**COBOL and ALGOL Financial Messaging System**

**Overview**

This project is a simulation of a financial transaction processing system, demonstrating how different programming languages like **COBOL** and **ALGOL** can be used together to handle specific tasks based on their strengths.

The system uses the **ISO 8583** standard to format messages for financial transactions, such as authorization requests and network management. COBOL is used for its powerful data handling and record management capabilities, making it ideal for building and parsing structured messages. ALGOL is used for its strengths in mathematical and algorithmic computation, making it suitable for cryptographic calculations.

**Programs**

This repository contains the following key programs:

**1. ISO8583-GEN.cbl & ISSAUTH.cbl**

* **Purpose**: This pair of programs demonstrates a fundamental client-server model for online transaction authorization.
* **ISO8583-GEN.cbl (The Acquirer)**: Acts as the "acquirer" or Point-of-Sale terminal. It builds an 0200 financial authorization request based on user input (PAN, amount) and sends it for processing.
* **ISSAUTH.cbl (The Issuer)**: Acts as the "issuer" or financial institution. It is called by ISO8583-GEN, receives the request, applies simple business logic (e.g., declining transactions over a certain amount), and returns an 0210 response message with an approval or decline code.

**2. ISO8583-FLOW.cbl & ALGOL\_MAC\_CALC.alg**

* **Purpose**: This pair demonstrates a more advanced and realistic architecture, incorporating both the **Request/Advice** message flow and a **multi-language call** for security processing.
* **ISO8583-FLOW.cbl (The Acquirer/Flow Controller)**: This is the main program that simulates a complete transaction lifecycle.
* **Request/Response (0100/0110)**: Simulates a real-time authorization where a response is required.
* **Advice/Acknowledgement (0220/0230)**: Simulates a notification message that only needs to be acknowledged.
* **Integration**: Before sending a message, it calls the ALGOL\_MAC\_CALC program to generate a security code (MAC).
* **ALGOL\_MAC\_CALC.alg (The Cryptographic Engine)**: This program is a specialized module written in ALGOL. Its sole purpose is to perform a complex mathematical calculation to generate a **Message Authentication Code (MAC)**. It accepts a message and a secret key from the COBOL program and returns the calculated 8-byte MAC.

**Architectural Approach**

The design of this system intentionally separates concerns:

* **COBOL**: Used for what it does best—handling structured business data, file I/O, and record layouts. It acts as the controller, building the main message bodies.
* **ALGOL**: Used as a specialized "engine" for a task it is better suited for—complex, logical, and mathematical algorithms. This reflects a common pattern in legacy systems where a primary business language would call out to a more specialized language for tasks like scientific, statistical, or, in this case, cryptographic computation.

**How to Run**

1. **Compile the Modules**: The external, callable programs must be compiled first.

* Compile ISSAUTH.cbl into a callable library/object file.
* Compile ALGOL\_MAC\_CALC.alg into a callable library/object file.

1. **Compile the Main Programs**:

* Compile ISO8583-GEN.cbl and link it with the ISSAUTH library.
* Compile ISO8583-FLOW.cbl and link it with the ALGOL\_MAC\_CALC library.

1. **Execute**:

* Run the ISO8583-GEN executable to test the basic authorization flow.
* Run the ISO8583-FLOW executable to test the advanced Request/Advice flow with MAC generation.