A UID NUMBERING SCHEME

Hemant Kanakia, Srikanth Nadhamuni and Sanjay Sarma

May, 2010

What is Unique ID?

A Unique Identification is merely a string assigned to an entity that identifies the entity uniquely. We plan to assign a Unique ID to every person residing in India. Biometric identification system and checks would be used to ensure that each individual is assigned one and only UID and the process of generating a new UID would ensure that duplicates are not issued as valid UID numbers.

Challenges in Design of UID

Designing a Unique Identity Number for a population of more than a billion plus poses incredible management and technical challenges. Along with the challenge comes a unique opportunity. The Multipurpose National Identification Card (MNIC) project undertaken circa 2002 has done excellent work to meet some of the difficulties inherent in such a national identity project. In our opinion, an important aspect of such efforts that we expand upon here is on how to design a robust and cost-effective Unique National ID *numbering* system.

This program to provide UID is an opportunity for India to design a 21st century system of identification that could be a significant improvement over the 19th and 20th century systems in use in developed countries like USA and Italy. The tradeoffs made in designing this system will continue to have a long-term impact. A good design for UID could last us for centuries without changes whereas suboptimal choices may dilute or even negate some of the presumed benefits and create future problems. In this white paper, first we will focus on issues involved in designing a nationwide unique identity system, particularly for a big, diverse country like India. Based on that discussion, we shall summarize the principles and the requirements of UID numbering format and propose a UID number that meets the requirements.

Building a unique numbering system has been done several times in the past and for a variety of applications. Bar codes used in supermarkets to identify products, Internet Addresses assigned to a computer on the Internet, and telephone numbers are some examples of unique numbering systems. Although the tradeoffs that were made in designing such systems depend heavily on the applications that use them, there are some common considerations in designing any unique numbering system. Moreover, not all of these designs have successfully stood the test of time. For instance, the explosive growth of the Internet has necessitated major changes in the Internet Address schema. Another example where a change was necessary was for the bar code used to identify con-

sumer products such as groceries and cereal boxes. In the USA, bar code design follows a standard called the Universal Product Code (UPC), whereas in Europe bar codes follow another standard called the European Article Number (EAN). In this era of global retail chains and multinational manufacturers it was found rather inconvenient to have two separate systems designed to serve essentially the same function operating on different continents. There were also some structural shortcomings in how UPC codes were designed. These problems led retailers and manufacturers to change over to a new global standard for bar codes called the Global Trade Item Number (GTIN). The reasons that many of these numbering systems have fallen short also have some commonalities. We also note that as these two previous examples suggest, once a system is widely in use, changing it is difficult, costly and fraught with delays and confusion.

Before outlining a specific proposal, we shall consider below various design trade-offs we considered while designing a unique national ID system for India.

Discussion of Design Trade-offs

Availability of new UIDs

We expect the UID system to live on for centuries. Thus, one needs to carefully plan the format, namely the structure and the length of the string as well as the method of issuing so as not to run out of available numbers to assign at some future date. Surprisingly, in the past, there have been several instances of unique numbering systems that had to be changed at a great cost after being in use for several years. The two main reasons have been 1) unanticipated growth (and types) of objects to which UIDs are assigned and 2) introduction of structure within the UID format that wastes large parts of the available space of UID values. The design of Internet addresses as proposed in Version 4 of Internet Protocols illustrates both of these issues. Initial design provided for 4 bytes to designate a unique address for a computer, called IP address. There was also an explicit hierarchical structure to the IP address. The design was undertaken in the days where mainframes and mini-computers dominated. No one anticipated that one day people would want to provide IP numbers to devices such as mobile phones and even smart appliances; nor did one anticipate that computers would become so small and so cheap that each person would have several computers available, each requiring an Internet Address. Initially, the structure was introduced to make it easy to perform address recognition easier. This meant that typically a large block of IP addresses had to be assigned to any collection of computers at a location not reflecting the actual number of actual computers to be used at that particular location (routing region).

Privacy and security concerns are two other reasons, described later, for avoiding having a structure embedded in the UID string.

Longevity

Systems that are to be as widely used and for multiple different applications as UID tend to be very sticky in the sense that these systems would be in active use for centuries. Once a billion plus people have been assigned a UID, and applications using the UID to conduct their transactions are evolved, anything that requires modifications to existing software applications and databases will cost a lot. Y2K problem where one did not anticipate that software systems would last beyond the turn of the century is a good example of unwittingly designing systems that would become obsolete. For instance, if one were to use a structure in UID that encodes as a part the year of birth for the assigned individual, one needs to make sure that Y2K type of problem is avoided. One cannot always anticipate what future needs may evolve that requires format changes but one should assume that it is likely and thus making the format such that it remains compatible with any future format changes (embodied typically as version numbers).

Privacy Issues

There are a number of ID cards and identification numbers around the world. They range in lengths from 8 to 16 digits depending on the size of the country. Some encode gender. Many involve a physical card that has a significant amount of personal content. Some use machine-readable technologies that range from barcodes to smartcard/RFID.

There has been a great deal written about the privacy implications of national ID cards in various countries. For example, consider the Electronic Privacy Information Center's (EPIC's) analysis of the US Department of Homeland Security REAL ID project.[1] Many of the issues raised in the EPIC analysis pertain to public policy and to the uses of the ID, especially as relevant to specific US concerns. A policy analysis of the privacy implications of the UID is beyond the scope of this paper. However, the EPIC document raises technical issues as well. We address the relevant ones below, as well as point out the specifics of the UID that were designed to protect the resident's privacy.

1. The UID program does not issue a smartcard or any type of card or mandate any machine-readable format such as a barcode or RFID. Even if a barcode were used on the UID letter delivered to the citizen, it would merely encode the UID itself, and not personal information pertaining to the citizen. For example, even the

[1] http://epic.org/privacy/id-cards/

name of the card-holder would not be in the barcode. This would make it difficult for an unauthorized entity to glean any information about the resident.

2. The UID number has been carefully designed to not disclose personal information about the resident including - the regional, ethnic and age information. The US Social Security Number, for example, has enough of a pattern that an expert can guess a person's number from their birth-date and from the location at which it was issued.[2] It is also possible to guess the date and location at which the card was issued from the Social Security Number. The UID is a random number that makes guesswork virtually impossible.
3. The UID approach is designed on an on-line system – data is stored centrally and authentication is done online. This is a forward-leaning approach that makes it possible to avoid the problems associated with many ID card schemes.

Process for De-duplication

One of the key features of the UID system is ensuring uniqueness in issuing the UID number. This means that each resident can get one and only one UID number and conversely the UID number can be used by one resident alone. The only way to ensure uniqueness with a high degree of accuracy covering a large population of 1.2billion is by the use of biometrics. Biometrics are physical markers of an individual that are unique to and individual such as fingerprints, iris patterns, face structure etc. By capturing and storing some of these biometric markers we will be able to uniquely identify residents and hence assign unique numbers to them and authenticate them accurately during service delivery. The UID Biometrics committee in the following report recommends the capture of all 10 fingerprints, photograph of the face and a photo of the Iris in order to de-duplicate and authenticate residents:
http://uidai.gov.in/documents/Biometrics_Standards_Committee%20report.pdf
Since biometric information contain no ordering and hence cannot be indexed like text based information, when a resident applies for a UID with his/her fingerprints, iris and photo of face, these biometrics have to be compared against the entire UID database (existing residents with UIDs) to ensure that this new applicant is indeed unique and has not already been allotted a UID (even under a different name, address etc). This 1:N biometric comparison (N=size of the UID database) is the most compute intensive operation of the UID server system.

[2] http://www.washingtonpost.com/wp-dyn/content/article/2009/07/06/AR2009070602955.html

Alpha-Numeric Values

There have been national numbering systems that have used alpha-numeric values for UID string. That has a presumed advantage of being more secure because it will be harder to guess, and provides more available numbers in compact length. We did not conside this as being a viable option for us given the multi-lingual society that also has high levels of functional illiteracy.

Memorization of UID

This section is about how long the string length should be. In short, the string has to be as short as possible but that meets density requirement and does not include alphabet characters, just numbers. It is important to keep the UID simple and small to help residents to remember their number. Since it is likely that increasingly the UID will be used by several service providers (government agencies, private institutions and NGOs) it is important for a resident to be able to remembers it in the absence of a token such as a card. Firstly the use of the hindu-arabic numeral system(0,1,2,3,4,5,6,7,8,9) is suggested since these numerals are recognized/used by the largest subset of people in the country. Secondly we suggest the use of 12 digits (11 + 1 check sum) since 11 digits gives us a 100billion number space which in turn can provide a low density of used numbers. Thirdly the UID authority can send a letter to the resident with the assigned UID number and the details of the person's record as recorded in the UID database. A small perforated portion can be used as a tear-off portion that can be used at the time of authentication.

De-activation of UIDs

When a person dies, eventually one would see a need to de-activate the UID associated with the person. One simple way to deal with that is to flag UID record as inactive once one confirms the death. In a country of a billion people, updating UID records based on the death register is not easy, especially since a large number of cases of death are not reported and moreover the registration of births and deaths is maintained at local distributed levels across the country making it difficult to update them at a central UID system. One way to ensure that UIDs are not misused by others after a person's death is to inactivate the UID if it has not been used say in 10 years (timeout can be changed). Using the lack of activity as being an indicator of being deceased is also not without its pitfalls. In the case that a UID in inactivated of a person who has simply not authenticated himself/herself in a long time, s/he can simply activate their UID by a simple re-activation procedure that involves authentication.

UID static PIN and dynamic PIN

In order to authenticate(ascertain it is who s/he claims to be) a resident needs to provide his/her UID number as well as say a biometric marker – such as a fingerprint. The UID authentication system will pull up the person's record using the UID number as key and will match the input fingerprint with the fingerprint template stored. If there is a match(above a certain threshold) then a positive response is returned to the authenticator such as a bank, ration shop etc.

In order to strengthen the authentication level other factors can be added such as a static PIN(known-to and changeable-by the resident only) code that is know to only the resident, much like a 4 digit PIN on a ATM card. A high value banking transaction could ask for a biometric+PIN authentication. If an even higher level of assurance is sought in authentication then a 4 digit dynamic PIN can be generated by the UID system and say sent to the resident's mobile phone via SMS, this dynamic PIN needs to be entered into the authentication device in addition to ones UID and fingerprints. The choice of assurance levels and hence the factors used in authentication is specified by the registrar or authentication user (typically service delivering agency) not the UID authority.

Recovering lost UID number

When a resident loses his/her UID number (and the associated UID letter) a process is needed to recover the UID number. This requires a 'Identity Check' which involves capturing the resident's biometric and comparing it against the entire UID database in order to locate the UID number of the resident – this is the same 1:N check that is undertaken during initial UID enrolment. Since this is an expensive compute intensive operation, the UID system needs to discourage repeated and frivolous applications of lost UID number – perhaps through a fee for the UID recovery service.

One should set-up a process to change some of the primary information, such as name change, change of primary residence based on the same KYR(know your resident) verification processes that was used for issuing a UID. The details of the KYR verification process is documented in the KYR committee report:
http://uidai.gov.in/documents/UID_DDSVP_Committee_Report_v1.0.pdf.

Types of Identifiers and the Identity Mapper

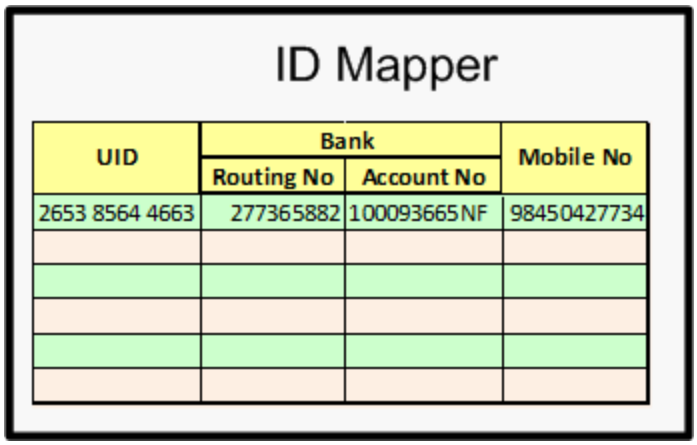
There are several different types of identifiers that are provided by different govt. and private agencies. Examples are driver license, ration card, election photo identity card (EPIC), PAN card, passport, NREGA job card, RSBY health insurance card etc. Examples of identifiers that are used for financial transactions include bank account, post office ac-

count numbers etc. Thirdly there are identifiers meant for communication such as mobile numbers, landline phone, email addresses etc.
While the above identifiers of an individual are relevant in specific sectors such as finance, health, communication etc, the UID is a pure identifier which is not tied to any particular sector or application and this abstract quality of the UID has distinct benefit in delivering cross sectoral benefits. There are distinct benefits in using a pure ID to stitch together transactions cutting across multiple sectors, some of the identifiers are also addressable in their local context and hence not global, the UID help make these identifiers globally addressable. The below example highlights the use of the UID for financial inclusion programs.

Financial Inclusion Example: Let's take the example of NREGA and look at how direct benefits can be delivered to the poor, by directly depositing their NREGA earnings to a bank account. Currently several innovative pilots have been implemented which integrate the NREGA payments workflow to direct bank deposit and also a payment solution that include business correspondents as well POS devices and/or mobile technologies to deliver NREGA cash payments at the village level. These solutions require a custom integration of the banking, mobile and NREGA backend that can be expensive and time consuming.
If there exists a backend financial inclusion infrastructure that is able to make direct deposit to a resident bank account at any bank using his/her UID through a national payments switch then it simplifies the way direct benefits is delivered by any service provider such as NREGA across the country. One of the key components of such a FI infrastructure is the ID mapper, that maps the key IDs of the resident such as UID, Bank Account and mobile number (optional) in order to make direct deposits easily.

ID Mapper : The ID Mapper links the various identifiers of a given resident. For instance the UID, Bank Account and mobile number are 3 key identifiers that are useful for financial inclusion as shown in the diagram below. Now it is easy to target payments to a UID since the ID Mapper can resolve a UID to a specific Bank Account and infact a SMS can be sent to the person's mobile number which is also available in the ID Mapper. The key benefit of a central ID Mapper is the resident has full flexibility to change the bank where he/she banks and automatically the direct benefits of the govt. will target the latest bank account that the resident has updated on the ID Mapper, similarly the resident has full flexibility to choose the mobile provider and device. An advanced ID Mapper can also be policy driven for instance the resident can upload multiple bank accounts and set policies as to which bank account to use for what type of transaction.

UID Numbering Scheme



Entity IDs - Unique Identification Number for Organizations

Institutions like Government departments, schools and even companies can benefit by using a UID like Identifier – this is called an Entity ID.

Since the UID will potentially be used as a primary identifier in several transactions in the financial, health, food distribution, job creation schemes and transactions it is important to assign an entity ID to the service delivery organization. For instance a financial transaction to transfer money might take the form:

TransferMoney(From_UID, To_UID, Amount);

Where the From_UID could be an entity UID of the block level NREGA entity and the To_UID can be that of the resident to who the amount is being transferred. This symmetric treatment of both *to* and *from* fields simplifies the end-to-end system.

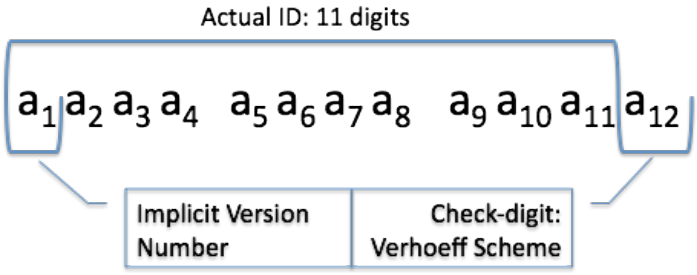
Principles and Requirements

This proposed design is based on several principles and requirements discussed among authors. The principles and requirements that emerged are:

- The number should be semantics-free. Apart from reserving two special values for the first digit and the last digit (used as check digit), the remaining number value will be generated using a randomized process at the time one requests a new ID.
- 12 decimal digits will be sufficient to satisfy all the requirements, outlined below.
- The number should support 80 to 100 billion ID's at the outset. In other words, there should be at least 11 digits available for actual numbering.

- The number should be extensible with backwards compatibility to have more decimal digits, for example, 20 digits, if a need should arise in future. This can be accomplished by reserving a value of 0 for first digit of the number
- There should be well known, reserved numbers for purposes outlined below.
- The number should have an error detecting (but not necessarily error correcting) check digit that detects as many data entry errors as possible.
- To make it harder to guess a correct valid ID assigned to anybody we propose a much larger space (80 billion IDs) than the maximum number of IDs needed at anytime (estimated to be about no more than 6 billion IDs). This notion is captured by the *density*, which we define as a ratio of total number of ID's assigned to the total number of IDs available. Our target is that by design we keep this ratio below 0.05 at all times. The sparseness of the numbers will make it difficult to simply guess numbers. The check digit is another form of sparseness but, being a publicly disclosed algorithm, will not prevent guessing. The check digit enables error detection at data entry.
- There may be a need to assign UIDs to entities such as schools, companies and other institutions to facilitate symmetry in transactions between residents and entities. For example, financial transactions for NREGA scheme may deposit NREGA wages direct from a Gram Panchayat to a particular resident. Such transactions could take the form: TransferMoney(GramPanchayat_UID, Resident_UID, Amount). While several other backend systems need to be in place for such transactions to take place, currently we are making a provision to assign up to ten billion entity UIDs. This requirement is met by reserving the value 1 for first digit (a_1=1) to entity UID's.

Number Design: 12 Digits



The format of 12-digit number is discussed below.

1. The Version Number: Some digits may be reserved for specific applications. This is an implicit form of a version number embedded into the numbering scheme. We recommend the following reservations:

   0- numbers ($a_1 = 0$) could be used as an "escape" for future extensions to the length of the number. For example, in future if we need 16 digit numbers, then we could say that 0 means that the number is 16-digits. As of now we can simply declare all 0-numbers as TBD (to be decided).

   1- numbers($a_1 = 1$) could be reserved for entities rather than individuals. Alternatively, 11- could be reserved for entities (or 111-) to match the size of the reserved space to the number of entities expected.

   We could use 2-9 numbers ($a_1 = 2,3...9$) right away to assign UIDs. That is 80 billion numbers -- plenty of space.

2. Number Generation: The numbers are generated in a random, non-repeating sequence. There are several approaches to doing this in the computer science literature. The algorithm and any"seed" chosen to generate IDs should not be made public and should be considered a national secret.

3. Lifetime: Individual UID is assigned once, at inception, and remain the same for the lifetime of the person, and for a specified number of years beyond. At this point there is no consideration of reusing numbers.

4. Entity ID's: We expect that entity ID numbers (1- numbers) will have different rules for periods of validity and retirement.

5. The Checksum: There are several schemes possible. We recommend the Verhoeff scheme. More on this in the section titled Checksum.

The Checksum

The point of the Checksum is to eliminate data-entry errors. There have been a number of studies of data-entry errors.

TABLE 1.1
Sample Errors

Error Type	Actual Number	Transmitted Error
Single digit	191433	191933
Transposition of adjacent digits	191433	191343
Jump transposition	191433	193413
Twin	191433	191455
Phonetic	191433	194033
Jump twin	191433	393433

TABLE 1.2
Common Error Patterns

Error Type	Form	Relative Frequency
Single digit	a → b	79.1%
Transposition of adjacent digits	ab → ba	10.2%
Jump transposition	abc → cba	0.8%
Twin	aa → bb	0.5%
Phonetic	a0 ↔ 1a*	0.5%
Jump twin	aca → bcb	0.3%

*For a = 2, . . . , 9.

For example, the tables above show errors as measured by Verhoeff, and reproduced in Kirtland.[3] The check digit must certainly guard against the top two errors first: *single digit errors* and *adjacent digit transposition*, while also detecting other errors with high probability. Many check digit approaches are in use in such identifiers as the EAN barcode, the ISBN number, the serial numbers on currency notes, check numbers and so on. Surprisingly, many of these schemes do *not* detect single-digit and transposition errors. Since we expect nearly a billion IDs to be issued by UID-AI, we must ensure that the check digit scheme we select will absolutely minimize data-entry errors; this will reduce server load, customer service load, and overall aggravation. There is one scheme that meets our requirements: the Verhoeff Scheme. This scheme is relatively complex, and in the days before ubiquitous computing, there was a tendency to avoid it in favor of simpler schemes. In this day and age however, and at the scale of the UID, precision must be the goal.

[3] Kirtland, Joseph. *Identification Numbers and Check Digit Schemes*. The Mathematical Association of America, 2001.

The Verhoeff scheme catches all single errors and all adjacent transpositions. It also catches >95% of twin errors and >94% of jump transpositions.[4]

#	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	*6*	*7*	*8*	*9*	*5*
2	2	3	4	0	1	*7*	*8*	*9*	*5*	*6*
3	3	4	0	1	2	*8*	*9*	*5*	*6*	*7*
4	4	0	1	2	3	*9*	*5*	*6*	*7*	*8*
5	5	*9*	*8*	*7*	*6*	0	*4*	*3*	*2*	*1*
6	6	*5*	*9*	*8*	*7*	*1*	0	*4*	*3*	*2*
7	7	*6*	*5*	*9*	*8*	*2*	*1*	0	*4*	*3*
8	8	*7*	*6*	*5*	*9*	*3*	*2*	*1*	0	*4*
9	9	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	0

Figure 1: The Dihedral Group Operator from (2). Red digits are not commutative.

We will not describe the Verhoeff scheme in detail here, but the basic idea is as follows. First, instead of using standard addition modulo 10 or 11, as is common in other check digit schemes, the Verhoeff Scheme uses the group operator shown in the table above. It can be seen that this operator, indicated by a #, is generally non-commutative. This helps capture transposition errors. The table derives from a group called the dihedral group D_5, which are symmetries of the pentagon.

Second, Verhoeff defines a permutation such as (0)(14)(23)(56789). Other permutations can also be used. The check digit "polynomial" works by applying the "nth power" of permutation to the nth element in the ID, and composing them together using the operator # defined above. The scheme can be implemented quite easily by an experienced programmer. A freeware version for PC's, cell phones and the web can easily be obtained.

It is important to emphasize that the checksum scheme is *not* intended to be a secret. On the contrary, it needs to be published widely so that and users of the UID system are able to detect keying and transmission errors.

[4] http://www.cs.utsa.edu/~wagner/laws/verhoeff.html

Other Issues

We have certain recommendations and comments:

- We recommend that UID-AI set up a number issuing system: when the registrar needs to issue a number, they submit the enrollment information and requests a number from the central server. The server generates the ID number returns it to the registrar.

  We suggest creating a "dummy UID" which is never issued to anyone. This number can be displayed in public but will never violate anyone's privacy.

- We recommend the creation of legislation around the display and publication of individuals' UID's.

- We recommend penal provisions to deter individuals from committing identity fraud such as trying to obtain multiple UIDs