"Aadhaar Face Authentication onboarding Audit checklist version 1.0" for certifying compliance with controls that the entity (AUA/KUA/Sub-AUA/Sub-KUA) is required to have in place Version 1.0 [issued in January 2025]

Control no.	Domain	Control description	Compliance status	Auditor's observations	Management comments (to be provided by RE)
A	**Access Control**				
AC.1	Access Authorization Mechanism	Verify that only authorised individuals are able to access information facilities such as the authentication application, audit logs, authentication servers, application, source code, information security infrastructure, etc., and Aadhaar processing related information. Insufficient authentication and authorization can lead to unauthorized access and misuse of sensitive functionality or data.			
AC.2	Privilege Access Mechanism	Check for privileges or bypass authorization controls by manipulating input parameters, changing user roles, or tampering with authorization tokens.			
AC.3	Multiple Systems Access	Ensure that one concurrent session per user is allowed at a time i.e., parallel login sessions are not allowed.			
AC.4	Account Lockout Mechanism	Check if app implements account lockout mechanisms after three successive log-in failures result in the user account being locked to prevent brute force attacks. End users / operators should not be able to login			

		until their account is unlocked and the password is reset.			
AC.5	Aadhaar Data Access	Check for effective access controls to Aadhaar and Aadhaar related data.			
AC.6	Role Based Mechanism	Check for defined role based access controls.			
AC.7	Dynamic Access Authorization	Check for implementation of dynamic access based control to authorize, limit or forbid access and strengthen authentication.			
AC.8	Assisted Device login Mechanism	Test in case of assisted devices and applications where operators need to mandatorily perform application functions, that the operator is authenticated using a multi-factor authentication scheme, such as user id, password, Aadhaar authentication, answer to personal security questions, soft token, hard token, one-time password, voice recognition, biometric data match and PIN.			
AC.9	Password management Guidelines	Test the app's authentication mechanism by attempting to create accounts with weak passwords, such as simple or common passwords.			
AC.10	Jailbreak and root detection	Implement control to detect and block access from jailbroken or rooted devices.			
B	**Algorithmic Security**				
AS.1	Resiliency against adversarial attacks	Check whether algorithms are optimized for performance (refresh rates, SSL pinning, low response time etc..) and resilience against adversarial attacks.			
AS.2	Reverse engineering	Use obfuscation tool , Detect and prevent debugging attempts			

AS.3	Restriction on designing/compiling malicious code	Entity personnel should not intentionally write, generate, compile, copy or attempt to introduce any computer code designed to damage or otherwise hinder the performance of, or access to, any Aadhaar information.			
AS.4	Hardcoding of security keys	Verify to ensure that the passwords, tokens, security keys and licenses are not hardcoded in the application code.			
AS.5	Cryptographic algorithmic security	Verify that app does not use cryptographic protocols or algorithms that are widely considered depreciated for security purposes.			
AS.6	Cryptographic protocol configuration	Verifying the Configuration of Cryptographic Standard Algorithms.			
C	**Audit Logging**				
AL.1	Forensic investigation of logs	Verify that login attempts or access violations are logged for monitoring and forensic investigations.			
AL.2	Log access mechanism	Verify that logs can only be accessed via authorized users.			

AL.3	Log process automation	All log transfer and processing must be automated.			
AL.4	Log data storage security	Check for use of secure methods such as digital signatures, cryptographic hashes or write once storage to ensure integrity of log data.			
AL.5	Sensitive Information Logging	Avoid logging of sensitive information to prevent inadvertent exposure of confidential information.			
AL.6	Vulnerability scanning and penetration testing	Conduct regular vulnerability scanning and Penetration testing for mobile and backend systems.			
D	**Authentication Flow**				
AF.1	Defense mechanism against replay attacks	Check strengthening against replay attacks and bypass techniques.			
AF.2	Use of encryption	Check for encryption during transmission between mobile device and information system.			
AF.3	Authorization check for sensitive data	Check app's API endpoints to ensure proper authorization checks are performed before accessing sensitive data or performing privileged operations.			
AF.4	Use of rate limiting	Check for rate limiting.			
AF.5	Safeguarded credential transmission	Check for user credential safeguarding during transmission.			

AF.6	Use of secure logging methods	Check for secure logout functionality for session termination.			
AF.7	Security header implementation	Check for utilization of security headers such as HSTS, CSP and X content type options.			
E	**Data Storage & Protection**				
DS.1	Data access mechanism	Check for data integrity and confidentiality (access control and authentication mechanism).			
DS.2	Encryption of data	Check for encryption for data at rest and in transit.			
DS.3	Data loss prevention	Check for DLP tool implementation.			
DS.4	Key management practice	Check for strong cryptographic algorithms and key management practices for sensitive, PII data.			
DS.5	Anonymization of secret data	Check for anonymization of secret data such as passwords, authentication tokens.			
F	**Error Handling**				
EH.1	Restriction of malicious code injection	Check for validation errors that could allow attackers to inject malicious code into the software application.			
EH.2	Restriction on display of sensitive data	Check for validation errors that could allow attackers to view sensitive data that is not intended to be displayed.			

EH.3	Sensitive information exposure	Analyze how the app handles errors and exceptions. Ensure that error messages and stack traces do not reveal sensitive information			
G	**Input Validation**				
IV.1	User input encoding	Verify that the application is properly encoding user-supplied input to prevent cross-site scripting (XSS) and other injection attacks such as LDAP (Lightweight Directory Access Protocol), SQL injection, etc.			
IV.2	Secure Application Development	Follow OWASP Mobile Security Project guidelines to build the mobile Application.			
IV.3	Input validation and sanitization	Check for proper input validation and sanitization on all inputs, including form fields, Facial Data, URLs, and APIs.			
IV.4	Input length check	Test for proper input length checking to prevent buffer overflows and other memory-related attacks			
IV.5	Code and log review	Review code and logs for any signs of improper input validation or exploitation attempts and conduct code reviews and automated testing to identify vulnerabilities.			
IV.6	Code related attack execution	Test for proper type checking to prevent the execution of arbitrary code and other code-related attacks.			
H	**Logging & Monitoring**				
LM.1	Monitoring of security logs	Check for application and transaction security logs retention			

LM.2	Security incident monitoring	Check for monitoring mechanism, log analysis			
LM.3	Security incident monitoring	Check for SIEM utilization for log management and analysis			
LM.4	Use of scalable monitoring solution	Check for efficacy and completeness of monitoring mechanism to record relevant security events			
LM.5	Regular Updates	Apply security patches to the mobile app and backend infrastructure at regular interval			
I	**Security Configuration**				
SC.1	Identification of media type format	Verify for Content-Type Header modification			
SC.2	Use of encryption for secure communication	Check for HSTS Header presence			
SC.3	Use of masquerading for malicious file execution	Check whether the application blocks files with multiple file extensions (filename.txt.exe).			
SC.4	Platform interaction in use	Web Views are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.			
SC.5	Use of secure communication protocol on network	Check for use of SSL for transmission of all sensitive data			

SC.6	Backend server security	Check for Server Side Request Forgery (SSRF)			
SC.7	Presence of common code vulnerability	Check for Cross site Request Forgery (CSRF)			
SC.8	Security Configuration	Review presence of Weak Cipher			
J	**Session Management**				
SM.1	Session log generation	Check app's session logs to ensure that session tokens or cookies are properly generated			
SM.2	Session invalidation against attacks	Check invalidation against session hijacking or fixation attacks			
SM.3	Unique Session Identifier	Check for reuse of session identifier			
SM.4	Idle session timeout	Entity to ensure that end-point devices used for developing, process and handling Aadhaar data and application timeout after a session is idle for more than 15 to 30 minutes, based on the criticality of the application.			
SM.5	Session logout mechanism	Ensure that session should not remain valid on server end after log out			
K	**Third Party Integration**				
TP.1	Testing for app permissions	Verify that app only requests the minimum set of permissions necessary			
TP.2	Sanitization of inputs from external sources	All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC			

		mechanisms such as intents, custom URLs, and network sources.			
TP.3	Export mechanism via custom URL	Verify app does not export sensitive functionality via custom URL schemes without proper protection.			
TP.4	Export mechanism via IPC facilities	The app does not export sensitive functionality through IPC facilities without proper protection.			
TP.5	Exposure of native methods	If native methods of the app are exposed to a WebView, that WebView only renders JavaScript contained within the app package.			
TP.6	Object Serialization	Object serialization, if any, is implemented using safe serialization APIs.			
TP.7	Third party API weaknesses	Check for Weaknesses in Third Party Libraries Verify that integration with third party API done securely or SDKs used in mobile app			
TP.8	Security Operations Centre	Check for anomaly detection			
TP.9	Use of security communication protocol	Entity to ensure message security and integrity between their servers and those of third party entities			

Note: The Audit shall only be conducted by a CERT-In empanelled Information Security Auditing Organisation, the list of which is published on the website of CERT-In.