```cpp
#include "Vigenere.h"

void Vigenere::initializeTable()
{
    for (char row = 0; row < CHARACTERS; row++)
    {
        char character = 'B' + row;

        for (char col = 0; col < CHARACTERS; col++)
        {
            if (character > 'Z')
            {
                character = 'A';
            }
            fMappingTable[row][col] = character++;
        }
    }
}

Vigenere::Vigenere(const std::string& aKeyword): fKeyword(aKeyword), fKeywordProvider(KeyProvider(aKeyword))
{
    initializeTable();
}

std::string Vigenere::getCurrentKeyword()
{
    std::string current_keyword;

    for (size_t i = 0; i < fKeyword.length(); i++)
    {
        current_keyword += *fKeywordProvider;
        fKeywordProvider << *fKeywordProvider;
    }
    return current_keyword;
}

void Vigenere::reset()
{
    fKeywordProvider.initialize(fKeyword);
}

char Vigenere::encode(char character)
{
    if (isalpha(character))
    {
        bool isLower = std::islower(character);
        char encoded = fMappingTable[*fKeywordProvider - 'A'][std::toupper(character) - 'A'];

        fKeywordProvider << character;
        if (isLower)
        {
            return static_cast<char>(std::tolower(encoded));
        }
        return encoded;
    }
    return character;
}

char Vigenere::decode(char character)
{
    if (isalpha((character)))
    {
        bool isLower = std::islower(character);
        char encoded = static_cast<char>(toupper(character));
        char decoded = 0;

        for (char col = 0; col < CHARACTERS; col++)
        {
            if (fMappingTable[*fKeywordProvider - 'A'][col] == encoded)
            {
                decoded = static_cast<char>(col + 'A');
                break;
            }
        }

        fKeywordProvider << decoded;
        if (isLower)
        {
            return static_cast<char>(std::tolower(decoded));
        }
        return decoded;
    }
    return character;
}
```