BAREKENG: Journal of Mathematics and Its Applications
June 2025 Volume 19 Issue 2 Page 1307-1316
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss2pp1307-1316

ON THE SECURITY OF GENERALIZED MULTILINEAR MAPS BASED ON WEIL PAIRING

Annisa Dini Handayani[1*], Indah Emilia Wijayanti[2], Uha Isnaini[3], Prastudy Fauzi[4]

[1,2,3]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada
Bulaksumur, Caturtunggal, Yogyakarta, 55281, Indonesia

[1]Department of Cryptography, Politeknik Siber dan Sandi Negara
Jln. Raya H. Usa, Ciseeng, Bogor, Jawa Barat, 16120, Indonesia

[4]Nanyang Technological University
50 Nanyang Avenue, Singapore, 639798, Singapore

*Corresponding author's e-mail: * **annisa.dini@polteksn.ac.id***

ABSTRACT

Article History:

Received: 11th October 2024
Revised: 2nd February 2025
Accepted: 5th March 2025
Published: 1st April 2025

Keywords:

Cryptography;
Diffie-Hellman Assumption;
Multilinear Map;
Multiparty Key Exchange;
Weil Pairing.

In 2017, Tran et al. proposed a multilinear map based on Weil pairings to realize the Boneh-Silverberg scheme. They proposed an algorithm to evaluate the Boneh-Silverberg multilinear map and showed that it could be used to establish a shared key in multipartite key exchange for five users. They claimed their scheme was secure and computable in establishing a shared key between 5 users. Unfortunately, they did not prove that their scheme meets three additional computational assumptions proposed by Boneh and Silverberg. In this paper, with some computational modifications, we show that the algorithm proposed by Tran et al. does not satisfy three security assumptions proposed by Boneh and Silverberg. Therefore, every user involved in this multipartite key exchange can obtain the shared key and other users' secret values. We also show that the computation to obtain a shared key is inefficient because it requires a lot of computation and time.

How to cite this article:

A. D. Handayani, I. E. Wijayanti, U. Isnaini and P. Fauzi., "ON THE SECURITY OF GENERALIZED MULTILINEAR MAPS BASED ON WEIL PAIRING," *BAREKENG: J. Math. & App.*, vol. 19, iss. 2, pp. 1307-1316, June, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Bilinear maps have emerged as an exciting area in cryptography, enabling many new protocols that were previously impossible [1]. It has unique properties allow it to be implemented in many new cryptographic protocols. One interesting application of bilinear maps in cryptography is Boneh and Franklin's identity-based encryption [2]. Boneh and Franklin used bilinear maps as building blocks to build an encryption scheme using user identities. Bilinear maps can also be used to construct short signature schemes, which were first proposed by Boneh, Lynn, and Sacham [3], and to exchange a shared key between three parties, which was proposed by Joux [4].

There are many kinds of bilinear maps used in cryptography. Two practical bilinear maps in cryptography are the Weil pairing and the Tate pairing. They map a pair of points on an elliptic curve into the multiplicative group of finite fields. The Weil pairing has an alternating property, which will map any two dependent points on the elliptic curve into a fixed value 1 [5]. Due to this alternating property, the Weil pairing cannot be used directly in cryptography. Therefore, we should use an additional map that can remove this property but retain the other properties of the Weil pairing. The Weil pairing with such an additional map is called the modified Weil pairing.

In 2002, Boneh and Silverberg showed that a multilinear map from the generalization of the Weil or Tate pairing would broadly impact on cryptography [6]. Multilinear maps can build secure broadcast encryption, a unique signature scheme, and a one-round multiparty key exchange. Another application of multilinear maps is group key exchange, witness encryption [7], and indistinguishability obfuscation [8]. For some cryptographic applications, the multilinear map should satisfy three security assumptions: the multilinear Diffie-Hellman assumption, the Diffie-Hellman inversion assumption, and the generalized Diffie-Hellman assumption. Unfortunately, their paper stated that it is not easy to construct a multilinear map that meets these three security assumptions, and this is still an open problem.

Garg, Gentry, and Halevi proposed the first interesting candidate for a multilinear map [9]. They introduced the concept of a graded encoding scheme applied to ideal lattices. However, Hu and Jia attacked this scheme [10]. Another candidate multilinear map construction was proposed by Coron, Lepoint, and Tibouchi [11] using the Chinese Remainder Theorem over the integers. Unfortunately, this scheme had also been attacked by Cheon et al. [12]. Other candidate multilinear maps have been proposed based on Weil pairing over elliptic curve [13], cohomology group [14], and nilpotent group [15].

In this paper, we show the usage of a multilinear map in simplifying the multipartite key exchange compared to a multipartite key exchange using an ordinary Diffie-Hellman key exchange. We also explore the multilinear maps proposed by Tran et al. [13]. They constructed a multilinear map based on Weil pairing over an elliptic curve. They showed that multilinear maps, especially 4-linear maps, can be built using Weil pairing and tensor products. They also proposed an algorithm to compute the exponentiation i of ζ, the generator of a multiplicative group of finite fields as a codomain group of Weil pairing. Finally, we show that the algorithm proposed by Tran et al. allows users not only to compute a shared key but also secret values from other users.

2. RESEARCH METHODS

This research focuses on multilinear maps using Weil pairing over elliptic curves and their application to establish the shared key between five users. This study consists of three stages.

a. Stage 1: Exploration of the definition of bilinear and multilinear maps ($\boldsymbol{k} = \mathbf{2}$), as well as the concept of Diffie-Hellman key exchange for two users.

b. Stage 2: Analyze algorithm 1 proposed by Tran et al. and evaluate whether the algorithm can be used to obtain the secret value of each user.

c. Stage 3: Formulation of claims and substantiation of research results.

Below, we provide some mathematical background related to the bilinear map, multilinear map, and the Diffie-Hellman key exchange as the result of the first step of our research.

Definition 1. *Let G_1 and G_2 be two cyclic groups of order q for some large prime q. The group G_1 can be viewed as an additive group and G_2 as a multiplicative group. The map $e: G_1 \times G_1 \to G_2$ is said* ***bilinear*** *if satisfy $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z$.*

A bilinear map used in cryptography should also satisfy these additional properties [16]*:*

a. **Non-degenerate**: the map does not send all pairs in $G_1 \times G_1$ to the identity in G_2.

b. **Computable**: there is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in G_1$.

Definition 2. *An elliptic curve over F_q is the set of solutions to an equation of the form:*

$$E: Y^2 = X^3 + AX + B, \ \textit{with } A, B \in F_q \textit{ satisfying } 4A^3 + 27B^2 \neq 0$$

which is denoted by [5]*:*

$$E\left(F_q\right) = \left\{(x, y): x, y \in F_q \ \ satisfy \ \ y^2 + x^3 + Ax + B\right\} \cup \{\mathcal{O}\}$$

The set of points of an elliptic curve forms a group under addition operation. This group can be used as a building block of a key exchange scheme, ElGamal encryption [17], and many cryptographic schemes and algorithms.

Definition 3. *Let $P, Q \in E[m]$ i.e., P, Q are points of order m in the group E. Let f_P and f_Q be rational functions on E satisfying*:

$$div(f_P) = m[P] - m[\mathcal{O}] \quad dan \quad div\left(f_Q\right) = m[Q] - m[P]$$

where $div(f_P)$ *and* $div\left(f_Q\right)$ are divisors on rational function f_P and f_Q respectively. The Weil pairing of P and Q is the quantity

$$e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \bigg/ \frac{f_Q(P - S)}{f_Q(-S)}$$

where $S \in E$ is any point satisfying $S \neq \{\mathcal{O}, P, -Q, P - Q\}$ [5].

The values of the Weil pairing satisfy $e_m(P, Q)^m = 1$ for all $P, Q \in E[m]$. In other words, the output of Weil pairing is an m^{th} root of unity. The Weil pairing is also alternating, which means that

$$e_m(P, P) = 1 \text{ for all } P \in E[m]$$

With this alternating property, if we want to evaluate the pairing at points $P_1 = aP$ and $P_2 = bP$ using the Weil pairing, it will output 1 as a fixed value for any random values $a, b \in Z$.

$$e(P_1, P_2) = e(aP, bP) = e(P, P)^{ab} = 1^{ab} = 1$$

The Weil pairing with this alternating property cannot be used directly in cryptography, because it will map any two dependent inputs into a fixed value 1. To overcome this drawback, we can choose an elliptic curve that has a “special” map $\phi: E \to E$ with the property that P and $\phi(P)$ are independent in $E[m]$. Then we can evaluate a pair of P and $\phi(P)$ and the output will not map into a fixed value.

$$e_m\big(P_1, \phi(P_2)\big) = e_m\big(aP, \phi(bP)\big) = e_m\big(aP, b\phi(P)\big) = e_m\big(P, \phi(P)\big)^{ab}$$

Definition 4. *Let $\ell \geq 3$ be a prime, let E be an elliptic curve, let $P \in E[\ell]$ be a point of order ℓ, and let $\phi: E \to E$ be a map from E to itself. We say that ϕ is an ℓ-distorsion map for P if it has the following two properties:*

(i) $\phi(nP) = n\phi(P)$ *for all* $n \geq 1$.

(ii) *The number* $e_\ell\big(P, \phi(P)\big)$ *is a primitive* ℓ^{th} *root of unity. This means that* $e_\ell\big(P, \phi(P)\big)^r = 1$ *if and only if* r *is a multiple of* ℓ.

The Weil pairing with this distortion map is called the modified Weil pairing, denoted by $\hat{e}$ [5].

Multilinear maps are a generalized form of bilinear maps. The bilinear map takes two elements in G_1 as its input while the multilinear map takes k-elements as its input. If the number of inputs is $k = 3$, it is called trilinear map otherwise if $k > 3$, it is called an k-linear map.

Definition 5. *A map* $e: \left(G_1^k \to G_1\right)$ *is an* k*-linear map if it satisfies the following properties* [6]:

(i) G_1 *and* G_2 *are groups of the same prime order.*

(ii) *If* $a_1, \cdots, a_k \in Z$ *and* $P_1, \cdots, P_k \in G_1$, *then* $e\left(P_1^{a_1}, \cdots, P_k^{a_k}\right) = e(P_1, \cdots, P_k)^{a_1 \cdots a_k}$

(iii) *The map* e *is non-degenerate in the following sense: if* $P \in G_1$ *is a generator of* G_1 *then* $e(P, \cdots, P)$ *is a generator of* G_2.

Definition 6. *Cryptographic* k*-linear map* $e: G_1^k \to G_2$ *is a multilinear map such that* [6]:

(i) *The group action in* G_1 *and* G_2 *is efficiently computable.*

(ii) *The map* e *is efficiently computable.*

(iii) *There is no efficient algorithm to compute discrete log in* G_1.

For some cryptographic applications, Boneh and Silverberg stated that multilinear maps must satisfy three additional computational assumptions [6]. They are the multilinear Diffie-Hellman assumption, the Diffie-Hellman inversion assumption, and the generalized Diffie-Hellman assumption.

Definition 7. **The multilinear Diffie-Hellman assumption**. *This assumption says that given* $g, g^{a_1}, \cdots, g^{a_{k+1}}$ *in* G_1, *it is hard to compute* $e(g, \cdots, g)^{a_1 \cdots a_{k+1}}$ *in* G_2.

Definition 8. The Diffie-Hellman inversion assumption. *The assumption says that given* $g, g^b \in G_1$ *it is hard to compute* $e(g, \cdots, g)^{1/b} \in G_2$.

Definition 9. The generalized Diffie-Hellman assumption. *The assumption says that given* $g^{a_1}, \cdots, g^{a_k}$ *in* G_1 *and given all the subset products* $g^{\prod_{i \in S} a_i} \in G_1$ *for any strict subset* $S \subset \{1, \cdots, k\}$, *it is hard to compute* $g^{a_1 \cdots a_k} \in G_1$.

Diffie-Hellman key exchange [5] is a scheme that allows two parties to establish a shared key over insecure communication. Let Alice and Bob want to communicate securely using symmetric cryptography. They need to have a shared key as the input of their agreed symmetric algorithm. Diffie-Hellman key exchange allows Alice and Bob to establish a shared key securely, as depicted in Figure 1.

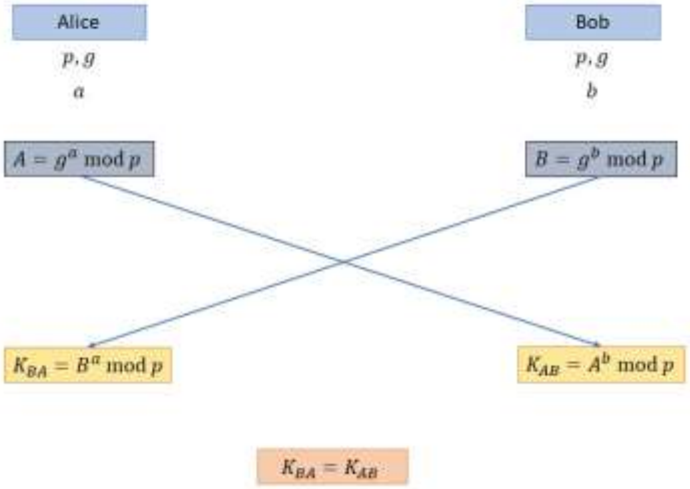

Figure 1. Diffie-Hellman Key Exchange

Alice and Bob agreed on some large prime $\boldsymbol{p}$ and $\boldsymbol{g}$, a generator of $\boldsymbol{Z_p}$. Alice and Bob choose a random secret value $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively, with $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{Z_p}$. Alice then computes $\boldsymbol{A = g^a} \textbf{ mod } \boldsymbol{p}$, and Bob computes $\boldsymbol{B = g^b} \textbf{ mod } \boldsymbol{p}$. Alice and Bob exchange $\boldsymbol{A}$ and $\boldsymbol{B}$ over public communication. To derive a shared key, Alice computes the received value from Bob with her secret value and Bob do the same thing. Alice computes the $\boldsymbol{K_{BA} = g^{ba}} \textbf{ mod } \boldsymbol{p}$ and Bob computes $\boldsymbol{K_{AB} = g^{ab}} \textbf{ mod } \boldsymbol{p}$. Now, Alice and Bob have the shared key.

$$\boldsymbol{K_{BA} = g^{ba}} \textbf{ mod } \boldsymbol{p = g^{ab}} \textbf{ mod } \boldsymbol{p = K_{AB}}$$

3. RESULTS AND DISCUSSION

This section describes the results of this research, including the description of multipartite key exchange using a multilinear map, particularly for a bilinear map. We also review Tran et al.'s scheme using 4-linear maps, including how to compute the shared key among the users. At the end of this section, we make some claims, proving that Tran et al.'s scheme is not secure under security assumptions and is inefficient in computation.

3.1 Multilinear Map in Multipartite Key Exchange

The k-linear map in cryptography can establish a shared key between $k + 1$ users more efficiently than using an ordinary Diffie-Hellman key exchange. We will show how the n-linear map can simplify a multipartite key exchange scheme compared to the ordinary Diffie-Hellman key exchange. First, we will show how the original Diffie-Hellman can establish a shared key between three users. It is followed by how the bilinear map (2-linear map) can establish a share key more efficiently than the previous one.

There are many proposed schemes to establish a shared key between k users, with $k > 2$, using generalized Diffie-Hellman key exchange. In this paper, we will describe a scheme proposed by Ingemarsson et al. [18]. Let Alice, Bob, and Carol wish to establish a shared key using Diffie-Hellman key exchange. They should choose the same large prime p and g, the generator of Z_p. To get a shared key, they should compute and exchange the public value in two rounds, as depicted in Figure 2.

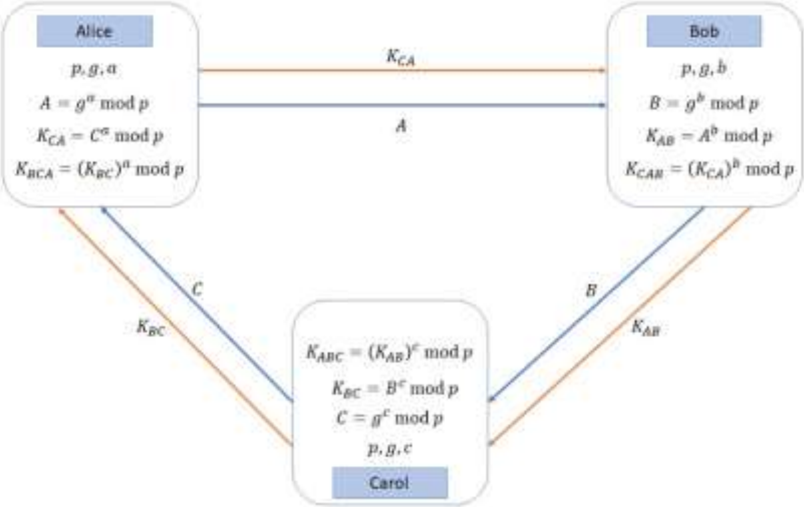

Figure 2. **Diffie-Hellman Key Exchange between 3 Users.**

The blue line in **Figure 2** describes the first round of computation. In this round, Alice, Bob, and Carol send $A, B,$ and C respectively to their neighbors. The second round is described by the orange line. In this round, each user computes their received value with their secret value and sends the result to the neighbors. After the second round, each user can compute a shared key. In the end of this scheme, Alice Bob, and Carol have a shared key.

$$K_{ABC} = K_{BCA} = K_{CAB}$$

If we want to establish a shared key between k users with this scheme, it will need $k - 1$ rounds to get a shared key. So, if there are 1 million users, it will need 999.999 rounds to get a shared key.

Next, we will show how to establish a shared key between three users using the bilinear map, as depicted in **Figure 3**.

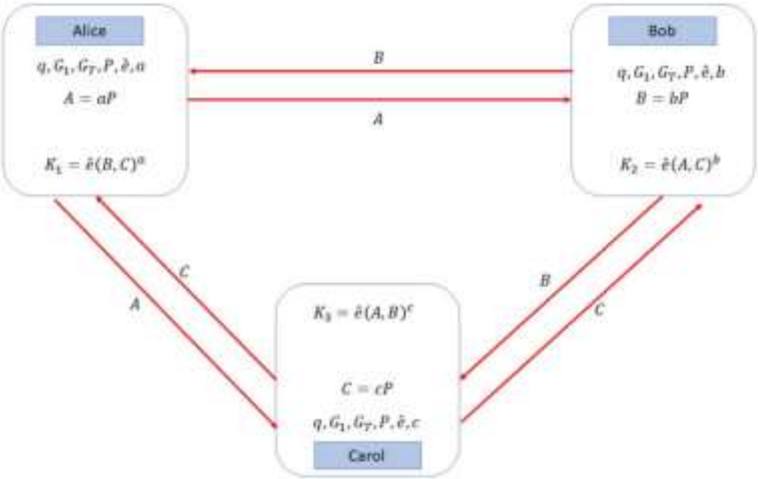

Figure 3. **Establish a Shared Key Between 3 Users Using Bilinear Map.**

When we want to use bilinear map to establish a shared key, we need to set the public parameter first. The public parameter consists of a large prime number q, two cyclic groups G_1 and G_T, an element $P \in G_1$ (a generator of G_1), and a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_T$. Then each user chooses a random secret number $a, b, c \in Z_q$ for Alice, Bob, and Carol respectively. With this scheme, each user does not need to carry out the computation in $k - 1$ rounds. They just broadcast their public value to other users. After received others public valued, each user can derive a shared key by computing $\hat{e}(B, C)^a$ for Alice, $\hat{e}(A, C)^b$ for Bob, and $\hat{e}(A, B)^c$ for Carol. The result computed by Alice, Bob, and Carol, will have the same value.

$$\hat{e}(A, B)^c = \hat{e}(B, C)^a = \hat{e}(A, C)^b = \hat{e}(P, P)^{abc}$$

3.2 Review of Tran et al.'s 4-linear Map Scheme

Now, we describe 4-linear map scheme proposed by Tran et al. [13] followed by its application on key exchange for 5-users.

Tran et al. used the fact that tensor product can be used to generalize a bilinear map to multilinear map, as stated by the following remark:

Remark 1. *If $V_1, \cdots, V_k$ is finite-dimensional F_m-vector spaces, there exists a natural one-to-correspondence between multilinear homomorphisms:*

$$h: V_1 \times \cdots \times V_k \to \mu_m$$

and linear homomorphisms:

$$\tilde{h}: V_1 \otimes \cdots \otimes V_k \to \mu_m$$

with $h(x_1, \cdots, x_k) = \tilde{h}(x_1 \otimes \cdots \otimes x_k)$.

In their paper, Tran et al. proposed a theorem that stated the existence of 4-linear map on an elliptic curve.

Theorem 1. *Let E be an elliptic curve defined over K. A map $e_{2,m}: E[m] \times E[m] \times E[m] \times E[m] \to \mu_m \otimes \mu_m$ given by $e_{2,m}(P_1, P_2, Q_1, Q_2) = e_m(P_1, Q_1) \otimes e_m(P_2, Q_2)$. Then,*

(i) *$e_{2,m}$ is a 4-linear mapping.*

(ii) *$e_{2,m}$ is non-degenerate.*

(iii) *$e_{2,m}(P, P, P, P) = 1$, for all $P \in E[m]$.*

Based on Remarks 1 and Theorem 1, we can fix an element $\zeta \in \mu_m$, the generator of μ_m, then there is an isomorphism $h_{2,m}: \mu_m \otimes \mu_m \to \mu_m$ given by $h_{2,m}(\zeta^{a_1} \otimes \zeta^{a_2}) = \zeta^{a_1 a_2}$. So, we can use this isomorphism to construct a 4-linear map as depicted in Figure 4.

$$\begin{tikzcd}
E[m] \times E[m] \times E[m] \times E[m] \arrow[r, "e_{2,m}"] \arrow[dr, "e_{2m}"'] & \mu_m \otimes \mu_m \arrow[d, dashed, "h_{2,m}"] \\
 & \mu_m
\end{tikzcd}$$

Figure 4. Isomorphism $h_{2,m}$

A 4-linear map $e_{2m}: E[m] \times E[m] \times E[m] \times E[m] \to \mu_m$ is defined as follows:

$$\begin{aligned}
e_{2m}(P_1, P_2, Q_1, Q_2) &= h_{2,m} \circ e_{2,m}(P_1, P_2, Q_1, Q_2) \\
&= h_{2,m}\big(e_m(P_1, Q_1) \otimes e_m(P_2, Q_2)\big) \\
&= \zeta^{a_1 a_2}
\end{aligned}$$

where $\zeta^{a_1} = e_m(P_1, Q_1)$ and $\zeta^{a_2} = (P_2, Q_2)$.

To evaluate 4-linear Weil pairing, Tran et al. proposed Algorithm 1 to compute exponentiation i of a Weil pairing. With the fact that μ_m is a cyclic group generated by ζ, then the Weil pairing is a power of ζ form. This algorithm also uses the fact that the Weil pairing has alternating property to find exponentiation i when e_m has no clear power of ζ.

Algorithm 1

Input: P, Q, m

Output: The exponentiation i of the $e_m(P, Q) = \zeta^i$.

```
$e_m = e_m(P, Q)$
if $e_m == 1$ then
  return 0
end if
for $i = 1$ to $m - 1$ do
  if $\zeta^i == e_m$ then
    return $i$
  else $\{\zeta^i * e_m == 1\}$
    return $-i \mod m$
  end if
end for
```

The 4-linear Weil pairing along with Algorithm 1 can be used to establish a shared key between five users, as depicted in **Figure 5**.

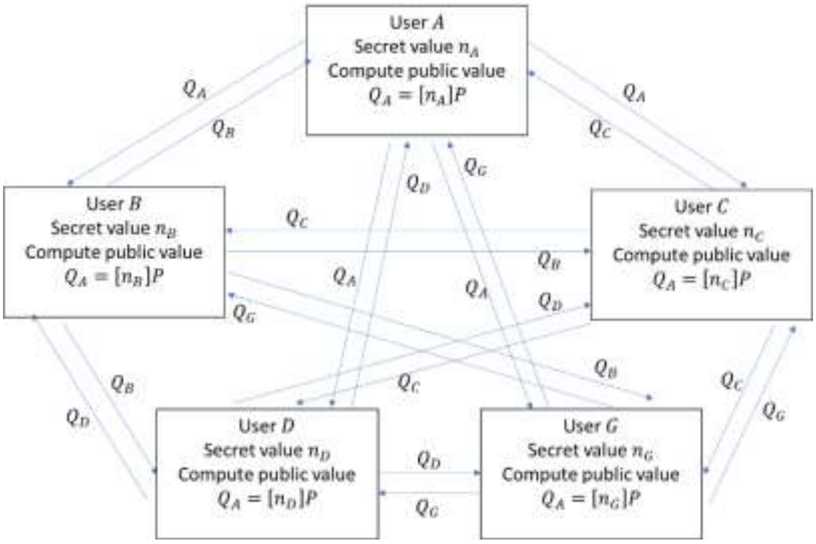

Figure 5. Key Exchange Between 5 Users

To compute a shared key, each user broadcasts their public values to other users. After receiving the public values from others, each user can compute a shared key using all received public values and their respective secret values. From user A's point of view, he can compute a shared key as follows:

Public parameter: $G_1 = E[m], G_2 = \mu_m,\ P,\ \hat{e}_{2,m},\ \hat{e}_{2m},\ \phi$

Received values: Q_B, Q_C, Q_D, Q_G

$$\begin{aligned}
\hat{e}_{2m}\big(Q_B, Q_C, \phi(Q_D), \phi(Q_G)\big)^{n_A} &= \Big(h_{2,m} \circ \hat{e}_{2,m}\big(Q_B, Q_C, \phi(Q_D), \phi(Q_G)\big)\Big)^{n_A} \\
&= \Big(h_{2,m}\Big(\hat{e}_m\big(Q_B, \phi(Q_D)\big) \otimes \hat{e}_m\big(Q_C, \phi(Q_D)\big)\Big)\Big)^{n_A} \\
&= \Big(h_{2,m}\Big(\hat{e}_m\big([n_B]P, \phi([n_D]P)\big) \otimes \hat{e}_m\big([n_C]P, \phi([n_G]P)\big)\Big)\Big)^{n_A} \\
&= \Big(h_{2,m}\Big(\hat{e}_m\big([n_B]P, [n_D]\,\phi(P)\big) \otimes \hat{e}_m\big([n_C]P, [n_G]\phi(P)\big)\Big)\Big)^{n_A} \\
&= \Big(h_{2,m}\Big(\hat{e}_m\big(P, \phi(P)\big)^{n_B n_D} \otimes \hat{e}_m\big(P, \phi(P)\big)^{n_C n_G}\Big)\Big)^{n_A} \\
&= \big(\zeta^{n_B n_D} \otimes \zeta^{n_C n_G}\big)^{n_A} \quad \text{(computed using Algorithm 1)}
\end{aligned}$$

$$= (\zeta^{n_B n_D n_C n_G})^{n_A}$$

$$= \zeta^{n_A n_B n_C n_D n_G}$$

To compute the shared key, $\zeta^{n_A n_B n_C n_D n_G}$, user A must compute the value of $n_B n_D$ and $n_C n_G$ using Algorithm 1. It means that each user must have Algorithm 1 and be able to run this Algorithm efficiently.

3.3 Security Analysis

In this section, we will describe that the 4-linear Weil pairing proposed by Tran et al. does not satisfy the security assumption of multilinear map proposed by Boneh and Silverberg.

Claim 1. *If Algorithm 1 can be run efficiently by each user, then the 4-linear Weil pairing proposed by Tran et al. does not satisfy the multilinear Diffie-Hellman assumption.*

Proof.

Algorithm 1 can be run efficiently by each user, it means that every user can compute the exponentiation i of ζ, such that $e_m(P,Q) = \zeta^i$. In other words, Algorithm 1 can solve discrete logarithm problem in G_2. It implies that each user can compute other's secret value. It will be shown how user A can compute n_B, n_C, n_D, and n_G, the secret value of user B, C, D, and G respectively, using public value received from another user.

User A can find n_B using Algorithm 1. By setting the input to Algorithm 1 as follows: $P, Q = \phi(Q_B)$, and m, Algorithm 1 will yield the exponentiation i of ζ, with $i = n_B$ and $\zeta = \hat{e}_m\big(P, \phi(P)\big)$, such that $\hat{e}_m(P,Q) = \hat{e}_m\big(P, \phi(Q_B)\big) = \zeta^{n_B}$. The same computation can be carried out by user A to derive n_C, n_D, n_G by setting the input to Algorithm 1 $(P, \phi(Q_C), m), (P, \phi(Q_D), m)$, and $(P, \phi(Q_G), m)$ respectively.

Claim 2. *If Algorithm 1 can be run efficiently by each user, then the 4-linear Weil pairing proposed by Tran et al. does not satisfy the Diffie-Hellman inversion assumption.*

Proof.

If Algorithm 1 can be run efficiently, every user can compute the exponentiation i of ζ. After getting the value of exponentiation i of ζ, one can compute the invers of i easily.

Assuming user A has P and $Q_B = [n_B]P$, the user B's public value. With Algorithm 1, user A can find n_B and compute the invers of n_B, say n_B^{-1}. Then Alice can easily compute $e_m(P, \cdots, P)^{n_B^{-1}} \in G_2$. ■

Claim 3. *Assuming each user can run Algorithm 1 efficiently, then the 4-linear Weil pairing proposed by Tran et al. does not satisfy the generalized Diffie-Hellman assumption.*

Proof.

Given all the public values of the users and all subset products $g^{\prod_{i \in S} a_i} \in G_1$ for any strict subset $S \subset \{1, \cdots, n\}$. Algorithm 1 can efficiently find all the exponentiation a_i (for $i = 1, \cdots, n$) or $\prod_{i \in S} a_i$. Then we can compute the product of all a_i or any $\prod_{i \in S} a_i$, such that their product equal to $a_1 a_2 \cdots a_n$. After getting the value of $a_1 a_2 \cdots a_n$, one can compute $g^{a_1 a_2 \cdots a_n} \in G_1$ easily. ■

Claim 4. *Algorithm 1 is not efficient in practice.*

Proof.

If this 4-linear pairing will be implemented in practice, it should use an elliptic curve equation with $q \geq 256$ bits to gain 128-bit security strength. The 4-linear pairing uses a subgroup of elliptic curves with order m. To gain security strength close to 128-bit, then we should choose $m \approx 256$ bit. Note that Algorithm 1 will try any possible values of m to find the correct value of i. It means that if we use $m \geq 256$ bit, Algorithm 1 will try 2^{256} possible values of m and it will take a very long time to get the correct value of i. If we have a supercomputer that can test 10^9 keys per second, we required $\approx 3.6734 \times 10^{60}$ years to try all possibilities. Hence, try all possible values of 2^{256} is effectively impossible with current technology. ■

4. CONCLUSIONS

We demonstrated that Tran et al.'s multilinear map construction fails to meet critical security assumptions and is inefficient for practical implementation. These findings emphasize the need for more robust and efficient multilinear map constructions. Many multilinear maps are proposed with various mathematical functions as their building blocks. Meanwhile, it still needs to be analyzed whether these multilinear maps are still secure and efficiently computed if they are applied in a cryptographic scheme, particularly in multiparty key exchange.

ACKNOWLEDGMENT

We would like to thank the LPDP Ministry of Finance of the Republic of Indonesia for providing scholarships and supporting this research.

REFERENCES

[1] D. Moody, R. Peralta, R. Perlner, A. Regenscheid, A. Roginsky, and L. Chen, "REPORT ON PAIRING-BASED CRYPTOGRAPHY," vol. 120, pp. 11–27, 2015.

[2] D. Boneh and M. Franklin, "IDENTITY-BASED ENCRYPTION FROM THE WEIL PAIRING," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2139 LNCS, no. 3, pp. 213–229, 2001, doi: 10.1007/3-540-44647-8_13.

[3] D. Boneh, B. Lynn, and H. Shacham, "SHORT SIGNATURES FROM THE WEIL PAIRING," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2248, pp. 514–532, 2001, doi: 10.1007/3-540-45682-1_30.

[4] A. Joux, "A ONE ROUND PROTOCOL FOR TRIPARTITE DIFFIE-HELLMAN," *J. Cryptol.*, vol. 17, no. 4, pp. 263–276, 2004, doi: 10.1007/s00145-004-0312-y.

[5] J. H. Hoffstein, Jeffrey; Pipher, Jill; Silverman, *AN INTRODUCTION TO MATHEMATICAL CRYPTOGRAPHY*. Springer Science+Bussiness Media, LLC., 2008. [Online]. Available: https://link.springer.com/book/10.1007/978-0-387-77993-5

[6] D. Boneh and A. Silverberg, "APPLICATIONS OF MULTILINEAR FORMS TO CRYPTOGRAPHY," vol. 0000, pp. 71–90, 2003, doi: 10.1090/conm/324/05731.

[7] S. Arita and S. Handa, "TWO APPLICATIONS OF MULTILINEAR MAPS: GROUP KEY EXCHANGE AND WITNESS ENCRYPTION," *ASIAPKC 2014 - Proc. 2nd ACM Work. ASIA Public-Key Cryptogr.*, pp. 13–22, 2014, doi: 10.1145/2600694.2600699.

[8] H. Lin and S. Tessaro, "INDISTINGUISHABILITY OBFUSCATION FROM TRILINEAR MAPS AND BLOCK-WISE LOCAL PRGS," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10401 LNCS, pp. 630–660, 2017, doi: 10.1007/978-3-319-63688-721.

[9] S. Garg, C. Gentry, and S. Halevi, "CANDIDATE MULTILINEAR MAPS FROM IDEAL LATTICES," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7881 LNCS, pp. 1–17, 2013, doi: 10.1007/978-3-642-38348-9_1.

[10] H. Jia and Y. Hu, "CRYPTANALYSIS OF MULTILINEAR MAPS FROM IDEAL LATTICES: REVISITED," *Des. Codes, Cryptogr.*, vol. 84, no. 3, pp. 311–324, 2017, doi: 10.1007/s10623-016-0266-8.

[11] J. S. Coron, T. Lepoint, and M. Tibouchi, "PRACTICAL MULTILINEAR MAPS OVER THE INTEGERS," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8042 LNCS, no. PART 1, pp. 476–493, 2013, doi: 10.1007/978-3-642-40041-4_26.

[12] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé, "CRYPTANALYSIS OF THE CLT13 MULTILINEAR MAP," *J. Cryptol.*, vol. 32, no. 2, pp. 547–565, 2019, doi: 10.1007/s00145-018-9307-y.

[13] M. H. T. Tran, T. T. Ton, T. D. Nguyen, N. T. Nguyen, N. V. Nguyen, and B. T. Nguyen, "MULTILINEAR MAPPINGS BASED ON WEIL PAIRING OVER ELLIPTIC CURVES," *2017 4th NAFOSTED Conf. Inf. Comput. Sci. NICS 2017 - Proc.*, vol. 2017-Janua, pp. 138–143, 2017, doi: 10.1109/NAFOSTED.2017.8108053.

[14] M. A. Huang, "TRILINEAR MAPS FOR CRYPTOGRAPHY," *Arxiv Cornell Univ.*, vol. 1, pp. 1–11, 2018, [Online]. Available: https://arxiv.org/abs/1803.10325

[15] D. Kahrobaei, A. Tortora, and M. Tota, "MULTILINEAR CRYPTOGRAPHY USING NILPOTENT GROUPS," *Gruyter Proc. Math.*, pp. 127–133, 2020, doi: 10.1515/9783110638387-013.

[16] A. Menezes, "AN INTRODUCTION TO PAIRING-BASED CRYPTOGRAPHY," pp. 47–65, 2009, doi: 10.1090/conm/477/09303.

[17] S. S. Carita and H. Kabetta, "MODIFICATION OF POLLARD RHO ALGORITHM USING NEGATION MAPPING," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 16, no. 4, pp. 1159–1166, 2022, doi: 10.30598/barekengvol16iss4pp1159-1166.

[18] C. Boyd, A. Mathuria, and D. Stebila, *PROTOCOLS FOR AUTHENTICATION AND KEY ESTABLISHMENT*. 2020.