QR code scanner v5

```
graph TD
    %% QR Code Generation
    A[QR Code Generation System] --> B[Cryptographic Signature Engine<br>RSA/ECC Signing]
    B --> C[AI Visual Watermark Generation<br>Generative AI Model]
    C --> D[Physical QR Code<br>Content + Signature + Visual Indicator]

    %% QR Code Scanning & Validation (Physical)
    D -->|QR Exchange Protocol| E[Physical QR Scanner/App]
    E --> F[Cryptographic Signature Validation]
    F --> G[AI Visual Watermark Validation]
    G --> H[QR Payload Validation via AI Backend Security Agent]

    %% Validation Outcome
    H --> I{All Validations Passed?}
    I -->|Yes| J[Display Trust Indicator ✅ in Scanner]
    I -->|No| K[Display Warning Indicator ❌ in Scanner]

    %% Node Styling
    style A fill:#BBDEFB,stroke:#0D47A1,stroke-width:2px
    style B fill:#D1C4E9,stroke:#5E35B1,stroke-width:2px
    style C fill:#B3E5FC,stroke:#0288D1,stroke-width:2px
    style D fill:#DCEDC8,stroke:#689F38,stroke-width:2px
    style E fill:#FFF9C4,stroke:#FBC02D,stroke-width:1px
    style F fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style G fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style H fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style I fill:#FFE0B2,stroke:#FB8C00,stroke-width:1px
    style J fill:#C5E1A5,stroke:#7CB342,stroke-width:1px
    style K fill:#FFCDD2,stroke:#E53935,stroke-width:1px
```

```
graph TD
    %% QR Code Generation
    A[QR Code Generation System] --> B[Cryptographic Signature Engine<br>RSA/ECC Signing]
    B --> C[AI Visual Watermark Generation<br>Generative AI Model]
    C --> D[Digital QR Code<br>Content + Signature + Visual Indicator]

    %% Digital QR Scanning & Validation
    D -->|QR Exchange Protocol| E[Digital QR Scanner/App]
    E --> F[Cryptographic Signature Validation]
```

```
    F --> G[AI Visual Watermark Validation]
    G --> H[QR Payload Validation via AI Backend Security Agent]

    %% Validation Outcome
    H --> I{All Validations Passed?}
    I -->|Yes| J[Display Trust Indicator ✅ in Scanner]
    I -->|No| K[Display Warning Indicator ❌ in Scanner]

    %% Updating Digital QR based on Validation
    J -->|QR Exchange Protocol| L[Update Digital QR Code Display]
    K -->|QR Exchange Protocol| L
    L --> M[Show Trust Indicator at center of Digital QR ✅ / ❌]
    L --> N[Show Last Verification Timestamp<br>in whitespace around Digital QR]

    %% Node Styling
    style A fill:#BBDEFB,stroke:#0D47A1,stroke-width:2px
    style B fill:#D1C4E9,stroke:#5E35B1,stroke-width:2px
    style C fill:#B3E5FC,stroke:#0288D1,stroke-width:2px
    style D fill:#DCEDC8,stroke:#689F38,stroke-width:2px
    style E fill:#FFF9C4,stroke:#FBC02D,stroke-width:1px
    style F fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style G fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style H fill:#E1F5FE,stroke:#03A9F4,stroke-width:1px
    style I fill:#FFE0B2,stroke:#FB8C00,stroke-width:1px
    style J fill:#C5E1A5,stroke:#7CB342,stroke-width:1px
    style K fill:#FFCDD2,stroke:#E53935,stroke-width:1px
    style L fill:#B2DFDB,stroke:#00796B,stroke-width:1px
    style M fill:#DCEDC8,stroke:#689F38,stroke-width:1px
    style N fill:#FFF59D,stroke:#FDD835,stroke-width:1px
```

```
sequenceDiagram
    participant QG as QR Generator System
    participant CSG as Cryptographic Signature Generator
    participant CSV as Cryptographic Signature Validator
    participant QS as QR Scanner/App

    %% Cryptographic Signing Process
    rect rgb(232,245,233)
    QG->>CSG: Request Signature (Payload + Timestamp)
    CSG->>CSG: Compute Hash (Payload + Timestamp) using SHA-256
    CSG->>CSG: Sign Hash using Private Key (RSA/ECC)
```

```
    CSG-->>QG: Return Digital Signature (base64 encoded)
    end

    %% QR Code Generated
    QG->>QG: Embed Digital Signature into QR Code

    %% Signature Validation (During Scanning)
    QS->>CSV: Validate Digital Signature (Payload + Timestamp + Signature)
    CSV->>CSV: Extract Payload, Timestamp, Signature from QR
    CSV->>CSV: Compute Hash (Payload + Timestamp) using SHA-256
    CSV->>CSV: Verify Hash with Public Key (RSA/ECC)
    alt Signature Valid
        CSV-->>QS: "Signature Verified ✅"
    else Signature Invalid
        CSV-->>QS: "Signature Invalid ❌"
    end
```

```
sequenceDiagram
    participant QG as QR Generator System
    participant VWG as AI Visual Watermark Generator
    participant VWV as AI Visual Watermark Validator
    participant QS as QR Scanner/App

    %% AI Visual Watermark Generation
    rect rgb(227,242,253)
    QG->>VWG: Request AI-generated Visual Watermark (Signed Payload)
    VWG->>VWG: Extract unique features from Signed Payload
    VWG->>VWG: Generate unique, hard-to-replicate watermark (colors/patterns/
icons) using GAN/AI model
    VWG->>VWG: Encode watermark visually around QR edges
    VWG-->>QG: Return QR Code with Embedded AI Visual Watermark
    end

    %% AI Visual Watermark Validation (During Scanning)
    QS->>VWV: Validate AI Visual Watermark from scanned QR
    VWV->>VWV: Extract visual watermark features (colors/patterns/icons) from QR
edges
    VWV->>VWV: Compare extracted features with expected AI-generated features
    alt Watermark Matches Expected Features
        VWV-->>QS: "Visual Watermark Valid ✅"
    else Watermark Mismatch Detected
        VWV-->>QS: "Visual Watermark Invalid ❌"
    end
```

```
sequenceDiagram
    participant QS as QR Scanner/App
    participant AI as AI Backend Security Agent
    participant DB as Threat Intelligence DB

    %% QR Payload Validation
    rect rgb(255,249,196)
    QS->>AI: Request Payload Validation (e.g., URL/content)
    AI->>AI: Apply heuristics (e.g., ML model analysis)
    AI->>DB: Check Payload against Phishing DB, Blacklists, Reputation DB
    DB-->>AI: Return Threat Analysis Result
    alt Payload Safe
        AI-->>QS: "Payload Safe ✅"
    else Payload Unsafe or Suspicious
        AI-->>QS: "Payload Unsafe ❌ (Reason provided)"
    end
    end
```

A. Secure Digital Signature Embedded QR Codes

Concept:

Embed a digital signature within the QR code payload using asymmetric cryptography (e.g., RSA or ECC). The digital signature guarantees the authenticity and integrity of QR code contents.

How it works:

- **QR Payload Structure:**
  [Content/URL] + [Timestamp] + [Digital Signature (RSA/ECC signed with Private Key)]
- **Cryptographic Signing Process:**
  - QR Generator System hashes the QR payload and timestamp using SHA-256.
  - Cryptographic Signature Generator signs this hash securely using the private key.
- **Validation Process (at scanning):**
  - QR Scanner extracts the payload, timestamp, and embedded digital signature.
  - Cryptographic Signature Validator recomputes hash and verifies signature using the corresponding public key.

Scanning Compatibility:

- **Standard QR scanners:**
  Can scan QR codes normally, seeing plaintext URL or text without obstruction.
- **Specialized Verification Apps:**

Perform cryptographic signature validation, instantly confirming QR code authenticity and integrity.

B. Visible Trust Indicator with AI-Generated Visual Watermark

Concept:

Integrate a dynamic, AI-generated visual watermark (unique colors, patterns, or icons) around QR code edges, making forgery extremely difficult. The watermark visually signifies authenticity linked directly to cryptographic data.

Implementation Details:

- **AI Visual Watermark Generation:**
  - Input: Cryptographically signed payload (content + timestamp + signature).
  - Output: Unique, hard-to-replicate visual watermark (GAN-generated distinct colors, patterns, icons) encoded around QR edges.
  - AI model ensures watermark uniqueness tied explicitly to the QR payload/signature combination.
- **Visual Watermark Validation (at scanning):**
  - QR Scanner/App extracts visual watermark features from QR edges.
  - AI Visual Watermark Validator compares extracted watermark features to AI-generated expected features.
  - Validates authenticity by confirming exact match.

User Benefits:

- Users visually identify trustworthy QR codes through distinct patterns without the need for initial scanning.
- Attackers cannot easily replicate the watermark due to cryptographic keys and AI watermark generation complexity.

C. AI-Powered Backend Security Agent for Real-Time Threat Detection

Concept:

Leverage an AI-powered security agent to provide real-time validation of QR payloads upon scanning. The AI agent identifies threats, misuse patterns, and emerging risks.

Implementation Details:

- **Real-Time AI Validation (at scanning):**
  - QR Scanner/App sends the QR payload (URL or content) to the AI Backend Security Agent.
  - AI Backend performs immediate threat validation:
    - Checks payload against updated phishing databases.
    - Verifies against malicious URL blacklists.
    - Applies heuristic and ML-driven analysis to identify suspicious or malicious patterns.

- **Validation Outcome:**
  - Provides instant feedback indicating safety or potential risks of QR code content.

Validation Flow:

- Validation performed server-side at the moment of scanning, ensuring real-time security assessment.
- Results are communicated instantly back to the scanner/app, enabling immediate visual trust indicators (green tick ✅ or red cross ❌).

Digital QR Codes Enhanced Validation Flow:

- **Upon successful validation:**
  - The trust indicator (✅) is visually embedded at the center of the digital QR code.
  - A timestamp of last successful verification is visibly displayed in the whitespace surrounding the QR.
- **Upon failed validation:**
  - A visual warning indicator (❌) is displayed clearly at the QR code's center.
  - The timestamp for last verification attempt is still displayed visibly.

Novelty:

1. Combined Cryptographic and AI-Based Visual Validation

- **First-of-its-kind integration** of asymmetric cryptography (RSA/ECC digital signatures) with AI-generated visual watermarks.
- Embeds cryptographic signatures directly within QR payload, securing content integrity and authenticity at scanning.

2. AI-Generated, Hard-to-Replicate Visual Watermarks

- Employs advanced Generative AI (GAN-based models) to produce unique visual watermarks (patterns/colors/icons).
- Watermarks visually encoded around QR edges, directly tied to cryptographic signatures, greatly reducing forgery risk.

3. Real-Time AI-Driven Threat Assessment

- Utilizes an AI Backend Security Agent that performs dynamic threat validation upon QR code scanning.
- Incorporates real-time heuristic checks, phishing DBs, blacklists, and ML-driven suspicious behavior detection.

4. Dynamic Trust Indicators within Digital QR Codes

- Implements real-time feedback via QR Exchange Protocol, dynamically updating digital QR codes post-validation.
- Displays validation results centrally (green tick/red cross) and

timestamps visibly around digital QR, enhancing user trust.

Why Novel?
This invention uniquely merges cryptographic security with AI-generated visual authenticity indicators, enabling **visual pre-validation** of QR codes, combined with **real-time AI threat assessment**, creating robust multi-layered security not currently available in standard QR-based solutions.

QR Exchange Protocol

Definition:

The **QR Exchange Protocol** is a secure communication standard used between QR scanners and QR codes, designed for real-time cryptographic and AI-based validation. It dynamically adapts as:

- **Unidirectional** (Physical QR): Scanner validates without feedback to QR.
- **Bidirectional** *(Digital QR)*: Scanner validates and updates QR visually, reflecting trust status and verification timestamp in real-time.

📤 Protocol Message Structure

✅ Request Message (Scanner → QR code / Backend)

Field	Description	Example
QR_Payload	QR code embedded content	https://trusted-domain.com/info
Timestamp	Original generation timestamp	2025-03-17T14:00Z
Signature	Digital signature (RSA/ECC)	Base64 encoded signature
Watermark Data	Extracted visual watermark features	Color patterns/icons features
Scanner ID	(Optional) Scanner app/device identifier	ScannerApp_v1.0

✅ Response Message (Backend → Scanner)

Field	Description	Example
Signature Status	Result of cryptographic signature verification	Valid / Invalid
Watermark Status	Result of AI visual watermark validation	Valid / Invalid

Payload Security	AI Backend payload validation result (Phishing DB, etc.)	Safe, Unsafe: phishing detected
Overall Status	Final combined validation status	Trusted ✅, Untrusted ❌
Verification Time	Timestamp of validation performed	2025-03-17T15:42:30Z
Recommended Action	Optional action/ instruction if validation fails	Block, Proceed with caution

🖥️ Dynamic Digital QR Update (Scanner → Digital QR Display)

(Applicable only for Digital QR scenario)

Field	Description	Example
Validation Status	Trust Indicator to display centrally on QR	✅ or ❌
Verification Time	Timestamp to display visibly around QR	2025-03-17T15:42 UTC

🖥️ Usage: Physical vs. Digital QR

Type	Communication Flow	Direction	Dynamic Visual Update
Physical QR	Uni-directional (Scanner only validates)	❌ (No QR update)	
Digital QR	Bi-directional (Scanner ↔ Digital QR)	✅ (QR updated dynamically)	

- **Physical QR**: Scanner validates internally; no changes made to QR code.
- **Digital QR**: QR visually updated via protocol messages to reflect trust indicators centrally and verification timestamps around QR edges.