Probability Theory for Pickpockets—ec-PIN Guessing

Markus G. Kuhn – mkuhn@acm.org – 1997-07-30
COAST Laboratory, Purdue University, West Lafayette, Indiana 47907-1398, USA

This abstract briefly describes an algorithm for determining the most likely 4-digit PINs associated with a debit card used at EuroCheque (ec) ATMs. We determine the probability of every PIN based on knowledge of the PIN-generation method and the data on the magnetic stripe. A card thief could use this strategy to optimally select the three PINs that he can try on a stolen card before it will be invalidated. The analysis shows a significant security problem of the PIN-generation algorithm, which allows the presented PIN-guess strategy to achieve a considerably higher success rate than a random guess would. The reader is assumed to be familiar with basic probability theory. The analyzed PIN-generation algorithm has been used by German banks from 1981 until 1997 according to documents available to the author.

Users of ec-cards cannot select their own PIN. The bank calculates the PIN for each customer as illustrated in the diagram. A 16-digit decimal number is formed by concatenating five digits of the bank routing number, the ten digit account number, and a single digit card sequence number. This number is transformed into a 64-bit pattern by encoding each digit with its 4-bit binary value (BCD). The result is encrypted using the DES algorithm with a secret 56-bit institute key K_I. The resulting 64-bit ciphertext can be written as a 16-digit hexadecimal number. We take the digits 3–6 and replace all occurrences of the letters A–F by digits 0–5 respectively. If the first of those four digits is a 0, we replace it by a 1. ATM networks owned by the card-issuing bank know K_I. They reconstruct the PIN the same way and compare it with what the customer has entered. ATM networks of other banks use a pool key K_{P1} instead, which results in a different PIN of course. The magnetic stripe of each card contains a 4-digit correction offset O_1 that an ATM using K_{P1} has to add without carry-over to the digits 3–6 of the decimalized DES result, to get the PIN known by the customer. In the decimalized DES result obtained with a pool key, a leading zero is not replaced. Since K_{P1} is known by all banks in Europe, it could be compromised more easily. Therefore, there exist two backup pool keys K_{P2} and K_{P3} and the card stripe stores two corresponding offsets O_2 and O_3. The emergency plan should K_{P1} be compromised one day is to switch to K_{P2} and overwrite O_1 on all cards at the next ATM visit. The problem that the designer of this PIN-handling system had not understood is that these pool key offsets provide valuable hints for someone who tries to guess a PIN.

From track 3 of the magnetic stripe of a card, we know the 12 offset digits

$$\begin{aligned}\text{Offset 1:}\quad & O_1 = (O_{1,1}, O_{1,2}, O_{1,3}, O_{1,4})\\ \text{Offset 2:}\quad & O_2 = (O_{2,1}, O_{2,2}, O_{2,3}, O_{2,4})\\ \text{Offset 3:}\quad & O_3 = (O_{3,1}, O_{3,2}, O_{3,3}, O_{3,4})\end{aligned}$$

Our goal is to determine four PIN digits

$$\hat{P} = (\hat{P}_1, \hat{P}_2, \hat{P}_3, \hat{P}_4)$$

that are most likely the actual PIN for this card.

Let $\tilde{P}_j$ denote the random variable representing the j-th digit of the actual PIN of a card, and let $\tilde{O}_{i,j}$ denote the random variable representing the j-th digit in offset number i (for all $1 \leq i \leq 3, 1 \leq j \leq 4$). We assume that all hexadecimal digits of the four DES results are mutually independent and that the 16 digits are uniformly distributed, a required characteristic of any good block-cipher algorithm such as DES. Then, the distributions of these random variables are due to the applied decimalization method (see diagram) as follows:

$$p(\tilde{P}_j = k) = \begin{cases} 0/16, & \text{if } j = 1 \text{ and } k = 0 \\ 4/16, & \text{if } j = 1 \text{ and } k = 1 \\ 2/16, & \text{if } j > 1 \text{ and } k \in \{0, 1\} \\ 2/16, & \text{if } k \in \{2, \ldots, 5\} \\ 1/16, & \text{if } k \in \{6, \ldots, 9\} \end{cases} \tag{1a}$$

$$p(\tilde{O}_{i,j} = k | \tilde{P}_j = l) = \begin{cases} 2/16, & \text{if } (l - k) \bmod 10 \in \{0, \ldots, 5\} \\ 1/16, & \text{if } (l - k) \bmod 10 \in \{6, \ldots, 9\} \end{cases} \tag{1b}$$

A most likely PIN $\hat{P}$ is a P for which the conditional probability $p(\tilde{P} = P | \text{ for all } i : \tilde{O}_i = O_i)$ is maximal. Since all digits of the PIN are determined independently of each other, we can determine a most likely j-th PIN digit $\hat{P}_j$ as a P_j that maximizes $p(\tilde{P}_j = P_j | \forall i : \tilde{O}_{i,j} = O_{i,j})$ and get a most likely PIN simply as the combination of the most likely digits for each position.

We can turn around this conditional probability as follows (Bayes' theorem)

$$\begin{aligned} p(\tilde{P}_j = P_j | \forall i : \tilde{O}_{i,j} = O_{i,j}) &= \frac{p(\tilde{P}_j = P_j \wedge \forall i : \tilde{O}_{i,j} = O_{i,j})}{p(\forall i : \tilde{O}_{i,j} = O_{i,j})} \\ &= \frac{p(\forall i : \tilde{O}_{i,j} = O_{i,j} | \tilde{P}_j = P_j) \cdot p(\tilde{P}_j = P_j)}{p(\forall i : \tilde{O}_{i,j} = O_{i,j})} \\ &= \frac{p(\forall i : \tilde{O}_{i,j} = O_{i,j} | \tilde{P}_j = P_j) \cdot p(\tilde{P}_j = P_j)}{\sum_{k=0}^{9} p(\forall i : \tilde{O}_{i,j} = O_{i,j} | \tilde{P}_j = k) \cdot p(\tilde{P}_j = k)} \end{aligned}$$

and since we assumed the DES results with the three pool keys to be mutually independent, we can replace the conditional probabilities for the combination of digits from all three offsets by the product of the probabilities for the individual offset digits, and thus we get

$$p(\tilde{P}_j = P_j | \forall i : \tilde{O}_{i,j} = O_{i,j}) = \frac{\prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} | \tilde{P}_j = P_j) \cdot p(\tilde{P}_j = P_j)}{\sum_{k=0}^{9} \prod_{i=1}^{3} p(\tilde{O}_{i,j} = O_{i,j} | \tilde{P}_j = k) \cdot p(\tilde{P}_j = k)}. \tag{2}$$

This formula uses only the known distributions given in (1). Based on it, we can easily write a small program to calculate $p(\tilde{P}_j = P_j | \forall i : \tilde{O}_{i,j} = O_{i,j})$ for all $P_j \in \{0, \ldots, 9\}$ given $O_{1,j}$, $O_{2,j}$, and $O_{3,j}$, and determine a $\hat{P}_j$ for which this probability is maximal. We do this for all four digit positions j and get this way a most likely PIN candidate $\hat{P}$. The probability that

this PIN is correct is the product of the probabilities that the individual digits $\hat{P}_j$ are each correct, as calculated above. It can get as high as $0.948\% \approx 1/105$.

We have so far described how to find a most likely PIN for a specific card for which we know the offsets, and we can calculate its success probability. We now calculate, what success probability we expect if we do not have the offsets of a specific card given, but if we pick a random card. This can be estimated per digit position j with another small program as follows. We try all 16^4 possible combinations for the four hexadecimal digits (W, X, Y, Z) in each of the four DES results that determine one digit in the PIN and one in each offset. We determine from this quadruple—like a bank does when a new card is issued—the j-th digit of the PIN and the three offsets as follows:

$$\begin{aligned} P_j &:= \begin{cases} W \bmod 10, & \text{if } W \bmod 10 > 0 \text{ or } j > 1 \\ 1, & \text{if } W \bmod 10 = 0 \text{ and } j = 1 \end{cases} \\ O_{1,j} &:= (P_j - X) \bmod 10 \\ O_{2,j} &:= (P_j - Y) \bmod 10 \\ O_{3,j} &:= (P_j - Z) \bmod 10 \end{aligned}$$

This way, we have generated a set of 16^4 simulated cards that has the same PIN and offset digit distribution that we expect from the set of all cards in circulation. Now, we determine a most likely PIN digit $\hat{P}_j$ as described above for each of those 16^4 cards. Since we know for each of these simulated cards the correct PIN digit P_j, we can count which fraction of the 16^4 calculated most likely PIN digits $\hat{P}_j$ is correct and equals the corresponding P_j.

The results of this program run are the following probabilities for a correct guess for each of the four PIN digit positions j:

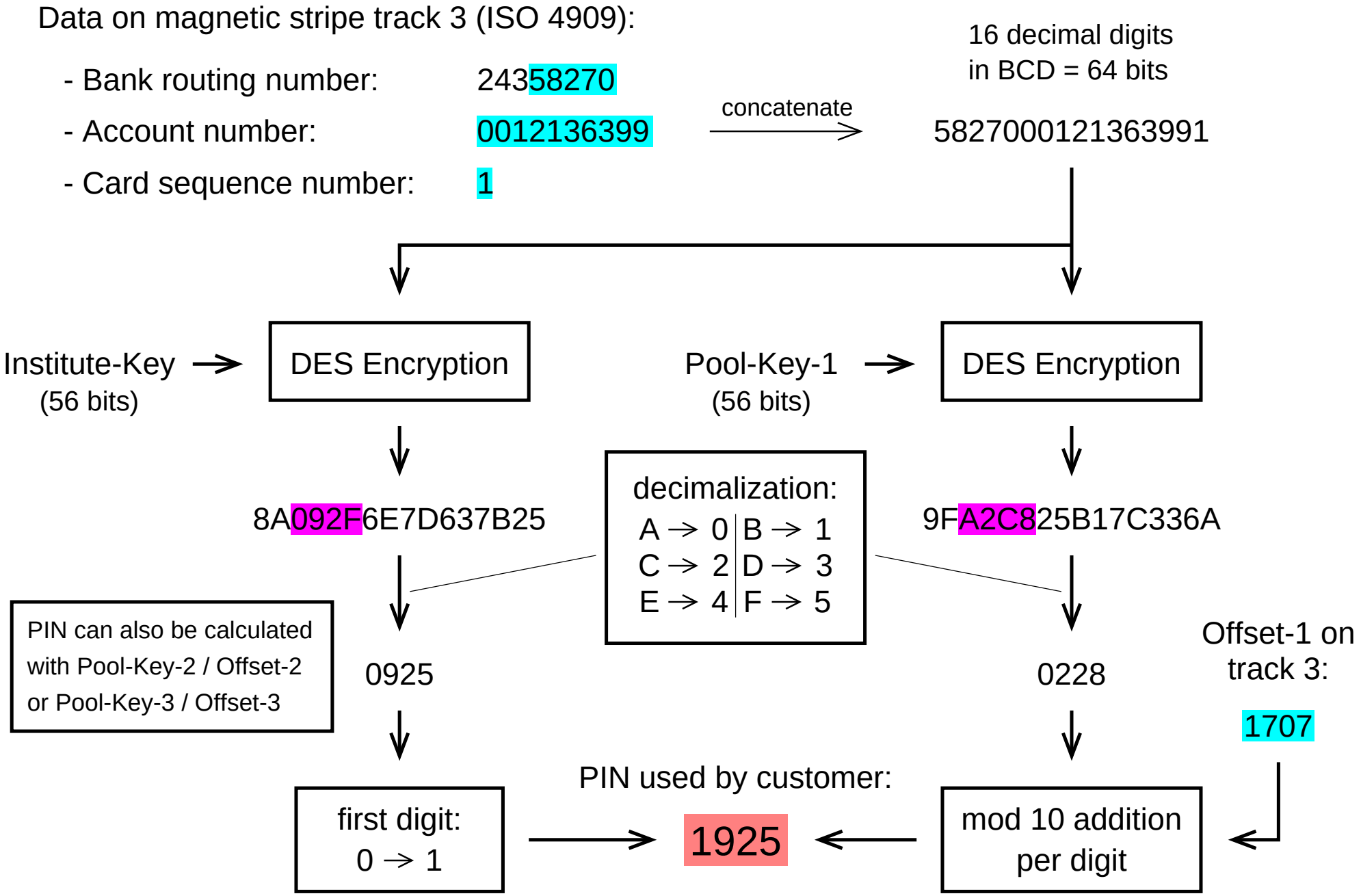
$$\begin{aligned} &\text{digit 1:} && 0.27856 \approx 28\% \approx 1/3.6 \\ &\text{digit 2:} && 0.20312 \approx 20\% \approx 1/4.9 \\ &\text{digit 3:} && 0.20312 \approx 20\% \approx 1/4.9 \\ &\text{digit 4:} && 0.20312 \approx 20\% \approx 1/4.9 \end{aligned}$$

Note that if the banks had used a good PIN generation algorithm, we would have expected a random guess success rate of 11% for the first digit (no leading zero) and 10% for the remaining three digits. By multiplying the actual four per-digit success probabilities above, we get a success probability of $0.0023346 \approx 0.233\% \approx 1/428$ for the most likely PIN. Since a thief has at least three attempts, and since most second or third best PINs have a similar success chance, the probability to get access to the account is roughly three times the success probability of the most likely PIN, this means in the order of $0.7\% \approx 1/150$. Had the banks used a good PIN-generation algorithm, we would have expected only a $1/3000 \approx 0.033\%$ success rate in three attempts, because there are 9000 possible PINs (1000–9999). In other words, the security of the ec-PIN system is worse than that of a good system with only three digit PINs, where we would expect a $1/300 \approx 0.33\%$ success rate in three attempts.

This text did not discuss techniques that allow more than three attempts to enter a PIN. It also did not discuss the cost of determining the DES keys using a brute-force search with special hardware. Both are in the author's opinion valid additional serious concerns regarding the security of the EC card system.

The author wishes to thank Bodo and Ulf Möller from the University of Hamburg for their help and for their suggestions in this analysis.

PIN Calculation for EuroCheque ATM Debit Cards



M. Kuhn