IAD (Issuer Application Data) — In-Depth Explanation

Issuer Application Data (IAD) is a tag (9F10) in EMV transactions, especially relevant in EMV contact and contactless cards. It contains issuer-defined data that helps verify transaction authenticity, apply risk management, and control the cryptogram generation and validation process.

🔹 High-Level Overview

- **Tag**: 9F10
- **Purpose**: Contains issuer-specific data used for:
    - Cryptogram generation/validation
    - Risk management
    - CVR (Card Verification Results)
    - Issuer action code decision making
- **Length**: Variable; often 32–36 hex characters (~16–18 bytes)

🔹 When and How is IAD Used?

📌 During EMV Transaction Flow:

1. **Generated by the card** during **first GAC** (GENERATE AC – Application Cryptogram).
2. **Returned to the terminal** in the GAC response.
3. **Sent to issuer** in the authorization request (ARQC).
4. **Issuer parses** IAD to:
    - Inspect card behavior (via CVR)
    - Decide online approval or decline
5. **Issuer may modify** some fields and send updated IAD in issuer scripts or ARPC.

🔹 Typical Format

IAD format is **issuer-specific**, but most follow **Visa**, **Mastercard**, or **Amex** specs. Below is a generic Visa-style IAD structure:

Field	**Bytes**	**Description**
Cryptogram Version No.	1	Indicates CVR layout format
CVR (Card Verification Results)	3–6	Bitwise flags that describe card decisions
Unpredictable No.	4	Random used in cryptogram generation
ATC (Application Transaction Counter)	2	Count of EMV transactions
Padding / Proprietary	Varies	Additional fields or padding by issuer/card

---

◆ CVR (Card Verification Results) — The Heart of IAD

The **CVR** inside the IAD is what tells the issuer what happened at the card level.

For example (Visa CVR – 3 bytes):

Byte	**Bit**	**Meaning (example)**
1	8	Offline PIN tried
1	7	Offline PIN successful
1	6	CVM unsuccessful
1	5	SDA failed
2	8	CDA failed
2	7	ARQC was generated
2	6	Unable to go online
3	8	Offline data authentication failed

Each card brand defines its own CVR layout, and it can be 3–6 bytes.

◆ Example Dump

Let’s analyze a sample IAD:

ini

CopyEdit

```
9F10  = 0E060112A00000010A01000000000000
```

- **0E**: Length = 14 bytes
- **06**: Cryptogram version
- **01 12 A0 00 00 01 0A 01 00 00 00 00 00 00** – this is the remaining IAD (usually includes CVR and ATC)

You need card brand documentation (like Visa's VCPS or Mastercard's M/Chip spec) to fully decode it.

◆ How IAD is Used in the Flow

✅ During First GAC (ARQC):

- Card creates IAD
- Includes:
    - Cryptogram Version
    - CVR (what card experienced)
    - ATC
    - Unpredictable number
- This is sent to terminal → then to issuer

🛡️ Issuer Validation:

- Parses IAD
- Reads CVR:
    - Was PIN entered? Correct?
    - Did CDA fail?
    - Is DDA skipped?
- Uses this info to determine:
    - If ARQC is valid
    - Whether to send ARPC or decline

🔄 During GAC2 (TC or AAC):

Issuer may send IAD back (optional) — updated with issuer's decision or additional flags.

🔍 Summary

Aspect	**Details**
Tag	9F10
Contains	Cryptogram Version, CVR, ATC, randoms
Created	By card, during first GAC
Purpose	Inform issuer of card decisions
Parsed by	Issuer host to approve/decline
Variable?	Yes — structure & content vary by brand

If you share your **card scheme (Visa/Mastercard)** and a real IAD dump, I can break it down line-by-line. Would you like that?

You said:

sample IAD value

ChatGPT said:

IAD (Issuer Application Data) – In-depth Explanation

The **Issuer Application Data (IAD)** is a field included in the EMV transaction process that carries issuer-specific data. It's a **proprietary data object** (tag **9F10**) created by the **issuer** and sent to the card during **GAC 1** (Generate AC – First) and/or **GAC 2** (Generate AC – Second) command responses.

🔹 Tag Details

- **Tag:** 9F10
- **Length:** Variable (commonly 32–48 bits, i.e., 4–6 bytes)
- **Source:** Issuer-defined format

- **Direction:** From card to terminal (usually in response to GAC 1 and GAC 2)
- **Used in:** Host authorization, risk management, cryptogram verification

---

◆ Purpose of IAD

The IAD contains data used by the **issuer's host system** to:

- Perform cryptographic validation
- Determine transaction risk
- Decide on further action (approve/decline)
- Identify which functions were supported and executed on the card (e.g., SDA, DDA, CDA)
- Check offline PIN results
- Check CVM results
- Validate ARQC or TC

◆ IAD Format

The structure of the IAD is **issuer-defined**, but most issuers and terminal vendors follow the **Common Core Definition (CCD)** or **Visa/Mastercard-defined formats**. These include fields such as:

Field	**Purpose**
Cryptogram Information Data (CID)	Type of cryptogram (ARQC, AAC, TC)
Application Transaction Counter (ATC)	Number of transactions completed
CVR (Card Verification Results)	Summary of card-side checks (PIN, risk, etc.)
Issuer Discretionary Data	Issuer-defined fields (custom logic, flags)

◆ Sample IAD Value Breakdown

Let's take a sample IAD value:

CopyEdit

9F10 12 06011203A0B800000000000000000000

Breakdown:

- 9F10 → Tag
- 12 → Length (18 bytes)
- 06 → Cryptogram Format Version Number
- 01 → Cryptogram Information Data (e.g., ARQC, TC, etc.)
- 12 → Application Transaction Counter (ATC) – part 1
- 03A0 → CVR (Card Verification Results)
- B800000000000000000000000000 → Issuer discretionary / padding

🔍 **CVR 03A0** might mean:

- Bit 8: Offline PIN performed
- Bit 6: CDA supported
- Bit 3: No issuer authentication

Note: Exact interpretation of each field requires issuer documentation.

🔸 When is IAD Used?

- **Card returns IAD in response to:**
    - GAC 1 (after Terminal Risk Management)
    - GAC 2 (after Online Authorization, if performed)
- **Issuer reads IAD:**
    - To verify card behavior
    - To confirm PIN validation result
    - To verify cryptogram

🔹 Real-Life Example Dump

Here’s a sample snippet from a GAC response APDU:

yaml

CopyEdit

```
77 2A
```

```
9F27 01 80       ; CID = 80 = ARQC

9F10 12 06011203A0B800000000000000000000

9F36 02 0012     ; ATC = 0012

9F26 08 2B9E8495A32F3991 ; Cryptogram
```

- 9F10 12 06011203A0… → IAD (explained above)
- 9F27 → CID (indicates type of cryptogram)
- 9F36 → ATC
- 9F26 → ARQC cryptogram

◆ Summary

- **IAD** is a compact, issuer-controlled field sent by the card to the terminal/issuer host.
- It provides **internal status and results** from card-side checks.
- **Decoding IAD** requires either:
  - CCD (Common Core Definition) knowledge
  - Issuer-specific documentation
- It is **crucial** for host-side risk management and decision-making in EMV.