```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/TRC20/ITRC20.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract Gratium is ITRC20, Ownable {
    uint256 private constant TOTAL_SUPPLY = 1_000_000_000 * (10 ** 18);
    uint256 private constant MAX_MESSAGE_LENGTH = 256; // Limit message length to prevent exce

    constructor() ITRC20("Gratium", "GRTM") {
        _mint(msg.sender, TOTAL_SUPPLY); // All tokens minted at deployment
    }

    event GratiumSent(address indexed sender, address indexed recipient, uint256 amount, strin

    /**
     * @dev Sends Gratium tokens along with a message. Messages are stored on-chain with a max
     * This aligns with the digital fortune cookie concept, where users can attach messages, r
     * or even hidden PollCoin amounts inside the Gratium transaction. The message remains sto
     * and can be retrieved by the recipient upon opening.
     *
     * Note: Messages are stored in plaintext and visible on-chain. Consider using off-chain e
     * or hashing mechanisms for privacy-sensitive messages in future upgrades.
     */
    function sendGratium(address recipient, uint256 amount, string memory message) external {
        require(balanceOf(msg.sender) >= amount, "Insufficient GRTM balance");
        require(bytes(message).length <= MAX_MESSAGE_LENGTH, "Message too long");

        _transfer(msg.sender, recipient, amount);
        emit GratiumSent(msg.sender, recipient, amount, message);
    }
}
```