India's "Aadhaar" Biometric ID: Structure, Security, and Vulnerabilities

Pratyush Ranjan Tiwari*[1], Dhruv Agarwal*[2], Prakhar Jain[3], Swagam Dasgupta[4], Preetha Datta[5], Vineet Reddy[6], and Debayan Gupta[7]

[1] Johns Hopkins University, [2] Microsoft Research, [3] Fractal Analytics, [4] Bastion Media, [5] Aalto University, [6] Northeastern University, [7] Ashoka University

Abstract. India's Aadhaar is the largest biometric identity system in history, designed to help deliver subsidies, benefits, and services to India's 1.4 billion residents. The Unique Identification Authority of India (UIDAI) is responsible for providing each resident (not each citizen) with a distinct identity—a 12-digit Aadhaar number—using their biometric and demographic details. We provide the first comprehensive description of the Aadhaar infrastructure, collating information across thousands of pages of public documents and releases, as well as direct discussions with Aadhaar developers. Critically, we describe the first known cryptographic issue within the system, and discuss how a workaround prevents it from being exploitable at scale. Further, we categorize and rate various security and privacy limitations and the corresponding threat actors, examine the legitimacy of alleged security breaches, and discuss improvements and mitigation strategies.

Keywords: Resident Identification · Biometric · Security & Privacy

1 Introduction

Resident identification systems are pervasive in the world today, with many using biometrics [15]. These systems hold and mediate vast amounts of private data, which in many cases is also used to facilitate welfare schemes and other public programs. Aadhaar is a 12-digit unique ID issued by the Indian government to each Indian resident (not citizen), using their demographic and biometric information. To date, over 1.3 billion residents have been enrolled [35]: it is the largest biometric identity system ever built and is linked to bank accounts, income tax numbers, social security schemes, etc. And while Aadhaar is technically not required for many things (such as getting a new cellular connection), its ubiquity has rendered it the default form of identification in India.

Though public trust in Aadhaar is crucial, the system has been relatively opaque, leading to much confusion and speculation. Civil activists [4] and media outlets [42] have alleged that Aadhaar is vulnerable to numerous types of breaches; corroborating these claims is difficult as there exists no comprehensive

* Indicates equal contribution. [1]pratyush@cs.jhu.edu, [2]t-dhaga@microsoft.com

resource detailing Aadhaar's system and security architecture. Public documentation about Aadhaar is outdated or ambiguous, and *no unified description of the infrastructure exists.* As a result, one has to collate information from multiple (often unreliable) sources. We present the first comprehensive description of Aadhaar, analyze all reported privacy or security breaches, and assess defenses against future attacks. We also report the first known[1] cryptographic issue (fortunately *not* exploitable at scale under current conditions) in the system.

Contributions *Comprehensive snapshot:* We outline the journey of an individual's data through the Aadhaar system and the entities involved (for data collection, processing, storage, and usage), covering the entire body of publicly available information on Aadhaar. Previous work has looked at authentication or verification, etc. [4,31], but none have covered the whole infrastructure.

Security flaws: We analyze all documentation made public by UIDAI — trawling through thousands of pages over time — as well as all alleged attacks to compile and analyze possible security issues. We find that the way Aadhaar generates IVs for AES (it uses AES-GCM) opens up the possibility to mount an identity forgery attack and steal data. We note that *the attack is not currently deployable*: we have made sure that this is not exploitable before publishing. However, any batching of queries or capture of multiple messages within the same second may still render the system insecure. Specifically, one could forge the identity of any individual whose Aadhaar number is available[2].

1.1 Paper Overview

Section 2 provides a brief background and discusses related work. A list of all abbreviations, in order of appearance, is provided in Appendix B. Section 3 describes Aadhaar's infrastructure in detail (along with data privacy and security policies)[3] This snapshot is divided into the following main sections: the Enrollment Ecosystem (Section 3.1), the Authentication Ecosystem (Section 3.2), the Central Identities Data Repository or CIDR (Section 3.3). Section 4 details the security of different endpoints at which an individual's data is vulnerable to attacks. Section 5 discusses information security in Aadhaar, using standard benchmarks. We define the threat model and discuss a cryptographic flaw we identified and its mitigation strategies (5.2). We use the threat model along with the snapshot, in Section 6, to filter legitimate attacks from our database of media allegations (Section 6.1). We discuss possible attacks, categorize the feasibility of these breaches based on the threat actor involved, cost (time and resources) and the level of security provided by Aadhaar (Section 6.2). Section 6.3 discusses

[1] Media reports have alleged flaws in associated organizations, or engineering/policy flaws (e.g., software bugs), but a cryptographic flaw within the Aadhaar infrastructure itself has never been discussed.

[2] Collections of Aadhaar numbers have been leaked at various times by multiple organizations, though never by UIDAI itself.

[3] We collate information from myriad technical reports, policy documents, Memoranda of Understanding (MoUs), and circulars published and signed by UIDAI and other organizations in Aadhaar infrastructure. We archive these reports here.

technical and structural mitigation strategies for each type of breach. A study of alleged attacks is provided in supplementary analysis C.

2 Background

The Unique Identification Authority of India (UIDAI) was established in January 2009. Its mission was to issue a unique identification (UID) number, an "Aadhaar Number," to every resident of the country. The UID's purpose was to be a one-stop identification that is eventually linked to every social service to make the disbursement of welfare services effective and efficient (by reducing leakages). The bill that provides legal backing to Aadhaar is called the "Aadhaar (Targeted Delivery of Financial and other Subsidies, benefits and services) Act." Apart from providing Indian residents with a unique identity (an Aadhaar number), the UIDAI is also responsible for providing a platform for residents to authenticate their physical presence [65] at a point of service. Aadhaar's policies regarding its vision, ethical implications, data security, and privacy have been under intense scrutiny [21]. This becomes all the more important with Aadhaar's ubiquity. It is different from login.gov [5,11], for example. It is not merely a single point of contact system for welfare. Aadhaar is what you can use to get on a plane, to open a bank account, to get a phone connection. Getting tested or vaccinated for COVID-19? Aadhaar. It is MOSIP [41] on steroids: closed-source, universal, and practically (although not officially) mandatory.

2.1 Related Work

National identification projects of many countries have attracted considerable academic research — Jamaica's attempt [34], Nepal's National Identity Project (NIDP) [3], UAE's ID system [6], Europe's e-ID systems [9], United States' Social Security Number [18], etc. Being the world's largest biometric ID system, India's Aadhaar has been an active research topic in the areas of ICTD, HCI, security, and privacy. Singh and Jackson [36] perform an ethnographic study of Aadhaar. They find exclusion of people in various phases: during enrollment, while authenticating, and while linking ("seeding") their Aadhaar numbers with existing public welfare databases (like the Public Distribution System database). Srinivasan and Johri [37] draw similarities between the legitimization and support tactics of Aadhaar and previously successful infrastructure projects like railroads in British India and dams in post-Independence India.

Prior security and privacy works have recommended using a Trust and Role-Based Access Control Model for internal Aadhaar processes and using cryptography to prevent illegal tracking and profiling [31]. Rajput and Gopinath [32] have analyzed the privacy of authentication workflows offered by Aadhaar and recommended new ones. The work of Agrawal, Banerjee and Sharma [4], though relatively informal, is the closest to ours. It provides a broad analysis of Aadhaar's vulnerabilities like faking biometrics, identification without consent, and illegal tracking by collation of data across service providers. Our work differs

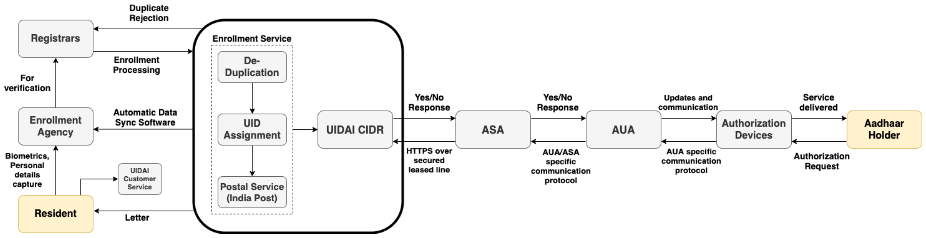

Fig. 1: Flowchart of Aadhaar's architecture. Yellow cells depict entry points into the enrollment (left) and authentication (right) ecosystems. Enrollment starts with the resident visiting the Enrollment Agency which uses an enrollment software provided by the Enrollment Service. The data is then sent to the Registrars for verification. If de-duplication succeeds, the data is stored in the CIDR and the user is enrolled. The authentication procedure starts with the Aadhaar holder's information reaching the CIDR via AUA and ASA. The biometric data is captured by the authorization devices, sent to the CIDR through AUA and ASA. The response is sent back by the CIDR via the same route.

from these: we present a detailed overview of the system and do not assume the correctness of media allegations and activism (which are essential in their own right). Instead, we analyze Aadhaar's security and allegations against it based on an extensive study of available documentation.

3 Snapshot: Aadhaar System Design

Aadhaar has three primary components: (1) the *Enrollment* ecosystem, (2) the *Authentication* ecosystem, and (3) the *CIDR* (Central Identities Data Repository). Enrollment handles onboarding and assigning of unique identity numbers. Authentication provides verification services when residents want to prove their identity. CIDR is a database that stores the collected biometric and demographic data. We provide an overview of a typical resident's interaction with the Aadhaar system and then discuss its usability and the three components.

System Overview. Let us start with Anita, a resident of India, who approaches an Enrollment Centre run by an Enrollment Agency (EA) to get registered into the Aadhaar system. She fills her personal details in the Enrollment Form and submits it to an Enrollment Officer (EO). The EO uses the Enrollment Client software to record her biometrics (photograph, iris scan, and fingerprints) and enter her demographic details into the system. Anita has carried her original documents as proof of identity and address, which were scanned and returned to her by the EO. Anita's personal information is encrypted and uploaded to the CIDR for deduplication to ensure that no one is enrolled twice. After successful deduplication, Anita receives a letter containing a randomly generated Aadhaar number, which her information can be authenticated against throughout her life.

Now, say Anita wants to draw her pension and needs to verify her identity. This is carried out by AUAs/KUAs (Authentication/e-Know Your Customer

User Agency): The pension office (an AUA/KUA) asks Anita for her Aadhaar number and fingerprints, and sends an authentication request to the CIDR, which returns a Yes/No response ("Yes, this is Anita"/not); it may similarly verify Anita's age. The UIDAI mandates the pension office (all AUA/KUAs) to have a local "Aadhaar Data Vault" to store Aadhaar data securely. (See Figure 1.) Since Anita shared her Aadhaar number with the pension office, UIDAI ensures that Anita's data is secure in the Aadhaar Data Vault and that its usage does not reveal any unknown information about Anita. The vault is located within the organization's infrastructure and contains Aadhaar numbers collected by any agencies for purposes under the *Aadhaar Act and Regulations, 2016*, accessible only on a "need-to-know" basis. Anita can update her Aadhaar data by visiting any Enrollment Centre. She must carry her original documents and pay a small fee to update her details. She may also update her demographic data (not biometric data) online by uploading required documents to the SSUP (Self Service Update Portal). She also can update her address via SSUP *without* official proof of address. In this case, UIDAI will send Anita an Address Validation Letter to her present address, which could be used as proof for an online update.

Usability of Aadhaar. The entire process assumes significant privilege: that a resident can read and speak fluently, has a phone (for many services, a smartphone), access to the internet, etc. Also, during the COVID pandemic, many centers are either fully or partially shut down: simple tasks such as linking a mobile number to one's Aadhaar for the first time have turned herculean. If one's Aadhaar number is lost (e.g., loss of card), there is no way to recover it for someone without a mobile phone (or an unlinked phone). This can result in loss of welfare [7], and restoring the UID is incredibly difficult. On the other hand, there is no way to remove one's data from the CIDR if the citizen wants/needs this (e.g., changing residency to another country). There are also on-ground issues like the prevalent use of the Aadhaar "card" or a photocopy as a visual proof of identity without biometric validation (e.g., at airports).

3.1 Enrollment Ecosystem

The Enrollment ecosystem (Figure 2) handles onboarding of residents into Aadhaar with the objective of providing each resident with a unique ID (UID). It also handles updating of demographic and biometric details of existing UID holders. Residents enroll only once but may request updates. The ecosystem is designed to work offline to allow enrollment of residents from areas that lack connectivity. There are two major actors: Registrars and Enrollment Agencies (EAs). UIDAI appoints Registrars, and each Registrar appoints EAs under it.

Registrar: UIDAI partners with various ministries, banks, public sector organizations, and other agencies that interact with Indian residents [63,68] to facilitate issuing Aadhaar numbers by enrolling residents and validating resident data during enrollment and updation. Registrars must take special measures to enroll women, children, persons with disabilities, unskilled workers, nomadic tribes, and people belonging to marginalized groups who cannot produce a valid Proof of Identity (PoI) and/or Proof of Address (PoA) [63]. "Introducers" are individuals (such as Registrar employees, members of local administrative and elected bodies, etc.) recognized by Registrars to confirm resident data without PoI or PoA. Registrars must follow protocols and standards prescribed by the UIDAI. They usually outsource these tasks to EAs. While they are responsible for the correct functioning of these EAs, there is no mention of Registrars having to inform UIDAI about the EAs. A Registrar uses a UIDAI developed Enrollment Client to enroll residents, and must follow the Demographic Data Standards and Verification Procedure (DDSVP) [44].

Security (Policy and Logs) The MoUs between Registrars and UIDAI specify that UIDAI periodically audits the Registrars and EAs (frequency not specified). Although the standard penalties are nowhere specified, if a Registrar fails to follow the security mandates, UIDAI will only make "reasonable attempts" [68] to discuss and resolve difficulties with the Registrar. Organizations have been penalized in the past: UIDAI terminated a Registrar's contract citing "enormous number of complaints of corruption and enrollment process violations against Aadhaar Enrollment/Update Centres under CSC e-Gov." [38]

Enrollment Agency Registrars employ third-party vendors called Enrollment Agencies (EA) to carry out enrollment services using tools and procedures [61] prescribed by the UIDAI. Sometimes, Registrars double up EAs instead of employing external EAs. For example, a bank may use its branches as EAs. In such cases, "Enrollment Agency" and "Enrollment Centre" become synonymous. As this is pervasive, we use these terms interchangeably in this paper. EAs are the on-ground functional arm of the Enrollment ecosystem and are responsible for providing operators and supervisors for each Enrollment Centre [62]. These Enrollment Operators (EOs) collect demographic and biometric data for enrollment or updation using UIDAI-approved equipment [54]. Before enrollment, EAs must verify the resident's PoA and PoI documents and ensure that the details entered in the Aadhaar Enrollment Client match. This verification is done by duly appointed officers at the EA called Verifiers [64].

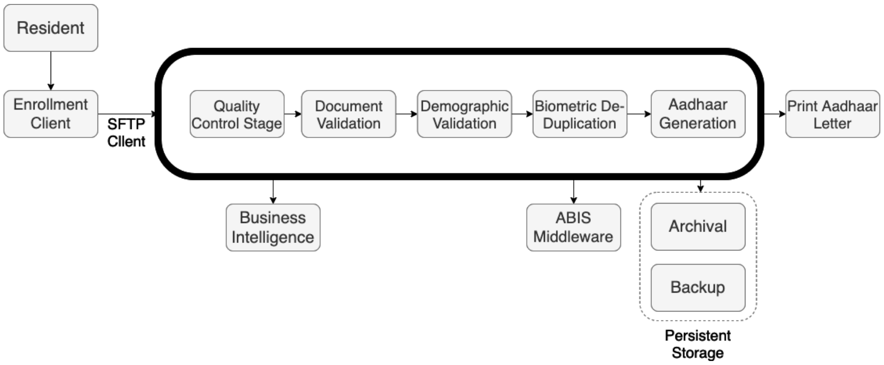

Fig. 2: Flowchart of the Aadhaar Enrollment Ecosystem. The resident's data is captured by the Enrollment Client and sent via the SFTP client for de-duplication. After multiple validity checks, an Aadhaar identity is generated and a physical card is printed.

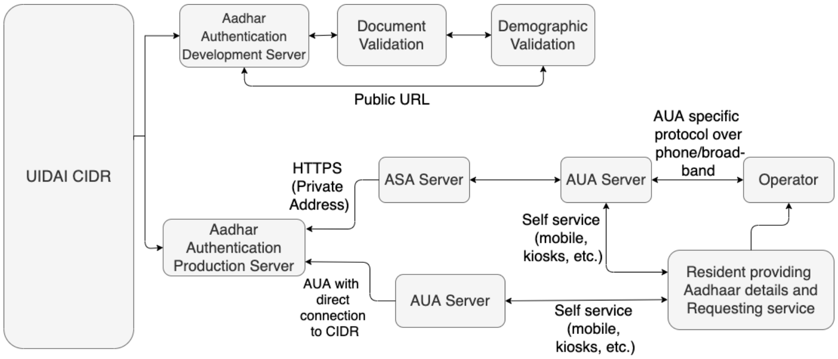
UIDAI CIDR
Aadhar Authentication Development Server
Document Validation
Demographic Validation
Public URL
AUA specific protocol over phone/broad-band
HTTPS (Private Address)
ASA Server
AUA Server
Operator
Self service (mobile, kiosks, etc.)
Aadhar Authentication Production Server
AUA with direct connection to CIDR
AUA Server
Resident providing Aadhaar details and Requesting service
Self service (mobile, kiosks, etc.)

Fig. 3: Flowchart of Aadhaar's Authentication Ecosystem. We start at bottom right with a resident requesting a service. Aadhaar details are sent to the CIDR either through an AUA Server directly to the Production Server or via an ASA server. The CIDR then authenticates this information and returns the results via the same route.

Security (Technical) **Enrollment Equipment** – UIDAI mandates Registrars to follow guidelines to set up the enrollment environment. Only certified equipment is allowed [50]. The Enrollment Client is equipped to work under "Indian conditions", which we assume means low lighting, lack of internet connectivity, dusty environments, etc. [27]. **Data Validation** – The resident's PoI and PoA documents are verified by the Verifier, and details are entered into the Enrollment Client by the EO, followed by biometric data capture and validation by the resident. Most onboarding happens offline — data is periodically synced with CIDR [54]. **Operator Activity Tracking** – Every EO using the Enrollment Client must sign each enrollment and update with their own biometrics. EO login involves a username, password, and the EO's biometrics [54].
Security (Policy and Logs) When a Registrar hires an EA, the EOs working there need training and certification. The UIDAI provides a questionnaire [45] and a presentation to ensure basic training. The "Training, Testing and Certification" team designs lessons to ensure that EOs can recognize the necessary documents for the first check [67]. Periodically, "Mega Training and Certification Programs" [51] are organized to facilitate mass onboarding of operators when there is high demand. Refresher courses are also organized.

3.2 Authentication Ecosystem

The Authentication ecosystem (Figure 3) provides paperless identity verification: **Authentication** – Uses an Aadhaar number and a one-time password (or biometrics) as a second factor to authenticate an individual. The CIDR returns a signed Yes/No [59]. **e-KYC** – identity verification via a signed and encrypted demographic record (name, age, address, etc.) from the CIDR.
AUAs and KUAs: A requesting entity is an agency that uses Aadhaar authentication and e-KYC facilities to provide services such as opening bank accounts,

LPG connections, purchasing mobile SIMs, etc [59]. There are two types of requesting entities [52,53]: an Authentication User Agency (AUA) uses only the authentication service, while a Know-Your-Customer User Agency (KUA) *also* uses the e-KYC service. When serving an individual, an AUA submits their Aadhaar number and demographic/biometric information to the CIDR for authentication [28]. An AUA connects to the CIDR through an Authentication Service Agency (ASA), which owns a secure connection to the CIDR. In response, the AUA receives a digitally signed response from the CIDR. A *sub-AUA* uses Aadhaar authentication to enable its services by contracting the services of an AUA. A KUA, *in addition to being an AUA*, uses e-KYC authentication facility to retrieve a resident's personal information from the CIDR.

When an Aadhaar holder wants to submit their KYC details to a KUA, they download a copy of their e-KYC in XML or QR Code format from the Aadhaar website. This is encrypted with a "Share Code" set by the user. To verify the submitted file, a request is sent to CIDR through a KSA. The KUA receives a "digitally signed [machine readable XML] e-KYC authentication response with encrypted e-KYC data [60]." The KUA uses this copy of the holder's KYC data retrieved from UIDAI to verify the offline copy the resident submitted. The encrypted XML file contains the resident name, download reference number, address, photo, gender, DoB/YoB, hash of mobile number, hash of email.

Security (Technical) Aadhaar numbers collected by an AUA/KUA are encrypted and stored locally in an "Aadhaar Data Vault" [13]. The encryption keys must be stored in a Hardware Security Module (HSM). The UIDAI does not mandate audits nor specifies repercussions if the vault stores plaintext. The implementation of the Data Vault is usually outsourced, and many third-party vendors [23] offer their own variants. An AUA/KUA can transmit biometric information over a network only after creating an encrypted Personal Identity Data (PID) block in accordance with UIDAI specifications [48]. The encrypted PID block cannot be stored except for buffered authentication (for up to 24 hours, after which it must be deleted from local storage) [26].AUA/KUAs send authentication and e-KYC requests to ASAs/KSAs (who relay them to the CIDR) via secure private lines or a secure channel (SSL, VPN) [43].

Security (Policy and Logs) Access to the application, audit logs, source code etc. is only given to authorized personnel [26]. The basis on which a person becomes authorized and the extent of access are unknown. AUAs/KUAs are required to maintain online logs of each authentication transaction for two years, for grievance and dispute redressal. After this, logs are archived offline for five more years and then deleted (unless required in a pending dispute). The logs record the Aadhaar number, auth request, CIDR's response, information disclosed upon authentication, and the person's consent for authentication [26, p. 12]. Logs do not store PID information. No encryption/safety standards are specified; we discuss the resultant privacy issues in Section 5.3. Aadhaar holders can self-generate Virtual IDs (VID) for privacy. VIDs are temporary, revocable 16-digit random numbers that are one-way mapped from the Aadhaar number [66]. This mapping should be secret and the Aadhaar number should not be recoverable from it. The algorithm used for generating VIDs is not specified.

AUAs/KUAs are required to ensure that their operations are audited, including information security controls and technical testing like vulnerability assessment, penetration tests, etc., especially for new technologies introduced [26]. This audit must be done by a recognised body (presumably government empanelled auditors [12]) annually and on a need basis [26, p. 46] or by UIDAI itself to ensure compliance. Although UIDAI states that only authorized personnel can access the audit trails, selection criteria and security policies are unspecified.

ASAs and KSAs: Authentication/KYC Service Agencies (ASAs/KSAs) are public and private agencies that have an "established secure leased line connectivity with the CIDR" [59] in accordance with UIDAI's standards and specifications [26]. Only they can interact directly with the CIDR in the Authentication ecosystem. ASAs provide secure CIDR access to AUAs for authentication; KSAs are ASAs with additional e-KYC permissions and therefore serve KUAs. Hence, ASAs/KSAs act as enabling intermediaries between an AUA/KUA and the CIDR as shown in Figures 1 and 3. There are 27 live ASAs/KSAs [58].

Security (Technical) Servers used by ASAs to connect to the CIDR must be located within India. ASA/KSA server host must be within a segregated network segment. It should be isolated from the rest of the network of the ASA/KSA. The ASA/KSA server host is solely dedicated to Aadhaar authentication. The PID block includes the keys generated by the ASAs/KSAs (sensitive and must never be stored). ASAs perform key generation, distribution, and storage.

Security (Policy and Logs) Access control, communication policies, log maintenance and expiration, and audit protocols are the same as those of AUAs/KUAs (Refer to Section 3.2). The logs can be accessed by UIDAI or the requesting entity solely for grievance and dispute redressal and contain the following information: identity of the requesting entity, parameters of authentication request submitted, and parameters received as authentication response.

3.3 CIDR (Central Identities Data Repository)

The Central Identities Data Repository (CIDR) is a centralized database that stores all Aadhaar numbers and corresponding demographic and biometric data. Maintained by UIDAI and distributed across multiple servers throughout India, CIDR is the core of Aadhaar and interacts with both the Enrollment and Authentication ecosystems. CIDR is also (indirectly) responsible for deduplication as deduplication servers access biometric data residing in the CIDR to check for matches before enrolling a new resident. Post-enrollment access to the CIDR comprises mainly authentication and e-KYC requests (see Section 3.2).

Security (Technical) **Enrollment Client**: The connection between the CIDR and the Enrollment Client is protected using SSL. The enrollment data (XML) is POSTed to the CIDR [27,46]. To ensure only certified operators and Enrollment Clients connect to the CIDR, each time an operator logs into the client, an XML document containing the machine identifier, enrollment agency code, and station number is sent to the CIDR for validation. The CIDR then sends back a security token, which is used to send subsequent enrollment data. The XML document containing the enrollment data is sent in the form of packets to the CIDR, *each* of which is encrypted using a public key published by UIDAI, and signed by the sender (to avoid wasting resources on extracting packets without a valid signature [27]). This packet encryption phase is handled by the Client Security module of the Enrollment Client, which also stores certificates and manages keys. The key management uses public-key style encryption where *two* sets of public keys are maintained – one for data exchange between the Enrollment Client and the CIDR, and another for data exchange between the Registrar and the CIDR. The CIDR is classified as a Protected System under the IT Act, and the link between the CIDR and the Enrollment Client is encrypted using 2048 bit PKI. **Deduplication**: Deduplication at the billion scale has never been previously attempted [27]. For risk mitigation, UIDAI has *three* independent ABIS (Automatic Biometric Identification System) providers performing biometric deduplication. At enrollment, Aadhaar first does a demographic and reduced biometric check for matches. The Aadhaar enrollment server integrates the ABIS solutions using an ABIS API and dynamically allocates deduplication requests to the 3 ABIS servers. Then, ABIS deduplication servers are sent packages of size 3-5 MB. The enrollment packet (containing all demographic, biometric, and metadata) is encrypted at the client side and then sent to CIDR; the CIDR interacts with the ABIS servers and sends them these packages. Only the Enrollment Server (maintained by CIDR) can decrypt the enrollment packet. It does this in memory; the decrypted packet is **never** sent to storage. Original biometric data is archived and sent to offline storage and is not available on an online network. 2048-bit PKI is used throughout. See supplementary analysis C for more details. When a registered device is called, it captures, processes, and encodes the digitally signed biometric record. The biometric data received by the CIDR is essentially a Base-64 of the DSA signature of a hash (SHA-256) of the biometric data and a timestamp, device code, and device private key.

4 Security Landscape

We consider the security of different endpoints at which an individual's data could be vulnerable and the steps Aadhaar takes to prevent any attacks.

4.1 Hardware Security and Certification

Biometric data is first collected during registration, and subsequently used to verify that individual's identity. These biometric devices, therefore, are a critical component of Aadhaar. The official documentation [50] specifies two types of devices. *Public Devices* are biometric capture devices that can be attached to the Aadhaar application provided to AUA/Sub-AUA to capture Aadhaar compliant biometric data. The application then encrypts the data before authentication. *Registered Devices* (RD) have three key additional features over public devices. Each RD has a unique device identifier, biometric data is signed with the device key to ensure liveness and encrypted on-device rather than on the host application, and lastly, the RD service is certified regardless of the device provider. "RD service" refers to the process of capturing biometrics, signing them, and forming a personal identity data (PID) block before returning to the application.

Device Compliance Levels. The RD service is certified over two levels. *Level 0 Compliance* ensures that the implementation of signing and encryption of biometrics is within the software zone at host's OS level. This includes ensuring that the associated private keys are not compromised through access via any external applications within the OS, and the biometric data can not be injected maliciously. *Level 1 Compliance* enhances security by ensuring that the signing and encryption take place within a Trusted Execution Environment (TEE). The private keys and the biometrics are stored in, and accessed via, the TEE.

Pre-certified Hardware: Any provider of an L1 compliant device needs to supply "Pre-certified" Hardware (PCH) and accompanying system software. This must protect against Hardware Cloning, Hardware Tampering (Physical, voltage, frequency, temperature attacks on crypto blocks), Differential Power analysis, Probing, Memory segregation of cryptographic operations, Cryptography implementation vulnerability, Attacks against Secure Boot and Secure Upgrade and TEE, and Secure processor OS attacks.

Certification: The agencies responsible for the certification are UIDAI and Standardization Testing and Quality Certification (STQC) Directorate (which is an attached office of the Ministry of Electronics and Information Technology). The certification process is exhaustive and combines testing over multiple, widely regarded industry and government standards like NIST's FIPS [39] for the security of cryptographic modules, PCI PTS [30] and PED for physical and software tampering, GlobalPlatform certification for the TEE, and other dedicated hardware for L1, like secure boot, secure upgrade, etc. More details are available in [50]. UIDAI and STQC also check for tamper responsiveness: these devices can detect box-open tampering, chemical tampering, etc. and destroy sensitive data upon detection. However, a small part of hardware and system software is vendor self-certified. We were unable to find any reasoning for this; it is unclear how a vendor can verfiably self-certify a lack of backdoors!

4.2 Key Management and Device Registration

Each device provider must register and obtain a device provider ID via UIDAI. UIDAI then signs a public-key certificate procured by the device provider from a certificate authority(CA) licensed by the Govt. of India's Controller of Certifying Authorities (CCA). These certificates are X.509 v3 compliant. Furthermore, the UIDAI policy specifies time periods after which device keys have to be rotated.

L1 compliant devices store their signing and encryption keys in PCH. There exists a hardware key-store in these devices. The certificate issued for the device, called the *Chip Identity Certificate*, is stored therein and must be non-clonable. The signing and encrypting key-pair generation and the cryptographic operations happen within this hardware key-store. However, L0 compliant devices have a software-based key-store provided by the OS. Common software security practices are specified and required for this key-store in [50]. All accesses to this key-store are logged. The private key is not extractable in any format, and the key-store is cleared and zeroed if the RD service is deleted. The key-store password is auto-generated using some random data, user credentials, and device identities of hardware like hard disk serial number, processor ID, and other device IDs. This key derivation is not public and obfuscated to prevent attacks. *We note that this can be dangerous.* Historically, security by obscurity has been a terrible idea [40], and has meant that bad security went uncriticized.

4.3 Biometric Deduplication and Locking

Since Aadhaar has the face, fingerprint, and iris biometrics for enrolled residents, it can combine these for de-duplication upon enrollment. With ten fingerprints and a facial image, a 95% de-duplication rate could be achieved over a population of 50 million. To increase the de-duplication rate to 99%, usage of iris biometrics was proposed. However, there is *no documentation* about the matching algorithms running at the ABIS and how well they perform. The accuracy listed above implies that authentication for valid Aadhaar numbers and corresponding residents might fail for a small fraction of requests. While UIDAI has not released any documentation about the de-duplication process, we discovered the following information from our interviews of Aadhaar personnel: The deduplication problem is viewed and solved as a multi-class classification problem where there are as many classes as there are individuals in the Aadhaar database. Using deep learning techniques, the set consisting of Aadhaar IDs, ten fingerprints, iris and face biometric data is pre-processed before classification. Since this is a huge dataset, this process is optimized by reducing some features. If candidate duplicates are discovered, they are checked using some more features along with a combination of manual assistance. The biometric algorithms used were described as standard ones from the works of Jain et al. [22,69]

Biometric Locking [57] is meant to give residents more control over their biometric data. The mAadhaar app allows users to lock or unlock their biometrics. The app signs in residents using their Aadhaar number and a one-time password (OTP) on their Aadhaar-linked phone number. Once locked, UIDAI will not authenticate the residents' Aadhaar number via biometrics. Since fingerprints and other biometrics are subject to forgery and spoofing attacks [20], such fine-grained access prevents these attacks in two ways: (1) the attacker now needs to access the victim's mAadhaar application; (2) residents can unlock their biometrics only when they anticipate using their biometrics for authentication.

5 Security, Privacy and Attacks

Defining "security" and "privacy" in the context of Aadhaar is nontrivial. It's easy to provide stringent requirements, but those would almost certainly result in the exclusion of large sections of marginalized people in India, who may not have much documentation — precisely those we want to help. Many Indians also routinely use different spellings for their names (and other data) and may need to update the same without requiring a complicated court process (names in various Indian languages can be anglicized in multiple ways). Therefore, any realistic treatment of security (and attacks) cannot be too broad; we detail our Aadhaar-specific interpretations of the CIA (Confidentiality, Integrity, and Availability) information security triad in this section. We also explicitly list a variety of threat actors and their abilities (see supplementary analysis C).

Classifying Attacks. We use the existing CIA standard for information security. Any attack must violate one or more of: **Confidentiality** – Access to a resident's data (demographic or biometric) collected at the time of enrollment or updation is granted only to authorized individuals within UIDAI and its partner organizations. **Integrity** – A resident's information within the CIDR or during transmission is not modified or lost in an unauthorized manner. **Availability** – A resident's data is available to authorized entities within UIDAI and its partner organizations when required.

5.1 Threat Actors

We conduct a threat actor analysis to identify possible threats as an individual's data travels through the system. In Table 3, we classify threat actors based on their *capability*, *motivation*, and *damage caused* and give low/medium/high ratings for each. The threat actors we identified are described below.

Rogue Enrollment Operator: The first barrier an individual's information has to the central repository is the enrollment operator, which has the responsibility of asking the individual their information and verifying its authenticity. A rogue agent can possibly enroll the individual with faulty data or, worse, make a copy of their data and enroll a fake resident instead.

Rogue agency seeking AUA/ASA services: AUA/ASA provide services to agencies seeking to become requesting entities for authentication. Aadhaar specifies the criteria for such agencies [47]. However, in some cases, the authentication devices are operator-assisted: a service might be provided without authentication or based on identity forgery. E.g., an operator at a cellular agency could authenticate twice by using Anita's Aadhaar details (when she applies for a new SIM) and keep one connection for themselves.

Rogue Enrollment agent: A rogue enrollment agent can help generate fake Aadhaar cards; in practice, there is little oversight in place.

Rogue UIDAI official: The access privileges of a high-ranking UIDAI official, if misused, can result in identity theft, fake voter IDs, and more.

External parties: Governments, IT companies, and curious residents could try to access confidential Aadhaar information for varying motives. The resources possessed by all these external parties can vary quite a bit.

5.2 Forbidden Attack: A Cryptographic Challenge

We describe a possible cryptographic attack on Aadhaar; note that carrying out such an attack would be illegal, as Aadhaar is classified as a "protected system" under Section 70 of the Indian IT Act, 2000 [1]. We reported this attack to UIDAI, which validated its correctness and ensured its mitigation.

Aadhaar's API security document [55, p. 29] details that packaged biometrics are sent for authentication as a `Pid` (Personal Identity Data) element, which is a base-64 encoded block. Before base-64 encoding, the `Pid` blocks are encrypted with a dynamic session key using AES-256 symmetric algorithm, using the Galois Counter Mode (GCM). Refer Appendix A.1 for details about GCM. One major issue discussed by Antoine Joux in his comments to NIST on GCM [8] is *A forbidden attack with repeated IV*. If an adversary sees two different messages encrypted with the same IV, it can inject malicious content into the communication channel. One such attack is demonstrated in detail by Böck et. al. [10].

The document [55] describes exactly how Aadhaar instantiates AES GCM: *"The last 12 bytes of the* `ts` *(string formatted date) is used as the IV or nonce."* The `ts` attribute (timestamp) is described as follows [55, p. 15]: *"Timestamp at the time of capture of authentication input. This is in the format YYYY-MM-DDThh:mm:ss (derived from ISO 8601)."* The implementation available on the Github repo [25] and the old Aadhaar developer portal [49], and our interviews with Aadhaar officials confirm this timestamp format. So, suppose the timestamp is `2020-06-22T19:47:30`. Then last 12 bytes are `-22T19:47:30` and the string used as IV for AES GCM comprises just the day-of-month and the time. Trivially, the IV is reused if multiple messages are sent within the same second, or if messages are buffered or batched. Further, the IV `-22T19:47:30` repeats at time `19:47:30` on the `22` date of each month, leading to monthly IV reuse. We describe this forbidden attack formally in Appendix A.2. Briefly: an adversary can exchange their invalid biometrics with valid data and still authenticate. (They cannot recover keys, but we want to protect the data, not just the keys.) Authentication requests can be altered over the channel due to IV reuse. As a consequence, a malicious party can open a bank account, fly domestically, get a SIM card, etc. — all in someone else's name.

Benchmarking. Using data published by the Govt. of India [56], we estimate how many times AES-GCM is used for encrypting requests. One source of such requests is the Authentication API; the other is e-KYC, which also uses AES-GCM in the exact same way [46]. Between October 2016 to September 2019, 7.9 billion requests were made for e-KYC; on average, the IV was reused ~ 83 times per second. Consequently, the malleability of the encrypted plaintext becomes a major security issue, and hence, all chosen ciphertext attacks become feasible.

Mitigation. The IV for AES-GCM is 96-bits (12 bytes) and we need to prevent IV reuse. Currently, the IV is of the form `-22T19:47:30` (day-of-month and time). In this format, the IV takes $<\sim 2^{22}$ different values (since the dates vary in range 1-31, range of hours is 0-23 and minutes and seconds are in range 0-59 each). Instead, if a simple counter is utilized, it would take values in the whole 2^{96} range space (as IV length is 96 bits). However, the communication complexity of a synchronized task across 30 million devices is infeasible: maintaining it proved impractical and so the Aadhaar team decided[4] to use timestamps as IVs due to the availability of this information across all devices. *To mitigate this attack, all AES-GCM communications now occur over secure channels with unique session keys. This prevents the attack from being exploitable.* Note that the UIDAI encrypts all communications and storage across Aadhaar. UIDAI policy is to use RSA [33] with 2048 bit keys for public key and AES with 256-bit keys for symmetric key encryption [48].

5.3 Privacy Issues

Aadhaar's policy for logging requests and responses creates two issues. (1) the privacy of registered individuals in the event of a breach; and (2) the possibility of surveillance. The logs are rich spatio-temporal data on almost everyone in India. Obviously, a leak would be catastrophic if the data is not anonymized; but even "anonymized" spatio-temporal data can be used to uniquely identify a very large fraction of the individuals, as demonstrated by de Montjoye et al. [29]. Therefore, the use of virtual IDs (see Section 3.2) is essential. However, existing documentation is ambiguous as to whether virtual IDs are used by default for authentication requests. Further, while all communication of Aadhaar's biometric templates is end-to-end encrypted, they remain vulnerable to social engineering attacks and the like at ECs; the privacy loss inherent in the storage of biometric templates for a national ID is beyond the scope of this work.

Non-KYC operations should not reveal anything beyond verification (yes/no). If an entity has knowledge of $a_1, a_2, ...a_k$ columns of a person's Aadhaar information, they should not be able to gain knowledge of the a_{k+1}^{th} column, including brute-forcing by checking against the same column multiple times (given someone's name and phone number, an entity should not be able to query multiple times with different dates of birth). Services using aggregated data must be differentially private. The work of Wilson et al. [70] focuses on this approach and provides extensive theoretical and practical analysis. This gives a scalable method which is generic enough to apply to all national ID systems including Aadhaar. Extensive data logging for almost a decade means that such a system can very easily be used to track registered individuals. Differentially private (DP) anonymized logs can be used to protect against such tracking. While such logs and streams have been studied in some detail [24,16], it remains to be seen if such proposals would be feasible at this scale (see 5.2). The closest (in scale) DP system is the recent work of the US Census Bureau [17] which shows that DP is not a one-size fits-all solution [19]. Aadhaar is meant to ensure the targeted delivery of benefits and services to Indian citizens. Verification of a resident's existence to receive a service must not leak personally identifiable information. Another solution to mitigate privacy concerns is via brokered identification [11]. Here, a centralized hub mediates communication between an identity authority and a user with identity credentials. The US FCCX [2] and GOV.UK Verify proposed

[4] This was discussed during our interviews of Aadhaar personnel.

using this, but were unable to ensure all the properties required (see [11]). Using such a mechanism would mitigate the possibility of surveillance using Aadhaar authentication requests.

6 Media Allegations Analysis

6.1 Filtering Legitimate Breaches

Our primary database of media allegations consists of 36 reports from various news outlets. We filter breaches that are "legitimate" based on our knowledge of the Aadhaar infrastructure and our definitions of security and privacy. This yielded 17 legitimate security breaches and 10 privacy breaches, which were further analyzed. (Security and privacy breaches are not mutually exclusive.) Additionally, for each legitimate security breach, we ascertain whether or not there was a breach of Confidentiality, Integrity, or Availability of data in the Aadhaar infrastructure. (See Table 2 in supplementary material C.)

According to our analysis, the prevalent breach is of **confidentiality**; this usually entails a subset of Aadhaar data being made public. Prevention goes back to ensuring that data is secured in encrypted "data vaults" and access is limited. Breach of **integrity** is also common. It compromises the quality of the central database. They typically occur at an individual level, involving a small set of rogue insider-agents or the hacking of individual accounts. This is easily detected if performed repeatedly, while for a specific use-case like introducing certain individuals into the database, the breach is virtually undetectable. OTP-based security, standardized punishments, and closing some known structural gaps could mitigate this. Breaches of **availability** are rare and occurs only in cases of insider attacks. The CIDR repository itself is reasonably secure, and removing/editing information is hard to do illegally. Internal attacks can be mitigated by using a decentralized system of checks and balances where no individual can commit edits [14]. For example, all operations by high-level employees could require approval by randomly chosen officers (anonymously).

6.2 Attack Analysis

We define three broad classes of attacks: **(1) Server compromise:** Hacking of the UIDAI or Partner software/database. **(2) Infrastructural loopholes**: Access via legitimate UIDAI channels. **(3) Sub-par hardware**: UIDAI hardware tricked into approving false biometrics as genuine due to flaws or backdoors.

We analyze the feasibility of attacks based on the cost (time and resources used) and the effort required to protect against it. We then suggest mitigation strategies to ensure robust security. A detailed breakdown of our examination is provided in C. Aadhaar is predominantly vulnerable to "Infrastructure Loopholes." These breaches exploit the general negligence to set or adhere to security protocols. As discussed, agents of Aadhaar, such as and especially EOs, can effectively be a threat to the security of the database if their credentials are not stored properly (multiple instances of this have occurred). This is a breach that is detected often, but measures taken to curb it are seemingly nonexistent. Complimentary and robust security standards like OTPs and Iris scans for these Aadhaar agents may be effective in ensuring accountability. The CIDR Database is secure and there exist no reports of it being hacked, but data in UIDAI's partner organizations are regularly stored insecurely. We recommend that the UIDAI sets stricter standards and enforce them across the board. No one should store any Aadhaar data except the CIDR. Any queries to the database should go through the CIDR, and local copies should not be stored.

6.3 Privacy Breach Analysis

Listing the various allegations of Aadhaar privacy breaches, we find that limited access to the database and illegal or insecure storage of Aadhaar information are common. These are primarily due to improper or inefficient handling of data by UIDAI's partner organizations. We summarize the number and type of privacy breaches in the attached supplementary material C. In either case, the pivotal issue is that an individual can be identified, resulting in the misuse of their data by malicious actors. This can include surveillance, profiling, or creating new services (without consent) by the state or other private actors. Most security breaches happen within the Enrollment Ecosystem; privacy breaches largely appear in the Authentication Ecosystem. For Aadhaar to be effective in the targeted delivery of subsidies, it needs to ensure that resident data is private beyond enrollment. If organizations require Aadhaar data to analyze aggregated trends, we strongly recommend differentially private systems be used. For other cases that require individual instances of data, it must be constitutionally valid (given India's Right to Privacy) and must be complemented by effective protocols to ensure that the individual is not identified after processing.

7 Conclusion

We analyze Aadhaar, the world's largest digital biometric identification system, and provide the first detailed, unified description of the infrastructure. We conclude that the framework does not have glaring security flaws of the kind suggested by media reports. Almost all the issues we found were due to a set of challenges unique to a system at Aadhaar's scale. While we discussed mitigations for any flaws we found, we did not make any policy recommendations in this paper: if we had to make one, it would be for the system to be *significantly more transparent and open-source.* Throughout its lifetime, Aadhaar has been subject to multiple allegations that have made national headlines in India. We list, analyze, and classify these allegations to allow for a more balanced view of Aadhaar, identifying which ones are likely to be legitimate. (We note that most of the alleged attacks are now infeasible.)

We emphasize that our focus remained on the strengths and vulnerabilities of the technology, structure, and policy behind Aadhaar, and not *issues with large-scale biometric ID schemes in general.*

References

1. Information technology act, 2000, https://www.meity.gov.in/writereaddata/files/act2000n_0.doc
2. Fccx briefing (2014), https://csrc.nist.gov/csrc/media/events/ispab-june-2014-meeting/documents/ispab_jun2014_fccx-briefing_glair.pdf
3. Adhikari, G.P.: National id project of nepal: Future challenges. In: Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance. p. 379–380. ICEGOV '11, Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/2072069.2072151, https://doi.org/10.1145/2072069.2072151
4. Agrawal, S., Banerjee, S., Sharma, S.: Privacy and security of aadhaar: A computer science perspective. Economic and Political Weekly **52**, 93–102 (09 2017)
5. Aiemworawutikul, W., Datla, M.V., Lee, J.C.S., Wen, T., Zhang, Y.: Vulnerability assessment in national identity services (2019)
6. Al-Khouri, A.M.: Facing the challenge of enrolment in national id schemes. In: Brömme, A., Busch, C. (eds.) BIOSIG 2010: Biometrics and Electronic Signatures. Proceedings of the Special Interest Group on Biometrics and Electronic Signatures. pp. 13–28. Gesellschaft für Informatik e.V., Bonn (2010)
7. Anil, V., Dreze, J.: Without aadhaar, without identity (2021), https://indianexpress.com/article/opinion/columns/flaw-in-aadhaar-architecture-uidai-card-enrolment-7389133/
8. Antoine Joux: Authentication Failures in NIST version of GCM (2006), https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/joux_comments.pdf
9. Arora, S.: National e-id card schemes: A european overview. Information Security Technical Report **13**(2), 46 – 53 (2008). https://doi.org/https://doi.org/10.1016/j.istr.2008.08.002, http://www.sciencedirect.com/science/article/pii/S1363412708000241
10. Böck, H., Zauner, A., Devlin, S., Somorovsky, J., Jovanovic, P.: Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS. In: 10th USENIX Workshop on Offensive Technologies, WOOT 16, Austin, TX, USA, August 8-9, 2016. USENIX Association, Austin, TX (2016)
11. Brandão, L.T., Christin, N., Danezis, G., et al.: Toward mending two nation-scale brokered identification systems. Proceedings on Privacy Enhancing Technologies **2015**(2), 135–155 (2015)
12. CERT-In: Empanelled Information Security Auditing Organizations (2018), https://www.cert-in.org.in/PDF/Empanel_org.pdf
13. Compliance Uncovered: Aadhaar Data Vault - To whom it applies (Sep 2018), https://complianceuncovered.com/2018/09/03/aadhar-data-vault-to-whom-it-applies/
14. Cybersecurity, Agency, I.S.: Insider threat mitigation (2019), https://www.dhs.gov/cisa/insider-threat-mitigation
15. Electronic Frontier Foundation: Mandatory national ids and biometric databases (2021), https://www.eff.org/issues/national-ids
16. Elkoumy, G., Pankova, A., Dumas, M.: Mine me but don't single me out: Differentially private event logs for process mining. In: ICPM 2021. pp. 80–87 (2021)
17. Garfinkel, S.: Implementing differential privacy for the 2020 census. USENIX Association (2021)
18. Garfinkel, S.L.: Risks of social security numbers (1995)

19. Garfinkel, S.L., Abowd, J.M., Powazek, S.: Issues encountered deploying differential privacy (2018)
20. Geller, B., Almog, J., Margot, P., Springer, E.: A chronological review of fingerprint forgery. Journal of Forensic Sciences **44**(5), 12024J (Sep 1999)
21. Goel, V.: 'Big Brother' in India Requires Fingerprint Scans for Food, Phones and Finances (Apr 2018), `https://www.nytimes.com/2018/04/07/technology/india-id-aadhaar.html`
22. Jain, A.K., Flynn, P.J., Ross, A.A.: Handbook of biometrics. Springer (2010)
23. JISA Softech Pvt Ltd: Aadhaar Data Vault (2018), `https://www.jisasoftech.com/aadhaar-data-vault/`
24. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. Proc. VLDB Endow. **7**(12), 1155–1166 (2014)
25. Limited, G.: Source code for Aadhaar v1.6. `https://github.com/GeoAmida/AadhaarAuth1.6` (2011), accessed: 2021-01-19
26. MeitY and UIDAI: Compendium Of Regulations, Circulars & Guidelines For ASA and AUA (2018), `https://uidai.gov.in/images/resource/compendium_auth_19042018.pdf`
27. Ministry of Electronics and Information Technology: Aadhaar technology & architecture (2014), `https://archive.org/details/Aadhaar-Technology-Architecture/page/n2`
28. Ministry of Law and Justice and Government of India: The Aadhaar (Targeted Delivery of Financial and Other Subsidies, Benefits and Services) Act, 2016 (2016), `https://uidai.gov.in/images/targeted_delivery_of_financial_and_other_subsidies_benefits_and_services_13072016.pdf`
29. de Montjoye, Y.A.A., Hidalgo, C.D., Verleysen, M., Blondel, V.: Unique in the crowd: The privacy bounds of human mobility (2013), `https://www.nature.com/articles/srep01376`
30. (PCI), P.C.I.: Payment Card Industry PTS POI Security Requirements v4.0 (June 2013), `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf`
31. Rajput, A., Gopinath, K.: Towards a more secure aadhaar. In: Information Systems Security - 13th International Conference, ICISS 2017, Mumbai, India, December 16-20, 2017, Proceedings. pp. 283–300 (2017)
32. Rajput, A., Gopinath, K.: Analysis of newer aadhaar privacy models. In: Information Systems Security - 14th International Conference, ICISS 2018, Bangalore, India, December 17-19, 2018, Proceedings. pp. 386–404 (2018)
33. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems (reprint). Commun. ACM **26**(1), 96–99 (1983)
34. S., D.H.: Risking identity: a case study of jamaica's short-lived national id system. Journal of Information, Communication and Ethics in Society **18**(3), 329–338 (Jan 2020). https://doi.org/10.1108/JICES-04-2020-0040, `https://doi.org/10.1108/JICES-04-2020-0040`
35. Service, I.A.N.: 125 Crore Aadhaar Cards Issued Since 2009: Centre (Dec 2019), `https://www.ndtv.com/india-news/centre-says-125-crore-aadhaar-cards-issued-till-date-2155184`
36. Singh, R., Jackson, S.J.: From Margins to Seams: Imbrication, Inclusion, and Torque in the Aadhaar Identification Project, p. 4776–4824. Association for Computing Machinery, New York, NY, USA (2017), `https://doi.org/10.1145/3025453.3025910`
37. Srinivasan, J., Johri, A.: Creating machine readable men: Legitimizing the '¡i¿aadhaar¡/i¿' mega e-infrastructure project in india. In: Proceedings of

the Sixth International Conference on Information and Communication Technologies and Development: Full Papers - Volume 1. p. 101–112. ICTD '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2516604.2516625, https://doi.org/10.1145/2516604.2516625

38. Srivas, A.: Millions of Rural Indians May be Hit as UIDAI Ends Contract With CSC Network For Aadhaar Enrolment (Feb 2018), https://thewire.in/tech/millions-may-affected-uidai-centres-csc-network-clash-renewal-aadhaar-services-contract
39. of Standards, N.I., (NIST), T.: FIPS 140-2: Security Requirements for Cryptographic Modules (May 2001), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.140-2.pdf
40. Swire, P.P.: A theory of disclosure for security and competitive reasons: Open source, proprietary software, and government systems. Hous. L. Rev. **42**, 1333 (2005)
41. Team, M.: Modular Open Source Identity Platform (MOSIP) Documentation. https://docs.mosip.io/platform/ (2021), accessed: 2021-01-19
42. Tech2News Staff: Aadhaar Security Breaches: Here are the major untoward incidents that have happened with Aadhaar and what was actually affected (Sep 2018), https://www.firstpost.com/tech/news-analysis/aadhaar-security-breaches-here-are-the-major-untoward-incidents-that-have.-happened-with-aadhaar-and-what-was-actually-affected-4300349.html
43. Thales: Complying with UIDAI's AADHAAR Number Regulations (2018), http://go.thalesesecurity.com/rs/480-LWA-970/images/Thales-UIDAI-AADHAAR-cb.pdf
44. UIDAI: Demographic Data Standards and Verification procedure (DDSVP) Committee Report (2009), https://uidai.gov.in/images/UID_DDSVP_Committee_Report_v1.0.pdf
45. UIDAI: Questionnaire - UIDAI Operators (2011), https://uidai.gov.in/images/training-2019/QuestionBank-Operator-510/English_510QB_24012019.pdf
46. UIDAI: Aadhaar E-KYC Specification - Version 2.0 (2016), https://uidai.gov.in/images/aadhaar_ekyc_api_2_0.pdf
47. UIDAI: Eligibility criteria for appointment as requesting entities (2016), https://uidai.gov.in/images/resource/eligibility_criteria_for_aua_kua_17122016.pdf
48. UIDAI: Aadhaar Authentication API Specification – Version 2.0 (2017), https://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_2_0.pdf
49. UIDAI: Aadhaar Developer Portal. https://web.archive.org/web/20170326113654/https://authportal.uidai.gov.in/web/uidai/developer (2017), accessed: 2017-03-26
50. UIDAI: Aadhaar Registered Devices - Technical Specification, vol. 2.0. MeitY, New Delhi, Delhi, 1 edn. (2017), https://uidai.gov.in/images/resource/aadhaar_registered_devices_2_0_09112016.pdf
51. UIDAI: Request for Empanelment of Enrolment Agencies. Empanelment Of Enrolling Agencies, MeitY, New Delhi, India (2017), https://uidai.gov.in/images/RFE_SEPT_Final_11092017.pdf
52. UIDAI: List of Live Authentication User Agencies (AUAs) (Aug 2018), https://uidai.gov.in/images/list_of_live_aua.pdf
53. UIDAI: List of Live KUAs (Aug 2018), https://uidai.gov.in/images/list_of_live_kua.pdf

54. UIDAI: Setting up and Managing an Enrolment Centre (2018), http://www.nictcsc.com/images/AadhaarProjectTrainingModule/EnglishTrainingModule/module_3a_settingup_managing_enrolment_centre17122012.pdf

55. UIDAI: AADHAAR AUTHENTICATION API SPECIFICATION - VERSION 2.5 (2019), https://uidai.gov.in/images/resource/aadhaar_authentication_api_2_5.pdf

56. UIDAI: Aadhaar Authentication Service Questions at Lok Sabha(Unstarred 2600) (2019), https://uidai.gov.in/images/loksabha/LSPQ_2600_Unstarred.pdf

57. UIDAI: Biometric Lock/Unlock FAQs (Jan 2019), https://uidai.gov.in/contact-support/have-any-question/925-faqs/aadhaar-online-services/biometric-lock-unlock.html

58. UIDAI: List of Live Authentication Service Agencies (ASAs) (2019), https://uidai.gov.in/images/list_of_live_asa.pdf

59. UIDAI: Authentication Requesting Agency (Live), https://uidai.gov.in/ecosystem/authentication-ecosystem/authentication-requesting-agency.html

60. UIDAI: Operation Model (Live), https://uidai.gov.in/ecosystem/authentication-ecosystem/operation-model.html

61. UIDAI: Aadhaar FAQ (Live web page), https://www.uidai.gov.in/298-faqs/enrolment-update/enrolment-partners-ecosystem-partners/2014-what-are-the-fifteen-commandments-that-an-operator-must-remember.-during-resident-enrolment.html

62. UIDAI: Enrolment Agencies (Live web page), https://uidai.gov.in/ecosystem/enrolment-ecosystem/enrolment-agencies.html

63. UIDAI: Registrars - Enrolment Ecosystem (Live web page), https://uidai.gov.in/ecosystem/enrolment-ecosystem/registrars.html

64. UIDAI: Roles and Responsibilities of Verifier and Introducer (Live web page), https://www.uidai.gov.in/images/training_nov_17/Roles_Responsibility_Verifier_Introducer_05122017.pdf

65. UIDAI: Vision & Mission (Live web page), https://uidai.gov.in/about-uidai/unique-identification-authority-of-india/vision-mission.html

66. UIDAI and MeitY: Circular No. 1 of 2018: Enhancing Privacy of Aadhaar holders - Implementation of Virtual ID, UID Token and Limited KYC (2018), https://uidai.gov.in/images/resource/UIDAI_Circular_11012018.pdf

67. UIDAI and MeitY: Training, Testing and Certification (2019), https://uidai.gov.in/aadhaar-eco-system/training-testing-certification-ecosystem.html

68. UIDAI, IDBI Bank: Memorandum of Understanding - UIDAI and IDBI Bank (2011), https://uidai.gov.in/images/mou/partners/mou_idbi.pdf

69. Wayman, J., Jain, A.K., Maltoni, D., Maio, D.: Biometric systems: technology, design and performance evaluation. Springer (2005)

70. Wilson, R.J., Zhang, C.Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., Gipson, B.: Differentially private sql with bounded user contribution. Proc. Priv. Enhancing Technol. (PETS) **2020**

A Background

A.1 AES-GCM

AES GCM (Galois/Counter Mode) is a block-cipher mode of operation which encrypts the plaintext by using the counter mode. For authentication, a hash function called GHASH is used, which computes over the Galois Field $GF(2^{128})$. For a comprehensive description of AES GCM we suggest referring to the work of Böck et al. [10].

A.2 Forbidden Attack

Consider a passive adversary $\mathcal{A}$ which only sees ciphertext data, including initialization vector (IV), associated data, and authentication tag. The authentication key is $H = \mathsf{Enc}_k(0^{128})$ where k is the secret key for encryption. The authentication tag t is the evaluation of a polynomial g at the authentication key H. The coefficients of polynomial g depend on the ciphertext blocks and the constant coefficient is the nonce. Suppose that $\mathcal{A}$ finds two messages m_1 and m_2 encrypted using the same IV. $\mathcal{A}$ now has two polynomials with known coefficients (the ciphertext is public) and the same constant coefficient. Let these polynomials be $g_1(\cdot)$ and $g_2(\cdot)$. For the two authentication tags, t_1 and t_2

$$g_1(H) = t_1, g_2(H) = t_2$$

The adversary $\mathcal{A}$ now knows two polynomials $g_1(x) - t_1$ and $g_2(x) - t_2$ with a common root H, and they can recover a short list of candidates for the authentication key. In theory, this list could be as long as the degree of the polynomial, but is relatively short in practice. The GCD of the two polynomials gives $\mathcal{A}$ a polynomial of small degree with H as a root. Similarly, by finding more IV reuses, the possible number of candidate H keeps reducing, and eventually, H is found. **Now that H is known, $\mathcal{A}$ can substitute any information they like and replace a valid ciphertext**. For a more detailed analysis and description of the attack we refer readers to the work of Joux [8] and Böck et al. [10].

B Abbreviations

See Table 1 for a list of abbreviations used in the paper.

C Supplementary Material

This work summarizes a long ongoing effort to provide the most comprehensive view of Aadhaar. Hence, we attach a short version of our analysis in the tables below for brevity. The full version contains many more examples.

Abbreviation	Full form
UIDAI	Unique Identification Authority of India
MoUs	Memoranda of Understanding
CIDR	Central Identities Data Repository
UID	Unique Identification
KYC	Know Your Customer
EA	Enrollment Agency
EO	Enrollment Officer
AUA	Authentication User Agency
KUA	KYC User Agency
SSUP	Self Service Update Portal
PoI	Proof of Identity
PoA	Proof of Address
DDSVP	Demographic Data Standards and Verification Procedure
HSM	Hardware Security Module
PID	Personal Identity Data
VID	Virtual ID
ABIS	Automatic Biometric Identification System
GCM	Galois Counter Mode
RD	Registered Devices
TEE	Trusted Execution Environment
PCH	Pre-Certified Hardware
STQC	Standardization Testing and Quality Certification
CA	Certificate Authority
CCA	Controller of Certifying Authorities
EC	Enrollment Center
DP	Differentially Private
OTP	One-Time Password
CIA	Confidentiality, Integrity, Availability

Table 1: Summary of abbreviations used in the paper (in order of appearance)

Table 2: Example of the Aadhaar Security Breach Analysis.

Each article was analyzed with regards to the type of security breach (**C**: Confidentiality, **I**: Integrity, **A**: Availability or **L**: Legitimacy) with our reasoning. This table consists of a few examples. The entire list is skipped for brevity.

Allegation	**C**	**I**	**A**	**L**	**Reason**
UIDAI reveals 210 govt websites made Aadhaar details public, did not specify when breach took place	Yes	No	No	Yes	The UIDAI has confirmed this breach of data confidentiality through an RTI (Right to Information) request. It is important to note that the UIDAI itself did not leak this data. It was posted on the websites of over 200 central and state government entities and educational institutes. How the Aadhaar data was accessed is unclear.
Three Gujarat websites including government portal made Aadhaar details public	Yes	No	No	Yes	The Ministry of Electronics and Information Technology confirmed the breach. This is not a direct breach of UIDAI data, but could not have happened without the initial data collection done by Aadhaar.
Indane Leaked Millions of Aadhaar Numbers, Claims French Researcher	Yes	No	No	Yes	Same as above.
UIDAI Blacklists Centre That Leaked Details of a sportsperson	Yes	No	No	Yes	Human error with regards to the enrollment officer/agency
Rs 500, 10 minutes, and you have access to billion Aadhaar details	Yes	No	No	Yes	UIDAI sued The Tribune for exposing this breach and accessing data. This suggests that it was a real attack.
How a SIM Card Operator in Hyderabad Apparently Created His Own Aadhaar Database	Yes	No	No	Yes	He got the information from other sources but the Aadhaar infrastructure should be strong enough to not accept biometrics that are not live.
Aadhaar data: French hacker exposes flaws in its Android app, asks people not to use it	Yes	No	No	Yes	The video released along with the hack demonstrates legitimacy.
Aadhaar's Dirty Secret Is Out, Anyone Can Be Added as a Data Admin	Yes	Yes	Yes	Yes	Access to Aadhaar details after enrollment or updation requires acccess to the CIDR. Therefore, this alleged breach points to an insider incident.

Table 3: Threat actor analysis — Security Breach.
We chose *capability* as a metric to account for the existing checks-and-balances in the system but recognize threats that exist despite them. High capability means that the actor's expertise (technical, political, etc.) may overpower existing security; medium means there are measures but loopholes exist, and low means that existing security measures are enough to protect against this actor.
Motivation addresses the value of Aadhaar data for the actor if they were able to gain access to it. It is low when hampering the system doesn't provide direct gains to the actor, medium if they can disrupt services for others, and high if they can if they can gain a financial or social advantage from the attack.
Damage caused measures the impact of a breach on UIDAI and/or the Indian state. It is low when there's no threat to C, I, or A (as defined in Section 5), medium when there's a threat to the integrity of the system, and high if there's a threat to multiple pillars of CIA.

Threat Actors	**Capability**	**Motivation**	**Damage Caused**	**Comments**
Enrolling Operator	Medium	Low	Medium	Damage caused is medium since rogue individuals can be entered into the database by the EO, thus hampering the **integrity** of the data.
AUA	Medium	Low	Low	AUA can store Aadhaar data provided by the resident when they come to get authenticated (Aadhaar welfare etc.), which is a threat to **confidentiality**.
ASA	Low	Low	Low	The ASA infrastructure has sound measures against Security Breaches.
EA	Low	Medium	Low	EA has medium motivation since the agencies can make fake Aadhaars, even though we are unsure of the number of cards they can generate. This is a threat to the **integrity**.
Registrars	Medium	Medium	Medium	Registrars control multiple agencies. Therefore, they can carry out the same type of breach as an EA on a larger public sphere.
UIDAI official (high ranking)	Medium	Low	High	A high ranking UIDAI official can potentially gain access to the database through internal connections, threatening data **availability**. They can also generate voter IDs via Aadhaar numbers, threatening data **integrity**. As discussed, Aadhaar is more vulnerable to insider attacks than from the outside.
External Governments*****	High	Low	High	The damage caused is high due to the sheer scale of the breach given the threat actor's high capabilities. This could compromise all 3—**Confidentiality, Integrity** and **Availability** of data.
Tech Companies*****	High	Medium	High	Same as above along with the fact that Aadhaar data might be extremely useful for the working of numerous big data companies.
Digitally Literate Citizens	Low	Low	Low	These breaches take a lot of effort and time, especially if information is collated from physical documents. This endangers the **Confidentiality** of data.

*** Potential** actors that could take advantage of vulnerabilities