Commutativity, Associativity, and Public Key Cryptography

Jacques Patarin[1] and Valérie Nachef[2(✉)]

[1] Laboratoire de Mathématiques de Versailles, UVSQ, CNRS, Université de Paris-Saclay, 78035 Versailles, France
jpatarin@club-internet.fr

[2] Department of Mathematics, University of Cergy-Pontoise, UMR CNRS 8088, 95000 Cergy-Pontoise, France
valerie.nachef@u-cergy.fr

Abstract. In this paper, we will study some possible generalizations of the famous Diffie-Hellman algorithm. As we will see, at the end, most of these generalizations will not be secure or will be equivalent to some classical schemes. However, these results are not always obvious and moreover our analysis will present some interesting connections between the concepts of commutativity, associativity, and public key cryptography.

Keywords: Diffie-Hellman algorithms · Chebyshev polynomials New public key algorithms

1 Introduction

Classical Diffie-Hellman Key-Exchange Algorithm. The Diffie-Hellman algorithm [5] was the first published key exchange algorithm (1976). In fact, it is rather a two-party key establishment protocol, which also has "ephemeral public key" features. The new functionalities it offers has created a whole new area of science and engineering: public-key cryptography. Since 1976, many more algorithms have been found, and some of them can be seen as generalizations of the original Diffie-Hellman algorithm, for example when the computations are done in an elliptic curve instead of (mod p), where p is a prime number. In this paper, we will study some other possible generalizations and the link between this problem and commutativity or associativity in some mathematical structures (with one way properties).

Let us first recall what was the original Diffie-Hellman algorithm. Let p be a prime number and g be an element of $\mathbb{Z}/p\mathbb{Z}$ such that $x \mapsto g^x \pmod{p}$ is (as far as we know) a one way function. Typically p has more than 1024 bits and g can be a generator of $\mathbb{Z}/p\mathbb{Z}$. Let Alice and Bob (as in the original paper of Diffie and Hellman) be the two persons who want to communicate. Alice randomly chooses a secret value a between 1 and $p-1$, and she sends the value $A = g^a \pmod{p}$ to Bob. Similarly, Bob randomly chooses a secret value b between 1 and $p-1$ and

J. Kaczorowski et al. (Eds.): NuTMiC 2017, LNCS 10737, pp. 104–117, 2018.
https://doi.org/10.1007/978-3-319-76620-1_7

sends $B = g^b \pmod p$ to Alice. Then Alice and Bob are both able to compute a common key $K = g^{a.b} \pmod p$ (Alice by computing $K = B^a \pmod p$), and Bob by computing $K = A^b \pmod p$). However, if an adversary, Charlie is a passive observer of the messages exchanged on the line, he will obtain A and B, but, if $x \mapsto g^x \pmod p$ is one way, he will not obtain a and b, and if the so called "Diffie-Hellman" problem is difficult, he will not be able to compute K.

Remark 1. If Charlie is also able to send messages, it is well known that this simple algorithm can be attacked by a man in the middle attack. So the messages MUST be authenticated somewhat, for example in usual HTTPS web, the problem is solved. However, this is not the aim of this paper.

DH in a More General Frame. We can state this problem in a more general frame as proposed by Couveignes [4]:

If A is a (semi-)group and G is a set, then a (left) group action φ of A on G is a function:

$$\varphi \colon A \times G \to X \colon (a, g) \mapsto \varphi(a, g)$$

that satisfies the following two axioms (where we denote $\varphi(a, g)$ as $a \cdot g$):

- Identity $e \cdot g = g$ for all g in G. (Here, e denotes the identity element of A if A is a group).
- Compatibility $(ab) \cdot g = a \cdot (b \cdot g)$ for all $a, b \in A$ and all $g \in G$.

We say that A acts transitively on G. We now suppose that A is abelian. We require that the action $(a, g) \to a \cdot g$ is easy to compute but that given g and h in G, it is difficult to compute a such that $a \cdot g = h$. Now we can state DH in this more general frame. Let $g \in G$. Alice choose randomly a secret value $a \in A$ and send the value $a \cdot g$ to Bob. Similarly, Bob choose a secret value $b \in A$ and send $b \cdot g$ to Alice. Then Alice and Bob will share the common value $ab \cdot g$.

In this paper, we look for specific constructions that allow to use the algorithm, and we will assume that Charlie remains a passive attacker and does not create/modify/suppress any messages.

We propose a first type of construction with $A = \mathbb{N}$. For this it is enough to have a set G with a associative composition $*$. The structure of $\mathbb{N}$-set is given by $g^n := n-1$-fold composition of g with itself. It is obvious that $(g^a)^b = g^{a \cdot b} = (g^b)^a$ and so we can use it for key exchange if, and only if, $g \mapsto g^n$ is one way to make the system secure.

In this paper we shall construct compositions $*$ on (affine) curves of genus 0 over finite fields. To find them we first go to such curves over $\mathbb{R}$ and use addition formulas for trigonometric functions to define compositions over $\mathbb{R}$. The next step then is to describe these compositions given by transcendental functions algebraically over the finite fields. Since associativity is inherited, we can use them to define a $\mathbb{N}$-set.

Remark 2

- We recall that the algebraic addition law on elliptic curves E (i.e. curves of genus 1) over any field is modeled after the addition theorems of elliptic functions, e.g. the Weierstrass $\wp$-function and $\wp'$-function.
- The $\wp$-function alone yields a partial $\mathbb{N}$-structure on $\mathbb{F}_q$ by the well-known formulas for the $X-$coordinate of the point $n \cdot (x, y)$ for $(x, y) \in E(\mathbb{F}_q)$.
- A generalization for hyperelliptic curves of genus ≥ 2 is, at least in principle, given by Theta-functions. For $g = 2$ this becomes very efficient [7].

Another possible constructions is to choose A as a family of functions with the composition law and that are pairwise commuting, defined on a space X. The action is then defined by $(f, x) \in A \times X \mapsto f(x)$. Here, A, is not necessary a group, but we can observe that commutativity is needed to be able to have DH in this context. In general, it is quite easy to design a very general commutative internal law on the elements (for example if $a \leq b$ we define $a * b$ as a fixed random element $\varphi(a, b)$ and if $b < a$, we define $a * b$ as $\varphi(b, a)$), but we want here associativity, not commutativity. On the opposite, for functions $f(x) = x^a$ and $g(x) = x^b$, we want $f \circ g = g \circ f$, i.e. commutativity. Here the composition of functions $\circ$ is always associative, but we want commutativity.

	Commutativity	Associativity
On the elements	Easy	What we want
On the functions	What we want	Easy

Quantum Computing on These Structures. We know that the quantum Shor's algorithm for factoring number or computing discrete logarithm $\pmod{p}$ in a finite field is polynomial. In the more general frame of groups operating on sets, when the group is abelian, one can only expect subexponential security [6]. Thus in our constructions, one cannot expect to obtain exponential security against quantum computing. This justifies the Feo [9] system using isogenies of supersingular elliptic curves.

Organisation of the Paper. In part I, we will concentrate on the first type of constructions, i.e. on the "associativity" property. In part II, we will concentrate on the second type on constructions, i.e. on the "commutativity" property, to generalize the fact that $(g^a) \circ (g^b) = (g^b) \circ (g^a)$, on the mathematical structure $(G, \circ)$.

Part I: Associative Properties on the Elements

In this part, we focus on the first type of construction and we present two examples. We work on affine curves of genus 0. Thus will end up with algebraic linear group of dimension 0. Indeed, we can get only tori or additive groups. This implies that we come to discrete logarithm in the multiplicative group of finite fields, as our examples will show.

2 Associativity with $a\sqrt{1+b^2}+b\sqrt{1+a^2}$

To generalize the Diffie-Hellman algorithm by working in a structure $(G,*)$ different from $(\mathbb{Z}/p\mathbb{Z},\times)$, we want:

- $*$ to be associative
- $x \mapsto g^x$ to be one way (from the best known algorithms, the existence of proven one way functions is an open problem since it would imply $P \neq NP$).

Moreover, we would like G to be as small as possible, but with a security greater than 2^{80}. Therefore, elements of G would have typically between 80 bits (or 160 bits if from a collision $g^x = g^y A$, we can find z such that $g^z = A$) and 2048 bits for example, since the computation of $a*b$ is expected to be fast. This is what we have on elliptic curves, but is it possible to suggest new solutions? Ideally, it would be great to generate a "random associative" structure on elements of size, say, about 200 bits for example. It is very easy to generate "random commutative" structures on elements of such size. Let for example a and b be two elements of 256 bits. If $a \leq b$, we can choose $a*b$ to be anything (for example $a*b = AES - CBC_k(a\|b)$ where k is a random value of 128 bits to be used as the AES key) and if $b < a$ then to define $b*a$ as $a*b$. However here we want to design a "random associative" structure on elements of about 200 bits and not a "random commutative" structure, and this is much more difficult! In fact, for associativity structure of this size, we do not know how to get them if we do not create a specific mathematical structure that gives the associativity. But then, there is a risk that such a structure could be used to attack the scheme. In this section, we will study an example of associativity created in this way. More precisely, we will study the operation $a*b = a.\sqrt{1+b^2}+b.\sqrt{1+a^2}$ on a set G where $.,+$ and $\sqrt{}$ can be defined (we will see examples below). Let us first see why $*$ is associative on various G.

2.1 Associativity in $(\mathbb{R},*)$

Definition 1. $\forall a,b \in \mathbb{R},\ a*b = a.\sqrt{1+b^2}+b.\sqrt{1+a^2}$.

We will see that $(\mathbb{R},*)$ is a group. In fact the only difficult part in the proof is to prove the associativity of $*$. We will see 3 different proofs of this fact, since all of these proofs are interesting.

Associativity of *: Proof n°1. A nice way to prove the associative property is to notice that sinh function is a bijection from $\mathbb{R}$ to $\mathbb{R}$ that satisfies: $\forall a \in \mathbb{R}$, $\forall b \in \mathbb{R}$, $\sinh(a+b) = \sinh(a)*\sinh(b)$ (since $\sinh(a+b) = \sinh a \cosh b + \sinh b \cosh a$). This shows that sinh is an isomorphism from $(\mathbb{R},+)$ to $(\mathbb{R},*)$ and therefore $*$ is associative and $(\mathbb{R},*)$ is a group.

Associativity of *: Proof n°2

Theorem 1

$$\forall a \in \mathbb{R}, \forall b \in \mathbb{R}, \left(a\sqrt{b^2+1}+b\sqrt{a^2+1}\right)^2+1=\left(ab+\sqrt{a^2+1}\sqrt{b^2+1}\right)^2$$

Proof. It is obvious by developing the two expressions.

Theorem 2

$$\forall a,b,c \in \mathbb{R},\ (a*b)*c = a*(b*c)$$

Proof Let $\alpha = a\sqrt{b^2+1}+b\sqrt{a^2+1}$. Then $A=(a*b)*c=\alpha*c=\alpha\sqrt{c^2+1}+c\sqrt{\alpha^2+1}$. Now from Theorem 1, $\sqrt{\alpha^2+1}=ab+\sqrt{a^2+1}\sqrt{b^2+1}$ (this is true even when $a<0$ or $b<0$). Therefore $(a*b)*c=(a\sqrt{b^2+1}+b\sqrt{a^2+1})\sqrt{c^2+1}+abc+c\sqrt{a^2+1}\sqrt{b^2+1}$. Similarly, let $\beta=b\sqrt{c^2+1}+c\sqrt{b^2+1}$. Then $B=a*(b*c)=a*\beta=a\sqrt{\beta^2+1}+\beta\sqrt{a^2+1}$. Then from Theorem 1, $\sqrt{\beta^2+1}=bc+\sqrt{b^2+1}\sqrt{c^2+1}$. Therefore $B=a*(b*c)=abc+a\sqrt{b^2+1}\sqrt{c^2+1}+(b\sqrt{c^2+1}+c\sqrt{b^2+1})\sqrt{a^2+1}$. Thus we obtain $A=B$.

Associativity of *: Proof n°3. Here, we will define a law on $\mathbb{R}^2$, called "Domino Law" and represented by $\boxminus$.

Definition 2. *Let* $(a,\alpha)\in\mathbb{R}^2$ *and* $(b,\beta)\in\mathbb{R}^2$. *Then the* $\boxminus$ *law is defined by*

$$(a,\alpha)\boxminus(b,\beta)=(a\beta+b\alpha, ab+\alpha\beta)$$

We can notice that $\boxminus$ is very similar to the multiplication in $\mathbb{C}$, except that we have $ab+\alpha\beta$ instead of $ab-\alpha\beta$. Here $a\beta+b\alpha$ is the analog of the imaginary part and $ab+\alpha\beta$ is the analog of the real part.

Proposition 1. *The* $\boxminus$ *law is associative:*

$$\forall (a,\alpha),(b,\beta),(c,\gamma),\quad (a,\alpha)\boxminus[(b,\beta)\boxminus(c,\gamma)]=[(a,\alpha)\boxminus(b,\beta)]\boxminus(c,\gamma)$$

Proof. It is easy to see that

$$\begin{aligned}(a,\alpha)\boxminus[(b,\beta)\boxminus(c,\gamma)] &= [(a,\alpha)\boxminus(b,\beta)]\boxminus(c,\gamma)\\ &= (abc+a\beta\gamma+b\alpha\gamma+c\alpha\gamma, ab\gamma+ac\beta+\alpha bc+\alpha\beta\gamma)\end{aligned}$$

Corollary 1. *The* $*$ *law is associative.*

Proof. First, using Theorem 1, we notice that $(a,\sqrt{1+a^2})\boxminus(b,\sqrt{1+b^2})=(a*b,\sqrt{1+(a*b)^2})$. Therefore, the associativity of $\boxminus$ implies the associativity of $*$, since $*$ is the restriction of $\boxminus$ on the curve $b^2=a^2+1$.

2.2 Application to Finite Fields: A New Group $(P,*)$ for Cryptography

Let K be a finite field. Let $P(K)=\{x\in K, \exists\alpha\in K, 1+x^2=\alpha^2\}$. When $a\in P$, let $\sqrt{a^2+1}$ denote any value α such that $\alpha^2=a^2+1$ (we will choose later if $\sqrt{a^2+1} = \alpha$ or $\sqrt{a^2+1} = -\alpha$). At this stage, we will only need that $\sqrt{a^2+1}$ denotes always the same value, α, or $-\alpha$ when a is fixed. When there is no ambiguity, $P(K)$ will be simply denoted by P.

Theorem 3

$$\forall a \in P,\, \forall b \in P,\, (a\sqrt{b^2+1} + b\sqrt{a^2+1})^2 + 1 = (ab + \sqrt{a^2+1}\sqrt{b^2+1})^2$$

Proof. As with Theorem 1, the proof is obvious: we just have to develop the two expressions.

Definition 3. *When $a \in P$ and $b \in P$, we will denote by $a * b = a\sqrt{b^2+1} + b\sqrt{a^2+1}$.*

Remark 3. For $\sqrt{a^2+1}$ we have two possibilities in K, α and $-\alpha$, and for $\sqrt{b^2+1}$, we also have two possibilities, β and $-\beta$. Therefore, for $a * b$, we have so far 4 possibilities. So far we just assume that one of these possibilities is choosen, and later (at the end of this Sect. 2.2) we will see how to choose one of these 4 possibilities in order to have a group $(P, *)$. Moreover we will always choose $\sqrt{1} = 1$.

Theorem 4. $*$ *is associative on P.*

Proof. This comes directly from Theorem 3 with the same proof as proof n°2 on $(\mathbb{R}, *)$.

Therefore, we can design a variant of the Diffie-Hellman scheme on $(P, *)$. To be more precise, we will now explain how to compute $\sqrt{1+a^2}$ explicitly.

Theorem 5. *We have the following properties:*
$\forall a \in P,\, a * 0 = 0 * a = a$ $\quad$ $\forall a, b \in P,\, (-a) * (-b) = -(a * b)$
$\forall a \in P,\, a * (-a) = (-a) * a = 0$ $\quad$ $\forall a, b \in P,\, (-a) * b = -(a * (-b))$

Proof. This comes immediately from $\sqrt{1} = 1$ and from the fact that $\sqrt{a^2+1}$ will always be the same value in all the expressions used for $*$.

Theorem 6. $\forall a, b \in P,\ a * b \in P$.

Proof. From Theorem 3, $1 + (a * b)^2$ is a square.

Theorem 7

$$[\forall a, b \in P,\, \sqrt{(ab + \sqrt{a^2+1}\sqrt{b^2+1})^2} = ab + \sqrt{a^2+1}\sqrt{b^2+1}]$$
$$\Longrightarrow \forall a, b, c \in P,\ a * (b * c) = (a * b) * c$$

Proof. Let $A = (a * b) * c$ and $B = a * (b * c)$. Let $\alpha = a\sqrt{b^2+1} + b\sqrt{a^2+1}$. Let $\beta = b\sqrt{c^2+1} + c\sqrt{b^2+1}$. From Theorem 7 we have $\sqrt{\alpha^2+1} = \pm ab + \sqrt{a^2+1}\sqrt{b^2+1}$ and similarly $\sqrt{\beta^2+1} = \pm bc + \sqrt{b^2+1}\sqrt{c^2+1}$. Therefore $A = (a\sqrt{b^2+1} + b\sqrt{a^2+1})\sqrt{c^2+1} \pm c(ab + \sqrt{a^2+1}\sqrt{b^2+1})$ and $B = (b\sqrt{c^2+1} + c\sqrt{b^2+1})\sqrt{a^2+1} \pm a(bc + \sqrt{b^2+1}\sqrt{c^2+1})$. We see that if here we will have two "+", then $A = B$, i.e. a sufficient condition to have $A = B$ is to have $\forall a, b \in P,\, \sqrt{(ab + \sqrt{a^2+1}\sqrt{b^2+1})^2} = ab + \sqrt{a^2+1}\sqrt{b^2+1}$.

We will denote by $\sharp$ this condition

$$\forall a, b \in P,\ \sqrt{(ab+\sqrt{a^2+1}\sqrt{b^2+1})^2} = ab+\sqrt{a^2+1}\sqrt{b^2+1}\ (\sharp)$$

From Theorem 3, $\sharp$ also means:

$$\forall a, b \in P,\ \sqrt{1+(a*b)^2} = ab+\sqrt{1+a^2}\sqrt{1+b^2} \quad (\sharp\sharp)$$

From $(\sharp\sharp)$ and $a*b = a\sqrt{1+b^2}+b\sqrt{1+a^2}$, we see that from $(a, \sqrt{1+a^2})$, $(b, \sqrt{1+b^2})$, we can compute $\left(a*b, \sqrt{1+(a*b)^2}\right)$ with 4 multiplications and 2 additions in K. With $a=b$ in $(\sharp)$, we obtain:

$$\forall a \in P,\ \sqrt{(2a^2+1)^2} = 2a^2+1\ (\natural)$$

2.3 A Toy Example for $(P, *)$

Here we have $K = \mathbb{Z}/19\mathbb{Z}$ with $p = 19$ ($p \equiv 3 \pmod 4$ as wanted). The set of all the squares of K is $C = \{0, 1, 4, 5, 6, 7, 9, 11, 16, 17\}$.

$\forall a \in K, a^2+1$ is a square $\Leftrightarrow a^2 \in \{0, 4, 5, 6, 16\} \Leftrightarrow a \in P$ with $P = \{0, 2, 4, 5, 9, 10, 14, 15, 17\}$. We denote by P this set. Therefore in P we have 9 values (i.e. $\frac{p-1}{2}$ values). For example, let assume that we want to compute $5*9$. We have: $5*9 = 5\sqrt{82}+9\sqrt{26} = 5\sqrt{6}+9\sqrt{7}$. Now $\sqrt{6}$ can be 5 or 14, and $\sqrt{7}$ can be 8 or 11, so for $5*9$ we have 4 possibilities here. In order to see what the exact values are for $\sqrt{6}$ and $\sqrt{7}$, we use the formula: $\forall a \in P,\ \sqrt{(2a^2+1)^2} = 2a^2+1\ (\natural)$. To compute $\sqrt{6}$, we first solve the equation $(2a^2+1)^2 = 6$. This gives $2a^2+1 = 5$ or 14, thus $2a^2 = 4$ or 13. Since $2^{-1} = 10 \pmod{19}$), we obtain $a^2 = 40$ or 130, i.e. $a^2 = 2$ or 16. This gives $a = 4$ or 15. Now, $(\natural)$ with $a = 4$ (or 15) gives: $\sqrt{6} = 14$.

Similarly, to compute $\sqrt{7}$ we first solve the equation $(2a^2+1)^2 = 7$. This gives $2a^2+1 = 11$ or 8. Thus we have $2a^2 = 10$ or 17 and $a^2 = 5$ or 13. Thus $a = 9$ or 10. Now $(\natural)$ with $a = 9$ (or 10) gives: $\sqrt{7} = 11$. Finally $5*9 = 5\sqrt{6}+9\sqrt{7} = 17$. All the values $a*b$ with $a, b \in P$ can be computed in the same way. We obtain like this the table below of the group $(P, *) = P(\mathbb{Z}/19\mathbb{Z})$.

2.4 A More General Context

Definition and Properties. The Domino Law can be defined also on $P \times P$. It is still associative (the proof is similar to the one given for $\mathbb{R}^2$) (Table 1).

Proposition 2. *Let $(a, b) \in P \times P$, then $(a, b) \boxminus (a, b) = (2ab, a^2+b^2)$. If $(a, b)^2_{\boxminus} = (A, B)$, then $A+B = (a+b)^2$.*

More generally, $\forall k \in \mathbb{N}$, if $(a, b)^k_{\boxminus} = (A, B)$ then $A+B = (a+b)^k$.

Proof. For $k = 2$, the computation is straightforwards. Then, the proof is done by induction.

Corollary 2. *Proposition 2 shows that computing logarithms in $(P \times P, \boxminus)$ is equivalent to computing logarithms in $(K, .)$*

Proof. The proof is obvious.

Table 1. $P(\mathbb{Z}/19\mathbb{Z})$

$*$	0	2	4	5	9	10	14	15	17
0	0	2	4	5	9	10	14	15	17
2	2	17	5	10	14	4	15	9	0
4	4	5	9	14	2	15	17	0	10
5	5	10	14	15	17	9	0	2	4
9	9	14	2	17	5	0	10	4	15
10	10	4	15	9	0	14	2	17	5
14	14	15	17	0	10	2	4	5	9
15	15	9	0	2	4	17	5	10	14
17	17	0	10	4	15	5	9	14	2

Application to $a * b = a\sqrt{1+b^2} + b\sqrt{1+a^2}$

Proposition 3. *We have:* $(a, \sqrt{1+a^2}) \boxminus (b, \sqrt{1+b^2}) = \left(a * b, \sqrt{1+(a*b)^2}\right)$.
Hence $\forall k,\ (a, \sqrt{1+a^2})^k_{\boxminus} = (a^k_*, \sqrt{1+(a^k_*)^2})$

Corollary 3. *This proposition shows that computing logarithms in* $(P, *)$ *is equivalent to computing logarithms in* $(K, .)$*.*

Proof. We want to compute k such that $a^k_* = \alpha$ (a and α are known). We first choose β such that $\beta^2 = \alpha^2 + 1$. Then, we want to find (a, b) satisfying $b^2 = a^2 + 1$ such that $(a, b)^k_{\boxminus} = (A, B)$. Since $\alpha + \beta = (a + b)^k$, this equation gives k by using the discrete log.

Therefore the cryptographic scheme based on $(P, *)$ is essentially similar to the classical cryptographic scheme based on discrete logarithms on $(K, .)$.

3 Associativity Based on the Hyperbolic Tangent

3.1 The General Case

In this section, we will use the tanh function to obtain associativity. This function is a bijection from $\mathbb{R}$ to $]-1, 1[$ and we have the formula

$$\tanh(a+b) = \frac{\tanh a + \tanh b}{1 + \tanh a \tanh b}$$

Thus if we define on $]-1, 1[$ the following law: $a * b = (a+b)(1+ab)^{-1}$ we obtain a group since tanh is an isomorphism from $(\mathbb{R}, +)$ to $(]-1, 1[, *)$. Similarly, we will work on finite fields. Let K be a finite field. We suppose that in K, -1 is not a square. When we can perform the computation (i.e. when $ab \neq -1$), we define:

$$a * b = (a+b)(1+ab)^{-1}$$

We have the following properties:

Proposition 4. *1.* $\forall a \in K,\ a * 0 = a$.
2. $\forall a \in K \setminus \{-1\}$, $a * 1 = 1$ *and* $\forall a \in K \setminus \{1\}$, $a * (-a) = 0$.
3. $\forall a, b,\ ab \neq -1,\ (-a) * (-b) = -(a * b)$.
4. $\forall a, b, c$, $(a * b) * c = a * (b * c)$ *when the computation is possible, i.e.* $*$ *is associative.*

Proof. Properties 1, 2 and 3 are straightforward. We will prove that $*$ is associative.

$$(a * b) * c = [(a + b)(1 + ab)^{-1} + c][1 + (a + b)(1 + ab)^{-1}c]^{-1}$$

We multiply by $(1 + ab)(1 + ab)^{-1}$. This gives:

$$(a * b) * c = [((a + b)(1 + ab)^{-1} + c)(1 + ab)][(1 + (a + b)(1 + ab)^{-1}c)(1 + ab)]^{-1}$$

$$(a * b) * c = [a + b + c + abc][(1 + ab + bc + ac]^{-1}$$

Similarly

$$a * (b * c) = [a + (b + c)(1 + bc)^{-1}][1 + a(b + c)(1 + bc)^{-1}]^{-1}$$

Here we multiply by $(1 + bc)(1 + bc)^{-1}$ and we obtain

$$a * (b * c) = [a + b + c + abc][(1 + ab + bc + ac]^{-1}$$

Remark 4. There is an analog with the addition law of the speed in simple relativity: $\frac{v_1+v_2}{1+\frac{v_1 v_2}{c}}$. From this, it is also possible to justify associativity from physical considerations.

3.2 A Toy Example

In Table 2, we give the example of the construction of a group denoted $(Q(K), *)$ when $K = \mathbb{Z}/19\mathbb{Z}$ and $*$ is the law based on the tanh function. Here -1 is not a square since $19 \equiv 3 \pmod 4$. We already know that 1 and 18 are not elements of $Q(K)$. When we do the computations, we obtain that for $Q(K) = \{0, 2, 3, 4, 7, 12, 15, 16, 17\}$. We also have that $Q(K) = \langle 3 \rangle$.

3.3 Computing Log with $*$ (Analog of tanh)

We will now study the power for $*$ of an element of K. We will use the following notation: $a_*^k = \underbrace{a * a * \ldots * a}_{k \text{ times}}$.

Proposition 5. *Suppose that we can perform the computations (i.e. we never obtain the value* -1 *during the computations).* $\forall a \in K, \forall k,\ a_*^k = s_k t_k^{-1}$ *with* $s_k = (1 + a)^k - (1 - a)^k$ *and* $t_k = (1 + a)^k + (1 - a)^k$*. Then* $s_k + t_k = 2(1 + a)^k$.

Table 2. $(Q(\mathbb{Z}/19\mathbb{Z}), *)$

$*$	0	2	3	4	7	12	15	16	17
0	0	2	3	4	7	12	15	16	17
2	2	16	17	7	12	15	3	4	0
3	3	17	12	2	16	4	7	0	15
4	4	7	2	15	3	17	0	12	16
7	7	12	16	3	17	0	2	15	4
12	12	15	4	17	0	2	16	3	7
15	15	3	7	0	2	16	4	17	12
16	16	4	0	12	15	3	17	7	2
17	17	0	15	16	4	7	12	2	3

Proof. We have $a_*^1 = a * 0$. Then $a_*^2 = a * a = 2a(1+a^2)^{-1}$. Since $s_2 = 2a$ and $t_2 = 2(1+a^2)$, we have $a_*^2 = s_1 t_2^{-1}$. Suppose that $a_*^{k-1} = s_{k-1}t_{k-1}^{-1}$. Then $a_*^k = a * a_*^{k-1} = (a + s_{k-1}s_{k-1}^{-1})(1 + as_{k-1}s_{k-1})^{-1}$. We multiply this expression by $t_{k-1}t_{k-1}^{-1}$. We obtain that $a_*^k = s_k s_k^{-1}$ with $s_k = at_{k-1} + s_{k-1}$ and $t_k = t_{k-1} + as_{k-1}$. Thus we can write:

$$\begin{bmatrix} s_k \\ t_k \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} \begin{bmatrix} s_{k-1} \\ t_{k-1} \end{bmatrix}$$

This gives:

$$\begin{bmatrix} s_k \\ t_k \end{bmatrix} = A^{k-1} \begin{bmatrix} s_1 \\ t_1 \end{bmatrix}$$

with $A = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}$. By diagonalizing the matrix A, we obtain that:

$$a_*^k = s_k t_k^{-1} \text{ with } s_k = (1+a)^k - (1-a)^k \text{ and } t_k = (1+a)^k + (1-a)^k$$

Then we get $u_k + v_k = (1+a)^k$. This can also be proved by induction.

Corollary 4. *If a_*^k exists, then $(-a)_*^k = -a_*^k$.*

Corollary 5. *Let $a \in K$.*

1. *If there exists $k(a) \in \mathbb{N}^*$ such that $\forall k < k(a),\ s_k \neq 0,\ t_k \neq 0$ and $s_{k(a)} = 0, t_{k(a)} \neq 0$, then $(\langle a \rangle, *)$ is a group.*
2. *If there exists $k'(a) \in \mathbb{N}^*$ such that $\forall k < k'(a),\ s_k \neq 0,\ t_k \neq 0$ and $t_{k'(a)} = 0$, then a does not generate a group.*

We recall the results obtained in Proposition 5: $\forall a \in K$, $\forall k$, $a_*^k = s_k t_k^{-1}$ with $s_k = (1+a)^k - (1-a)^k$ and $t_k = (1+a)^k + (1-a)^k$. Then $s_k + t_k = 2(1+a)^k$. Let $\alpha = a_*^k$. It is possible to compute k from α and a like this:

$$\alpha = a_*^k = s_k t_k^{-1} = \frac{(1+a)^k - (1-a)^k}{(1+a)^k + (1-a)^k}$$

$$\alpha = \frac{1 - \left(\frac{1-a}{1+a}\right)^k}{1 + \left(\frac{1-a}{1+a}\right)^k}$$

Then we can find $\left(\frac{1-a}{1+a}\right)^k$ and finally we obtain k by using the discrete log. This shows that computing logarithms for the $*$ law is essentially the same as for the classical case. Therefore the cryptographic scheme based on this law $*$ (analog to tanh) is again essentially similar to the classical cryptographic scheme based on the discrete logarithm.

4 Widen the Range

As pointed out by Jérôme Plût to us, it seems that there is a little hope to find "magic algebraic curves" that are more efficient than elliptic curves. In particular, our curve $b^2 = a^2 + 1$ had little chance to be useful due to general results on the classification of algebraic groups. For any abelian algebraic group, there exist unique decompositions:

- $0 \to G^0 \to G \to \pi_0(G) \to 0$ where G^0 is connexe and $\pi(G)$ is étale.
- $0 \to L \to G^0 \to A \to 0$ where A is an abelian variety and L is a linearizable group.
- $0 \to U \to L \to T \to 0$ where T is a torus, and U is unipotent.

The first and the third decompositions are rather simple. The second one is more complicated and can be found in [1].

Therefore the only possibility to get more efficient systems is to use curves of genus larger than 1 and varieties related to their Jacobians, which are accessible to effective computation. But because of security reasons it is very doubtful that one can use curves of genus larger than 3 (key word: index-calculus). As said already in Remark 2, Theta functions lead to the very efficient Kummer surfaces for $g = 2$.

Part II: Commutative Properties on the Functions

5 Chebyshev Polynomials

To generalize the Diffie-Hellman Algorithm by using $(f \circ g)(a) = (g \circ f)(a)$, we want:

- f and g to be one way
- f and g to be easy to compute
- $f \circ g = g \circ f$, i.e. commutativity.

The value a is typically between 80 and 2048 bits (as in Sect. 2). Ironically, here (unlike in Part I) associativity is very easy, since $\circ$ is always associative, but we want commutativity on f and g, and this is not easy to obtain. In part I, we had a law $*$ on elements of G with about 160 bits, but here, we work with functions f and g on G and we have more functions from G to G than elements of G. Moreover $a_*^i = \underbrace{a * a * \ldots * a}_{i \text{ times}}$ can be computed in $O(\ln i)$ with square and multiply, while $f^i(a) = f[f \ldots f(a))]$ would generally require $O(i)$ computations of f. An interesting idea is to use the Chebyshev polynomials (cf. [2,8,10–12,15] for example). In [14], the structure of Chebyshev polynomials on $\mathbb{Z}/p\mathbb{Z}$ is also studied. However, as mentioned in some of these papers, and as we will see below, public key schemes based on Chebyshev polynomials have often exactly the same security than public key schemes based on monomials. We will present here only a few properties.

Some Properties of Chebyshev Polynomials on $\mathbb{R}$

The Chebyshev polynomials T_n can be defined as the polynomials such that:

$$\cos nx = T_n(\cos x) \tag{1}$$

Since $\cos a + \cos b = 2\cos(\frac{a+b}{2})\cos(\frac{a-b}{2})$, we have: $\cos(n+1)x + \cos(n-1)x = 2\cos x \cos nx$, and therefore we have:

$$T_{n+1}(X) = 2XT_n(X) - T_{n-1}(X). \tag{2}$$

For example, the first polynomials are: $T_0 = 1$, $T_1 = X$, $T_2 = 2X^2 - 1$, $T_3 = 4X^3 - 3X$, $T_4 = 8X^4 - 8X^2 + 1$. From 1, we can see that the Chebyshev polynomials commute: $(T_n(T_m(X)) = T_m(T_n(X))$ since $\cos(nm)x = \cos(mn)x$. Therefore, we can design analog of the Diffie-Hellman or RSA schemes by using Chebyshev polynomials instead of the monomial transformation $X \mapsto X^a$. Moreover, from 2, we can write:

$$\begin{bmatrix} T_n(X) \\ T_{n+1}(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix} \begin{bmatrix} T_{n-1}(X) \\ T_n(X) \end{bmatrix}$$

and this gives

$$\begin{bmatrix} T_n(X) \\ T_{n+1}(X) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2X \end{bmatrix}^n \begin{bmatrix} 1 \\ X \end{bmatrix} \tag{3}$$

Now from 3 we can obtain:

$$T_n(X) = U \circ X^n \circ U^{-1} \tag{4}$$

with $U(X) = \frac{X+\frac{1}{X}}{2}$ if $X \neq 0$ and $U^{-1}(X) = X + \sqrt{X^2-1}$ if $X > 1$. Therefore, if $|X| \geq 1$, $T_n(X) = \frac{1}{2}\left(\left(X - \sqrt{X^2-1}\right)^n + \left(X + \sqrt{X^2-1}\right)^n\right)$. Property (4) is very nice since it shows that we can compute $T_n(X)$ about as fast as a X^n (and we use an analog of the square and multiply algorithm), so we can compute

$T_n(X)$ efficiently even when n has a few hundred or thousands of bits. However, property 4 also shows that $T_n(X)$ and X^n are essentially the same operation since U and U^{-1} can be considered as public.

Properties of Chebyshev Polynomials on Other Spaces.
For cryptographic use, it has been suggested to use Chebyshev polynomials on various spaces. In fact, it could be assumed that the analysis of Chebyshev polynomials properties for cryptography would depend on the type of space where the computations are done (finite fields with characteristic equal or not equal to 2, computations modulo n with n prime or not prime, etc.). However, most of the time, the above properties on real numbers suggest that public key cryptography based on Chebyshev polynomials is essentially the same as (classical) public key cryptography based on X^n (see [8,10–12,14,15] for details).

Remark 5. After our presentation at the NuTMiC conference (Warsaw 2017), Gérard Maze pointed out to us that in his PhD Thesis (Chap. 6) [13], he had also studied how to use Chebyshev polynomials for public key cryptography. His conclusions were similar to ours, i.e. when the Chebyshev polynomials are properly used, the resulting schemes are essentially the same as schemes based on discrete log.

6 Commutativity with Other Polynomials

We first give the definition of a commutative family of polynomials.

Definition 4. *Let (Q_n) be a family of polynomials. We say that we have a family of polynomials that commute if $\forall n$, $\forall m$, $Q_n \circ Q_m = Q_m \circ Q_n$.*

If we look for infinite family of polynomials satisfying commutativity, the Block and Thielman theorem [3] shows that we do not have many solutions. More precisely:

Theorem 8 *(Bloch and Thielman 1951)***.** *Let (Q_n) be a polynomial of degree n. If $(Q_n)_{n \geq 1}$ is a family of polynomials that commute, then there exists a polynomial of degree 1, U, such that, either for all n, $Q_n = U \circ X^n \circ U^{-1}$ or for all n, $Q_n = U \circ T_n \circ U^{-1}$, where T_n is the Chebyshev polynomial of degree n.*

For cryptographic use, we may look for "sufficiently large" families of polynomials that commute (instead of "infinite families") but it seems difficult to find new large families. Some suggestions are given in [13], but more possibilities should exist and could be the subject of further work.

7 Conclusion

In this paper, we investigated several methods to construct algebraic generalizations of the Diffie-Hellman key exchange algorithm. However, after our analysis, it appears that the proposed schemes are essentially equivalent to the classical ones. Nevertheless, the study showed that there are interesting connections between associativity, commutativity and the construction of such algorithms. We also explained that there is little hope to find "magic algebraic curves" more efficient than elliptic curves and we suggested to study "large" but not infinite families of polynomials that commute for further analysis.

Acknowledgment. The authors want to thank Jérôme Plût, Gerhard Frey and Gérard Maze for very useful comments and particularly Gerhard Frey for his help to improve the presentation of this paper. We have met Gerhard Frey and Gérard Maze at NuTMiC 2017 in Poland.

References

1. Barsotti, I.: Un Teorema di structura per le variettà di gruppali. Rend. Acc. Naz. Lincei **18**, 43–50 (1955)
2. Bergamo, P., D'Arco, P., de Santis, A., Kocarev, L.: Security of Public Key Cryptosystems based on Chebyshev Polynomials. arXiv:cs/0411030v1, 1 February 2008
3. Block, H.D., Thielman, H.P.: Commutative Polynomials. Quart. J. Math. Oxford Ser. **2**(2), 241–243 (1951)
4. Couveignes, J.M.: Hard Homogeneous Spaces. Cryptology ePrint archive: 2006/291: Listing for 2006
5. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976)
6. Frey17: Deep Theory, efficient algorithms and surprising applications. In: NuTMiC (2017)
7. Gaudry, P., Lubicz, D.: The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines. Finite Fields Appl. **15**(2), 246–260 (2009)
8. Hunziker, M., Machiavelo, A., Parl, J.: Chebyshev polynomials over finite fields and reversibility of σ-automata on square grids. Theor. Comput. Sci. **320**(2–3), 465–483 (2004)
9. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 19–34. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_2
10. Kocarev, L., Makraduli, J., Amato, P.: Public key encryption based on Chebyshev polynomials. Circ. Syst. Sign. Process. **24**(5), 497–517 (2005)
11. Li, Z., Cui, Y., Jin, Y., Xu, H.: Parameter selection in public key cryptosystem based on Chebyshev polynomials over finite field. J. Commun. **6**(5), 400–408 (2011)
12. Lima, J.B., Panario, D., Campello de Sousa, R.M.: Public-key cryptography based on Chebyshev polynomials over $G(q)$. Inf. Process. Lett. **111**, 51–56 (2010)
13. Maze, G.: Algebraic Methods for Constructing One-Way Trapdoor Functions. Ph.D. thesis - University oof Notre Dame (2003). http://user.math.uzh.ch/maze/
14. Rosen, J., Scherr, Z., Weiss, B., Zieve, M.: Chebyshev mappings over finite fields. Amer. Math. Monthly **119**, 151–155 (2012)
15. Sun, J., Zhao, G., Li, X.: An improved public key encryption algorithm based on Chebyshev polynomials. TELKOMNIKA **11**(2), 864–870 (2013)