Match On Card Implementation Challenges

PRESENTED BY:
Powell Benedict
powell.benedict@identix.com
10 March 2004
Match On Card

- Program Size
  - GSA/DMDC CAC-A Limited 2 Templates and Algorithm to 5K
  - Variable Type/Size Limitations
    - Signed Shorts and Bytes
  - Reduced Algorithmic Complexity
  - Must Maintain Good Error Rates
    - Refining Match Algorithms
    - Multiple Templates / Fail-over Matching
- Current Identix MOC Implementation
  - Template Size < 255 Bytes
  - Code Size < 1900 Bytes
  - Error Rates Comparable to BioEngine 2.x
Match On Card

• Implementation Speed
  – Includes Data Transmission Times
    ■ USB Interfaces are an Improvement
  – Best Case / Worst Case Execution Time
  – User Requirements Depending on Use
    ■ Physical Access
    ■ Logical Access

• Memory Type Considerations
  – Volatile Memory (RAM)
    ■ Very fast
    ■ Used for Stack Allocation, Etc.
    ■ Very little available (<255 Bytes)
  – Non-volatile (EEPROM)
    ■ Very Slow for Calculations and Results Aggregation
Match On Card

- Language Selection
  - Portability
    - Java Applets / Java Card 2.1
    - JVM Adds Significant Execution Time
  - Speed
    - C Code to Native Code Compilers
    - Manufacturer-specific Native Code
      - No portability
Match On Card

- Data Security In Transmission
  - Template Encryption
    - Enrolled Template
      - Safe on Card
      - Transmitted to Card in Controlled Environment
    - Verification Template
      - Transmit in the Clear – or –
      - Decrypt on Card
  - MOC: Enrolled Template Never Leaves Card