation from $F_{\mathbf{A}}$ by the corresponding operation from $F_{\mathbf{B}}$, we have an expression of

$\varphi(x)$ by the homomorphism property of φ. The value of $\varphi(x)$ can be computed by evaluating the expression. $\square$

For a key agreement scheme, we require a set of homomorphisms such that if a particular homomorphism φ is known, then $\varphi(x)$ can be efficiently computed for any $x \in \mathbf{A}$ even without a factorization of x. In the following, we call such a set of homomorphism *efficiently computable.*

Scheme 1. *Let the participants be Alice and Bob. Let* $\mathbf{A} = (X_{\mathbf{A}}, F_{\mathbf{A}})$ *and* $\mathbf{B} = (X_{\mathbf{B}}, F_{\mathbf{B}})$ *be public algebras and let* K *be a set of efficiently computable homomorphisms from* $\mathbf{A}$ *to* $\mathbf{B}$*.*

Step 1: *Alice randomly samples a finite subset of distinct elements*

$$A = \{a_1, a_2, \ldots, a_n\} \subseteq X_{\mathbf{A}}.$$

and a random private homomorphism $\alpha \in K$*. She computes*

$$\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$$

and transmits the pairs

$$(a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n))$$

to Bob.

Step 2: *Bob randomly applies the operations of* $F_{\mathbf{B}}$ *on the elements*

$$\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$$

to obtain $\alpha(b) \in X_{\mathbf{B}}$*. Bob also applies the corresponding sequence of operations in* $F_{\mathbf{A}}$ *on the corresponding* $a_1, a_2, \ldots, a_n$ *to obtain* $b \in X_{\mathbf{A}}$*, which he transmits to Alice.*

Step 3: *Alice computes* $\alpha(b)$*, which is the common secret element.*

Proposition 2. *If an eavesdropper Eve is able to feasibly compute the common secret element, then she is able to feasibly solve the CHIP on* $\mathbf{A}_A$*.*

Proof. By observing the exchanged messages during the execution of the scheme, Eve knows

$$(a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)) \text{ and } b,$$

which is an instance of the CHIP on the subalgebra generated by A. If Eve is able to feasibly compute $\alpha(b)$, then she is able to solve this particular instance. Since α and b are randomly chosen, she is able to feasibly solve an arbitrary instance of the CHIP on $\mathbf{A}_A$. $\square$

If the CHIP is infeasible, then it is infeasible for an eavesdropper to fully recover the common element.

We can show that the Diffie-Hellman key agreement scheme is a special case of Scheme 1. Presenting the scheme in this form allows us to eliminate many of the algebraic properties needed by the original Diffie-Hellman scheme. For example, the underlying structure does not need to be a group and derivation of the common element does not require the commutativity of the exponentiation operation. Nevertheless, the original Diffie-Hellman scheme takes the following form. Let $\mathbf{A} = \mathbf{B} = G$, a cyclic group and let $K = \mathrm{Aut}\,(G)$.

Step 1: Alice generates a random $A = \{g\}$, where g is a generator of G, and a random private automorphism $\alpha : x \mapsto x^a$, where $a \in \{1, 2, \ldots, |G|\}$. Using the only generator g, Alice computes $\alpha(g) = g^a$ and transmits the pair (g, g^a) to Bob.

Step 2: Bob randomly applies the only binary operation of G on g^a to obtain a secret element

$$g^{ab} = \underbrace{g^a g^a \cdots g^a}_{b \text{ times}}.$$

Bob also applies the corresponding sequence of operations on g to obtain an element

$$g^b = \underbrace{gg \cdots g}_{b \text{ times}},$$

which he transmits to Alice.

Step 3: Alice computes $\alpha(g^b) = g^{ab}$, which is the common element.

In this case, the FP asks for a factorization of g^b using g, which is equivalent to the discrete logarithm problem. The CHIP asks to compute $\alpha(g^b) = g^{ab}$ given g^b and (g, g^a), which is equivalent to the computational Diffie-Hellman problem. Of course, whether any information can be deduced about g^{ab} depends on the decision Diffie-Hellman assumption on G. A similar assumption is clearly needed for $\mathbf{A}, \mathbf{B}$ and K that are used for Scheme 1.

4 Provably secure key agreement protocol

To be able to securely agree on a key, Alice and Bob need to be able to authenticate each other. We shall not consider authentication schemes in this paper. Instead, we first devise a protocol that is secure in the AM and then apply an authenticator. This approach allows us to devise a secure protocol in a clear manner.

4.1 Decision homomorphic image assumption

We establish the security of our protocol under a *decision homomorphic image* (DHI) assumption, which is analogous to the decision Diffie-Hellman assumption. We formulate the DHI assumption for algebras by following [5].

Definition 4. *A* family of algebras and homomorphisms

$$\mathbb{A}_n \to \mathbb{B} = \{(\mathbf{A}_i, \mathbf{B}_i, K_i)\}$$

is a set consisting of ordered 3-tuples $(\mathbf{A}_i, \mathbf{B}_i, K_i)$*, where* $\mathbf{A}_i$ *and* $\mathbf{B}_i$ *are finitely generated algebras of the same type,* $\mathbf{A}_i$ *is generated by* n *elements for every* i*,* K_i *is a set of homomorphisms from* $\mathbf{A}_i$ *to* $\mathbf{B}_i$ *such that* $\alpha(\mathbf{A}_i) = \mathbf{B}_i$ *for every* $\alpha \in K_i$ *and* i *ranges over an infinite index set.*

Remark 1. *The requirements on this definition could be relaxed by requiring that* $A = \{a_1, a_2, \dots, a_n\} \subseteq \mathbf{A}_i$ *only generates a subalgebra* $(\mathbf{A}_i)_A$ *of* $\mathbf{A}_i$ *and the image of* $(\mathbf{A}_i)_A$ *under* α *may be different for each* $\alpha \in K_i$*. However, in such a case the cardinality of the algebra generated by* $\alpha(A)$ *might be a priori unknown.*

An *instance generator* $IG_{\mathbb{A}_n \to \mathbb{B}}$ for the family $\mathbb{A}_n \to \mathbb{B}$ is a randomized algorithm that given a security parameter k runs in polynomial time and outputs an index i and a random set of distinct elements $A = \{a_1, a_2, \dots, a_n\} \subseteq \mathbf{A}_i$ such that $\mathbf{A}_i$ is generated by A.

Using this notation, the decision Diffie-Hellman assumption can be formulated in the following way. Let $\mathbb{G}_1 \to \mathbb{G} = \{(G_p, G_p, \mathrm{Aut}\,(G_p))\}$, where each G_p is a cyclic group and p ranges over prime numbers. An instance generator for this family outputs an index p and a random generator $g \in G_p$.

Definition 5. *A probabilistic polynomial time algorithm* $\mathcal{D}$ *is a decision Diffie-Hellman (DDH) distinguisher for* $\mathbb{G}_1 \to \mathbb{G}$*, if for a fixed* $a > 0$ *and a sufficiently large* k*,*

$$|\Pr[\mathcal{D}(p, (g, \alpha(g)), (g^y, \alpha(g^y)) = 1] - \Pr[\mathcal{D}(p, (g, \alpha(g)), (g^y, g^z) = 1]\,| > \frac{1}{k^a},$$

where g *is a generator of* G_p *and* $\alpha \in \mathrm{Aut}\,(G)$*. The probability is taken over*

- *the random choice of* p, g *according to the distribution induced by the instance generator* $IG_{\mathbb{G}_1 \to \mathbb{G}}(k)$*,*
- *the random choice of* y, z *in* $\{1, 2, \dots, |G_p|\}$*,*
- *the random choice of* $\alpha \in \mathrm{Aut}\,(G)$*, that is, the random choice of* $x \in \{1, 2, \dots, |G_p|\}$ *since* $\alpha : g \mapsto g^x$ *for some* x*,*
- *the random bits of* $\mathcal{D}$*.*

The family $\mathbb{G}_1 \to \mathbb{G}$ *satisfies the DDH assumption if there is no DDH distinguisher for* $\mathbb{G}_1 \to \mathbb{G}$*.*

We assume that a distinguisher $\mathcal{D}$ outputs 1 if it thinks that it was given $\alpha(g^y) = g^{xy}$ instead of a random element g^z. The DHI assumption can be formulated in an analogous way.

Definition 6. *A probabilistic polynomial time algorithm $\mathcal{D}$ is an n-DHI distinguisher for $\mathbb{A}_n \to \mathbb{B}$ if, for a fixed $a > 0$ and a sufficiently large k, it satisfies*

$$\begin{aligned} &|\Pr[\mathcal{D}(i,(a_1,\alpha(a_1)),(a_2,\alpha(a_2)),\ldots,(a_n,\alpha(a_n)),(b,\alpha(b)))=1] \\ &-\Pr[\mathcal{D}(i,(a_1,\alpha(a_1)),(a_2,\alpha(a_2)),\ldots,(a_n,\alpha(a_n)),(b,c))=1]\,| > \tfrac{1}{k^a}, \end{aligned}$$

where $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbf{A}_i$ is a set of generators of $\mathbf{A}_i$. The probability is taken over

- *the random choice of $i, a_1, a_2, \ldots, a_n$ according to the probability distribution induced by the instance generator $IG_{\mathbb{A}_n \to \mathbb{B}}(k)$,*
- *the random choice of $\alpha \in K_i$,*
- *the random choice of b from $\mathbf{A}_i$,*
- *the random choice of c from $\mathbf{B}_i$,*
- *the random bits of $\mathcal{D}$.*

The family $\mathbb{A}_n \to \mathbb{B}$ satisfies the n-DHI assumption if there is no n-DHI distinguisher for $\mathbb{A}_n \to \mathbb{B}$.

Remark 2. *For $\mathbb{G}_1 \to \mathbb{G}$, the* 1*-DHI assumption is equivalent to the DDH assumption.*

4.2 The protocol

Based on Scheme 1 and the n-DHI assumption, we formulate a two-party key agreement protocol that is secure in the AM. The proof of the security is in fact very similar to the proof of Theorem 8 in [8].

The established session key is often required as a binary string. Since we have no requirements for the representations of the algebras, we assume that for any particular representation of an algebra $\mathbf{A}$, there is an injective public function B that maps elements of the algebra to binary strings. This function can be used to derive a valid session key.

Protocol 1. *Common information: a member $(\mathbf{A}_k, \mathbf{B}_k, K_k)$ from a family of algebras and homomorphisms $\mathbb{A}_n \to \mathbb{B}$ and an injective function $B : \mathbf{B}_k \to \{0,1\}^*$.*

Step 1: *The principal P_i on input $(P_i, P_j, s, initiator)$*

- *randomly samples a sequence $(a_1, a_2, \ldots, a_n)$ of distinct elements of $\mathbf{A}_k$ such that $\mathbf{A}_k$ is generated by these elements,*
- *randomly samples a homomorphism $\alpha \in K_k$,*
- *computes $\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$,*
- *transmits $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$ to P_j.*

Step 2: *After receiving* $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$*, the responder* P_j

- *randomly applies the finitary operations of* $\mathbf{B}_k$ *on* $\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$ *to obtain an element* $\alpha(b)$*,*
- *applies the corresponding sequence of operations of* $\mathbf{A}_k$ *on* $a_1, a_2, \ldots, a_n$ *to obtain* b*,*
- *transmits* (P_j, P_i, s, b) *to* P_i*,*
- *computes* $B(\alpha(b))$*,*
- *erases* $\alpha(b)$*,*
- *erases the sequence of operations,*
- *outputs the session key* $B(\alpha(b))$ *under the session identifier* s*.*

Step 3: *After receiving* (P_j, P_i, s, b)*, the principal* P_i

- *computes* $B(\alpha(b))$*,*
- *erases* α*,*
- *outputs the session key* $B(\alpha(b))$ *under the session identifier* s*.*

Proposition 3. *Protocol 1 is secure in the AM under the n-DHI assumption for* $\mathbb{A}_n \to \mathbb{B}$*.*

Proof. If P_i and P_j complete the protocol uncorrupted they both establish the same key for the session identified by s. Therefore, the first requirement of Definition 1 is satisfied.

Suppose that there is an adversary $\mathcal{A}$ in the AM against Protocol 1 such that $\mathcal{A}$ distinguishes with probability $1/2 + \epsilon(t)$, where $\epsilon(t)$ is a non-negligible function on the security parameter t, whether the response to a test query is real or randomly chosen. We construct an n-DHI distinguisher $\mathcal{D}$ for $\mathbb{A}_n \to \mathbb{B}$ using $\mathcal{A}$. Let the input to $\mathcal{D}$ be one of the following, each with probability 1/2:

$$(i, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)), (b, \alpha(b))),$$

or

$$(i, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)), (b, c)),$$

where c is chosen uniformly at random from $\mathbf{B}_k$.

Let us denote by l an upper bound for the number of sessions that $\mathcal{A}$ possibly invokes in any interaction and let us consider the following description for the distinguisher $\mathcal{D}$ on input $(i, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)), (b, x))$:

1. Choose r uniformly at random from $\{1, 2, \ldots, l\}$.
2. Run $\mathcal{A}$ and simulate an interaction of principals $P_1, P_2, \ldots, P_m$ using Protocol 1.

3. Whenever $\mathcal{A}$ activates a principal for a session that is not the r-th one, follow the protocol instructions on behalf of that principal. Whenever a session expires at a principal, erase the corresponding session key from its memory. If a principal is corrupted, give all information stored in its memory to $\mathcal{A}$. If a session that is not the r-th one is exposed, give all information corresponding to that session to $\mathcal{A}$.

4. When $\mathcal{A}$ invokes the r-th session, with session number s, to establish a key between P_i and P_j with P_i as the initiator, let P_i send the message $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$ to P_j.

5. When $\mathcal{A}$ invokes P_j to receive $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$, let P_j respond with (P_j, P_i, s, b).

6. If the session with session number s is chosen as the test session by $\mathcal{A}$, provide $B(x)$ to $\mathcal{A}$ as an answer to the query.

7. If the r-th session is not chosen as the test session, or if $\mathcal{A}$ halts without choosing a test session, choose a bit b' uniformly at random, output b' and halt.

8. If $\mathcal{A}$ halts with an output bit b', output b' and halt.

Clearly the run of $\mathcal{A}$ by $\mathcal{D}$ is identical to a normal run of $\mathcal{A}$ against Protocol 1. Let us consider the two cases depending on whether the test-session chosen by $\mathcal{A}$ is the r-th one.

1. Let the test-session be the r-th session. In this case, $\mathcal{A}$ is given

$$(a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)), b \quad \text{and} \quad B(x).$$

   If the input x was $\alpha(b)$, then $\mathcal{A}$ was given the actual key agreed between P_i and P_j for session s. If x was a random element c, then $\mathcal{A}$ was provided with a binary string corresponding to a random element from $\mathbf{B}_k$. That is, a random value was given to $\mathcal{A}$ from the distribution of session keys. Each of these cases happens with probability $1/2$, which satisfies the requirement for Definition 1. Since $\mathcal{A}$ distinguishes these cases with probability $1/2 + \epsilon(t)$ and $\mathcal{D}$ outputs the same bit, $\mathcal{D}$ distinguishes the inputs with probability $1/2 + \epsilon(t)$.

2. Suppose that the test session is not the r-th session. Then $\mathcal{D}$ outputs a random bit and its probability to guess correctly is $1/2$.

The probability that case 1 happens is $1/l$, while the probability that case 2 happens is $1 - 1/l$. Therefore, the probability of $\mathcal{D}$ to succeed to distinguish the input distributions is $1/2 + \epsilon(t)/l$, which is non-negligible. $\square$

Protocol 1 can be transformed into a protocol that is secure in the UM using any authenticator. In the following, we apply a signature based authenticator described in [4].

Protocol 2. *Common information: an instance* $(\mathbf{A}_k, \mathbf{B}_k, K_k)$ *from a family of algebras and homomorphisms* $\mathbb{A}_n \to \mathbb{B}$ *and a function* $B : \mathbf{B}_k \to \{0,1\}^*$. *Each participant* P_i *also has a private key for a signature algorithm* Sign *and the public verification keys of the other participants.*

Step 1: *The initiator* P_i *on input* $(P_i, P_j, s, initiator)$

- *randomly samples a sequence* $(a_1, a_2, \ldots, a_n)$ *of distinct elements from* $\mathbf{A}_k$ *such that* $\mathbf{A}_k$ *is generated by these elements,*
- *randomly samples a homomorphism* $\alpha \in K_k$,
- *computes* $\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$,
- *computes a signature*

$$s_1 = \mathsf{Sign}(i, (P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))),$$

- *transmits* $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$ *and* s_1 *to* P_j.

Step 2: *After receiving* $(P_i, P_j, s, (a_1, \alpha(a_1)), (a_2, \alpha(a_2)), \ldots, (a_n, \alpha(a_n)))$ *and* s_1*, the responder* P_j

- *verifies the signature* s_1 *and all of the received values. If the verification fails,* P_j *aborts. If it succeeds,* P_j *continues and*
- *randomly applies the binary operations of* $\mathbf{B}_k$ *on* $\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n)$ *to obtain an element* $\alpha(b)$,
- *in parallel applies the corresponding sequence of operations of* $\mathbf{A}_k$ *on* $a_1, a_2, \ldots, a_n$ *to obtain* b,
- *computes a signature* $s_2 = \mathsf{Sign}(j, (P_j, P_i, s, b))$,
- *transmits* (P_j, P_i, s, b) *and* s_2 *to* P_i,
- *computes* $B(\alpha(b))$,
- *erases* $\alpha(b)$,
- *erases the sequence of operations,*
- *outputs the session key* $B(\alpha(b))$ *under the session identifier* s.

Step 3: *After receiving* (P_j, P_i, s, b) *and* s_2*, the principal* P_i

- *verifies the signature* s_2 *and all of the received values. If the verification fails,* P_i *aborts. If it succeeds,* P_i *continues and*
- *computes* $B(\alpha(b))$,
- *erases* α,
- *outputs the session key* $B(\alpha(b))$ *under the session identifier* s.

5 Protocol based on an additively homomorphic symmetric encryption scheme

In this Section, we give an instantiation of the devised protocol based on a symmetric encryption scheme of Armknecht and Sadeghi [3] (hereby abbreviated as the AS scheme) that is homomorphic (in a restricted sense) over a vector space. We start by briefly describing the AS scheme and considering a suitable choice of parameters. Then, we construct a family of algebras and homomorphisms and prove that the r-DHI assumption holds for the family provided that the AS scheme is IND-CPA secure for $r+1$ encryptions. Finally, we give a description of the complete protocol.

5.1 The Armknecht-Sadeghi scheme

Armknecht and Sadeghi suggest in [3] a homomorphic symmetric encryption scheme that is a modification of a non-homomorphic encryption scheme suggested by Kiayias and Yung in [15]. The AS scheme supports an unlimited number of additions but a very limited number of multiplications. Furthermore, the ciphertext size grows exponentially with the number of encryptions. Ciphertext indistinguishability under chosen plaintext attack (IND-CPA or semantic security [13]) of the AS scheme is based on the problem of decoding interleaved Reed-Solomon codes. The problem is also called the synchronized polynomial reconstruction problem (SPRP) and it is defined in the following way in [10]. Let $[m] = \{1, 2, \ldots, m\}$ and let $\mathbb{F}$ be a field.

Definition 7 (SPRP)**.** *Given $k, t, r \in \mathbb{N}$, a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$ with $x_i \neq x_j$ for $i \neq j$ and r vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r$ such that $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,n}) \in \mathbb{F}^n$ for every $i \in [r]$, output a sequence of polynomials $(p_{\mathbf{y}_1}, p_{\mathbf{y}_2}, \ldots, p_{\mathbf{y}_r})$ and a set of indices I (error-free entries) that satisfy*

- *$p_{\mathbf{y}_i} \in \mathbb{F}[x]$ and $\deg(p_{\mathbf{y}_i}) < k$ for every $i \in [r]$,*
- *$I \subseteq [n]$ and $|I| = t$,*
- *$p_{\mathbf{y}_i}(x_j) = y_{i,j}$ for all $i \in [r], j \in I$.*

Instances of the SPRP can be efficiently sampled by an instance generator IG_{SPR}. On input $(\mathbf{x}, k, t, r)$ choose uniformly at random a subset $I \subset [n]$ of size t and r polynomials $p_{\mathbf{y}_i} \in \mathbb{F}[x]$ with $\deg(p_{\mathbf{y}_i}) < k$ for every $i \in [r]$. Set $y_{i,j} = p_{\mathbf{y}_i}(x_j)$ for every $i \in [r], j \in I$ and choose $y_{i,j}$ uniformly at random from $\mathbb{F} \setminus \{p_{\mathbf{y}_i}(x_j)\}$ for $i \in [r], j \notin I$. The generated instance is $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r)$.

A decision problem can be formulated based on the SPRP by considering the indices I. Informally, the goal is to determine with non-negligible probability whether a given index $i \in I$. Clearly, a solution to the SPRP yields a solution to this decision problem.

Definition 8 (Decision SPRP)**.** *A probabilistic polynomial time algorithm $\mathcal{D}$ is an $(\mathbf{x}, k, t, r)$-SPR distinguisher if for a fixed a and a suitably large k, it satisfies*

$$|\Pr[\mathcal{D}(i, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r) = 1] - \Pr[\mathcal{D}(j, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r) = 1]| > \frac{1}{k^a},$$

where $i \in I, j \in [n] \setminus I$ and $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r)$ is an instance of SPRP. The probability is taken over

- *the random choice of $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_r)$ according to the instance generator IG_{SPR},*
- *the random choice of $i \in I$,*
- *the random choice of $j \in [n] \setminus I$,*
- *the random bits of $\mathcal{D}$.*

The $(\mathbf{x}, k, t, r)$-SPR assumption holds if there is no $(\mathbf{x}, k, t, r)$-SPR distinguisher.

If p is a polynomial and $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$, let us denote $p(\mathbf{x}) = \mathbf{y}$ if and only if $p(x_i) = y_i$ for every $i \in [n]$. The AS scheme encrypts vectors over $\mathbb{F}$ to instances of the SPRP using the index set I as a key. The scheme consists of the following five operations.

Setup: The input consists of the security parameter and two positive integers r, an upper bound on the number of encryptions, and μ, the number of supported multiplications. The setup algorithm chooses integers n, k, t such that $\mu{\cdot}k < t < n$ and the appropriate security conditions are met. See Section 5.2 for details. It also selects an index set $I \subset [n]$ such that $|I| = t$ and two vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$ and $\mathbf{z} = (z_1, z_2, \ldots, z_{\lfloor k/2 \rfloor}) \in \mathbb{F}^{\lfloor k/2 \rfloor}$ with all entries pairwise distinct.

Encrypt: The input consists of a plaintext $\mathbf{m} \in \mathbb{F}^{\lfloor k/2 \rfloor}$ and a key I. The encryption algorithm chooses a random polynomial $p \in \mathbb{F}[x]$ of degree $\leq k$ such that $p(\mathbf{z}) = \mathbf{m}$. The ciphertext $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{F}^n$ is constructed by setting $c_i = p(x_i)$ for $i \in I$ and choosing c_i uniformly at random from $\mathbb{F} \setminus \{p(x_i)\}$ for $i \notin I$. The output is $(\mathbf{c}, 1)$, where the second entry is a counter to keep track of multiplications.

Decrypt: The input consists of a ciphertext $(\mathbf{c}, ctr)$ and a key I. The decryption algorithm interpolates a polynomial $p_{\mathbf{c}}$ of degree $\leq k$ that satisfies $c_i = p_{\mathbf{c}}(x_i)$ for $i \in I$ and outputs $p_{\mathbf{c}}(\mathbf{z})$.

Add: Compute the sum of two ciphertexts $(\mathbf{c}_1, ctr_1)$ and $(\mathbf{c}_2, ctr_2)$ by outputting $(\mathbf{c}_1 + \mathbf{c}_2, \max(ctr_1, ctr_2))$.

Multiply: Compute the product of two ciphertexts $(\mathbf{c}_1, ctr_1)$ and $(\mathbf{c}_2, ctr_2)$ by outputting $(\mathbf{c}_1 \bullet \mathbf{c}_2, ctr_1 + ctr_2)$, where $\bullet$ denotes componentwise product.

5.2 Choice of parameters

The parameters $t = |I|, n = |\mathbf{x}|$ and k need to be chosen correctly to ensure the security of the AS scheme. The scheme is proved IND-CPA secure in [3, Theorem 3] for $t = \mu \cdot k$ whenever the $(\mathbf{x}', \lfloor k/2 \rfloor, t, r)$-SPR assumption holds. The vector $\mathbf{x}' \in \mathbb{F}^{n-1}$ is derived from $\mathbf{x}$ by removing a coordinate. Two lower bounds for $(n-1)/k$ are stated in [3]:

$$\frac{n-1}{k} \geq \frac{(2\mu-1)^{r+1}}{2} \quad \text{and} \quad \frac{n-1}{k} \geq \frac{(r+1)\mu - r}{2}.$$

The first one is derived from a bound in [10] that describes the current state-of-the-art probabilistic polynomial time algorithm to solve the SPRP. The algorithm succeeds with probability $1 - O(n^{O(r)}/|\mathbb{F}|)$ whenever

$$t > k(\frac{n}{k})^{1/(r+1)} + k + 1. \tag{1}$$

This entails an exponential growth on the ciphertext size in terms of the number of encryptions if multiplication needs to be supported. Since we only need to support addition, we set $\mu = 0$ and omit *ctr* from the ciphertexts. We also want to choose the smallest t possible due to (1). To allow unique decryption it is necessary that $t > k$ and we set $t = k + 1$. This means that (1) is not satisfied for any choice of n and k.

Remark 3. *Since the AS scheme is based on the Reed-Solomon code, it has inherent error correction properties. Considering the t error free entries, with our choice of parameters we in fact have a linear code of length t and dimension $k = t - 1$, which means that the error correcting property is lost.*

5.3 A family of algebras and homomorphisms

Let $\mathbb{F}$ be a field with

$$\log(|\mathbb{F}| - 1) \geq \frac{\log(\binom{n}{t}) + s}{t - k}.$$

This bound guarantees that the decryption function of the AS scheme obtains a unique interpolation polynomial with probability at least $1 - 2^{-s}$ [15]. For the rest of the paper, let us assume that a unique solution is obtained for every decryption. Let

$$\mathbb{D}_t = \{D_I : I \subset [n], |I| = t\},$$

be the set of functions $\mathbb{F}^n \to \mathbb{F}^{\lfloor k/2 \rfloor}$ arising from the decryption algorithm of the AS scheme over $\mathbb{F}$. Let also

$$\begin{aligned} \operatorname{Ker}\left(D_{[n]}\right) &= \{\mathbf{w} \in \mathbb{F}^n : D_{[n]}(\mathbf{w}) = \mathbf{0}\} \\ &= \{\mathbf{w} \in \mathbb{F}^n : \exists\, p \in \mathbb{F}[x], \mathbf{w} = p(\mathbf{x}), p(\mathbf{z}) = \mathbf{0}, \deg p \leq k\}, \end{aligned}$$

where $\mathbf{x}$ and $\mathbf{z}$ are the vectors agreed in the setup phase of the AS scheme. Note that whenever $\mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$, then $\mathbf{w} \in \operatorname{Ker}\left(D_I\right)$ for any $I \subseteq [n], t \leq |I| \leq n$, since the erroneous entries are discarded by D_I.

Let $\mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$ and let us define the following binary operation on $\mathbb{F}^n$:

$$\mathbf{u} *_{\mathbf{w}} \mathbf{v} = \mathbf{u} + \mathbf{v} + \mathbf{w} \tag{2}$$

for every $\mathbf{u}, \mathbf{v} \in \mathbb{F}^n$.

Proposition 4. *D_I is a homomorphism $(\mathbb{F}^n, *_{\mathbf{w}}) \to (\mathbb{F}^{\lfloor k/2 \rfloor}, +)$ for every $D_I \in \mathbb{D}_t, \mathbf{w} \in Ker\left(D_{[n]}\right)$.*

Proof. Every $D_I \in \mathbb{D}_t$ is a mapping $\mathbb{F}^n \to \mathbb{F}^{\lfloor k/2 \rfloor}$. Since the AS scheme is additively homomorphic, every $D_I \in \mathbb{D}_t$ satisfies $D_I(\mathbf{c}_1 + \mathbf{c}_2) = D_I(\mathbf{c}_1) + D_I(\mathbf{c}_2)$ for every $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{F}^n$. Furthermore, $D_I(\mathbf{w}) = \mathbf{0}$ for every $\mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$. This means that

$$D_I(\mathbf{u} *_{\mathbf{w}} \mathbf{v}) = D_I(\mathbf{u} + \mathbf{v} + \mathbf{w}) = D_I(\mathbf{u}) + D_I(\mathbf{v}) + D_I(\mathbf{w}) = D_I(\mathbf{u}) + D_I(\mathbf{v})$$

for every $\mathbf{u}, \mathbf{v} \in \mathbb{F}^n, \mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$. ☐

Let $r \leq \lfloor k/2 \rfloor$. Let U be an r-dimensional subspace of $\mathbb{F}^n$ such that there is a basis $B_U = \{\mathbf{b}_1, \ldots, \mathbf{b}_r\}$ of U such that $D_{[n]}(B_U) = \{D_{[n]}(\mathbf{b}_1), \ldots, D_{[n]}(\mathbf{b}_r)\}$ forms a basis B_V of an r-dimensional subspace V of $\mathbb{F}^{\lfloor k/2 \rfloor}$. Let

$$Q = \left(U \setminus \operatorname{Ker}\left(D_{[n]}\right)\right) \oplus \operatorname{Ker}\left(D_{[n]}\right),$$

where $\oplus$ denotes internal direct sum. Consider an algebra over Q with a set of finitary operations consisting of $*_{\mathbf{w}}$ for every $\mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$. Clearly, a set $W = \{\mathbf{b}_1 + \mathbf{w}_1, \ldots, \mathbf{b}_r + \mathbf{w}_r\}$ is a set of generators of such an algebra for any $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r \in \operatorname{Ker}\left(D_{[n]}\right))$. Let

$$\mathbf{Q}_{n,k,W} = (Q, \{*_{\mathbf{w}} : \mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)\}).$$

Let $\mathbf{A}_{\lfloor k/2 \rfloor, B_V} = \mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}$ denote an algebra over V of the same type as $\mathbf{Q}_{n,k,W}$ such that each of the $|\operatorname{Ker}\left(D_{[n]}\right)|$ operations is (the normal) vector sum on V. That is, $\mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}$ is essentially the additive group $(V, +)$ of the r-dimensional subspace V of $\mathbb{F}^{\lfloor k/2 \rfloor}$. By Proposition 4, $\mathbb{D}_t$ is a set of homomorphisms $\mathbf{Q}_{n,k,W} \to \mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}$. Finally, let

$$\mathbb{Q}_r \to \mathbb{A} = \{(\mathbf{Q}_{n,k,W}, \mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}, \mathbb{D}_t)\}$$

be a family of algebras and homomorphisms parametrized by n, k, W and t.

We have explicitly augmented U with the binary operations of the type (2) for the following reason. Considering the AS scheme, adding a random element from $\operatorname{Ker}\left(D_{[n]}\right)$ corresponds to the randomization of the ciphertexts required for the IND-CPA security of an encryption scheme. To see this, consider the encryption algorithm of the scheme. Instead of choosing a random polynomial p, it is possible to choose p in a completely deterministic way, proceed as normal and finally add a random element from $\operatorname{Ker}\left(D_{[n]}\right)$ as the last step. This property is closely related to random self-reducibility and has been previously noted in [22, 15]. Regarding Protocol 2, it allows Bob to randomize the element b he is about to transmit to Alice in Step 2 without knowing the key I.

Proposition 5. *The r-DHI assumption holds for $\mathbb{Q}_r \to \mathbb{A}$ under the assumption that the AS scheme is IND-CPA secure for parameters $(\mathbf{x}, k, t, r+1)$.*

Proof. The AS scheme is provably IND-CPA secure for $r+1$ encryptions for the parameters $(\mathbf{x}, k, t, r+1)$ under the $(\mathbf{x}', \lfloor k/2 \rfloor, t, r+1)$-SPR assumption, where $\mathbf{x}' \in \mathbb{F}^{n-1}$ is derived from $\mathbf{x}$ by removing a coordinate [3]. Suppose that the r-DHI assumption does not hold for $\mathbb{Q}_r \to \mathbb{A}$. Then there is an r-DHI distinguisher $\mathcal{D}$ for $\mathbb{Q}_r \to \mathbb{A}$ that succeeds with non-negligible probability. We will construct an adversary $\mathcal{A}$ for the AS scheme that uses $\mathcal{D}$ as a subroutine and succeeds to distinguish the challenge ciphertexts with non-negligible probability and thus violates the IND-CPA assumption. Let $\mathcal{A}$ engage in a security game with a challenger and consider the following description of $\mathcal{A}$.

1. Choose a set B_V of r linearly independent plaintext vectors $\mathbf{p}_i \in \mathbb{F}^{\lfloor k/2 \rfloor}$ for $1 \leq i \leq r$ uniformly at random.

2. Query the encryption oracle to obtain a set W of ciphertexts $\mathbf{c}_i$ for every plaintext $\mathbf{p}_i$.

3. Compute two distinct elements $\mathbf{m}_0$ and $\mathbf{m}_1$ by randomly summing $\mathbf{p}_i$ for $1 \leq i \leq r$. That is,

$$\mathbf{m}_0 = \sum_{h \in H_0} \mathbf{p}_h, \quad \mathbf{m}_1 = \sum_{h \in H_1} \mathbf{p}_h,$$

   where H_0 and H_1 are random multisets containing elements from $[r]$.

4. Query the challenger with $\mathbf{m}_0, \mathbf{m}_1$ for a challenge ciphertext $\mathbf{c}$.

5. Run $\mathcal{D}$ on input

$$((n, k, W, t), (\mathbf{c}_1, \mathbf{p}_1), (\mathbf{c}_2, \mathbf{p}_2), \ldots, (\mathbf{c}_r, \mathbf{p}_r), (\mathbf{c}, \mathbf{m}_1))$$

   to obtain a bit b.

6. Output b.

We will first establish that the input to $\mathcal{D}$ at Step 5 is a valid input to an r-DHI distinguisher for $\mathbb{Q}_r \to \mathbb{A}$. By the choice of B_V, the generated subspace V of $\mathbb{F}^{\lfloor k/2 \rfloor}$ is r-dimensional and W is a set of generators of $\mathbf{Q}_{n,k,W}$. This means that we have a valid random member $(\mathbf{Q}_{n,k,W}, \mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}, \mathbb{D}_t)$ from the family $\mathbb{Q}_r \to \mathbb{A}$. The challenge ciphertext $\mathbf{c}$ obtained in Step 4 is one of the valid ciphertexts of either $\mathbf{m}_0$ or $\mathbf{m}_1$. Since D_I is additively homomorphic, we have that $\mathbf{c}$ is one of

$$\mathbf{c}_{\mathbf{m}_0} = \left(\sum_{h \in H_0} \mathbf{c}_h \right) + \mathbf{w}_0, \quad \mathbf{c}_{\mathbf{m}_1} = \left(\sum_{h \in H_1} \mathbf{c}_h \right) + \mathbf{w}_1,$$

where $\mathbf{w}_0, \mathbf{w}_1$ are random elements of Ker $\left(D_{[n]}\right)$. There are factorizations for $\mathbf{c}_{\mathbf{m}_0}$ and $\mathbf{c}_{\mathbf{m}_1}$ in terms of $\mathbf{c}_i$ for $1 \le i \le r$:

$$\mathbf{c}_{\mathbf{m}_0} = \mathbf{c}_{i_1} *_{\mathbf{0}} \mathbf{c}_{i_2} *_{\mathbf{0}} \mathbf{c}_{i_3} *_{\mathbf{0}} \cdots \mathbf{c}_{i_{l-1}} *_{\mathbf{w}_0} \mathbf{c}_{i_l},$$

$$\mathbf{c}_{\mathbf{m}_1} = \mathbf{c}_{j_1} *_{\mathbf{0}} \mathbf{c}_{j_2} *_{\mathbf{0}} \mathbf{c}_{j_3} *_{\mathbf{0}} \cdots \mathbf{c}_{j_{m-1}} *_{\mathbf{w}_1} \mathbf{c}_{j_m},$$

where $\{i_1, i_2, \ldots, i_l\} = H_0$ and $\{j_1, j_2, \ldots, j_m\} = H_1$. This means that $\mathbf{c}$ is a random element of $\mathbf{Q}_{n,k,W}$. Furthermore, $\mathbf{m}_1$ is either the image of $\mathbf{c}$ under D_I or the image of a random element of $\mathbf{A}_{\lfloor k/2 \rfloor, D_{[n]}(W)}$, each with probability 1/2, and

$$((n,k,W,t), (\mathbf{c}_1, \mathbf{p}_1), (\mathbf{c}_2, \mathbf{p}_2), \ldots, (\mathbf{c}_r, \mathbf{p}_r), (\mathbf{c}, \mathbf{m}_1))$$

is a valid input to an r-DHI distinguisher for $\mathbb{Q}_r \to \mathbb{A}$.

Exactly $r+1$ encryption queries were made to the encryption oracle including the challenge ciphertext, which does not violate the parameters chosen for the AS scheme. Let s_W denote the size of the binary representation of W and consider the two cases that can happen.

1. Suppose that $\mathbf{m}_0$ is the image of $\mathbf{c}$ under D_I. Then the input to $\mathcal{D}$ was a random element and $\mathcal{D}$ succeeds with probability 1/2.

2. Suppose that $\mathbf{m}_1$ is the image of $\mathbf{c}$ under D_I. Then $\mathcal{D}$ succeeds with probability $1/2 + \epsilon(n,k,s_W,t)$, where $\epsilon(n,k,s_W,t)$ is a non-negligible function.

Both cases happen with probability 1/2. Since $\mathcal{A}$ outputs the same bit as $\mathcal{D}$, the probability of $\mathcal{A}$ to win the security game is $1/2 \cdot (1/2 + \epsilon(n,k,s_W,t)) + 1/2 \cdot 1/2 = 1/2 + \epsilon(n,k,s_W,t)/2$, which is non-negligible. □

5.4 The protocol

Assuming that the AS scheme is IND-CPA secure for the parameters $(\mathbf{x}, k, t, r+1)$, Protocol 2 is secure in the UM by Proposition 3 over $\mathbb{Q}_r \to \mathbb{A}$. A similar construction is clearly possible for any IND-CPA secure additively homomorphic encryption scheme with the property that ciphertexts can be re-randomized without the key. For the sake of completeness, we present the final UM secure protocol here using the signature based authenticator [4].

Protocol 3. *Common information: A field $\mathbb{F}$, nonnegative integers n, k, r and vectors $\mathbf{x}, \mathbf{z} \in \mathbb{F}^n$ such that $(\mathbf{x}, k, t, r+1)$ are valid and secure parameters for the AS scheme. Participants have also agreed on a function $B : \mathbb{F}^{r\lfloor k/2 \rfloor} \times \mathbb{F}^{\lfloor k/2 \rfloor} \to \{0,1\}^*$ that given a basis of an r-dimensional vector space maps elements of the vector space to binary strings. Each participant P_i also has a private key for a signature algorithm* Sign *and the public verification keys of the other participants.*

Step 1: *The initiator P_i on input $(P_i, P_j, s, initiator)$*

- *randomly samples a sequence $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r)$ of linearly independent vectors from $\mathbb{F}^{\lfloor k/2 \rfloor}$,*

- *randomly samples a key* $I \subset [n]$ *with* $|I| = t$,
- *encrypts* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r$ *using the AS scheme with key* I *to obtain a sequence of vectors* $(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r)$ *from* $\mathbb{F}^n$ *such that* $\mathbf{b}_i = D_I(\mathbf{a}_i)$ *for every* $1 \le i \le r$,
- *computes a signature*

$$s_1 = \mathsf{Sign}(i, (P_i, P_j, s, (\mathbf{a}_1, D_I(\mathbf{a}_1)), (\mathbf{a}_2, D_I(\mathbf{a}_2)), \ldots, (\mathbf{a}_r, D_I(\mathbf{a}_r)))),$$

- *transmits* $(P_i, P_j, s, (\mathbf{a}_1, D_I(\mathbf{a}_1)), (\mathbf{a}_2, D_I(\mathbf{a}_2)), \ldots, (\mathbf{a}_r, D_I(\mathbf{a}_r)))$ *and* s_1 *to* P_j.

Step 2: *After receiving* $(P_i, P_j, s, (\mathbf{a}_1, D_I(\mathbf{a}_1)), (\mathbf{a}_2, D_I(\mathbf{a}_2)), \ldots, (\mathbf{a}_r, D_I(\mathbf{a}_r)))$ *and* s_1, *the responder* P_j

- *verifies the signature* s_1 *and all of the received values. If the verification fails,* P_j *aborts. If it succeeds,* P_j *continues and*
- *generates a random multiset* H *of elements from* $[r]$ *and a random element* $\mathbf{w} \in Ker\left(D_{[n]}\right)$,
- *computes*

$$\mathbf{b} = \sum_{h \in H} \mathbf{a}_h + \mathbf{w},$$

- *computes*

$$D_I(\mathbf{b}) = \sum_{h \in H} D_I(\mathbf{a}_h),$$

- *computes a signature* $s_2 = \mathsf{Sign}(j, (P_j, P_i, s, \mathbf{b}))$,
- *transmits* $(P_j, P_i, s, \mathbf{b})$ *and* s_2 *to* P_i,
- *computes* $B(D_I(\mathbf{b}))$
- *erases* $H, \mathbf{w}$ *and* $D_I(\mathbf{b})$,
- *outputs the session key* $B(D_I(\mathbf{b}))$ *under the session identifier* s.

Step 3: *After receiving* $(P_j, P_i, s, \mathbf{b})$ *and* s_2, *the principal* P_i

- *verifies the signature* s_2 *and all of the received values. If the verification fails,* P_i *aborts. If it succeeds,* P_i *continues and*
- *computes* $B(D_I(\mathbf{b}))$,
- *erases* I,
- *outputs the session key* $B(D_I(\mathbf{b}))$ *under the session identifier* s.

Suppose that Alice and Bob are the initiator and the responder, respectively. Then, for Bob, the most complex parts of the protocol are the computation of the signature (or another authentication procedure), the generation of a polynomial p with $\deg p \le k$ such that $p(\mathbf{z}) = \mathbf{0}$ and the evaluation of the polynomial on $\mathbf{x}$. The rest consists of computing additions over a vector space. Therefore, the protocol is relatively light for Bob compared to Alice. If a list of pseudo-random elements $\mathbf{w} \in \operatorname{Ker}\left(D_{[n]}\right)$ can be precomputed, then Bob only needs to compute a signature and additions over a vector space. Furthermore, Bob only needs to transmit a single n-dimensional vector $\mathbf{b}$ to Alice. As a downside, Alice needs to transmit r vectors that are n-dimensional and r vectors that are $\lfloor k/2 \rfloor$-dimensional to Bob. Nevertheless, such a protocol could be preferable in a setting where computational resources and energy consumption of one of the participants need to be minimized while there are no requirements for the other participant. Such a situation could occur for example in a wireless sensor network. The protocol also has support for parallelization for both of the principals. Alice can compute $D_I(\mathbf{a}_i)$ independently for each i. Bob can compute $\mathbf{b}$ and $D_I(\mathbf{b})$ in parallel. Furthermore, both $\mathbf{b}$ and $D_I(\mathbf{b})$ are computed using vector space addition which is easily parallelized.

6 Conclusion and final remarks

We presented a key agreement scheme that generalizes the Diffie-Hellman scheme from exponentiation in a cyclic group to computing homomorphisms between two algebras. We also showed that the Diffie-Hellman scheme is a special case of our scheme implemented over a cyclic group. We also formulated the computational and the decision homomorphic image problems and provided a key agreement protocol that is secure in the Canetti-Krawczyk model under the decision homomorphic image assumption. We also gave an implementation of our protocol based on an additively homomorphic symmetric encryption scheme. The protocol is secure under the assumption that the encryption scheme is IND-CPA secure.

As is the case with the Diffie-Hellman scheme, our scheme can be also implemented as a non-interactive protocol. Such a version requires that a set of generators $a_1, a_2, \ldots, a_n$ of $\mathbf{A}$ is fixed and public. The public key of Alice consists of

$$\alpha(a_1), \alpha(a_2), \ldots, \alpha(a_n) \quad \text{and} \quad a,$$

where α is a secret homomorphism from $\mathbf{A}$ to $\mathbf{B}$ and a is obtained by randomly applying the operations of $\mathbf{A}$ on the generators $a_1, a_2, \ldots, a_n$. The private key of Alice consists of α together with the sequence used to generate a. Let the public key of Bob be

$$\beta(a_1), \beta(a_2), \ldots, \beta(a_n) \quad \text{and} \quad b.$$

If Alice and Bob have previously shared their public keys, two shared secrets, $\alpha(b)$ and $\beta(a)$, can be derived between Alice and Bob without exchanging messages.

As a final note, we remark that it is possible that non-group based families of algebras and homomorphisms provide computationally more efficient platforms for the protocol. Unfortunately, non-associative algebraic structures and their homomorphisms have not been extensively studied in the cryptographic literature.

7 Acknowledgements

Financial support from Infotech Oulu Graduate School and the following foundations is gratefully acknowledged: Finnish Foundation of Technology Promotion, the Nokia Foundation, Tauno Tönning Foundation, Walter Ahslträm Foundation and The Finnish Foundation for Economic and Technology Sciences – KAUTE.

References

[1] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, New key agreement protocols in braid group cryptography, *Topics in cryptology—CT-RSA 2001*, *Lecture Notes in Computer Science*, vol. 2020, Springer, 2001, pp. 13–27.

[2] I. Anshel, M. Anshel, and D. Goldfeld, An algebraic method for public-key cryptography, *Mathematical Research Letters*, vol. 6, no. 3-4(1999), pp. 287–291.

[3] F. Armknecht and A.-R. Sadeghi, A new approach for algebraically homomorphic encryption, Cryptology ePrint Archive, Report 2008/442, 2008, URL `http://eprint.iacr.org/2008/422`.

[4] M. Bellare, R. Canetti, and H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract), *Proceedings of the thirtieth annual ACM symposium on Theory of computing – STOC '98*, ACM, 1998, pp. 419–428.

[5] D. Boneh, The decision Diffie-Hellman problem, *Algorithmic Number Theory*, *Lecture Notes in Computer Science*, vol. 1423, Springer, 1998, pp. 48–63.

[6] S. Burris and H. P. Sankappanavar, *A course in universal algebra*, *Graduate Texts in Mathematics*, vol. 78, Springer, New York, 1981.

[7] R. Canetti and H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, *Advances in Cryptology – EUROCRYPT 2001*, *Lecture Notes in Computer Science*, vol. 2045, Springer, 2001, pp. 453–474.

[8] R. Canetti and H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels (full version), Cryptology ePrint Archive, Report 2001/040, 2001, `http://eprint.iacr.org/2001/040`.

[9] D. Coppersmith, A. Odlzyko, and R. Schroeppel, Discrete logarithms in GF(p), *Algorithmica*, vol. 1(1986), pp. 1–15.

[10] D. Coppersmith and M. Sudan, Reconstructing curves in three (and higher) dimensional space from noisy data, *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing – STOC 2003*, ACM, 2003, pp. 136–142.

[11] P. Dehornoy, Braid-based cryptography, *Contemporary Mathematics*, vol. 360(2004), pp. 5–33.

[12] W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. IT-22, no. 6(1976), pp. 644–654.

[13] S. Goldwasser and S. Micali, Probabilistic encryption & how to play mental poker keeping secret all partial information, *Proceedings of the fourteenth annual ACM symposium on Theory of computing – STOC '82*, ACM, 1982, pp. 365–377.

[14] A. Joux, A one round protocol for tripartite Diffie-Hellman, *Journal of Cryptology*, vol. 17(2004), pp. 263–276.

[15] A. Kiayias and M. Yung, Cryptographic hardness based on the decoding of reed-solomon codes, *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP 2002, Springer, 2002, pp. 232–243.

[16] 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, *Advances in cryptology—CRYPTO 2000*, *Lecture Notes in Computer Science*, vol. 1880, Springer, 2000, pp. 166–183.

[17] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, vol. 48, no. 177(1987), pp. 203–209.

[18] N. Koblitz, Hyperelliptic cryptosystems, *Journal of Cryptology*, vol. 1(1989), pp. 139–150.

[19] G. Maze, C. Monico, and J. Rosenthal, Public key cryptography based on semigroup actions, *Advances in Mathematics of Communications*, vol. 1(2007), pp. 489–507.

[20] V. S. Miller, Use of elliptic curves in cryptography, *Advances in cryptology—CRYPTO '85*, *Lecture Notes in Computer Science*, vol. 218. Springer, 1986, pp. 417–426.

[21] C. Monico, *Semirings and semigroup actions in public-key cryptography*, Ph.D. thesis, University of Notre Dame, 2002.

[22] M. Naor and B. Pinkas, Oblivious transfer and polynomial evaluation, *Proceedings of the thirty-first annual ACM symposium on Theory of computing – STOC '99*, ACM, 1999, pp. 245–254.

[23] S.-H. Paeng, K.-C. Ha, J. Kim, S. Chee, and C. Park, New public key cryptosystem using finite non abelian groups, *Advances in Cryptology–CRYPTO 2001*, *Lecture Notes in Computer Science*, vol. 2139, Springer, 2001, pp. 470–485.

[24] V. Shpilrain and G. Zapata, Combinatorial group theory and public key cryptography, *Applicable Algebra in Engineering, Communication and Computing*, vol. 17(2006), pp. 291–302.

[25] 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*, vol. 48(1994), pp. 384–386.

[26] A. Stolbunov, Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves, *Advances in Mathematics of Communications*, vol. 4(2010), pp. 215–235.