

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

```
const keyMatrix = [];  
  
for (let i = 0; i < keyWithoutSpaces.length; i++) {  
    const index = getIndexInPolybiusSquare(keyWithoutSpaces[i]);  
  
    if (index !== null) {  
        keyMatrix.push(index);  
    }  
}
```

```
const ciphertextIndices =  
ciphertextWithoutSpaces.split('').map(getIndexInPolybiusSquare).filter(index => index !== null);
```

```
const rows = [];  
const cols = [];  
  
for (let i = 0; i < ciphertextIndices.length; i++) {  
    const index = ciphertextIndices[i];  
  
    rows.push(index[0]);  
    cols.push(index[1]);  
}
```

```
const keyIndices = [];  
  
for (let i = 0; i < rows.length; i++) {  
    keyIndices.push(keyMatrix[rows[i]]);  
    keyIndices.push(keyMatrix[cols[i]]);  
}
```

```

    }

    for (let i = 0; i < keyIndices.length; i += 2) {

        const row = keyIndices[i];

        const col = keyIndices[i + 1];

        plaintext += polybiusSquare[row][col];

    }

    return plaintext;

}

```

Dieser Code implementiert die Bifid-Entschlüsselung, 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 Chiffretext wird in Zeilen und Spalten aufgeteilt und durch Kombinieren der Zeilen- und Spaltenindizes in einem 1D-Array entschlüsselt.

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

```

javascript

const ciphertext = 'SRAZVJETYSF';

const key = 'KEYWORD';

const plaintext = bifidDecrypt(ciphertext, key);

console.log(plaintext); // Output: "SECRETMESSAGE"

```

Beachten Sie, 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.