symmetry

MDPI

Article

Public Key Protocols over Twisted Dihedral Group Rings

María Dolores Gómez Olvera *, Juan Antonio López Ramos and Blas Torrecillas Jover

Department of Mathematics, University of Almería, 04120 La Cañada de San Urbano, Spain
* Correspondence: gomezolvera@ual.es

Received: 9 July 2019; Accepted: 3 August 2019; Published: 7 August 2019

Abstract: Key management is a central problem in information security. The development of quantum computation could make the protocols we currently use unsecure. Because of that, new structures and hard problems are being proposed. In this work, we give a proposal for a key exchange in the context of NIST recommendations. Our protocol has a twisted group ring as setting, jointly with the so-called decomposition problem, and we provide a security and complexity analysis of the protocol. A computationally equivalent cryptosystem is also proposed.

Keywords: cryptography; non-commutative algebra; public key cryptography; key management; group ring

1. Introduction

In recent years, intense research has been made in cryptography, especially in relation to new public key protocols. In August 2015, the USA's National Security Agency (NSA) announced plans to upgrade security standards. Improvements in quantum computation make it necessary to replace current protocols with secure quantum ones. In a NIST report [1], there are six proposals to be quantum safe: lattice-based, code-based, multivariate-based, hash-based, isogeny-based, and group-based cryptographic schemes.

In this work, we make a proposal for a quantum-safe public key protocol. In the context of group-based proposals, it is believed that problems such as the conjugate search problem (CSP) are not solvable using quantum computers. We propose the so-called decomposition problem (DP), which is a generalization of the CSP, and the multiplicative monoid of a twisted group ring as a setting in our aim to find a quantum-safe key exchange in the context of group-based cryptography.

> **Decomposition Problem.** Given $(x, y) \in G^2$ and $S \subset G$, the problem is to find $z_1, z_2 \in S$ such that $y = z_1 x z_2$.

Note that the CSP is a special case of this problem where $z_2 = z_1^{-1}$, and that for the DP we do not need invertible elements.

The idea is that even in asymmetric cryptography (more usually called public key cryptography), characterized by having both a secret and public key to encrypt and decrypt (in contrast with symmetric cryptography, which uses the same key for both procedures), the first and last steps in the algorithm use the same key, which is the secret key, i.e., both generation of the public key and computation of the shared key. In terms of Diffie–Hellman key exchange generalization using semigroup actions [2], this would be the following.

Let S be a finite set, G be an abelian semigroup, and ϕ a G-action on S, and a public element $h \in S$. The extended Diffie–Hellman key exchange in (G, S, ϕ) is the following protocol:

1. Alice chooses $a \in G$ and computes $\phi(a,h)$. Alice's private key is a, and her public key is $p_A = \phi(a,h)$.
2. Bob chooses $b \in G$ and computes $\phi(b,h)$. Bob's private key is b, and his public key is $p_B = \phi(b,h)$.
3. Their common secret key is then

$$\phi(a,p_B) = \phi(a,\phi(b,h) = \phi(ab,h) = \phi(ba,h) = \phi(b,\phi(a,h) = \phi(b,p_A).$$

So we can see that both Alice and Bob use their secret key in the first and last steps of the algorithm. In contrast, our purpose would work as follows.

Let S be a finite set, G be a non-abelian semigroup, and ϕ a G-action on S, and a public element $h \in S$. The extended Diffie–Hellman key exchange in (G,S,ϕ) is the following protocol:

1. Alice chooses $a \in G$ and computes $\phi(a,h)$. Alice's private key is a, and her public key is $p_A = \phi(a,h)$.
2. Bob chooses $b \in G$ and computes $\phi(b,h)$. Bob's private key is b, and his public key is $p_B = \phi(b,h)$.
3. Their common secret key is then

$$\phi(a^*,p_B) = \phi(a^*,\phi(b,h) = \phi(a^*b,h) = \phi(b^*a,h) = \phi(b^*,\phi(a,h) = \phi(b^*,p_A),$$

where a^*, b^* depend on a, b and also on the algebraic setting, in our case, in the cocycle of the twisted group ring. In this way, the symmetry that we found using the secret key twice during the key exchange does not occur, and we can show that this is an advantage, for example, when facing attacks like the decomposition attack [3].

The rest of this paper is organized as follows: In Section 2, we give an algebraic setting of twisted group rings. In Section 3, we provide our proposed key exchange and a security analysis of this protocol. Section 4 shows a computationally equivalent cryptosystem. In Section 5, we extend our proposal for n users, where we can observe clearly that there is a lack of symmetry concerning the action of every user. Finally, conclusions are presented in Section 6.

2. Algebraic Setting

In this section, twisted group rings are defined, and we also show some properties that make the key exchange possible. Firstly, we recall the definition of 2-cocycles, which will allow us to define the twisted multiplication.

Definition 1. *Let G be a group and A be an abelian group. An application*

$$\alpha : G \times G \to A$$

is a 2-cocycle if

1. $\alpha(g,1) = \alpha(1,g) = 1$, *for all* $g \in G$,
2. $\alpha(g,h)\alpha(gh,k) = \alpha(g,hk)\alpha(h,k)$, *for all* $g,h,k \in G$.

Now we define twisted group rings as follows.

Definition 2. *Let K be a ring, G be a multiplicative group, and α be a cocycle in $U(K)$. The group ring $K^\alpha G$ is defined to be the set of all finite sums of the form*

$$\sum_{g_i \in G} r_i g_i,$$

where $r_i \in K$ and all but a finite number of r_i are zero.

The sum of two elements in $K^{\alpha}G$ is given by

$$\Big(\sum_{g_i \in G} r_i g_i\Big) + \Big(\sum_{g_i \in G} s_i g_i\Big) = \sum_{g_i \in G} (r_i + s_i) g_i. \tag{1}$$

And multiplication, which is twisted by a cocycle, is given by

$$\Big(\sum_{g_i \in G} r_i g_i\Big) \cdot \Big(\sum_{g_i \in G} s_i g_i\Big) = \sum_{g_i \in G} \Big(\sum_{g_j g_k = g_i} r_j s_k \, \alpha(g_j, g_k)\Big) g_i.$$

When the cocycle α is trivial, $R^{\alpha}G$ is the classical group ring $R[G]$.

As an example, consider the field K, its primitive root of unity t, and the dihedral group of $2m$ elements, $D_{2m} =< x, y : x^m = y^2 = 1, yx^a = x^{m-a}y >$. The group ring $R = K^{\alpha}D_{2m}$, where α is

$$\alpha : D_{2m} \times D_{2m} \to K$$

with $\alpha(x^i, x^j y^k) = 1$ and $\alpha(x^i y, x^j y^k) = t^j \; i, j = 1, ..., 2m-1$, is a twisted group ring.

This example is our concrete proposal for the key exchange, the twisted dihedral group ring $K^{\alpha}D_{2m}$, where K is a finite field of characteristic p such that $p \mid 2m$. This is required in order to R is not a semisimple ring, which has its consequence in the security analysis.

Once we have defined our setting, we establish some useful properties that will allow us to make our key exchange possible.

Definition 3. *Let $R = K^{\alpha}D_{2m}$, where t is the primitive root of unity that generates K and α is the cocycle defined above. Given $h \in R$,*

$$h = \sum_{\substack{0 \le i \le m-1 \\ k=0,1}} r_i x^i y^k,$$

where $r_i \in K$ and $x, y \in D_{2m}$. We define $h^* \in K^{\alpha}D_{2m}$:

$$h^* = \sum_{\substack{0 \le i \le m-1 \\ k=0,1}} r_i t^{-i} x^i y^k,$$

where $r_i \in K$ and $x, y \in D_m$.

Note that $R = K^{\alpha}D_{2m}$ can be written as

$$R = R_1 \oplus R_2,$$

where $R_1 = KC_m$ and $R_2 = K^{\alpha}C_m y$, and C_m is a cyclic group of order m. In this context, we can define $A_j \le R_j$ as

$$A_j = \Big\{ \sum_{i=0}^{m-1} r_i x^i y^k \in R_j : r_i = r_{m-i} \Big\}.$$

Proposition 1. *Given $h_1, h_2 \in R$,*

- *If $h_1, h_2 \in R_1$, then $h_1 h_2 = h_2 h_1$;*
- *If $h_1, h_2 \in A_2$, then $h_1 h_2^* = h_2 h_1^*$, and $h_1^* h_2 = h_2^* h_1$;*
- *If $h_1 \in A_1, h_2 \in A_2$, then $h_1 h_2 = h_2 h_1^*$.*

The proof can be seen in Appendix A.

3. Twisted Key Exchange Protocol

In this section, we explain the key exchange proposed, over the twisted group ring $R = K^{\alpha}D_{2m}$, and discuss the security and complexity of the protocol.

Let $h \in R$ be a random public element. The key exchange between Alice and Bob is as follows:

1. Alice selects a secret pair $s_A = (g_1, k_1)$, where $g_1 \in R_1, k_1 \in A_2 \leq R_2$.
2. Bob selects a secret pair $s_B = (g_2, k_2)$, where $g_2 \in R_1, k_2 \in A_2 \leq R_2$.
3. Alice sends Bob $p_A = g_1 h k_1$, and Bob sends Alice $p_B = g_2 h k_2$.
4. Alice computes $K_A = g_1 p_B k_1^*$, and Bob computes $K_B = g_2 p_A k_2^*$, and they get the same secret shared key.

We can easily check that they get the same shared key, computing

$$K_A = g_1 p_B k_1^* = g_1 g_2 h k_2 k_1^* = g_2 g_1 h k_1 k_2^* = g_2 p_A k_2^* = K_B,$$

since we had $g_1 g_2 = g_2 g_1$ and $k_1 k_2^* = k_2 k_1^*$ by Proposition 1.

Security Analysis of the Protocol

Security of the protocol described above is based on the assumption that the following problem is computationally hard:

> *Given $R = K^{\alpha}D_{2m} = R_1 \oplus R_2$, $A_2 \leq R_2$, and the public elements $h, y \in R$, and the map $* : R \to R$, find $a \in R_1$, $b \in A_2 \leq R_2$ such that $ahb^* = y$.*

The stronger decisional version of this assumption would be:

> *Given $R = K^{\alpha}D_{2m} = R_1 \oplus R_2$, $A_2 \leq R_2$, and the public elements $h, y \in R$, and the map $* : R \to R$, it is hard to distinguish $a \in R_1$, $b \in A_2 \leq R_2$ such that $ahb^* = y$ from a random element of the form ghk, where $g \in R_1$ and $k \in R_2$.*

Now we discuss the security of our protocol against various types of attacks in the literature. The first attack given by [4] takes advantage of the algebraic setting; the second one, from [3], involves the underlying computational problem; the third attack is a variation on our case, and finally, we check the brute force attack.

1. Attacks using decomposition of group rings. Our proposed protocol over $K^{\alpha}D_{2m}$ is not susceptible to this kind of attack because in our case, $char(K) = p \mid 2m = |D_{2m}|$, so our group ring is never semisimple.
2. Decomposition attack (Roman'kov). This attack by Roman'kov cannot be applied directly since secret keys in our case are not selected in that way. But we propose the necessary changes for it to be applicable (mainly, where the secret key belongs). Our proposal is robust against this attack, as can be observed in the following example.

Example 1. *Let $R = GF(2^2)^{\alpha}D_6$, the public element $h = t + (t+1)x + x^2 + ty + xy + x^2y$, and the secret keys*

$$s_A = (1 + tx + tx^2, y + (t+1)xy + (t+1)x^2y),$$
$$s_B = (t + (t+1)x + (t+1)x^2, (t+1)y + xy + x^2y).$$

Then Alice and Bob obtain their public keys

$$p_A = t + tx + (t+1)x^2 + ty + xy + (t+1)x^2y, \qquad p_B = t + x + tx^2 + (t+1)y + txy + x^2y,$$

and the shared key

$$K = (t+1) + x + tx^2 + ty + (t+1)xy + (t+1)x^2y.$$

A passive eavesdropper, Eve, might obtain a basis B of $R_1hR_2 \ni p_A$,

$$B = \{1+tx, 1+x+x^2, (t+1)x+x+tx^2, y+(t+1)xy+tx^2y\}.$$

So she can see p_A *as*

$$p_A = \sum \beta_i a_i h b_i,$$

where

$$a_1 = 1+tx, \quad a_2 = 1+x+x^2, \quad a_3 = 1, \quad a4 = 1+tx+(t+1)x^2,$$

and

$$b_1 = x^2y, \quad b_2 = x^2y, \quad b3 = y+xy+x^2y, \quad b4 = y+xy$$

if she applies the attack, obtains

$$\sum_{i=1}^{3} \alpha_i a_i p_B b_i = t+(t+1)x+x^2+(t+1)y+(t+1)xy \neq K.$$

We can see that this attack would not work since our shared key K cannot be expressed in terms of the basis B.

3. Decomposition as 1-side multiplication. This decomposition is not always possible, and if that is the case, it does not necessarily imply breaking our protocol. We show it by using the following example.

Example 2. *Let us consider* R, h, s_A, s_B *as in the preceding example. A passive eavesdropper, Eve, would try to recover K from* p_A *and* p_B. *Let us assume that she can find* γ *such that*

$$\gamma \cdot h = p_A.$$

In this case, Eve finds $\gamma = y$. *But applying this* γ *to* p_B *is helpless,*

$$\gamma \cdot p_B = (t+1)+(t+1)x+(t+1)x^2+ty+txy+x^y \neq K,$$

$$p_B \cdot \gamma = (t+1)+x+tx^2+ty+txy+x^y \neq K.$$

4. Brute force attack. The complexity of our algorithm is $\mathcal{O}(p^{\frac{3}{4}k})$ for a k-bits long key.

Complexity can be obtained by computing the number of possible keys. Given h public, we have that the set of private keys is R_1hA_2 and the set of shared keys is R_1hA_1. Recall that $R_1 = KC_m$, $R_2 = K^\alpha C_m y$, and $A_j = \left\{ \sum_{i=0}^{m-1} r_i x^i y^k \in R_j : r_i = r_{m-i} \right\}$. In both cases, we have

$$|R_1| = (p^n)^m \qquad |A_j| = (p^n)^{\frac{m}{2}},$$

so an eavesdropper would have to try $(p^n)^m \cdot (p^n)^{\frac{m}{2}} = p^{\frac{3}{2}nm}$ for an nm-bits long key, i.e., $p^{\frac{3}{4}k}$ possibilities for a k-bits long key. In the example proposed in Appendix B, $GF(2^4)^\alpha D_{32}$, for a 128-bits long key, we obtain a security of $\mathcal{O}(2^{96})$.

In terms of complexity, we could say that our protocol is not as good as other protocols in group rings, such as the key exchange proposed in [5] (in our case, the key should be larger for the same security against a brute force attack), but it is still competitive, and it is resistant against attacks such as [4], which breaks the proposal in [5].

Finally, note that we have studied passive attacks, but in case of an active attack, such as a man-in-the-middle attack, we would need extra security in our protocol. It could be solved by using an authenticated channel, with digital signatures.

4. A Public Key Cryptosystem

In this section, we show a computationally equivalent cryptosystem.

Let $R = K^{\alpha}D_{2m}$ be a twisted group ring by the 2-cocycle α, and recall $R = R_1 \oplus R_2$. We consider the following Elgamal-type cryptosystem for encryption and decryption. Suppose Bob wants to send a message to Alice. We have R, and a random element $h \in R$, both public. Alice establishes her public key as follows: she selects $g_1 \in R_1$ and $k_1 \in R_2$, and computes $p_A = g_1hk_1$.

Encryption. *To encrypt a message, Bob executes the following steps:*

1. *Bob selects two secret elements $g_2 \in R_1$ and $k_2 \in R_2$ and computes $x_1 = g_2hk_2$.*
2. *Bob represents the message as an element $m \in R$.*
3. *Bob computes $x_2 = m + g_2p_Ak_2^*$ and sends (x_1, x_2) to Alice.*

Decryption. *Alice decrypts the message m by calculating*

$$m = x_2 + g_1x_1k_1^* = m + g_2(g_1hk_1)k_2^* - g_1(g_2hk_2)k_1^*,$$

given that $g_1g_2 = g_2g_1$ and $k_1k_2^* = k_2k_1^*$ by Proposition 1.

Proposition 2. *Breaking the cryptosystem above is equivalent to breaking the key exchange proposed.*

Proof. Assume that an eavesdropper, Eve, can solve the key exchange, and she wants to get m from the pair

$$(x_1, x_2) = (g_2hk_2, m + g_2p_ak_2^*).$$

Since she is able to break the key exchange, knowing Alice's public key g_1hk_1 and Bob's g_2hk_2 allows her to get $g_2g_1hk_1k_2^*$ (the shared key). So she can recover the message

$$m = x_2 - g_2p_Ak_2^* = m + g_2g_1hk_1k_2^* - g_2g_2hk_1k_2^*.$$

Now, assume that Eve can solve the cryptosystem. Then she can obtain any message m if she knows $h, g_1hk_1, g_2hk_2, m + g_2g_1hk_1k_2^*$. Eve encrypts a message m using g_2hk_2, obtaining

$$(x_1, x_2) = (g_2hk_2, m + g_2g_1hk_1k_2^*).$$

Since she can break the cryptosystem, she recovers m,

$$m = x_2 - g_2g_1hk_1k_2^*,$$

and obtains the shared key by computing

$$K = g_2g_1hk_1k_2^* = m - x_2.$$

□

5. Group Key Management

In this section, we present a key exchange protocol for n users. As stated before, we observe very clearly the lack of symmetry concerning the action of every user. We also discuss the rekeying process.

Let $h \in R = R_1 \oplus R_2$, described above. For $i = 1, ..., n$, user U_i has a secret pair $s_i = (g_i, k_i)$, where $g_i \in R_1$ and $k_i \in A_2 \leq R_2$. Let $\phi(s_i, h) = g_ihk_i$, 2-sided multiplication. We will denote $s_i^* = (g_i, k_i^*)$.

1. For $i = 1, ..., n$, user U_i sends to user U_{i+1} the message

$$\{C_i^1, C_i^2, ..., C_i^{i+1}\},$$

where $C_1^1 = h$, $C_1^2 = g_1hk_1$ and

- for $i > 1$ even, $C_i^j = \phi(s_i, C_{i-1}^j)$, when $j < i$, $C_i^i = C_{i-1}^i$, $C_i^{i+1} = \phi(s_i^*, C_{i-1}^i)$,
- for $i > 1$ odd, $C_i^j = \phi(s_i^*, C_{i-1}^j)$, when $j < i$, $C_i^i = C_{i-1}^i$, $C_i^{i+1} = \phi(s_i, C_{i-1}^i)$.

2. User U_n computes $\phi(s_n, C_{n-1}^n)$ if n is odd and $\phi(s_n^*, C_{n-1}^n)$ if n is even.
3. User U_n broadcasts

$$\{C_n^1, C_n^2, ..., C_n^n\}.$$

4. User U_i computes $\phi(s_i, C_n^i)$ if n is odd or $\phi(s_i^*, C_n^i)$ if n is even, and gets the shared key.

Proposition 3. *After this protocol, users $U_1, ..., U_n$ agree on a common key.*

Proof. Users $U_1, ...U_n$ agree on a common key. Now we prove it for an even or odd n number of users.

Firstly, we consider n odd. Let us show that users $U_1, ...U_{n-1}$ get the same key and that this is equal to U_n key. To do so, we will prove by induction that

$$\phi(s_i, C_n^i) = \phi(s_j, C_n^j)$$

for $i \neq j$, $i, j \in \{1, ..., n-1\}$. And this also equals U_n key, $\phi(s_n, C_{n-1}^n)$. For $s = 3$,

$$\begin{aligned} \phi(s_1, C_3^1) &= \phi(s_1, g_3 g_2 h k_2 k_3^*) \\ &= g_1 g_3 g_2 h k_2 k_3^* k_1 \\ &= g_2 g_3 g_1 h k_1 k_3^* k_2 \\ &= \phi(s_2, g_3 g_1 h k_1 k_3^*) \\ &= \phi(s_2, C_3^2) \end{aligned}$$

using the commutativity rules given by Proposition 1,

$$k_2 k_3^* k_1 = k_2 k_1^* k_3 = k_1 k_2^* k_3 = k_1 k_3^* k_2.$$

Moreover, $\phi(s_3, C_2^3) = g_3 g_2 g_1 h k_1 k_2^* k_3 = g_2 g_3 g_1 h k_1 k_3^* k_2 = \phi(s_2, C_3^2)$.

Now, suppose that

$$\phi(s_i, C_n^i) = \phi(s_j, C_n^j).$$

Then we have

$$\begin{aligned} \phi(s_i^*, C_{n+1}^i) &= \phi(s_i^*, \phi(s_{n+1}, C_n^i)) \\ &= \phi(s_i^* s_{n+1}, C_n^i) \\ &= \phi(s_{n+1}^* s_i, C_n^i) \\ &= \phi(s_{n+1}^*, \phi(s_i, C_n^i)) \\ &= \phi(s_{n+1}^*, \phi(s_j, C_n^j)) \\ &= \phi(s_{n+1}^* s_j, C_n^j) \\ &= \phi(s_j^* s_{n+1}, C_n^j) \\ &= \phi(s_j^*, \phi(s_{n+1}, C_n^j)) \\ &= \phi(s_j^*, C_{n+1}) \end{aligned}$$

and

$$\phi(s_n, C_{n-1}^n) = \phi(s_n, \phi(s_{n-1}^*, C_{n-1}^{n-2}) = \phi(s_n s_{n-1}^*, C_{n-1}^{n-2}) = \phi(s_{n-1} s_n^*, C_{n-1}^{n-1})$$
$$= \phi(s_{n-1}, \phi(s_n^*, C_{n-1}^{n-1}) = \phi(s_{n-1}, C_n^{n-1}).$$

So all users $U_1, ..., U_n$ get the same key for n odd.

Secondly, we show that this also works for n even. We prove by induction that

$$\phi(s_i^*, C_n^i) = \phi(s_j^*, C_n^j)$$

for $i \neq j$, $i, j \in \{1, ..., n-1\}$. And this also equals U_n key, $\phi(s_n, C_{n-1}^n)$. For $s = 4$,

$$\begin{aligned}
\phi(s_1^*, \phi(s_4, C_3^1)) &= \phi(s_1^*, g_4 g_3 g_2 h k_2 k_3^* k_4) \\
&= g_1 g_4 g_3 g_2 h k_2 k_3^* k_4 k_1^* \\
&= g_2 g_4 g_3 g_1 h k_1 k_3^* k_4 k_2^* \\
&= \phi(s_2^*, g_4 g_3 g_1 h k_1 k_3^* k_4) \\
&= \phi(s_2^*, \phi(s_4, C_3^2)),
\end{aligned}$$

$$\begin{aligned}
\phi(s_1^*, \phi(s_4, C_3^1)) &= \phi(s_1^*, g_4 g_3 g_2 h k_2 k_3^* k_4) \\
&= g_1 g_4 g_3 g_2 h k_2 k_3^* k_4 k_1^* \\
&= g_3 g_4 g_2 g_1 h k_1 k_2^* k_4 k_3^* \\
&= \phi(s_3^*, g_4 g_2 g_1 h k_1 k_2^* k_4) \\
&= \phi(s_3^*, \phi(s_4, C_3^3))
\end{aligned}$$

using that $g_i \in R_1$ commute and

$$k_2 k_3^* k_4 k_1^* = k_3 k_2^* k_4 k_1^* = k_3 k_4^* k_2 k_1^* = k_3 k_4^* k_1 k_2^* = k_3 k_1^* k_4 k_2^* = k_1 k_3^* k_4 k_2^*,$$

$$k_2 k_3^* k_4 k_1^* = k_2 k_4^* k_1 k_3^* = k_2 k_1^* k_4 k_3^* = k_1 k_2^* k_4 k_3^*.$$

In addition, $\phi(s_4^*, C_3^4) = g_4 g_3 g_2 g_1 h k_1 k_2^* k_3 k_4^* = g_3 g_4 k_2 k_1 h k_1 k_2^* k_4 k_3^* = \phi(s_3^*, \phi(s_4, C_3^3))$.

Suppose now that

$$\phi(s_i^*, C_n^i) = \phi(s_j^*, C_n^j).$$

Then we have

$$\begin{aligned}
\phi(s_i, C_{n+1}^i) &= \phi\big(s_i, \phi(s_{n+1}^*, C_n^i)\big) \\
&= \phi\big(s_i, \phi(s_{n+1}^*, C_n^i)\big) \\
&= \phi\big(s_i s_{n+1}^*, C_n^i)\big) \\
&= \phi\big(s_{n+1} s_i^*, C_n^i)\big) \\
&= \phi\big(s_{n+1}, \phi(s_i^*, C_n^i)\big) \\
&= \phi\big(s_{n+1}, \phi(s_j^*, C_n^j)\big) \\
&= \phi\big(s_{n+1} s_j^*, C_n^j)\big) \\
&= \phi\big(s_j s_{n+1}^*, C_n^j\big) \\
&= \phi\big(s_j, \phi(s_{n+1}^*, C_n^j)\big) \\
&= \phi(s_j, C_{n+1}^j).
\end{aligned}$$

So the shared key $\phi\big(s_i, \phi(s_{n+1}^*, C_n^i)\big)$ is the same for every $i \in \{1, ..., n-1\}$, and also,

$$\begin{aligned}
\phi(s_n^*, C_{n-1}^n) = \phi(s_n^*, \phi(s_{n-1}, C_{n-1}^{n-2}) = \phi(s_n^* s_{n-1}, C_{n-1}^{n-2}) = \phi(s_{n-1}^* s_n, C_{n-1}^{n-1}) \\
= \phi(s_{n-1}^*, \phi(s_n, C_{n-1}^{n-1}) = \phi(s_{n-1}^*, C_n^{n-1}),
\end{aligned}$$

and all users $U_1, ..., U_n$ have the same shared key, and we are done. □

An important issue in group key management is rekeying after the initial key agreement. There exist three situations: the first is due to key caducity, and the members of the group are the same; the second is when a user leaves the group, and the third is when a new user joins it. We describe these procedures in the following lines.

Let us consider the first situation. Every user U_i has the information C_n^i received from the user U_n. The rekeying process can be carried out by any of them, as is suggested in [6]. We call this user U_c. He chooses a new element $s_{c'} = (g_{c'}, k_{c'})$, where $g_{c'} \in R_1$ and $k_{c'} \in A_2$. If n is odd, he changes his private key to $s_{c'}^* s_c$ and broadcasts the message

$$\{\phi(s_{c'}^*, C_n^1), \phi(s_{c'}^*, C_n^2), ..., \phi(s_{c'}^*, C_n^{c-1}), C_n^c, \phi(s_{c'}^*, C_n^{c+1}), ..., \phi(s_{c'}^*, C_n^n)\}.$$

If n is even, he changes his private key to $s_{c'}s_c^*$ and broadcasts the message

$$\{\phi(s_{c'}, C_n^1), \phi(s_{c'}, C_n^2), ..., \phi(s_{c'}, C_n^{c-1}), C_n^c, \phi(s_{c'}, C_n^{c+1}), ..., \phi(s_{c'}, C_n^n)\}.$$

Then every user recovers the common key using the private key s_i if n is even, and s_i^* if n is odd.

We can easily check that this shared key is the same for every user. Recall that we proved that $\phi(s_i, C_n^i) = \phi(s_j, C_n^j)$ for $i, j = 1, ..., n-1$ and odd n. Now we have

$$\begin{aligned}
\phi(s_i^*, \phi(s_{c'}, C_n^i) &= \phi(s_i^* s_{c'}, C_n^i)) \\
&= \phi(s_{c'}^* s_i, C_n^i) \\
&= \phi(s_{c'}^*, \phi(s_i, C_n^i) \\
&= \phi(s_{c'}^*, \phi(s_j, C_n^j) \\
&= \phi(s_{c'}^* s_j, C_n^j) \\
&= \phi(s_j^* s_{c'}, C_n^j)) \\
&= \phi(s_j^*, \phi(s_{c'}, C_n^j),
\end{aligned}$$

and $\phi(s_n, C_{n-1}^n) = \phi(s_{n-1}, C_n^{n-1})$ implies that

$$\begin{aligned}
\phi(s_n^*, \phi(s_{c'}, C_n^n) &= \phi(s_n^* s_{c'}, C_n^n)) \\
&= \phi(s_{c'}^* s_n, C_n^n) \\
&= \phi(s_{c'}^*, \phi(s_n, C_n^n)) \\
&= \phi(s_{c'}^*, \phi(s_n, C_{n-1}^n)) \\
&= \phi(s_{c'}^*, \phi(s_{n-1}, C_n^{n-1})) \\
&= \phi(s_{c'}^* s_{n-1}, C_n^{n-1}) \\
&= \phi(s_{n-1}^* s_{c'}, C_n^{n-1})) \\
&= \phi(s_{n-1}^*, \phi(s_{c'}, C_n^{n-1})),
\end{aligned}$$

so all users get the same shared key. This can be proved analogously for odd n. Note that every time we rekey, we need to consider that a new user has been added to the key agreement (just to decide if we use the procedure for odd or even n), so the second time we rekey we will consider that they are $n+1$ users, and so on.

In the second case, when some user leaves the group, the corresponding position in the rekeying message is omitted.

In the last case, when a new user U_{n+1} joins the group, if n is odd, then U_c adds the element $\phi(s_{c'}, C_n^n)$ and sends the following to the new user:

$$\{\phi(s_{c'}, C_n^1), \phi(s_{c'}, C_n^2), ..., \phi(s_{c'}, C_n^{c-1}), C_n^c, \phi(s_{c'}, C_n^{c+1}), ..., \phi(s_{c'}, C_n^{n-1}), \phi(s_{c'}, C_n^n)\}.$$

If n is even, U_c adds the element $\phi(s_{c'}^*, C_n^n)$ and sends to U_{n+1} the following:

$$\{\phi(s_{c'}^*, C_n^1), \phi(s_{c'}^*, C_n^2), ..., \phi(s_{c'}^*, C_n^{c-1}), C_n^c, \phi(s_{c'}^*, C_n^{c+1}), ..., \phi(s_{c'}^*, C_n^{n-1}), \phi(s_{c'}^*, C_n^n)\}.$$

Finally, user U_{n+1} proceeds to step 3 of the group key protocol and sends the other users the information to obtain the shared key using their private keys.

Our next objective is showing that the security of this protocol for n users is equivalent to the security of the key exchange in the case of two users, as is the case of [6] and any other similar proposal such as [7] or more recently, [8], among many others.

6. Conclusions

Our contribution is proposing twisted group rings as interesting structures for key management, combined with the decomposition problem, since they seem to be quantum-safe for the time being. More specifically, we have introduced a key exchange protocol using the group ring $K^\alpha[D_{2m}]$ and have

shown a security and complexity analysis. We have also proposed an Elgamal cryptosystem and discussed its security. Finally, we have extended this protocol for several users.

Author Contributions: Investigation, M.D.G.O., J.A.L.R., and B.T.J.; Writing—original draft, M.D.G.O.; Writing—review & editing, J.A.L.R. and B.T.J.

Funding: This work has been supported by Ministerio de Economía y Competitividad grant MTM2017-86987-P. Both the second and third authors are also funded by Junta de Andalucía grant FQM221.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Proof of Proposition 1. Let us show that each of these equalities holds.

1. Given $h_1, h_2 \in R_1$, we have

$$r_i x^i s_j x^j = r_i s_j \alpha(x^i, x^j) x^i x^j = r_i s_j x^{i+j},$$

$$s_j x^j r_i x^i = s_j r_i \alpha(x^j, x^i) x^j x^i = r_i s_j x^{i+j}.$$

So then

$$h_1 h_2 = \sum_{i=0}^{m-1} (r_i x^i) \sum_{j=0}^{m-1} (s_j x^j) = \sum_{i,j=0}^{m-1} (r_i s_j x^{i+j}) = \sum_{j=0}^{m-1} (r_j x^j) \sum_{i=0}^{m-1} (r_i x^i) = h_2 h_1.$$

2. Given $h_1, h_2 \in A_2$, these elements can be written as

$$r_0 y + r_1 xy + r_2 x^2 y + ... + r_{\frac{m-1}{2}} x^{\frac{m-1}{2}} y + r_{\frac{m-1}{2}} x^{\frac{m+1}{2}} y + ... + r_2 x^{n-2} y + r_1 x^{n-1} y$$

$$= r_0 y + \sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y)$$

if m is odd, and

$$r_0 y + r_1 xy + r_2 x^2 y + ... + r_{\frac{m}{2}-1} x^{\frac{m}{2}-1} y + r_{\frac{m}{2}} x^{\frac{m}{2}} y + r_{\frac{m}{2}-1} x^{\frac{m}{2}+1} y + ... + r_2 x^{n-2} y + r_1 x^{n-1} y$$

$$= r_0 y + r_{\frac{m}{2}} x^{\frac{m}{2}} y + \sum_{i=1}^{\frac{m-2}{2}} (r_i x^i y + r_i x^{m-i} y)$$

if m is even.
It is worth showing now that the following equality holds:

$$\sum_{i=1}^{n} (r_i x^i y + r_i x^{m-i}) \sum_{j=1}^{n} (t^{-j} s_j x^j y + t^j s_j x^{m-j}) = \sum_{j=1}^{n} (s_j x^j y + s_j x^{m-j}) \sum_{i=1}^{n} (t^{-i} r_i x^i y + t^i r_i x^{m-i}). \quad \text{(A1)}$$

This is because we will need to use it in both bases, for even and odd m. Since we have that with basic elements we get

$$(r_i x^i y + r_i x^{-i} y) \cdot (t^{-j} s_j x^j y + t^j s_j x^{-j} y) = (s_j x^j y + s_j x^{-j} y) \cdot (t^{-i} r_i x^i y + t^i r_i x^{-i} y),$$

$$
\begin{aligned}
&(r_i x^i y + r_i x^{-i} y) \cdot (t^{-j} s_j x^j y + t^j s_j x^{-j} y) \\
=\;& r_i x^i y \cdot t^{-j} s_j x^j y + r_i x^i y \cdot t^j s_j x^{-j} y + r_i x^{-i} y \cdot t^{-j} s_j x^j y + r_i x^{-i} y \cdot t^j s_j x^{-j} y \\
=\;& r_i s_j t^{-j} \alpha(x^i y, x^j y) x^i y x^j y + r_i s_j t^j \alpha(x^i y, x^{-j} y) x^i y x^{-j} y \\
& + r_i s_j t^{-j} \alpha(x^{-i} y, x^j y) x^{-i} y x^j y + r_i s_j t^j \alpha(x^{-i} y, x^{-j} y) x^{-i} y x^{-j} y \\
=\;& r_i s_j t^{-j} t^j x^{i-j} + r_i s_j t^j t^{-j} x^{i+j} \\
& + r_i s_j t^{-j} t^j x^{-i-j} + r_i s_j t^j t^{-j} x^{-i+j} \\
=\;& r_i s_j (x^{i-j} + x^{i+j} + x^{-i-j} + x^{-i+j}),
\end{aligned}
$$

$$
\begin{aligned}
&(s_j x^j y + s_j x^{-j} y) \cdot (t^{-i} r_i x^i y + t^i r_i x^{-i} y) \\
=\;& s_j x^j y \cdot t^{-i} r_i x^i y + s_j x^j y \cdot t^i r_i x^{-i} y + s_j x^{-j} y \cdot t^{-i} r_i x^i y + s_j x^{-j} y \cdot t^i r_i x^{-i} y \\
=\;& s_j r_i t^{-i} \alpha(x^j y, x^i y) x^j y x^i y + s_j r_i t^i \alpha(x^j y, x^{-i} y) x^j y x^{-i} y \\
& + s_j r_i t^{-i} \alpha(x^{-j} y, x^i y) x^{-j} y x^i y + s_j r_i t^i \alpha(x^{-j} y, x^{-i} y) x^{-j} y x^{-i} y \\
=\;& s_j r_i x^{j+i} + s_j r_i x^{j-i} + s_j r_i x^{-j+i} + s_j r_i x^{-j-i} \\
=\;& s_j r_i (x^{j+i} + x^{j-i} + x^{-j+i} + x^{-j-i}) \\
=\;& r_i s_j (x^{i-j} + x^{i+j} + x^{-i-j} + x^{-i+j}).
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&\textstyle\sum_{i=1}^n (r_i x^i y + r_i x^{m-i} y) \sum_{j=1}^n (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) \\
=\;& \textstyle\sum_{i=1}^n (r_i x^i y \cdot t^{-j} s_j x^j y + r_i x^i y \cdot t^j s_j x^{m-j} y + r_i x^{m-i} y \cdot t^{-j} s_j x^j y + r_i x^{m-i} y \cdot t^j s_j x^{m-j} y) \\
=\;& \textstyle\sum_{i=1}^n (r_i s_j t^{-j} \alpha(x^i y, x^j y) + r_i s_j t^i \alpha(x^i y, x^{m-j} y) x^i y x^{m-j} y \\
& + r_i s_j t^{-j} \alpha(x^{m-i} y, x^j y) x^{m-i} y x^j y + r_i s_j t^j \alpha(x^{m-i} y, x^{m-j} y) x^{m-i} y x^{m-j} y) \\
=\;& \textstyle\sum_{i=1}^n (r_i s_j t^{-j} t^j x^i y x^j y + r_i s_j t^j t^{-j} x^i y x^{m-j} y + r_i s_j t^{-j} t^j x^{m-i} y x^j y + r_i s_j t^j t^{-j} x^{m-i} y x^{m-j} y) \\
=\;& \textstyle\sum_{i=1}^n (r_i s_j x^{i-j} + r_i s_j x^{i+j} + r_i s_j x^{-i-j} + r_i s_j x^{-i+j}) \\
=\;& \textstyle\sum_{i=1}^n r_i s_j (x^{i-j} + x^{i+j} + x^{-i-j} + x^{-i+j}) \\
=\;& \textstyle\sum_{j=1}^n (s_j x^j y + s_j x^{m-j} y) \sum_{i=1}^n (t^{-i} r_i x^i y + t^i r_i x^{m-i} y).
\end{aligned}
$$

2.1 Now we show that $h_1 h_2^* = h_2 h_1^*$.

- If m is odd:

$$
\begin{aligned}
h_1 h_2^* =\;& \Big(r_0 y + \textstyle\sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y)\Big) \Big(s_0 y + \sum_{j=1}^{\frac{m-1}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y)\Big) \\
=\;& r_0 y \cdot s_0 y + r_0 y \textstyle\sum_{j=1}^{\frac{m-1}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) + \sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y) s_0 y \\
& + \textstyle\sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y) \sum_{j=1}^{\frac{m-1}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) \\
=\;& r_0 s_0 + \textstyle\sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y) s_0 y + r_0 y \sum_{j=1}^{\frac{m-1}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) \\
& + \textstyle\sum_{i=1}^{\frac{m-1}{2}} (r_i x^i y + r_i x^{m-i} y) \sum_{j=1}^{\frac{m-1}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) \\
=\;& s_0 y \cdot r_0 y + s_0 y \textstyle\sum_{i=1}^{\frac{m-1}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y) + \sum_{j=1}^{\frac{m-1}{2}} (s_j x^j y + s_j x^{m-j} y) r_0 y \\
& + \textstyle\sum_{j=1}^{\frac{m-1}{2}} (s_j x^j y + s_j x^{m-j} y) \sum_{i=1}^{\frac{m-1}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y) \\
=\;& \Big(s_0 y + \textstyle\sum_{j=1}^{\frac{m-1}{2}} (s_j x^j y + s_j x^{m-j} y)\Big) \Big(r_0 y + \sum_{i=1}^{\frac{m-1}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y)\Big) \\
=\;& h_2 h_1^*,
\end{aligned}
$$

where we have used that

- $\sum_{i=1}^n (r_i x^i y + r_i x^{m-i}) \sum_{j=1}^n (t^{-j} s_j x^j y + t^j s_j x^{m-j}) = \sum_{j=1}^n (s_j x^j y + s_j x^{m-j}) \sum_{i=1}^n (t^{-i} r_i x^i y + t^i r_i x^{m-i})$ A1,
- $r_0 y \cdot \sum_{j=1}^{\frac{m-2}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) = \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y + s_j x^{m-j} y) \cdot r_0 y$ A2,
- $\sum_{i=1}^{\frac{m-2}{2}} (r_i x^i y + r_i x^{m-i} y) s_0 y = s_0 y \sum_{j=1}^{\frac{m-2}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y)$ A2,

$$
\begin{aligned}
r_0y\sum_{j=1}^{\frac{m-1}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}) &= \sum_{j=1}^{\frac{m-1}{2}}(r_0y\cdot t^{-j}s_jx^jy+r_0y\cdot t^js_jx^{m-j}y)\\
&= \sum_{j=1}^{\frac{m-1}{2}}(r_0s_jt^{-j}\alpha(y,x^jy)yx^jy+r_0s_jt^j\alpha(y,x^{m-j})yx^{m-j}y)\\
&= \sum_{j=1}^{\frac{m-1}{2}}(r_0s_jx^{m-j}+r_0s_jx^j) \qquad \text{(A2)}\\
&= \sum_{j=1}^{\frac{m-1}{2}}(s_jr_0\alpha(x^jy,y)x^jyy+s_jr_0\alpha(x^{m-j}y,y)x^{m-j}yy)\\
&= \sum_{j=1}^{\frac{m-1}{2}}(s_jx^jy+s_jx^{m-j})r_0y.
\end{aligned}
$$

- Analogously, if m is even, we have

$$
\begin{aligned}
h_1h_2^* = &\left(r_0y+r_{\frac{m}{2}}x^{\frac{m}{2}}y+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i})\right)\\
&\left(s_0y+t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y+\textstyle\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j})\right)\\
= &\ r_0y\cdot s_0y+r_0y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y+r_0y\textstyle\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)+r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot s_0y\\
&+r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y+r_{\frac{m}{2}}x^{\frac{m}{2}}y\textstyle\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)\\
&+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)\cdot s_0y+\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)t^{-\frac{m}{2}}\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y\\
&+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)\\
= &\ r_0s_0+r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot s_0y+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)\cdot s_0y+r_0y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y\\
&+r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)t^{-\frac{m}{2}}\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y\\
&r_0y\textstyle\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)+r_{\frac{m}{2}}x^{\frac{m}{2}}y\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)\\
&+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(r_ix^iy+r_ix^{m-i}y)\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)\\
= &\ s_0y\cdot r_0y+s_0y\cdot t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}y+s_0y\textstyle\sum_{j=1}^{\frac{m-2}{2}}(t^{-i}r_ix^iy+t^ir_ix^{m-i}y)+s_{\frac{m}{2}}x^{\frac{m}{2}}yr_0y\\
&+s_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}y+s_{\frac{m}{2}}x^{\frac{m}{2}}y\textstyle\sum_{i=1}^{\frac{m-2}{2}}(t^{-i}r_ix^iy+t^ir_ix^{m-i}y)\\
&+\textstyle\sum_{j=1}^{\frac{m-2}{2}}(s_jx^jy+r_jx^{m-j}y)r_0y+\sum_{j=1}^{\frac{m-2}{2}}(s_jx^jy+r_jx^{m-j}y)t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}y\\
&+\textstyle\sum_{j=1}^{\frac{m-2}{2}}(s_jx^jy+r_jx^{m-j}y)\sum_{i=1}^{\frac{m-2}{2}}t^{-i}(r_ix^iy+t^ir_ix^{m-i}y)\\
= &\ \left(s_0y+s_{\frac{m}{2}}x^{\frac{m}{2}}+\textstyle\sum_{j=1}^{\frac{m-2}{2}}(s_jx^jy+s_jx^{m-j})\right)\\
&\left(r_0y+t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}+\textstyle\sum_{i=1}^{\frac{m-2}{2}}(t^{-i}r_ix^iy+t^ir_ix^{m-i})\right)\\
= &\ h_2h_1^*,
\end{aligned}
$$

where we have used that

- $r_0y\cdot s_0y=r_0s_0=s_0r_0=s_0y\cdot r_0y$
- $r_0y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y=r_0s_{\frac{m}{2}}t^{-\frac{m}{2}}\alpha(y,x^{\frac{m}{2}}y)yx^{\frac{m}{2}}y=r_0s_{\frac{m}{2}}x^{\frac{m}{2}}=s_{\frac{m}{2}}r_0\alpha(x^{\frac{m}{2}}y,y)x^{\frac{m}{2}}yy=s_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot r_0y$,
- $r_0y\cdot\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)=\sum_{j=1}^{\frac{m-2}{2}}(s_jx^jy+r_jx^{m-j}y)\cdot r_0y$ A2,
- $r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot s_0y=r_{\frac{m}{2}}s_0\alpha(x^{\frac{m}{2}}y,y)x^{\frac{m}{2}}yy=r_{\frac{m}{2}}s_0x^{\frac{m}{2}}=s_0r_{\frac{m}{2}}t^{\frac{m}{2}}\alpha(y,x^{\frac{m}{2}}y)yx^{\frac{m}{2}}y=s_0y\cdot t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}y$,
- $r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot t^{-\frac{m}{2}}s_{\frac{m}{2}}x^{\frac{m}{2}}y=s_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot t^{-\frac{m}{2}}r_{\frac{m}{2}}x^{\frac{m}{2}}y$ A3,
- $r_{\frac{m}{2}}x^{\frac{m}{2}}y\sum_{j=1}^{\frac{m-2}{2}}(t^{-j}s_jx^jy+t^js_jx^{m-j}y)=\sum_{j=1}^{\frac{m-2}{2}}(r_{\frac{m}{2}}x^{\frac{m}{2}}y\cdot s_jx^jy+\cdot r_{\frac{m}{2}}x^{\frac{m}{2}}ys_jx^{m-j}y)=\sum_{j=1}^{\frac{m-2}{2}}(r_{\frac{m}{2}}s_jx^{\frac{m}{2}+j}+r_{\frac{m}{2}}s_jx^{\frac{m}{2}-j})$

$= \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y \cdot t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y + r_j x^{m-j} y \cdot t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y) = \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y + r_j x^{m-j} y) t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y,$

- $\sum_{i=1}^{\frac{m-2}{2}} (r_i x^i y + r_i x^{m-i} y) s_0 y = s_0 y \sum_{j=1}^{\frac{m-2}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y)$ A2,
- $\sum_{i=1}^{\frac{m-2}{2}} (r_i x^i y + r_i x^{m-i} y) t^{-\frac{m}{2}} s_{\frac{m}{2}} x^{\frac{m}{2}} y = s_{\frac{m}{2}} x^{\frac{m}{2}} y \sum_{i=1}^{\frac{m-2}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y)$ A3,
- $\sum_{i=1}^{\frac{m-2}{2}} (r_i x^i y + r_i x^{m-i} y) \sum_{j=1}^{\frac{m-2}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) = \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y + r_j x^{m-j} y) \sum_{i=1}^{\frac{m-2}{2}} (t^{-i} r_i x^i y + t^i r_i x^{m-i} y)$ A1,

$$
\begin{aligned}
r_{\frac{m}{2}} x^{\frac{m}{2}} y \sum_{j=1}^{\frac{m-2}{2}} (t^{-j} s_j x^j y + t^j s_j x^{m-j} y) &= \sum_{j=1}^{\frac{m-2}{2}} (r_{\frac{m}{2}} x^{\frac{m}{2}} y \cdot t^{-j} s_j x^j y + \cdot r_{\frac{m}{2}} x^{\frac{m}{2}} y s_j x^{m-j} y) \\
&= \sum_{j=1}^{\frac{m-2}{2}} (r_{\frac{m}{2}} s_j t^{-j} \alpha(x^{\frac{m}{2}}, x^j y) x^{\frac{m}{2}} x^j y + \\
&\quad r_{\frac{m}{2}} s_j t^j \alpha(x^{\frac{m}{2}}, x^{m-j} y) x^{\frac{m}{2}} x^{m-j} y) \\
&= \sum_{j=1}^{\frac{m-2}{2}} (r_{\frac{m}{2}} s_j x^{\frac{m}{2}+j} + r_{\frac{m}{2}} s_j x^{\frac{m}{2}-j}) \\
&= \sum_{j=1}^{\frac{m-2}{2}} (r_{\frac{m}{2}} s_j x^{\frac{m}{2}-j} + r_{\frac{m}{2}} s_j x^{\frac{m}{2}+j}) \qquad \text{(A3)} \\
&= \sum_{j=1}^{\frac{m-2}{2}} (s_j r_{\frac{m}{2}} t^{-\frac{m}{2}} \alpha(x^j y, x^{\frac{m}{2}}) x^j y x^{\frac{m}{2}} + \\
&\quad s_j r_{\frac{m}{2}} t^{-\frac{m}{2}} \alpha(x^{m-j} y, x^{\frac{m}{2}}) x^{m-j} y x^{\frac{m}{2}}) \\
&= \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y \cdot t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y + r_j x^{m-j} y \cdot t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y) \\
&= \sum_{j=1}^{\frac{m-2}{2}} (s_j x^j y + r_j x^{m-j} y) t^{-\frac{m}{2}} r_{\frac{m}{2}} x^{\frac{m}{2}} y.
\end{aligned}
$$

The proof of $h_1^* h_2 = h_2^* h_1$ and $h_1 h_2 = h_2 h_1^*$ can be made by using similar arguments.

□

Appendix B. Mathematica Implementation of 128-Bit Example

In this appendix, we include an example of our key exchange in $GF(2^4)^{\alpha} D_{32}$, where keys are 128 bits long, implemented in the software Mathematica.

```
In[52]:= h = RandomChoice[GR128];
         elección aleatoria
         g1 = RandomChoice[elementsR1];
         elección aleatoria
         g2 = RandomChoice[elementsR1];
         elección aleatoria
         k1 = RandomChoice[elementsA2];
         elección aleatoria
         k2 = RandomChoice[elementsA2];
         elección aleatoria

In[57]:= TwistedMult128[g1, g2]
         TwistedMult128[g2, g1]
         TwistedMult128[k1, StarApl[k2]]
         TwistedMult128[k2, StarApl[k1]]

Out[57]= {t + t^2 + t^3, 0, 1 + t + t^2, 1 + t + t^2 + t^3, t + t^3,
          t^2, 1, 1 + t + t^3, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, 0, 1, 0, t^3,
          t, 1 + t^2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[58]= {t + t^2 + t^3, 0, 1 + t + t^2, 1 + t + t^2 + t^3, t + t^3,
          t^2, 1, 1 + t + t^3, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, 0, 1, 0, t^3,
          t, 1 + t^2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[59]= {t^3, 1 + t + t^2 + t^3, t + t^2 + t^3, t + t^2 + t^3, t^3,
          1, 0, t + t^2, 0, t + t^2, 0, 1, t^3, t + t^2 + t^3, t + t^2 + t^3,
          1 + t + t^2 + t^3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Out[60]= {t^3, 1 + t + t^2 + t^3, t + t^2 + t^3, t + t^2 + t^3, t^3,
          1, 0, t + t^2, 0, t + t^2, 0, 1, t^3, t + t^2 + t^3, t + t^2 + t^3,
          1 + t + t^2 + t^3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

In[61]:= GetKey[a_, b_, c_] := TwistedMult128[a, TwistedMult128[b, c]]

In[62]:= publicA = GetKey[g1, h, k1]
         publicB = GetKey[g2, h, k2]

Out[62]= {t + t^3, 0, 1 + t + t^2, t, 1 + t + t^3, 1, t + t^2 + t^3, t^3, t^3, t, t + t^3, 1 + t^3, 0,
          1 + t + t^3, t^2 + t^3, 1 + t^2 + t^3, t^2, 1 + t^3, t, 1 + t, t + t^2 + t^3, 1 + t + t^3, t^3,
          t^2, 1 + t^3, 1 + t + t^3, t + t^2 + t^3, 1 + t + t^2 + t^3, 1 + t^3, t^2, 1 + t + t^3, t^3}

Out[63]= {t + t^3, 1, 1, t + t^2, t + t^2, 1 + t^3, 1 + t^3, 1 + t^2, t^2 + t^3, 1 + t + t^2,
          t^2, t + t^3, 1, 1 + t + t^2, 1 + t^2, 1 + t^3, 1 + t + t^2 + t^3, 1 + t + t^2,
          1 + t^2 + t^3, 1, t + t^2, t^3, t^2, 1 + t^2, 1 + t + t^3, 1 + t, 0, 1, t^3, 0, 0, t^2}

In[64]:= sharedA = GetKey[g1, publicB, StarApl[k1]]
         sharedB = GetKey[g2, publicA, StarApl[k2]]
         sharedA == sharedB

Out[64]= {t^2, 1, t^2 + t^3, t + t^2, 1 + t^3, 1 + t^2, 1 + t^2, 1 + t + t^2, 1 + t^2,
          1 + t, t^2, t + t^2, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, t + t^2 + t^3, t^2 + t^3,
          1 + t^2 + t^3, t + t^2, 0, 1 + t + t^2 + t^3, t, t + t^2, 1 + t^3, t^2 + t^3, 1 + t + t^3,
          t + t^3, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, 1 + t^2, t + t^2 + t^3, 1 + t + t^2, t^3}

Out[65]= {t^2, 1, t^2 + t^3, t + t^2, 1 + t^3, 1 + t^2, 1 + t^2, 1 + t + t^2, 1 + t^2,
          1 + t, t^2, t + t^2, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, t + t^2 + t^3, t^2 + t^3,
          1 + t^2 + t^3, t + t^2, 0, 1 + t + t^2 + t^3, t, t + t^2, 1 + t^3, t^2 + t^3, 1 + t + t^3,
          t + t^3, 1 + t + t^2 + t^3, 1 + t + t^2 + t^3, 1 + t^2, t + t^2 + t^3, 1 + t + t^2, t^3}

Out[66]= True
```

Figure A1. Implementation of the key exchange in Mathematica

References

1. Chen, L.; Jordan, S.; Liu,Y.-K.; Moody, D.; Peralta, R.; Perlner, R.; SmithTone, D. *Report on Post-Quantum Cryptography*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016; Volume 8150.
2. Maze, G.; Monico, C.; Rosenthal, J. Public key cryptography based on semigroup actions. *Adv. Math. Commun.* **2007**, *1*, 489–507.
3. Romankov, V. A general encryption scheme using two-sided multiplications with its cryptanalysis. *arXiv* **2017**, arXiv:1709.06282.
4. Eftekhari, M. Cryptoanalisis of Some Protocols using Matrices over Group Rings. In Proceedings of the International Conference on Cryptology in Africa, Dakar, Senegal, 24–26 May 2017; Volume 10239, pp. 223–229.
5. Kahrobaei, D.; Koupparis, C.; Shpilrain, V. Public key exchange using matrices over group rings. *Groups Complex. Cryptol.* **2013**, *5*, 97–115. [CrossRef]
6. Steiner, M.; Tsudik, G.; Waidner, M. Key agreement in dynamic peer groups. *IEEE Trans. Parallel Distrib. Syst.* **2000**, *11*, 769–780. [CrossRef]
7. Burmester, M.; Desmedt, I. A secure and scalable Group Key Exchange system. *Inf. Proc. Lett.* **2005**, *94*, 137–143. [CrossRef]
8. Lopez-Ramos, J.A.; Rosenthal, J.; Schipani, D.; Schnyder, R. An application of group theory in confidential network communications. *Math. Meth. Apply Sci.* **2018**, *41*, 2294–2298. [CrossRef]