Assigning Driver's License Numbers

JOSEPH A. GALLIAN
University of Minnesota
Duluth, MN 55812

"You know my name
look up the number"

John Lennon and Paul McCartney, *You know my name*,
single, B-side of *Let it be*, March 1970.

Introduction

Among the individual states, a wide variety of methods are used to assign driver's license numbers. The three most common methods, a sequential number, the social security number, and a computer-generated number, are uninteresting mathematically. On the other hand, many states encode data such as month and date of birth, year of birth, and sex in ways that involve elementary mathematics. Seven states go so far as to employ a check digit for possible detection of forgery or errors. Several states assign driver's license numbers by applying complicated hashing functions to the first, middle, and last names and formulas or tables for the month and date of birth. Surprisingly, the assignment of numbers is not always injective. In Michigan, for instance, there are 56 numbers whose inverse image has two or more members. New Jersey incorporates eye color into the number. Some states keep their method confidential. In a few instances, administrators of the license bureaus do not know the method used to assign numbers in their state! In this paper we discuss some of the methods we have uncovered.

Check Digit Schemes in General Use

Schemes for the assignment of identification numbers are extremely varied in methodology and in the information encoded. Most interesting to mathematicians are those that incorporate an extra digit for the detection of errors or fraud. Although the purpose of this paper is to analyze the methods used for driver's license numbers, it is worthwhile to begin with a brief survey of the methods employed to assign check digits to the most ubiquitous numbers in use and to provide a theoretical result that delineates their limitations.

The simplest and least effective method for assigning a check digit is to use the remainder or inverse of the remainder of the identification number modulo some number. For airline tickets, UPS packages, and Federal Express mail the check digit is the identification number modulo 7. At the bottom of FIGURE 1 we see the airline ticket number 17000459570 (the airline code 012 is not used in the calculation) is assigned a check digit 3 since $17000459570 \equiv 3 \bmod 7$.

U.S. postal money orders use the remainder modulo 9 while VISA traveler's checks use the inverse of the number modulo 9. Thus, the check digit for the VISA number 1002044679091 is 2 since $1002044679091 \equiv 7 \bmod 9$.

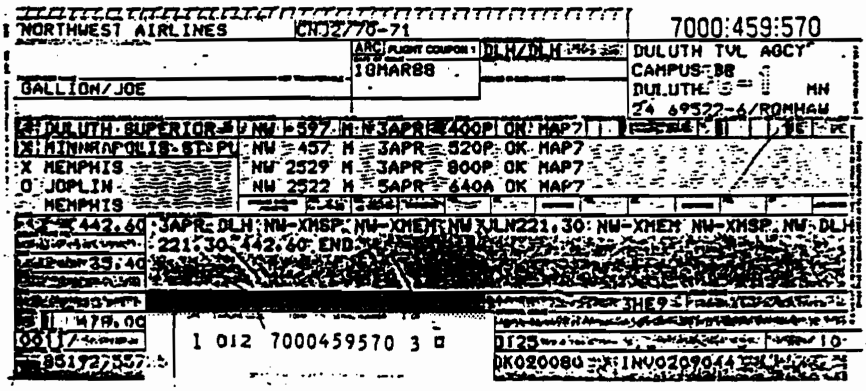

FIGURE 1
Airline ticket with number 17000459570 and check digit 3.

The modulo 7 schemes detect all errors involving a single digit except those where b is substituted for a and $|a-b|=7$. Likewise, an error of the sort $\cdots a_i \cdots a_j \cdots \rightarrow \cdots a_j \cdots a_i \cdots$ will go undetected if $|a_i - a_j| = 7$ or if 6 divides $j-i$.

The modulo 9 schemes are slightly better at detecting single-digit errors: Only a substitution of a 9 for a 0 or vice versa goes undetected. On the other hand, the only errors of the form $\cdots a_i \cdots a_j \cdots \rightarrow \cdots a_j \cdots a_i \cdots$ that are undetected are those involving the check digit itself. (A quick proof of this is to observe that the residue of a number modulo 9 is the residue of the sum of its digits modulo 9.)

Nearly all methods for assigning a check digit to a string of digits involve a scalar product of two vectors and modular arithmetic. For a string $a_1 a_2 \cdots a_{k-1}$ and a modulus n, many schemes assign a check digit a_k so that

$$(a_1, a_2, \ldots, a_k) \cdot (w_1, w_2, \ldots, w_k) \equiv 0 \bmod n.$$

We call such schemes *linear* and we call the vector $(w_1, w_2, \ldots, w_k)$ the *weighting vector*. The Universal Product Code (UPC) used on grocery items employs the weighting vector $(3,1,3,1,3,1,3,1,3,1,3,1)$ with $n=10$; the International Standard Book Number (ISBN) utilizes $(10,9,8,7,6,5,4,3,2,1)$ and $n=11$; banks in the U.S. use $(7,3,9,7,3,9,7,3,9)$ with $n=10$; many Western countries use $(7,3,1,7,3,1,\ldots)$ with $n=10$ to assign check digits to numbers on passports. Notice that the division schemes mentioned at the outset of this section are also linear with weighting vectors of the form $(10^{k-2}, 10^{k-3}, \ldots, 10^0, \pm 1)$.

The error-detecting capability of linear schemes is given by the following theorem.

THEOREM. *Suppose a number $a_1 a_2 \cdots a_k$ satisfies the condition $(a_1, a_2, \ldots, a_k) \cdot (w_1, w_2, \ldots, w_k) \equiv 0 \bmod n$. Then the single error occasioned by substituting a_i' for a_i is undetectable if and only if $(a_i' - a_i)w_i$ is divisible by n and a sole error of the form $\cdots a_i \cdots a_j \cdots \rightarrow \cdots a_j \cdots a_i \cdots$ is undetectable if and only if $(a_i - a_j)(w_i - w_j)$ is divisible by n.*

Proof. If a_i' is substituted for a_i, then the dot product of the correct number and the incorrect number differ by $(a_i' - a_i)w_i$. Thus, the error is undetected if and only if $(a_i' - a_i)w_i \equiv 0 \bmod n$.

Now consider an error of the form $\cdots a_i \cdots a_j \cdots \rightarrow \cdots a_j \cdots a_i \cdots$. Here the dot product of the correct number and the incorrect number differ by

$$(a_i w_i + a_j w_j) - (a_j w_i + a_i w_j) = (a_i - a_j)(w_i - w_j)$$

The conclusion now follows as before.

Since the most common moduli are 10 and 11, the following corollary is worth mention.

COROLLARY. *Suppose an identification number* $a_1 a_2 \cdots a_k$ *satisfies*

$$(a_1, a_2, \ldots, a_k) \cdot (w_1, w_2, \ldots, w_k) \equiv 0 \bmod n$$

where $0 \leqslant a_i < n$ *for each* i. *Then all single-digit errors occurring in the* i*th position are detectable if and only if* w_i *is relatively prime to* n *and all errors of the form* $\cdots a_i \cdots a_j \cdots \to \cdots a_j \cdots a_i \cdots$ *are detectable if and only if* $w_i - w_j$ *is relatively prime to* n.

The above theorem verifies our claims about the error-detection capability of the schemes used on money orders and airline tickets. It also explains why the bank and passport schemes will detect some errors of the form $\cdots abc \cdots \to \cdots cba \cdots$ while the UPC code will detect no such errors. Observe that because 11 is prime the ISBN code detects 100% of all single-digit errors and 100% of all errors involving the interchange of two digits. But there is a price to pay for using the modulus 11: The number a_k needed to satisfy the condition may be 10, which is two digits. In this case, an alphabetic character such as X or A is used or such numbers are not issued. As we will see below there are schemes that use the modulus 11 that do not resort to an alphabetic character, but there is a price to pay for this too: Not all transposition errors are detectable. More information about check digits schemes in use can be found in [1], [2], [3], [4], [6], [7], [8].

Check Digits on Driver's License Numbers

The state of Utah assigns an eight-digit driver's license number in sequential order, say $a_1 a_2 \cdots a_8$, then appends a check digit a_9 using a linear scheme with weighting vector $(9, 8, 7, 6, 5, 4, 3, 2, 1)$ and modulus 10. This method is identical to that used by the American Chemical Society for its chemical registry numbers. Assuming that all errors are equally likely,[1] this method detects 73/81 or 90.1% of all single-digit errors and 100% of all transposition errors (i.e., errors of the form $\cdots ab \cdots \to \cdots ba \cdots$).[2]

To verify the single-error detection rate, observe from our theorem that in positions 2, 4, 6, and 8 substitution of b for a will go undetected when $|a - b| = 5$; in position 5, a substitution of b for a will go undetected when a and b have the same parity. Thus in each of positions 2, 4, 6, and 8 there are 10 undetected errors among 90 possible errors while in the fifth position, 40 of the 90 possible errors are undetected.

[1] In practice all errors are not equally likely. One study [7, p. 15] revealed that a substitution of a "5" for a "3" was 17 times as likely as a substitution of a "9" for a "1." However, available data are insufficient to assign reliable probabilities to the various error possibilities.

[2] A highly publicized error of this kind recently occurred when Lt. Col. Oliver North gave U.S. Assistant Secretary of State Elliot Abrams an incorrect Swiss bank account number for depositing $10 million for the contras. The correct number begins with "386"; the number North gave to Abrams begins with "368."

So, in all, 80 of 810 errors are undetected.

Someone working for the Canadian province of Quebec, probably having seen a scheme like the one used by Utah somewhere, came up with the laughable weighting vector $(12, 11, 10, 9, \ldots, 2, 1)$ with modulus 10 to assign a check digit. Of course, any error in the third position is undetectable and weights of 12 and 11 have the same effect as the weights 2 and 1.

Newfoundland uses the weighting vector $(1, 2, 3, 4, 5, 6, 7, 8, 1)$ with modulus 10. This is nearly identical to the Utah scheme except that it will not detect the event that the first and last digit are interchanged.

Three states use a modified linear scheme with modulus 11. New Mexico and Tennessee append a check digit a_8 to $a_1 a_2 \cdots a_7$ as follows: First calculate

$$x = -(a_1, a_2, \ldots, a_7) \cdot (2, 7, 6, 5, 4, 3, 2) \bmod 11.$$

If $x = 0$, a_8 is 1; if $x = 10$, $a_8 = 0$; otherwise $a_8 = x$. This method catches 100% of all single-digit errors. Furthermore, the only errors of the form $\cdots a_i \cdots a_j \cdots \to \cdots a_j \cdots a_i \cdots$ that go undetected are those where $i = 1$ and $j = 7$ (an unlikely error indeed) and some involving the check digits 0 and 1. Assuming that all transposition errors are equally likely,[3] this method detects 98.2% of such errors. The Vermont scheme is the same as the one used by New Mexico except that when $x = 0$, the letter "A" is the check. This method, like the ISBN method, yields a 100% detection rate for both single-digit and transposition errors, but utilizes two formats for numbers. Notice that there would be nothing lost if the weighting vector began with 8 instead of 2 and there would be a slight gain since errors of the form $a_1 a_2 \cdots a_7 a_8 \to a_7 a_2 \cdots a_1 a_8$ would be detectable.

The state of Washington and the province of Manitoba use a check digit scheme devised by IBM in 1964 to assign a check digit. The license number is a blend of 12 alphabetic and numeric characters. To compute the Washington check digit, alphabetic characters are assigned numeric values as follows: $* \to \emptyset$, $A \to 1$, $B \to 2, \ldots, I \to 9$, $J \to 1$, $K \to 2, \ldots, R \to 9$, $S \to 2$, $T \to 3, \ldots, Z \to 9$. (Notice the aberration at S.) The 12-character license number, after an alphabetic to numeric conversion, then corresponds to a string of digits $a_1 a_2 \cdots a_{12}$ with a_{10} as the check digit calculated as $|a_1 - a_2 + a_3 - a_4 + \cdots + a_9 - a_{11} + a_{12}| \pmod{10}$. Interestingly, the use of the absolute value actually makes the method nonlinear and reduces the error detection capability of the scheme. It would have been better to use the linear scheme with weighting vector $(1, 9, 1, 9, \ldots, 1) \bmod 10$.

South Dakota and Saskatchewan employ another nonlinear scheme developed by IBM to assign its check digit. In South Dakota, a six-digit computer-generated string is assigned a check digit as follows. Each of the second, fourth, and sixth digits is multiplied by 2 and the digits of the resulting products are summed (e.g., a 7 yields $1 + 4 = 5$ while a 3 yields 6). This resulting total is then added to the digits in the first, third, and fifth positions. The check digit is the inverse modulo 10 of this tally. (Alternatively, the check digit is $(10 - ((\Sigma_{i\,\text{even}}\ (2a_i + \lfloor 2a_i/10 \rfloor) + \Sigma_{i\,\text{odd}}\ a_i) \pmod{10})) \bmod 10$.) Thus, the check digit for 263743 is $-(1 + 2 + 1 + 4 + 6 + 2 + 3 + 4) \bmod 10 = 7$. This method is used by credit card companies, many libraries, and drug stores in the U.S. and by banks in West Germany, although in some instances it is the digits in the odd positions that are multiplied by 2. It detects 100% of all

[3] In reality, the likelihood of a transposition error depends on the pair of digits as well as the positions. But as before, reliable data for these occurrences are unavailable.

single-digit errors and 97.8% of transposition errors. To see that all single-digit errors are detected, observe that distinct digits contribute distinct values to the sum. To compute the detection rate for errors of the form $\cdots ab \cdots \to \cdots ba \cdots$, suppose such an error is undetected. We consider four cases. For simplicity, assume a in the correct number is in position 2, 4 or 6. The alternative case gives the same result.

Case 1. $a, b < 5$
Then $2a + b \equiv 2b + a \bmod 10$.
Thus $a - b = 0$ and $a = b$.

Case 2. $a < 5$, $b \geqslant 5$
Then $2a + b \equiv 2b - 9 + a \bmod 10$.
It follows that $b - a = 9$ so that $b = 9$ and $a = 0$.

Case 3. $a \geqslant 5$, $b < 5$
Then $2a - 9 + b \equiv 2b + a \bmod 10$.
So, $a - b = 9$ and $a = 9$ and $b = 0$.

Case 4. $a \geqslant 5$, $b \geqslant 5$
Then $2a - 9 + b \equiv 2b - 9 + a \bmod 10$.
Thus $a - b = 0$ and $a = b$.

So all transposition errors except $09 \leftrightarrow 90$ are detected. Since there are 90 possible transposition errors, the error detection rate is 88/90 or 97.8%.

It is worth noting that Gumm [4] has shown that it is not possible to improve upon these rates with any system that uses addition modulo 10 to compute the check digit without utilizing an extra character, as was the case for the New Mexico scheme.

Wisconsin appends a check digit to a 13-digit string. Unfortunately, I have not been able to figure out how this scheme works. I know it isn't linear; for if so, the weighting vector $(w_1, w_2, \ldots, w_{13}, w_{14})$ could be determined by gathering up a large number of valid license numbers to produce a system of linear equations with the w_i's as the unknowns. I have done this for modulo 10 and 11 to no avail. To circumvent any peculiarity that might arise involving a check digit of 10 in a modulo 11 scheme (e.g., New Mexico), I avoided numbers with a check digit of 0 or 1.

Encoding Personal Data

Here is the driver's license number of a Wisconsin resident: E 425-7276-9176-07. What information about the holder can you deduce from this number: year of birth, day and month of birth, sex, name? None of these is obvious. Let's go the other direction. I am a resident of Minnesota. I was born on January 5, 1942, and my middle name is Anthony. From this can you deduce my driver's license number?

Eleven states assign their driver's license numbers with hashing functions applied solely to personal data. A good hash function should be fast and minimize collisions (see [5, pp. 506–544] for a detailed discussion of this topic). Of course, there will be occasions when two or more individuals have enough personal data in common that collisions will occur. Most states have a tie-breaking mechanism to handle this situation. Coding license numbers only from personal data enables automobile insurers, government entities, and law enforcement agencies to determine the numbers when necessary.

Washington uses a complicated blend of name, check digit, and codes for the month and date of birth to assign its numbers. This 12-digit identifier consists of the first five letters of the surname; the first and middle initials (* is used when a name has less than five characters, or there is no middle initial); the year of birth subtracted from 100 (we suspect this is done to disguise the year of birth); a check digit; a code for the month of birth; and a code for the day of birth. For instance, Fielding Mellish (no middle name) born on 10/29/42 receives the identifier MELLI F* 587P9. When checked against a file of 1.6 million items, this scheme yielded duplicates at the rate of 0.03% and only one number appeared as many as four times. (Most of the duplications represented twins.) To ensure that the correspondence between individuals and numbers is injective, 17 alternate codes for month and year of birth are available. For example, an S can be used instead of a B for January or a Z instead of a 9 for the year of birth. Interestingly, the check digit is invariant under all alternate coding. The primary code and one alternate for months is given in TABLE 1 and the code for the days is given in TABLE 2. Notice the absence of completely predictable patterns.

TABLE 1. Washington code for months.

Months	Codes	Alternate Codes
January	*B*	*S*
February	*C*	*T*
March	*D*	*U*
April	*J*	1
May	*K*	2
June	*L*	3
July	*M*	4
August	*N*	5
September	*O*	6
October	*P*	7
November	*Q*	8
December	*R*	9

TABLE 2. Washington code for days.

1 - *A*	7 - *G*	13 - *L*	19 - *R*	25 - 5	31 - *U*
2 - *B*	8 - *H*	14 - *M*	20 - 0	26 - 6	
3 - *C*	9 - *Z*	15 - *N*	21 - 1	27 - 7	
4 - *D*	10 - *S*	16 - *W*	22 - 2	28 - 8	
5 - *E*	11 - *J*	17 - *P*	23 - 3	29 - 9	
6 - *F*	12 - *K*	18 - *Q*	24 - 4	30 - *T*	

Illinois, Florida, and Wisconsin encode the surname, first name, middle initial, date of birth, and sex by a quite sophisticated scheme. The first character of the license number is the first character of the name. The next three characters are obtained by applying the "Soundex Coding System" to the surname as follows:

1. Delete all occurrences of h and w.
2. Assign numbers to the remaining letters as follows:
   $b, f, p, v \to 1$ $\qquad$ $l \to 4$
   $c, g, j, k, q, s, x, z \to 2$ $\qquad$ $m, n \to 5$
   $d, t \to 3$ $\qquad$ $r \to 6$
   (No values are assigned to a, e, i, o, u, and y.)
3. If two or more letters with the same numeric value are adjacent, omit all but the first. (Here a, e, i, o, u, and y act as separators.) For example, Schworer becomes Sorer and Hughgill becomes Ugil.

4. Delete the first character of the original name if still present.
5. Delete all occurrences of a, e, i, o, u, and y.
6. Use the first three digits corresponding to the remaining letters; append trailing zeros if less than three letters remain.

Here are some examples: Schworer → S-660; Hughgill → H-240; Skow → S-000; Sachs → S-200; Lennon → L-550; McCartney → M-263.

We parenthetically remark that the Soundex System was designed so that likely misspellings of a name would nevertheless result in the correct coding of the name. For example, frequent misspellings of my name are: Gallion, Gillian, Galian, Galion, Gilliam, Gallahan, and Galliam. Observe that all of these yield the same coding as Gallian. We also mention that the above method differs somewhat from the system called Soundex by Knuth in [5, p. 392].

The next three digits are determined by summing numbers that correspond to the first name and middle initial. The scheme for doing this begins with the block 000 for the letter A and makes jumps of 20 for especially common names and each subsequent letter of the alphabet. A small portion of this scheme is given in TABLE 3. The values assigned to the middle initial are given in TABLE 4.

So Aaron G. Schlecker would be coded as S426-007 (S426 from Schlecker; 000 for Aaron +7 for "G"), while Anne P. Schlecker would be coded as S426-055.

The last five digits of Illinois and Florida numbers capture the year and date of birth as well as the sex. In Illinois, each day of the year is assigned a three-digit number in sequence beginning with 001 for January 1. However, each month is assumed to have 31 days. Thus, March 1 is given 063. These numbers are then used to identify the month and day of birth of male drivers. For females, the scheme is identical except January 1 begins with 601. The last two digits of the year of birth, separated by a dash (probably for camouflage), are listed in the 5th and 4th positions from the end of the driver's license number. Thus, a male born on July 18, 1942, would have the last five digits 4-2204 while a female born on the same day would have 4-2804. When necessary, Illinois adds an extra character to avoid duplications.

TABLE 3. Illinois, Florida, Wisconsin given name or first initial code.

000	–	A
020	–	Albert, Alice
040	–	Ann, Anna, Anne, Annie, Arthur
060	–	B
080	–	Bernard, Bette, Bettie, Betty
100	–	C

TABLE 4. Illinois, Florida, Wisconsin middle initial code.

0 - none	10 - J
1 - A	11 - K
2 - B	12 - L
3 - C	13 - M
4 - D	14 - N, O
5 - E	15 - P, Q
6 - F	16 - R
7 - G	17 - S
8 - H	18 - T, U, V
9 - I	19 - W, X, Y, Z

Among the 9,397,518 licenses on file on January 1, 1987, this occurred in 14,856 instances. Of these, 55 numbers corresponded to three individuals (excluding the extra digit). No number corresponded to more than three people.

The scheme to identify birthdate and sex in Florida is the same as Illinois except each month is assumed to have 40 days and 500 is added for women. For example, the five digits 49583 belong to a woman born on March 3, 1949.

Wisconsin employs the same scheme as Florida to generate the first 12 of their 14 characters. The thirteenth character is an integer issued sequentially beginning with 0 to people who share the same first 12 characters. The fourteenth character is a check digit.

A Missouri driver's license number has 16 characters. The first 13 characters are obtained by applying a hashing function to the first five letters of the surname, the first three letters of the first name and the middle initial. (The method of encoding is similar to that used by Florida.) The final three characters are a function of the month and day of birth and sex. For a male born in month m and day d the three digits are $63m + 2d$. For a female, the corresponding formula is $63m + 2d + 1$. Thus the number of a woman born on March 4 has the final three characters 198. To avoid duplications, Missouri assigns a 17th character. Among the first 3,921,922 numbers issued, 31,719 have a 17th character.

Last, we discuss the scheme employed by Minnesota, Michigan, and Maryland. The number is a function of last name, first name, middle name, month and date of birth. The first four characters are determined by the Soundex System, as was the case for Illinois, Florida, and Wisconsin. The first and middle names account for the next six characters and the same algorithm is applied to both names. In the majority of cases the first two characters of the name determine the desired three digits for each name (see TABLE 5 for a sample); for common pairs of leading letters such as Al or Ja, the third letter is invoked (see TABLE 6); 11 three-digit numbers are uniquely assigned to the 11 most popular names (e.g., 189 ↔ Edward; 210 ↔ Elizabeth).

TABLE 5. Minnesota, Michigan, Maryland code for first and middle names beginning with A except *Al*.

		A	027		
Aa	028	*Aj*	037	*As*	072
Ab	029	*Ak*	038	*At*	073
Ac	030	*Al*	—	*Au*	074
Ad	031	*Am*	066	*Av*	075
Ae	032	*An*	067	*Aw*	076
Af	033	*Ao*	068	*Ax*	077
Ag	034	*Ap*	069	*Ay*	078
Ah	035	*Aq*	070	*Az*	079
Ai	036	*Ar*	071		

The final three digits are based on month and date of birth (but not year). Each day of the year is assigned a three-digit number in a monotonically increasing fashion. Although the usual pattern is to alternate increments of 3 and 2, there are numerous seemingly random increments at unpredictable dates. The month of March illustrates this behavior well. Notice from TABLE 7 that March 1 is assigned 159. Subsequent days are assigned values by increments of 3 and 2 in alternating fashion until March 8. Then there is an increment of 5. Notice the jump of 20 between March 19 and March 20.

These gaps serve a practical purpose. In the event that there are two or more individuals born on the same month and date and with names so similar that the

TABLE 6. Minnesota, Michigan, Maryland code for first and middle names beginning with *Al*.

		Al	039		
Ala	040	*Alj*	049	*Als*	058
Alb	041	*Alk*	050	*Alt*	059
Alc	042	*All*	051	*Alu*	060
Ald	043	*Alm*	052	*Alv*	061
Ale	044	*Aln*	053	*Alw*	062
Alf	045	*Alo*	054	*Alx*	063
Alg	046	*Alp*	055	*Aly*	064
Alh	047	*Alq*	056	*Alz*	065
Ali	048	*Alr*	057		

TABLE 7. Minnesota, Michigan, Maryland code for dates in March.

	March - 158	
1 - 159	11 - 187	21 - 229
2 - 162	12 - 189	22 - 232
3 - 164	13 - 192	23 - 234
4 - 167	14 - 194	24 - 237
5 - 169	15 - 197	25 - 239
6 - 172	16 - 199	26 - 242
7 - 174	17 - 202	27 - 244
8 - 177	18 - 204	28 - 247
9 - 182	19 - 207	29 - 249
10 - 184	20 - 227	30 - 252
		31 - 254

hashing function does not distinguish between them (e.g., Jill Paula Smith and Jimmy Paul Smythe), the first person who applies for a license is assigned the number given by the algorithm while the second person is assigned the next higher number thereby using one of the numbers in the gap for birthdays. For example, if Jill Paula Smith is born on March 2 and is the first to receive the combination S530-441-675-162 as determined by the algorithm, then the next person who yields the same number is assigned S530-441-675-163 instead. Once all of the higher numbers in a gap have been assigned, lower numbers are used. Thus the third applicant with a name yielding the combination S530-441-675 born on March 2 would be assigned the last three digits 161. As of 1984, this scheme had not yielded any duplications among 4,468,080 people in Maryland while of Michigan's 6,332,878 drivers by 1987 there are 56 that have a number not uniquely their own. In fact, Michigan has two numbers that are each shared by four individuals and three that are each shared by three individuals. A common cause of duplication is the custom of naming a son after the father. When both share the same birthday a duplication occurs.

Summary

TABLE 8 summarizes the information the author has discovered about the coding of driver's license numbers. Unfortunately our knowledge is incomplete. Several states (e.g., Florida, New York, Minnesota, Missouri, Wisconsin) keep their methods confidential. In some of these cases we were able to determine the coding scheme by examining data. A question mark after the letter *X* indicates the corresponding item is used in the coding, but we do not know the method involved. The expression (A) after an *X* indicates that the corresponding item is part of a scheme that is an alternative to the social security number.

.E 6. Minnesota, Michigan, Maryland code for first and middle names nning with *Al*.

		Al	039		
la	040	*Alj*	049	*Als*	058
lb	041	*Alk*	050	*Alt*	059
lc	042	*All*	051	*Alu*	060
ld	043	*Alm*	052	*Alv*	061
le	044	*Aln*	053	*Alw*	062
lf	045	*Alo*	054	*Alx*	063
lg	046	*Alp*	055	*Aly*	064
lh	047	*Alq*	056	*Alz*	065
li	048	*Alr*	057		

TABLE 7. Minnesota, Michigan, Maryland code for dates in March.

	March - 158	
1 - 159	11 - 187	21 - 229
2 - 162	12 - 189	22 - 232
3 - 164	13 - 192	23 - 234
4 - 167	14 - 194	24 - 237
5 - 169	15 - 197	25 - 239
6 - 172	16 - 199	26 - 242
7 - 174	17 - 202	27 - 244
8 - 177	18 - 204	28 - 247
9 - 182	19 - 207	29 - 249
10 - 184	20 - 227	30 - 252
		31 - 254

ınction does not distinguish between them (e.g., Jill Paula Smith and Jimmy the), the first person who applies for a license is assigned the number given ;orithm while the second person is assigned the next higher number thereby of the numbers in the gap for birthdays. For example, if Jill Paula Smith is March 2 and is the first to receive the combination S530-441-675-162 as :d by the algorithm, then the next person who yields the same number is S530-441-675-163 instead. Once all of the higher numbers in a gap have gned, lower numbers are used. Thus the third applicant with a name he combination S530-441-675 born on March 2 would be assigned the last its 161. As of 1984, this scheme had not yielded any duplications among people in Maryland while of Michigan's 6,332,878 drivers by 1987 there are ave a number not uniquely their own. In fact, Michigan has two numbers each shared by four individuals and three that are each shared by three .s. A common cause of duplication is the custom of naming a son after the hen both share the same birthday a duplication occurs.

y

ummarizes the information the author has discovered about the coding of cense numbers. Unfortunately our knowledge is incomplete. Several states ida, New York, Minnesota, Missouri, Wisconsin) keep their methods confi- ı some of these cases we were able to determine the coding scheme by ; data. A question mark after the letter *X* indicates the corresponding item the coding, but we do not know the method involved. The expression (A) *X* indicates that the corresponding item is part of a scheme that is an : to the social security number.

TABLE 8. Summary of Schemes for Assigning Driver's License Numbers.

State	Social Security Number	Computer or Sequential Number	Check Digit	Last Name Coded	First Name Coded	Middle Name Coded	Year of Birth Coded	Month of Birth Coded	Day of Birth Coded	Sex Coded
Alabama		X								
Alaska		X								
Arizona	X									
Arkansas	X	X(A)								
California		X								
Colorado		X								
Connecticut		X						X		
Delaware		X					X			
Florida				X	X	X	X	X	X	X
Georgia	X	X(A)								
Hawaii	X									
Idaho	X	X(A)								
Illinois				X	X	X	X	X	X	X
Indiana	X	X(A)								
Iowa	X									
Kansas		X								
Kentucky	X									
Louisiana		X								
Maine		X		X	X		X	X	X	
Maryland				X	X	X		X	X	
Massachusetts	X									
Michigan				X	X	X		X	X	
Minnesota				X	X	X		X	X	
Mississippi	X									
Missouri				X(?)	X(?)	X(?)		X	X	X
Montana	X				X (A)		X (A)	X (A)	X (A)	X (A)
Nebraska		X								
Nevada	X	X(A)								
New Hampshire				X	X		X	X	X	
New Jersey				X	X	X	X	X		X
New Mexico		X	X mod 11							
New York				X(?)	X(?)	X(?)	X	X(?)	X(?)	?
North Carolina		X								
North Dakota	X	X (A)								
Ohio		X								
Oklahoma	X	X(A)								
Oregon		X								
Pennsylvania		X								
Rhode Island		X								
South Carolina		X								
South Dakota		X	X mod 10				X	X		
Tennessee		X	X mod 11							
Texas		X								
Utah		X	X mod 10							
Vermont		X	X mod 11							
Virginia	X	X(A)								
Washington			X mod 10	X	X	X	X	X	X	
West Virginia		X								
Wisconsin			X(?)	X	X	X		X	X	X
Wyoming		X	X(?)							

REFERENCES

1. Steve Connor, The invisible border guard, *New Scientist* Jan. 5 (1984), 9–14.
2. Joseph A. Gallian, The zip code bar code, *The UMAP Journal* 7 (1986), 191–194.
3. Joseph A. Gallian and Steven Winters, Modular arithmetic in the marketplace, *Amer. Math. Monthly* 95 (1988), 548–551.
4. H. Peter Gumm, A new class of check digit methods for arbitrary number systems, *IEEE Transactions on Information Theory* 31 (1985), 102–105.
5. Donald E. Knuth, *The Art of Computer Programming*, Vol. 3, Addison-Wesley, Reading, MA, 1973.
6. Philip M. Tuchinsky, International standard book numbers, *The UMAP Journal* 5 (1985), 41–54.
7. J. Verhoeff, *Error Detecting Decimal Codes*, Mathematical Centrum, Amsterdam, 1969.
8. E. F. Wood, Self-checking codes—an application of modular arithmetic, *Mathematics Teacher* 80 (1987), 312–316.