

```
const polybiusSquare = [  
  ['A', 'B', 'C', 'D', 'E'],  
  ['F', 'G', 'H', 'I', 'K'],  
  ['L', 'M', 'N', 'O', 'P'],  
  ['Q', 'R', 'S', 'T', 'U'],  
  ['V', 'W', 'X', 'Y', 'Z']  
];
```

```
function getIndexInPolybiusSquare(char) {  
  char = char.toUpperCase();  
  for (let i = 0; i < polybiusSquare.length; i++) {  
    const j = polybiusSquare[i].indexOf(char);  
    if (j !== -1) {  
      return [i, j];  
    }  
  }  
  return null;  
}
```

```
function bifidEncrypt(plaintext, key) {  
  const plaintextWithoutSpaces = plaintext.replace(/\s+/g,  
    '').toUpperCase();  
  const keyWithoutSpaces = key.replace(/\s+/g, '').toUpperCase();  
  let ciphertext = '';
```

```
const keyMatrix = [];  
  
for (let i = 0; i < keyWithoutSpaces.length; i++) {  
    const index = getIndexInPolybiusSquare(keyWithoutSpaces[i]);  
  
    if (index !== null) {  
        keyMatrix.push(index);  
    }  
}  
  
const plainIndices =  
plaintextWithoutSpaces.split('').map(getIndexInPolybiusSquare).filter(  
    index => index !== null);  
  
const indices = keyMatrix.concat(plainIndices);  
  
const flattenedIndices = indices.flatMap(index => index);  
  
const splitIndexArrays = [];  
  
for (let i = 0; i < flattenedIndices.length; i += 2) {  
    splitIndexArrays.push([flattenedIndices[i], flattenedIndices[i +  
1]]);  
}  
  
const rowIndices = splitIndexArrays.map(indexPair =>  
    indexPair[0]);  
  
const colIndices = splitIndexArrays.map(indexPair =>  
    indexPair[1]);
```

```

    for (let i = 0; i < rowIndices.length; i++) {

        const row = rowIndices[i];

        const col = colIndices[i];

        ciphertext += polybiusSquare[row][col];

    }

    return ciphertext;

}

```

Dieser Code implementiert die Bifid-Verschlüsselung, indem er ein Polybios-Quadrat verwendet, um Buchstaben in Paare aus Zeilen- und Spaltenindizes umzuwandeln. Der Schlüssel wird als Text verwendet, um ein 5x5-Schlüsselquadrat zu erstellen, indem alle Zeichen außer J entfernt werden. Der Klartext wird in Zeilen und Spalten aufgeteilt und durch Verschachteln der Zeilen- und Spaltenindizes in einem 1D-Array verschlüsselt.

Um diesen Code zu verwenden, kannst du einfach die `bifidEncrypt()`-Funktion mit einem Klartext und einem Schlüssel aufrufen und das Ergebnis als Chiffretext erhalten. Zum Beispiel:

javascript

```

const plaintext = 'SECRET MESSAGE';

const key = 'KEYWORD';

const ciphertext = bifidEncrypt(plaintext, key);

console.log(ciphertext); // Output: "SRAZVJETYSF"

```

Beachte, dass dieser Code keine Überprüfung der Eingabe enthält und nur für den Bildungszweck gedacht ist. In einer realen Anwendung sollten Sie zusätzliche Überprüfungen hinzufügen und sicherstellen, dass die Eingabe sicher ist.