Reference Documents

Device Data Model Reference

This section provides a comprehensive reference for the device data model implemented within the solution. The model is centered around the DEVICES table, which serves as the authoritative registry for all devices interacting with the system.

Table: DEVICES

The DEVICES table resides in the SAM_AUTH_SAM schema. Each record represents a unique instance of an application installation on a physical or virtual device, linked to a specific user account.

Column Name	**Data Type**	**Description**
ID	VARCHAR2 (64)	Internal primary key. A unique identifier for the device record.
DEVICE_ID	VARCHAR2 (255)	Globally unique identifier for the device, generated and securely stored by the mobile application. This is the primary business key.
USER_ID	VARCHAR2 (64)	Foreign key linking the device to a specific user account in the USER table.
STATUS	NUMBER(10)	Current lifecycle state: 0 = PENDING, 1 = ACTIVE, 2 = SUSPENDED, 3 = REVOKED.
CREATED_AT	TIMESTAMP(6)	Timestamp when the device record was created.
UPDATED_AT	TIMESTAMP(6)	Timestamp when the device record was last modified.

Column Name	Data Type	Description
LAST_ACTIVITY	TIMESTAMP(6)	Timestamp of the last successful authentication or transaction from this device.
DEVICE_NAME	VARCHAR2(50)	User-friendly alias for the device (e.g., "My iPhone").
BRAND_NAME	VARCHAR2(50)	Manufacturer or brand of the device.
MODEL	VARCHAR2(255)	Specific model name or number.
OS	VARCHAR2(50)	Operating system name.
OS_VERSION	VARCHAR2(50)	Operating system version.
APP_VERSION	VARCHAR2(20)	Version of the mobile banking application.
APP_BUILD_CODE	VARCHAR2(20)	Build number of the mobile banking application.
FCM_TOKEN	VARCHAR2(255)	Push notification token for Firebase Cloud Messaging (FCM) or Apple Push Notification service (APNs).

Integrity and Environment Indicators

Column Name	Data Type	Description
IS_ROOTED	NUMBER(1)	Indicates if the device is rooted (Android) or jailbroken (iOS).
IS_EMULATOR	NUMBER(1)	Indicates if the application is running on an emulator or simulator.
IS_DEBUG_BUILD	NUMBER(1)	Indicates if the application is a debuggable build.
IS_HOOK_DETECTED	NUMBER(1)	Indicates the presence of hooking frameworks (e.g., Frida, Xposed).
IS_APP_TAMPERED	NUMBER(1)	Indicates if the application's signature has been modified.
VPN_DETECTED	NUMBER(1)	Indicates if a VPN connection is active.
PROXY_DETECTED	NUMBER(1)	Indicates if an HTTP proxy is configured.
TOR_DETECTED	NUMBER(1)	Indicates if the device is connected through the Tor network.
IS_MOCK_LOCATION	NUMBER(1)	Indicates if location spoofing is detected.

Advanced Device Fingerprinting

Column Name	Data Type	Description
COMPOSITE_HASH	VARCHAR2(64)	Composite cryptographic hash of hardware and software attributes for device fingerprinting.
HW_HASH	VARCHAR2(64)	Hash computed primarily from hardware attributes.
SW_HASH	VARCHAR2(64)	Hash computed primarily from software attributes.
FP_ALGORITHM	VARCHAR2(20)	Algorithm used to generate the fingerprint hashes.
FP_VERSION	VARCHAR2(10)	Version of the fingerprinting logic.

Network and Location Context

Column Name	Data Type	Description
IP_ADDRESS	VARCHAR2(255)	Public IP address of the device.
IP_COUNTRY	VARCHAR2(5)	Country code associated with the IP address.
IP_CITY	VARCHAR2(100)	City name associated with the IP address.
ASN	VARCHAR2(20)	Autonomous System Number of the network provider.
CARRIER_NAME	VARCHAR2(100)	Name of the mobile carrier.

Column Name	Data Type	Description
NETWORK_TYPE	VARCHAR2(20)	Type of network connection (e.g., WIFI, CELLULAR).

Usage in Policies and Risk Evaluation

The fields within the DEVICES table are actively consumed by the **Risk Engine Service** and the **Authentication Orchestration Layer** to make real-time access and transaction approval decisions.

- **Trust Evaluation:** A device is considered **trusted** if its STATUS is ACTIVE and its integrity flags do not violate configured security policies.
- **Risk Scoring:** The **Risk Engine** queries the device's integrity flags and fingerprint hashes to calculate a real-time risk score. A mismatch in COMPOSITE_HASH or a positive IS_HOOK_DETECTED flag will significantly increase the risk score.
- **Anomaly Detection:** Changes in attributes like IP_COUNTRY combined with an "Impossible Journey" time window are used to flag suspicious activity.
- **Compliance Reporting:** The detailed device context stored in this table provides the necessary data for generating audit trails required by **Circular 50/2024/TT-NHNN**.

Context and Risk Attribute Reference

These attributes originate from the device, the transaction, the user's behavior, and external risk engines.

Core Risk Decision Context

The RISK_DECISION_LOG table captures the outcome of each risk evaluation performed by the **Risk Engine Service**.

Attribute	Source	Description
DECISION_ID	Risk Engine	Unique identifier for the risk decision session.
USER_ID	Request Context	The identifier of the user initiating the action.
DEVICE_ID	Request Context	The identifier of the device used for the action.
SESSION_ID	Request Context	The active session identifier.
PAYMENT_ID	Transaction Context	The unique identifier of the payment or transaction, if applicable.
TRANSACTION_TYPE	Transaction Context	The type of transaction (e.g., TRANSFER, PAYMENT, LOGIN).
TRANSACTION_AMOUNT	Transaction Context	The monetary value of the transaction.
FMS_SCORE	Fms / FMS	The raw risk score (0.00 - 100.00) calculated by the external fraud management system.
FMS_ACTION	Fms / FMS	The recommended action from the fraud system (APPROVE, REVIEW, BLOCK_AND_REVIEW, DECLINE).

Attribute	Source	Description
APPLIED_POLICY_ID	Risk Engine	The identifier of the RISK_POLICY record that was matched and applied.
DECIDED_FACTORS	Risk Engine	A JSON structure detailing the required and fallback authentication factors.
STEP_UP_REQUIRED	Risk Engine	A boolean flag indicating whether strong customer authentication is mandated.
CONTEXT_SIGNALS	Aggregated	An encrypted binary field containing a comprehensive JSON payload of all context signals used in the evaluation.

Adaptive Context Signals

The **Risk Engine** evaluates real-time adaptive signals defined in the ADAPTIVE_CONTEXT_RULE table.

Signal Type	Description	Escalation Action
HIGH_VALUE_TXN	Transaction amount exceeds a defined threshold.	UPGRADE_FACTOR (e.g., to FIDO2)
NEW_DEVICE	The request originates from a device not previously	UPGRADE_FACTOR

Signal Type	Description	Escalation Action
	associated with the user.	
UNUSUAL_TIME	The transaction occurs outside of normal business hours (e.g., 22:00 - 06:00).	UPGRADE_FACTOR
MULTIPLE_FAIL	Multiple recent authentication failures have been detected for the user.	BLOCK or REQUIRE_ADMIN_REVIEW
HIGH_RISK_COUNTRY	The beneficiary account is located in a high-risk jurisdiction.	UPGRADE_FACTOR
FIRST_TIME_RECIPIENT	The user is sending funds to a beneficiary	UPGRADE_FACTOR

Signal Type	Description	Escalation Action
	for the first time.	

User Authentication Setup Context

The USER_AUTHEN_SETUP table provides the Risk Engine with information about the authentication factors a user has enrolled.

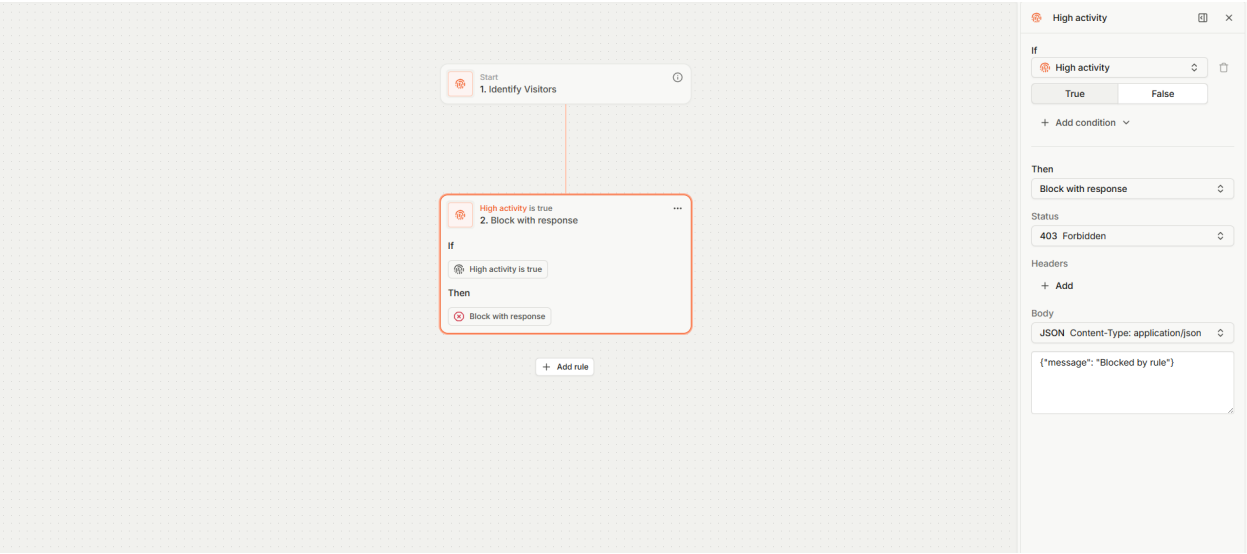
Attribute	Description
USER_ID	The user's identifier.
FACTOR_TYPE	The type of authentication factor (FIDO2, SMART_OTP, SMS_OTP, etc.).
STATUS	The enrollment status (ACTIVE, INACTIVE, REVOKED).
DEVICE_ID	The device on which the factor was registered (for device-bound factors).
PRIORITY	The user's preferred order of factors.
LAST_USED_AT	Timestamp of the last successful use of this factor.

Integration with Curity

These attributes are made available to the Curity orchestration layer through Authentication Actions and Context Attributes. This allows Curity to make dynamic policy decisions, such as triggering a Conditional Multi-Factor step or denying access based on a high-risk score.

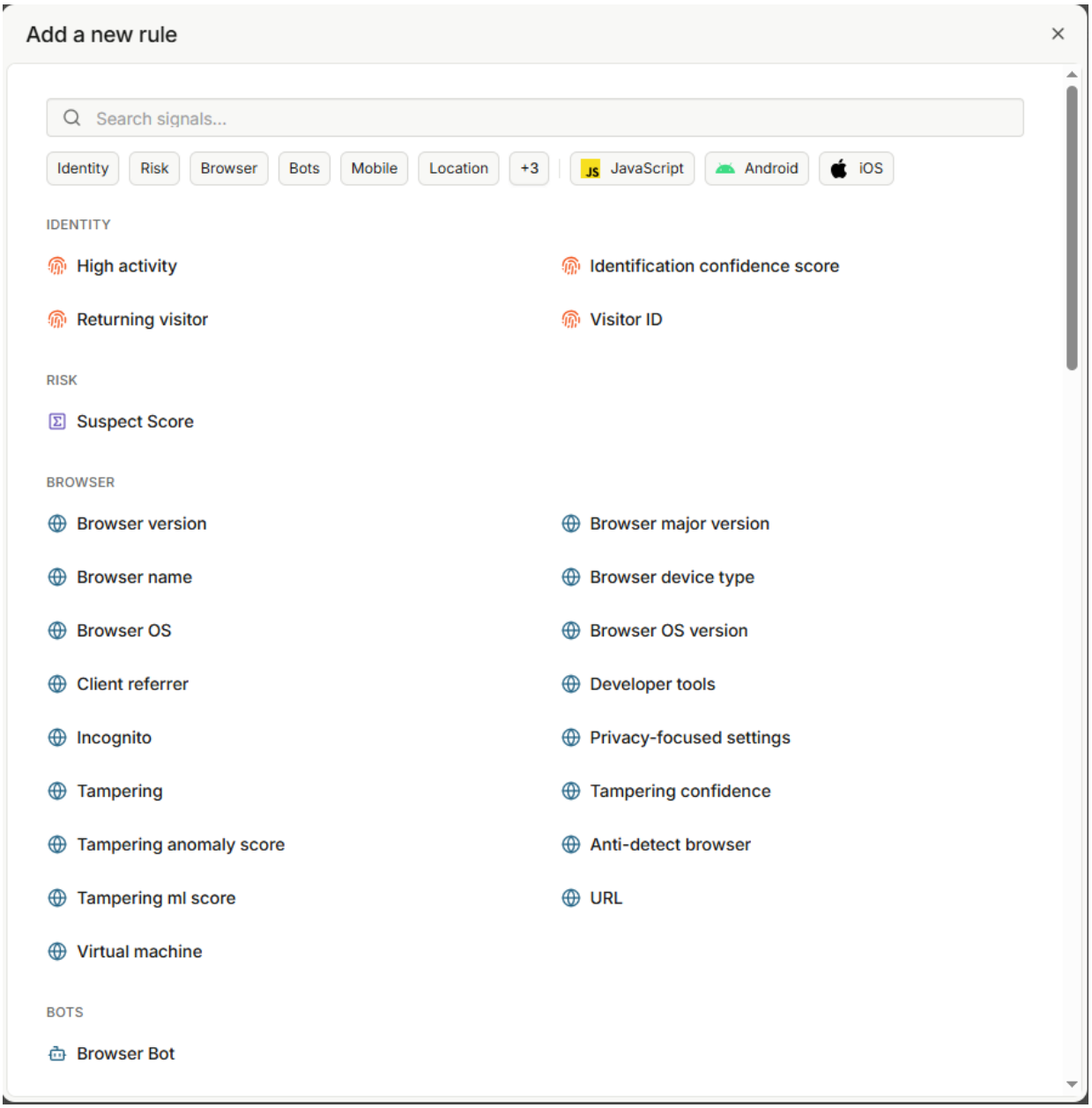
Configuration Rule Base

The two images illustrate the solution's Risk Rule Configuration interface within the administrative portal. This visual, low-code/no-code rule builder empowers security and fraud teams to define granular, context-aware policies for adaptive authentication and transaction authorization without writing custom code.



The first image shows a high-level, flow-chart style rule builder. It provides a visual representation of a security policy:

- Start Node: 1. Identify Visitors – Represents the initial point where the system begins evaluating an incoming request.
- Condition Node: If High activity is true – A decision point based on a specific risk signal (High activity). This signal is one of many available attributes that the Risk Engine can evaluate in real-time.
- Action Node: Block with response – The consequence that is executed if the condition is met. The interface clearly indicates that the action is not just a simple "block," but a customizable response.
- + Add rule Button: Allows administrators to easily extend the policy with additional conditions and branches, creating complex decision trees.



The second image provides a detailed view of the configuration panel used to define a new rule. It highlights the extensive library of contextual signals and the flexible response options available to policy authors.

Identity-Device-Credential Relationship Model

User (Identity) , **Device**, and **Credential**. These relationships are physically enforced through the database schema and are fundamental to implementing trusted device policies, strong authentication, and transaction binding.

Key Features Demonstrated:

A. Extensive Library of Contextual Signals

The left panel displays a categorized, searchable list of signals that can be used as conditions in a rule. This demonstrates the richness of the solution's device and behavioral intelligence.

Category	Example Signals Available
Identity	High activity, Returning visitor, Identification confidence score, Visitor ID
Risk	Suspect Score (a consolidated risk score from the fraud engine)
Browser	Browser name, Browser version, Developer tools (detection), Incognito, Tampering, Anti-detect browser
Bots	Browser Bot (detection)
Mobile	Platform-specific signals (e.g., Android, iOS)
Location	Geolocation and IP-based signals (implied in the +3 category)

This comprehensive library allows administrators to create highly specific rules, such as *"Block access if the* Suspect Score *> 80 AND the browser is running in* Incognito *mode."*

B. Flexible Response Actions

The interface explicitly states that the action for a triggered rule is Block with response. The configuration allows for deep customization of this response, going far beyond a simple HTTP 403 error. Administrators can configure:

- HTTP Status Code: A dropdown (shown in the mockup context) allows selection from various standard HTTP error codes (e.g., 403 Forbidden, 429 Too Many Requests, 503 Service Unavailable).

- Custom Response Payload: The system supports defining a custom JSON or text response body to be returned to the client. This is crucial for providing informative and controlled error messages to the user or calling application without exposing internal security logic.

3.3.1 Relationship Descriptions

Relationship	Cardinality	Description
User → Device	One-to-Many	A single user account (USER) can own multiple trusted devices (DEVICES). The USER_ID foreign key in DEVICES establishes this ownership.
Device → FIDO2 Credential	One-to-Many	A single device can create multiple WebAuthn/Passkey credentials (FIDO_IDENTITY). The DEVICE_ID field links the credential to the specific device that generated it.
Device → Signing Key	One-to-Many	A single device can be provisioned with one or more device signing keys (KEY_DATA). The DEVICE_ID field links the public key to the device.
User → Transaction	One-to-Many	A user performs many transactions (TRANSACTION).
Device → Transaction	One-to-Many	Each transaction is associated with the specific DEVICE_ID from which it was initiated, enabling traceability and non-repudiation.

3.3.2 Policy Implications

- **Trusted Device Enforcement:** An authentication or transaction request is only allowed if it originates from a DEVICE_ID that is associated with the user and has a STATUS of ACTIVE.
- **Credential Binding:** A WebAuthn assertion is only considered valid if the CREDENTIAL_ID is linked to the correct USER_ID and, optionally, if the presenting DEVICE_ID matches the DEVICE_ID on which the credential was registered.
- **Revocation Cascade:** When a device's STATUS is set to REVOKED, all associated credentials (FIDO_IDENTITY, KEY_DATA) and active sessions for that device are effectively invalidated, preventing a lost or stolen device from being used for access or approval.

Event, Audit, and Security Logging Reference

The solution implements a multi-layered logging strategy to ensure a complete, non-repudiable audit trail for all security-relevant events, meeting the requirements of Circular 50/2024/TT-NHNN, 64/2024/TT-NHNN, PCI DSS, and ISO 27001.

Audit Log Tables

The following tables serve as the primary audit trail for user, device, and transaction activities.

Table Name	Schema	Purpose
TRANSACTION	SAM_AUTH_SAM	Records the details of every financial transaction, including the user, device, authentication method, and outcome.
ACTION_LOG	SAM_AUTH_SAM	Logs critical user actions such as login attempts, device registration, and changes to security settings.

Table Name	Schema	Purpose
USER_HISTORY	SAM_AUTH_SAM	Tracks changes to a user's account status and profile information.
DEVICE_HISTORY	SAM_AUTH_SAM	Provides an immutable history of device state changes (e.g., ACTIVE → REVOKED), including the actor and reason.
RISK_DECISION_LOG	RISK_ENGINE	Contains a detailed record of every risk evaluation and policy decision made by the Risk Engine Service.
DECISION_OUTCOME_EVENT	RISK_ENGINE	Stores a chronological timeline of state changes for each RISK_DECISION_LOG record (e.g., CREATED, FULFILLED, EXPIRED).

Security Event Logging

The solution generates structured JSON logs for all API requests and system events, which are designed for ingestion by SIEM/SOC platforms.

- **Format:** All logs are output in a single-line JSON format using **Logstash Logback Encoder**, facilitating easy parsing and indexing.
- **Traceability:** Every log entry includes a traceId (propagated via the X-Trace-Id header) to correlate all actions and events across different services for a single user request.
- **PII Protection:** Sensitive data such as userId, deviceId, and paymentId are never logged in plaintext. They are replaced with a salted SHA-256 hash before being written to the logs.

- **Key Events:** The following critical events are logged at INFO level or higher and can be configured to trigger real-time alerts:
    - RISK_DECISION_CREATED / FULFILLED / DECLINED
    - SECURITY_REPLAY_ATTEMPT
    - SECURITY_UNAUTHORIZED_FULFILL
    - SETUP_FACTOR_REVOKED
    - ADAPTIVE_SIGNAL_TRIGGERED

Kafka Event Streaming

For loose coupling and fan-out to downstream systems, the solution publishes key events to a Kafka cluster using the **Transactional Outbox Pattern**.

Topic	Producer	Consumer(s)	Description
sam.factor.registered	SAM Service	Risk Engine	Notifies that a user has enrolled a new authentication factor.
risk-engine.decision.created	Risk Engine	Audit-Log, Analytics	Published when a new risk decision is created.
risk-engine.decision.fulfilled	Risk Engine	RAS, Fms, Audit	Published when a user successfully fulfills a step-up authentication challenge.
risk-engine.decision.declined	Risk Engine	Fms, Alert, Audit	Published when a transaction is declined due to high risk or policy violation.

This event stream provides a real-time feed for fraud monitoring, compliance reporting, and integration with the bank's enterprise data lake.

Sensitive Data at Rest Protection

To comply with Circular 50/2024/TT-NHNN, Circular 64/2024/TT-NHNN and Circular 77/2025/TT-NHNN, all sensitive customer and transaction data stored in the database is protected using strong encryption. The solution employs an envelope encryption scheme that combines the performance of symmetric encryption with the secure key management of asymmetric cryptography.

Protected Data Categories

The following categories of data, as mandated by the State Bank of Vietnam, are encrypted at rest within the database:

Category	**Description**	**Example Columns**
Customer Personal Information	Information used to identify or authenticate a customer.	USER.IDENTITY_NUMBER, USER.PHONE_NUMBER, USER.EMAIL
Authentication Secrets	Credentials used for user authentication.	USER.PIN, DEVICES.USER_PIN, DEVICES.PIN_SALT, USER.OTP_SECRET
Transaction Details	Detailed information about financial transactions.	TRANSACTION.PAYMENT_INFO
Risk Evaluation Context	Aggregated signals and context data used for risk scoring.	RISK_DECISION_LOG.CONTEXT_SIGNALS

Encryption Mechanism: Envelope Encryption (RSA-OAEP-256 + AES-256-CBC)

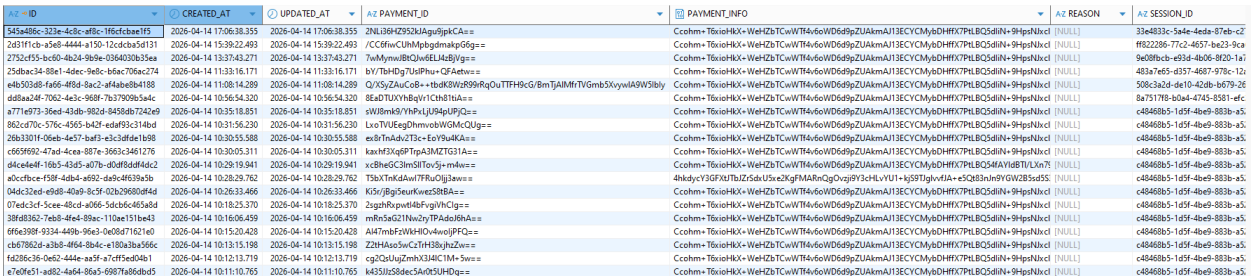
The encryption framework is implemented as a custom JPA AttributeConverter that transparently encrypts data before writing to the database and decrypts it after reading. The mechanism operates as follows for each record:

1. **Generate a Unique Data Encryption Key (DEK):** A cryptographically secure random **AES-256** key is generated for each sensitive field (or record).
2. **Generate a Unique Initialization Vector (IV):** A random **16-byte IV** is generated for AES encryption in CBC mode.
3. **Encrypt the Plaintext:** The sensitive data is encrypted using **AES-256-CBC** with the generated DEK and IV.
4. **Encrypt the DEK:** The AES DEK is then encrypted using the system's **RSA Public Key (2048-bit)** with **RSA-OAEP-256** padding.
5. **Store as a Binary Package:** The final stored value in the database is a concatenated binary package consisting of:
    - [IV (16 bytes)] + [Encrypted AES Key (256 bytes)] + [Encrypted Data (variable)]
6. **Encode for Storage:** This binary package is then Base64-encoded and stored in a VARCHAR2 column, or stored directly as RAW/BLOB for columns expecting binary data.

This envelope encryption model ensures that the heavy-lifting AES key is itself protected by RSA, and the RSA private key is the only secret that must be strictly guarded. The RSA private key is stored securely in an external **Vault** or **Hardware Security Module (HSM)** and is never persisted in the application configuration or database.

Visual Evidence of Encryption

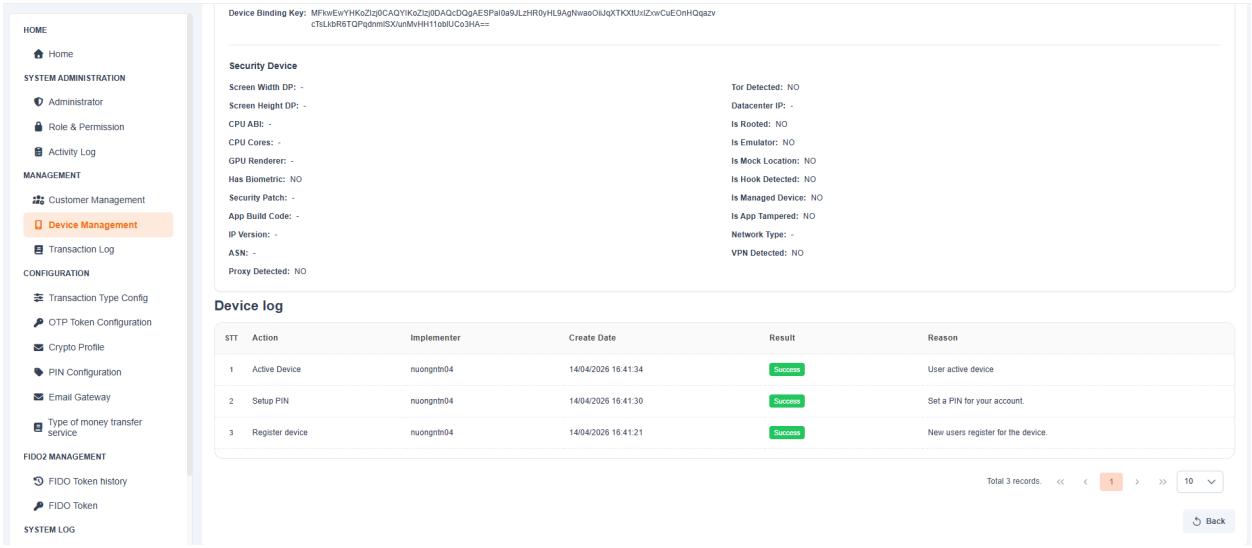
The screenshot below, taken directly from the solution's database client, illustrates how encrypted data appears at rest. The PAYMENT_INFO column in the TRANSACTION table contains only Base64-encoded ciphertext, rendering the underlying transaction details unreadable to anyone with direct database access.



ID	CREATED_AT	UPDATED_AT	PAYMENT_ID	PAYMENT_INFO	REASON	SESSION_ID
545a486c-323e-4c8c-af8c-1f6cfcbae1f5	2026-04-14 17:06:38.355	2026-04-14 17:06:38.355	2NLi36HZ952kJAgu9jpkCA==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	33e4833c-5a4e-4eda-87eb-c2
2d31f1cb-a5e8-4444-a150-12cdcba5d131	2026-04-14 15:39:22.493	2026-04-14 15:39:22.493	/CC6fiwCUhMpbgdmakpG6g==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	ff822286-77c2-4657-be23-9ca
2752cf55-bc60-4b24-9b9e-0364030b35ea	2026-04-14 13:37:43.271	2026-04-14 13:37:43.271	7wMynwJBtQJw6ELJ4zBjVg==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	9e08fbcb-e93d-4b06-8f20-1a7
25dbac34-88e1-4dec-9e8c-b6ac706ac274	2026-04-14 11:33:16.171	2026-04-14 11:33:16.171	bY/TbHDg7UsIPhu+QFAetw==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	483a7e65-d357-4687-978c-12
e4b503d8-fa66-4f8d-8ac2-af4abe8b4188	2026-04-14 11:08:14.289	2026-04-14 11:08:14.289	Qj/XSyZAuCoB++tbdK8WzR99iRqOuTTFH9cG/BmTjAlMfrTVGmb5XvywlA9W5lbly	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	508c3a2d-de10-42db-b679-26
dd8aa24f-7062-4e3c-968f-7b37909b5a4c	2026-04-14 10:56:54.320	2026-04-14 10:56:54.320	8EaDTUXYhBqVr1Cth81tiA==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	8a7517f8-b0a4-4745-8581-efc
a771e973-36ed-43db-982d-8458db7242e9	2026-04-14 10:35:18.851	2026-04-14 10:35:18.851	sWJ8mk9/YhPxLjU94pUPjQ==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
862cd70c-576c-4565-b42f-edaf93c314bd	2026-04-14 10:31:56.230	2026-04-14 10:31:56.230	LxoTVUEegDhmvobWGMcQUg==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
26b3301f-06eb-4e57-baf3-e3c3dfde1b98	2026-04-14 10:30:55.588	2026-04-14 10:30:55.588	ex8rTnAdvZT3c+EoY9u4KA==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
c665f692-47ad-4cea-887e-3663c3461276	2026-04-14 10:30:05.311	2026-04-14 10:30:05.311	kaxhf3Xq6PTrpA3MZTG31A==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
d4ce4e4f-16b5-43d5-a07b-d0df8ddf4dc2	2026-04-14 10:29:19.941	2026-04-14 10:29:19.941	xcBheGC3ImSlITov5j+m4w==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ54fAYIdBTI/LXn7!	[NULL]	c48468b5-1d5f-4be9-883b-a5.
a0ccfbce-f58f-4db4-a692-da9c4f639a5b	2026-04-14 10:28:29.762	2026-04-14 10:28:29.762	T5bXTnKdAwI7FRuOIjj3ww==	4hkdycY3GFXtJTbJZr5dxU5x2KgFMARnQgOvzji9Y3cHLvYU1+kj59TJgIvvfJA+e5Qt83nJm9YGW2B5sd5S	[NULL]	c48468b5-1d5f-4be9-883b-a5.
04dc32ed-e9d8-40a9-8c5f-02b29680df4d	2026-04-14 10:26:33.466	2026-04-14 10:26:33.466	Ki5r/jBgiSeurKwez58tBA==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
07edc3cf-5cee-48cd-a066-5dcb6c465a8d	2026-04-14 10:18:25.370	2026-04-14 10:18:25.370	2sgzhRxpwtI4bFvgiVhCIg==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
38fd8362-7eb8-4fe4-89ac-110ae151be43	2026-04-14 10:16:06.459	2026-04-14 10:16:06.459	mRn5aG21Nw2ry7PAdsJ6hA==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
6f6e398f-9334-449b-96e3-0e08d7162fe0	2026-04-14 10:15:20.428	2026-04-14 10:15:20.428	AiH7mbFzWHHOv4wojPFQ==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
cb67862d-a3b8-4f64-8b4c-e180a3ba566c	2026-04-14 10:13:15.198	2026-04-14 10:13:15.198	ZZtHAsq5wCzTrH38vjhzZw==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
fd286c36-0e62-444e-aa5f-a7cff5ed04b1	2026-04-14 10:12:13.719	2026-04-14 10:12:13.719	cg2QsUujZmhX3J4IC1M+5w==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.
e7e0fe51-ad82-4a64-86a5-6987fa86dbd5	2026-04-14 10:11:10.765	2026-04-14 10:11:10.765	k435IJzS8dec5Ar0t5UHDg==	Ccohm+T6xioHkX+WeHZbTCwWTf4v6oWD6d9pZUAkmAJ13ECYCMybDHffX7PtLBQ5dliN+9HpsNlxcl	[NULL]	c48468b5-1d5f-4be9-883b-a5.

Figure: The PAYMENT_INFO *column in the* TRANSACTION *table stores AES-256-CBC encrypted data, encoded as a Base64 string. Without the RSA private key, the data cannot be decrypted.*

3.4.5 Device history, transaction history

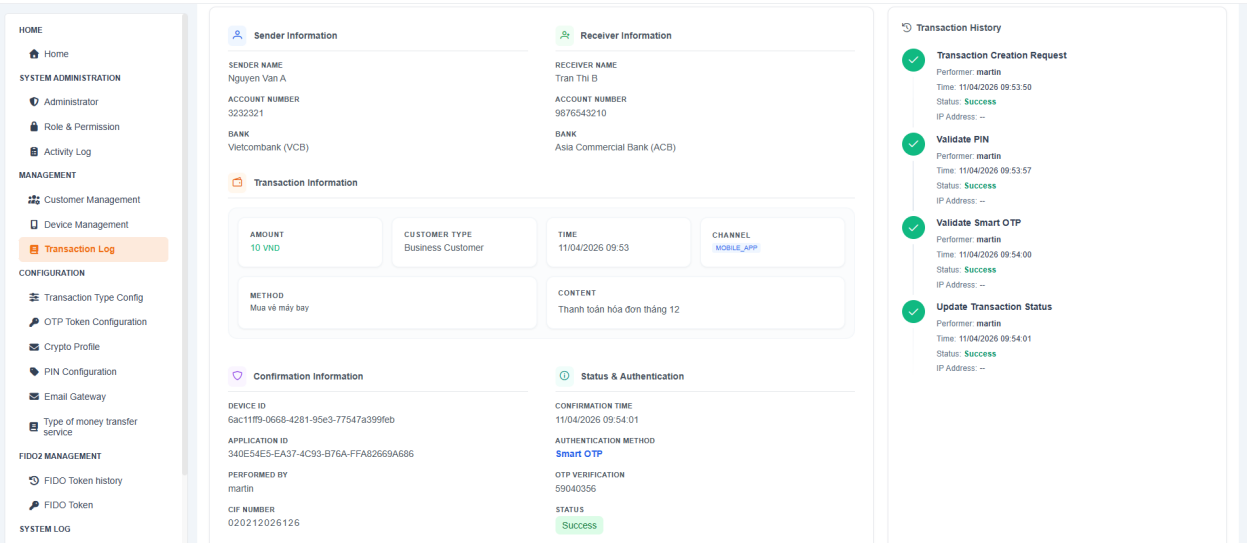


The image illustrates the Device Management interface within the solution's administrative portal, specifically highlighting the Device Log (audit trail) feature. This view provides a comprehensive and non-repudiable history of all significant lifecycle events for a specific registered device.

Key elements shown in the image include:

- Device Context: The top section displays key identifiers for the device being viewed, including its unique Device Binding Key (a public key used for cryptographic device attestation) and a summary of its security posture (e.g., Has Biometric: NO, Proxy Detected: NO).
- Audit Log Table (Device Log): The main table provides a chronological record of all actions performed on or by this device. Each entry is a structured audit record containing:

- STT (Row Number): Sequential identifier for the log entry.
- Action: The specific operation performed. Examples visible include Active Device, Setup PIN, and Register device. This corresponds to the ACTION column in the DEVICE_HISTORY database table.
- Implementer: The identity of the actor who performed the action. In this case, all actions were performed by the end-user nuongnt04. This provides clear accountability and supports non-repudiation.
- Create Date: A precise timestamp of when the event occurred (format: DD/MM/YYYY HH:MM:SS), enabling forensic timeline analysis.
- Result: The outcome of the action (e.g., Success), corresponding to the RESULT field in the audit table.
- Reason: A descriptive, human-readable reason for the action (e.g., "New users register for the device", "Set a PIN for your account"). This field is critical for understanding the business context of the event.



The image depicts the Transaction Log detail view within the solution's administrative portal. This interface provides a complete, forensic-level audit trail for a specific financial transaction, demonstrating how the solution captures every step of the authorization lifecycle to ensure accountability, non-repudiation, and compliance with Circular 50/2024/TT-NHNN.

Key elements shown in the image are organized into logical sections:

1. Transaction Context (Sender, Transaction, Receiver Information)
The top section provides a summary of the business context for the transaction being audited. This includes:

- Sender Information: Name (Nguyen Van A), Account Number (3232321), and Bank (Vietcombank).
- Transaction Information: Amount (10 VND), Customer Type (Business Customer), and Method/Description (Mua vé máy bay - flight ticket purchase).
- Receiver Information: Name (Tran Thi B), Account Number (9876543210), and Bank (Asia Commercial Bank).

  This context anchors the subsequent audit events to a specific, identifiable business transaction.

2.Transaction History Timeline
This is the core audit trail, presented as a chronological sequence of actions. Each step in the transaction authorization flow is logged as a distinct.

This timeline directly corresponds to the data stored in the TRANSACTION_LOG and ACTION_LOG database tables. It provides clear evidence of:

- What action was performed.
- Who performed it (the authenticated user martin).
- When it occurred (with millisecond precision).
- What the outcome was (Success).

3. Confirmation and Device Binding Information
The bottom section captures the final confirmation details, establishing a strong link between the transaction, the user, and the specific trusted device used for approval:

- Device ID: 6ac111f9-0668-4281-95e3-77547a399feb – Uniquely identifies the physical device that authorized the transaction. This directly supports the device binding requirements of Circular 50.
- Application ID: 340E545E-EA37-4C93-B76A-FFA82669A686 – Identifies the client application (e.g., the mobile banking app).
- Performed By: martin – The user account that executed the final approval.
- CIF Number: 020212026126 – Links the action to the core banking customer record.
- Confirmation Time: 11/04/2026 09:54:01 – The exact moment the transaction was finalized.

- Authentication Method: Smart OTP – The specific factor used for strong customer authentication (SCA).
- OTP Verification: 59040356 – The one-time password code that was validated.
- Status: Success – The final outcome of the authorization.