Schwächen des ec-PIN-Verfahrens
— ENTWURF —

Bodo Möller
<3moeller@informatik.uni-hamburg.de>

18. Februar 1997

Zusammenfassung

Das in [3] festgelegte Verfahren zur Berechnung, Verschlüsselung und Prüfung von PINs im institutsübergreifenden Geldausgabeautomaten-System hat Entwurfsfehler, durch die das Erraten von PINs so erleichtert wird, daß pro ca. 1049, 608 bzw. 428 Karten (je nachdem, ob 1, 2 oder 3 „Offsets“ vorliegen) ein Rateerfolg beim ersten Versuch zu erwarten ist. Bei einem hinsichtlich der Sicherheit optimalen Verfahren sollte nur ein Erfolg pro 9000 Karten möglich sein. Darüberhinaus ist die Benutzung des Verschlüsselungsalgorithmus DES mit 56-Bit-Schlüsseln heute unzureichend.

1 Einführung

Am 1. Mai 1979 schlossen der Bundesverbank der Deutschen Volksbanken und Raiffeisenbanken (BVR), der Bundesverband deutscher Banken (BdB), der deutsche Sparkassen- und Giroverband (DSGV) und die deutsche Bundespost die „Vereinbarung für das institutsübergreifende Geldausgabeautomaten-System“ ab; im März 1981 wurden in Berlin die ersten zwölf Geldausgabeautomaten in Betrieb genommen ([1]).

Diese Untersuchung des PIN-Verfahrens für deutsche eurocheque-Karten (und andere, damit kompatible Bankkarten; im folgenden sprechen wir immer von ec-Karten) basiert auf [3] und [2], zwei Anhängen zu den „Richtlinien für das institutsübergreifende Geldausgabeautomaten-System“. Die vorliegenden Fassungen von [3] und [2] sind nicht datiert[1], und da die Titelseite nicht vorliegt, ist nicht völlig klar, ob es sich um gültige Fassungen der beiden Anhänge oder um Entwürfe handelt[2].

[3] beschreibt ein Verfahren zur Erzeugung und Überprüfung von persönlichen Identifikationsnummern (PINs) bei mit Magnetstreifen ausgestatteten ec-Karten für die Benutzung an Geldausgabeautomaten. Dabei werden über den Aufbau der 3. Spur des Magnetstreifens (siehe [2]) bestimmte Voraussetzungen gemacht.

In Abschnitt 2 wird das Verfahren aus [3] vorgestellt; die Darstellung ist dabei nicht aus der Vorlage übernommen, sondern wurde in Hinblick auf die darauffolgenden Untersuchungen gewählt. Abschnitt 3 zeigt, wie mit geringem Aufwand die Trefferwahrscheinlichkeit beim „Erraten“ von PINs gegenüber dem „wilden Raten“ deutlich verbessert werden kann. (Hierfür wird angenommen, daß der Inhalt des Magnetstreifens der ec-Karte bekannt ist. Geeignete Lesegeräte sind problemlos verfügbar.) Abschnitt 4 geht auf auf die Möglichkeit ein, durch erschöpfende Suche einen DES-Schlüssel zu finden, mit dem die PIN jeder beliebigen ([2] und [3] entsprechenden) ec-Karte sofort sicher bestimmt werden kann.

2 Beschreibung des Verfahrens

Wir vereinbaren folgende Schreibweisen:

$$\mathbb{X} := \{0, 1, \ldots, 9, \mathrm{A}, \ldots, \mathrm{F}\}$$

sei das Alphabet der Hexadezimalziffern. Es sei

$$\mathbb{Z}_{10} := \mathbb{Z}/_{10\,\mathbb{Z}},$$

aufgefaßt als additive Gruppe. Für die Elemente von $\mathbb{Z}_{10}$ schreiben wir einfach die Ziffern $0, 1, \ldots, 9$. In beiden Fällen bedeute $a_1 a_2 \ldots a_n$ die *Verkettung* der Elemente a_1 bis $a_n \in \mathbb{X}$ bzw. $\in \mathbb{Z}_{10}$ zu einem entsprechend langen Tupel. (Mehrdeutigkeiten sind dadurch nicht zu befürchten, denn die aus $\mathbb{Z}$ induzierte Multiplikation wird nicht benötigt.) Die Menge $\mathbb{Z}_{10}$ fassen wir gelegentlich auch als Teilmenge von $\mathbb{X}$ auf, wobei die additive Struktur natürlich verlorengeht. (Das entspricht der BCD-Codierung, wenn man die hexadezimalen Ziffern den entsprechenden 4-Bit-Worten gleichstellt.)

Das PIN-Verfahren benutzt den in [7] definierten Verschlüsselungsalgorithmus DES. DES-Schlüssel sind 64

[1] Es gibt jedoch Hinweise: [2] enthält Hinweise auf die Norm DIN ISO 4909; die Internationale Norm ISO 4909 wurde erstmals 1980 als DIN-Norm übernommen ([4]). Außerdem verweist [2] auf die Normen ISO 3554 und ISO 2894. Beide sind heute ungültig; die letzten Fassungen, ISO 3554:1976 und ISO 2894:1980, wurden 1985 zurückgezogen ([5]).

[2] Die Kurzbeschreibung in [6] legt jedoch nahe, daß das Verfahren in dem hier zugrundegelegten Text [3] zutreffend dargestellt ist:

> „Prüfung der PIN-Nummer (PIN-verify)
>
> - [...]
> - Encrypt der PIN-Ausgangsdaten mit dem Instituts-Key (oder Pool-Key)
> - Extrahieren und dezimalisieren des PIN
> - Vergleichen des errechneten und des empfangenen PIN (‚clear PIN‘)
>
> Falls mit dem EC-Pool-Key der PIN errechnet wird, ist die Differenz zum empfangenen PIN der ‚Offset‘. Der Offset ist in den PIN-Ausgangsdaten enthalten. Mit dem ec-Pool-Key werden nur institutsfremde ec-PIN’s geprüft.“ [6]

Bits (8 Bytes) lang. Hiervon gehen nur 56 Bits tatsächlich in die kryptographischen Operationen ein, denn in jedem Byte ist ein Parity-Bit vorgesehen. (Deshalb spricht man bei DES meist auch von 56-Bit-Schlüsseln.) Folgende Schlüssel sind in [3] vorgesehen:

- Drei *„Poolschlüssel“*, die wir mit K_1, K_2, K_3 bezeichnen. Diese Poolschlüssel sind institutsübergreifend festgelegt.
- Für jedes kartenausgebende Institut ein *„Institutsschlüssel“*, hier mit K' bezeichnet. Er soll nur beim jeweiligen Institut bekannt sein.

Auf dem Magnetstreifen der Karte befinden sich u. a. folgende Angaben: Die Bankleitzahl (8 Stellen), die Kontonummer (10 Stellen) sowie die einstellige „Kartenfolgenummer“. Letztere, eine Ziffer von 0 bis 9, dient „zur Unterscheidung zwischen einzelnen Karten (gleichzeitig oder nacheinander ausgestellt) mit derselben Kontonummer“; „beim Ausstellen einer Zweit- oder Ersatzkarte ist die nächsthöhere Ziffer zu verwenden“ [2]. Als Eingabeblock für die DES-Verschlüsselung werden die Stellen 4 bis 8 der BLZ, die Kontonummer und die Kartenfolgenummer aneinandergehängt. Diesen 16stelligen (d. h. 64 Bits langen) Wert bezeichnen wir mit D.

Die DES-Verschlüsselung mit dem Schlüssel K schreiben wir als Abbildung

$$\mathrm{DES}_K\colon\ \mathbb{X}^{16} \to \mathbb{X}^{16}.$$

Im PIN-Verfahren wird auf DES-Ergebnisse folgende Projektion angewendet:

$$\begin{aligned} \Pi\colon\ \mathbb{X}^{16} &\to \mathbb{X}^4 \\ \Pi(x_1x_2x_3\ldots x_{14}x_{15}x_{16}) &:= x_3x_4x_5x_6 \end{aligned}$$

Die Funktion zur „Dezimalisierung“ von hexadezimalen Ziffern sei gegeben durch

$$\begin{aligned} \mathrm{dec}\colon\ \mathbb{X} &\to \mathbb{Z}_{10} \\ \mathrm{dec}(0) &:= 0 \\ \mathrm{dec}(1) &:= 1 \\ &\ldots \\ \mathrm{dec}(9) &:= 9 \\ \mathrm{dec}(\mathrm{A}) &:= 0 \\ &\ldots \\ \mathrm{dec}(\mathrm{F}) &:= 5 \end{aligned}$$

Der Zahlenwert der hexadezimalen Ziffern wird von dec also modulo 10 reduziert. Als Variante hiervon brauchen wir ferner

$$\begin{aligned} \mathrm{dec}'\colon\ \mathbb{X} &\to \mathbb{Z}_{10} \\ \mathrm{dec}'(n) &:= \begin{cases} 1 & \text{falls } \mathrm{dec}(n) = 0 \\ \mathrm{dec}(n) & \text{sonst} \end{cases} \end{aligned}$$

Wir definieren

$$\mathrm{dec}, \mathrm{dec}'\colon\ \mathbb{X}^4 \to {\mathbb{Z}_{10}}^4$$

durch

$$\begin{aligned} \mathrm{dec}(x_1x_2x_3x_4) &:= \mathrm{dec}(x_1)\mathrm{dec}(x_2)\mathrm{dec}(x_3)\mathrm{dec}(x_4), \\ \mathrm{dec}'(x_1x_2x_3x_4) &:= \mathrm{dec}'(x_1)\mathrm{dec}(x_2)\mathrm{dec}(x_3)\mathrm{dec}(x_4). \end{aligned}$$

Ist n eine natürliche Zahl, so sei

$$\underline{n} := \{x \in \mathbb{N} \mid 1 \le x \le n\}.$$

2.1 Erzeugung der PIN und der Offsets

Zu einer Karte berechnet das kartenausgebende Institut die PIN mit Hilfe des Institutsschlüssels K' aus den Kartendaten D wie folgt:

$$\mathit{PIN} := (\mathrm{dec}' \circ \Pi \circ \mathrm{DES}_{K'})(D) \tag{1}$$

Für jeden Poolschlüssel ($p \in \underline{3}$) wird ein Wert berechnet, den [3] als „Ergebnis des DES-Algorithmus mit Poolschlüssel“ bezeichnet:

$$\mathit{EDP}_p := (\mathrm{dec} \circ \Pi \circ \mathrm{DES}_{K_p})(D) \tag{2}$$

Hieraus wird ein „Offset“ mit vier Dezimalstellen berechnet:

$$\mathit{offset}_p := \mathit{PIN} - \mathit{EDP}_p \tag{3}$$

(Die Substraktion erfolgt hierbei komponentenweise in $\mathbb{Z}_{10}$, anders gesagt stellenweise modulo 10.) Diese Berechnung des Offsets kann man als Verschlüsselung in einer Variante des Cipher Feedback Mode von DES (mit D als Initialisierungswert) ansehen.

2.2 Prüfen der PIN

Beim Einsatz einer ec-Karte an einem Geldausgabeautomaten sind für die Prüfung der an der Tastatur eingegebenen PIN zwei Fälle zu unterscheiden. Falls es sich um einen Automaten des jeweiligen kartenausgebenden Institutes handelt, kann mit dem Institutsschlüssel K' und den Kartendaten D gemäß (1) die PIN berechnet und dann mit der Eingabe verglichen werden. Andernfalls muß der Geldausgabeautomat den jeweils gültigen Poolschlüssel K_p verwenden, um gemäß (2) den Wert EDP_p auszurechnen. Dann ist wegen (3)

$$\mathit{PIN} = \mathit{EDP}_p + \mathit{offset}_p,$$

und dieser Wert kann mit der vom Kunden eingegebenen PIN verglichen werden.[3]

Die Geldausgabeautomaten enthalten immer nur einen der Poolschlüssel:

> „In den institutsübergreifenden Geldausgabeautomaten werden maximal 2 Schlüssel vorrätig gehalten und wahlweise eingesetzt: Der Institutsschlüssel [...] und der [jeweils gültige] Poolschlüssel. [...] Bei Start des Systems wird vereinbart, welcher der drei Poolschlüssel eingesetzt wird.“ [3].

[3] Diese Überprüfungsmethode über Poolschlüssel und Offsets wird auch für die Online-Prüfung von PINs bei EFT/PoS-Terminals („Electronic Funds Transfer am Point of Sale“) verwendet; hierbei übernehmen zentrale Autorisierungsrechner das Prüfen der PIN ([6]).

Auf dem Magnetstreifen der ec-Karte sind aber drei Felder für Offsets vorgesehen ([2]). Es können also gleichzeitig *offset*$_1$, *offset*$_2$ und *offset*$_3$ zur Verfügung gestellt werden:

> „Die [. . .] offsets sind in den Feldern 13.2, 27.2 und 27.3 des Magnetstreifens niedergelegt. Der Aufruf des jeweils gültigen Feldes erfolgt über eine Vorlaufinformation, die beim Versand des gültigen Poolschlüssels von der schlüsselverwaltenden Stelle mitgegeben wird. Nicht mehr gültige offsets sind vom institusübergreifenden Geldausgabeautomaten bei der nächsten Benutzung der Karte auf ‚0' zu setzen" [3]

Eine klare Aussage darüber, wieviele Poolschlüssel tatsächlich gleichzeitig „in Betrieb" sein sollen – wieviele Offsets man also tatsächlich auf Karten vorfindet – enthält [3] nicht. Denkbar ist, daß ein „Rotieren" der Poolschlüssel ermöglicht werden soll: Von Zeit zu Zeit könnte der „jeweils gültige" Poolschlüssel für ungültig erklärt werden und reihum der nächste Poolschlüssel Gültigkeit erlangen. Für den freigewordenen Platz könnte ein völlig neuer Schlüssel erzeugt werden. Neu ausgegebene Karten könnten Offsets für den gerade aktuellen Poolschlüsselvorrat erhalten. Die Gültigkeitsdauer der ec-Karten wäre damit so abzustimmen, daß bis Ablauf der Karte höchstens zwei solche Schlüsselwechsel stattfinden (wonach noch ein „gültiger Offset" auf der Karte stünde). Ein Erneuern der Poolschlüssel wird in [3] allerdings nicht erwähnt.

Laut [2] heißt das Feld 27.2 bei Karten mit internationaler Freizügigkeit „ec-PVV" („PIN-verificiation-value"); nur für Karten ohne internationale Freizügigkeit gilt die Bezeichnung „Offset 2". Der ec-PVV hat wie die Offsets vier Stellen. Er wird in [3] nicht erwähnt (dort wird, wie oben zitiert, das Feld 27.2 als Position eines der Offsets angegeben). Ob dieser Wert auf gleiche Art und Weise zustandekommt wie der Offset 2, ist daher nicht ganz klar. Gäbe es Unterschiede, so wäre jedoch das PIN-Prüfungs-Verfahren aus [3] bei international freizügigen Karten nicht anwendbar, wenn gerade der zu Offset 2 gehörige Poolschlüssel der „gültige Poolschlüssel" ist. (Falls dieser Poolschlüssel nie zum „gültigen Poolschlüssel" bestimmt würde, hätte Offset 2 keine Funktion.)[4]

3 Analyse

Bei unbekannten (und paarweise verschiedenen) Schlüsseln K', K_1, K_2 und K_3 können wir die Werte $\mathrm{DES}_{K'}(D)$ und $\mathrm{DES}_{K_p}(D)$ $(p = 1, 2, 3)$ als gleichverteilte Zufallszahlen aus $\mathbb{X}^{16}$ ansehen[5]. (Die gleiche Annahme könnte man auch machen, wenn als Verschlüsselungsalgorithmus anstelle von DES z. B. Triple-DES in einer seiner Spielarten, z. B. Encrypt-Decrypt-Encrypt mit einem 112-Bit-Schlüssel, eingesetzt würde.) Damit sind auch die hexadezimal vierstelligen Werte $(\Pi \circ \mathrm{DES}_K)(D)$ für $K = K', K_1, K_2, K_3$ stochastisch unabhängig gleichverteilt. Mit den Bezeichnungen

$$\begin{aligned} & X_1X_2X_3X_4 \quad \text{für} \quad (\Pi \circ \mathrm{DES}_{K'})(D) \\ \text{und} \quad & Y_{p,1}Y_{p,2}Y_{p,3}Y_{p,4} \quad \text{für} \quad (\Pi \circ \mathrm{DES}_{K_p})(D) \end{aligned}$$

$(p = 1, 2, 3)$ können wir also im Rahmen dieser Betrachtung von stochastisch unabhängigen, gleichverteilten Zufallsvariablen $X_1, \ldots, X_4,\ Y_{1,1}, \ldots, Y_{1,4},\ Y_{2,1}, \ldots, Y_{2,4},\ Y_{3,1}, \ldots, Y_{3,4}$ mit $\mathbb{X}$ als Wertebereich ausgehen.

Entsprechend den Formeln (1) bis (3) erhalten wir hieraus folgende Zufallsvariablen:

Für *PIN*:

$$\begin{aligned} & M_1M_2M_3M_4 \\ & := \quad \mathrm{dec}'(X_1X_2X_3X_4) \\ & = \quad \mathrm{dec}'(X_1)\mathrm{dec}(X_2)\mathrm{dec}(X_3)\mathrm{dec}(X_4) \end{aligned}$$

Für *EDP*$_p$:

$$\begin{aligned} & N_{p,1}N_{p,2}N_{p,3}N_{p,4} \\ & := \quad \mathrm{dec}(Y_{p,1}Y_{p,2}Y_{p,3}Y_{p,4}) \\ & = \quad \mathrm{dec}(Y_{p,1})\mathrm{dec}(Y_{p,2})\mathrm{dec}(Y_{p,3})\mathrm{dec}(Y_{p,4}) \end{aligned}$$

Für *offset*$_p$ = *PIN* − *EDP*$_p$:

$$\begin{aligned} & O_{p,1}O_{p,2}O_{p,3}O_{,p4} \\ & := \quad M_1M_2M_3M_4 - N_{p,1}N_{p,2}N_{p,3}N_{p,4} \end{aligned}$$

Wenn man diese vierstelligen Objekte in einzelne Ziffern zerlegt, ergibt sich:

$$\begin{aligned} M_i &= \begin{cases} \mathrm{dec}'(X_1) \text{ für } i = 1 \\ \mathrm{dec}(X_i) \text{ für } i > 1 \end{cases} \\ N_{p,i} &= \mathrm{dec}(Y_{p,i}) \\ O_{p,i} &= M_i - N_{p,i} \\ &= \begin{cases} \mathrm{dec}'(X_1) - \mathrm{dec}(Y_{p,1}) \text{ für } i = 1 \\ \mathrm{dec}(X_i) - \mathrm{dec}(Y_{p,i}) \text{ für } i > 1 \end{cases} \end{aligned}$$

Hierbei ist $p = 1, 2, 3$ und $i = 1, \ldots, 4$.

An dieser Stelle zeichnet sich schon deutlich ab, wie die Kenntnis der Offsets einer ec-Karte beim Raten der PIN-Ziffern helfen kann. Von vornherein ist wegen der Dezimalisierung klar, daß die Werte $0, \ldots, 5$ bessere Aussichten haben als $6, \ldots, 9$ (bei der ersten Stelle, also $i = 1$, ist 0 unmöglich und dafür 1 am wahrscheinlichsten). Angenommen, wir wissen außerdem, daß der Offset 1 an vierter Stelle die Ziffer 8 hat (d. h. $O_{1,4} = 8$). Dann werden wir uns beim Raten der vierten PIN-Ziffer M_4 eher für 3 entscheiden als für 4: Denn es ist $N_{1,4} = M_4 - O_{1,4}$ (in $\mathbb{Z}_{10}$), und $M_4 = 3$ würde somit $N_{1,4} = 3 - 8 = 5$ implizieren, $M_4 = 4$ aber $N_{1,4} = 4 - 8 = 6$. Letzteres ist weniger wahrscheinlich, da $N_{1,4} = \mathrm{dec}(Y_{1,4})$ mit in $\mathbb{X}$ gleichverteiltem $Y_{1,4}$ ist.

[4] Es läßt sich vermuten, daß die Bezeichnung „ec-PVV" für 27.2 im Gegensatz zu den Feldern der Offsets international – im Rahmen der eurocheque-Organisation – vereinbart ist. Geldausgabeautomaten im Ausland könnten diesen Wert zusammen mit der eingegebenen PIN an eine Verifizierungsstelle in Deutschland schicken, die den ec-PVV als Offset 2 ansehen und den dazugehörigen Poolschlüssel einsetzen könnte.

[5] Eines der Entwurfsziele für Verschlüsselungsalgorithmen der Form von DES ist, daß die Verschlüsselungsoperation DES_K für einen Kryptoanalytiker bei einem ihm unbekannten Schlüssel K wie eine zufällige Permutation auf der Menge aller möglichen 64-Bit-Blöcke erscheinen soll (obwohl es tatsächlich nur 2^{56} verschiedene Abbildungen DES_K gegenüber insgesamt $2^{64}!$ Permutationen gibt). Wenn es gelingt, den benutzten Schlüssel K herauszufinden, können natürlich alle Funktionswerte von DES_K ohne weiteres bestimmt werden; dazu siehe Abschnitt 4.

Diese Überlegung läßt sich darauf ausweiten, daß sämtliche Ziffern von s Offsets bekannt sind (wobei $s = 1, 2, 3$, je nach Anzahl der auf der Karte vorhandenen, gültigen Offsets[6]). Ohne Beschränkung der Allgemeinheit seien die Offsets $1, \ldots, s$ bekannt. Man kennt also Ziffern $o_{p,i} \in \mathbb{Z}_{10}$ ($p \in \underline{s}$, $i \in \underline{4}$) und möchte zum möglichst erfolgreichen Raten der PIN die Ziffern $n_1, \ldots, n_4$ so wählen, daß die bedingte Wahrscheinlichkeit

$$\mathcal{P}(\, N_1N_2N_3N_4 = n_1n_2n_3n_4 \mid \forall p \in \underline{s}, i \in \underline{4}\colon\ O_{p,i} = o_{p,i})$$

maximal wird. Da die vier Stellen voneinander unabhängig sind, ist diese Wahrscheinlichkeit ist auch als Produkt

$$\prod_{i \in \underline{4}} \mathcal{P}(N_i = n_i \mid \forall p \in \underline{s}\colon\ O_{p,i} = o_{p,i})$$

darstellbar[7].

Damit können wir die n_i einzeln angehen. Wir haben also als Teilziel, für festes i durch geeignete Wahl einer Ziffer n_i die Wahrscheinlichkeit

$$\mathcal{P}(N_i = n_i \mid \forall p \in \underline{s}\colon\ O_{p,i} = o_{p,i}) \tag{4}$$

zu maximieren. Ihr Wert in Abhängigkeit von den Offset-Ziffern $o_{1,i}, \ldots, o_{s,i}$ und von der gewählten PIN-Ziffer n_i läßt sich durch ein einfaches Computerprogramm bestimmen, das alle Werte der Zufallsvariablen $X_i, Y_{1,i}, \ldots, Y_{s,i}$ durchspielt (hierbei gibt es 16^{1+s} Möglichkeiten, die nach unseren Annahmen jeweils die gleiche Wahrscheinlichkeit haben).

Das in Perl geschriebene Programm `offsets`, das diese Berechnung durchführt, findet sich in Anhang A. Neben den Wahrscheinlichkeiten (4) wird auch ermittelt, welche Erfolgsaussicht man hat, mit dieser Methode für eine zufällig ausgewählte Karte mit s Offsets die i-te PIN-Stelle mit einem Versuch richtig zu raten. Bei dieser Wahrscheinlichkeit handelt es sich um den Wert

$$\begin{aligned} & E_i(s) \\ &:= \sum_{(o_{1,i},\ldots,o_{s,i}) \in (\mathbb{Z}_{10})^s} \mathcal{P}(\forall p\colon\ O_{p,i} = o_{p,i}) \\ &\quad \cdot \max_{n_i \in \mathbb{Z}_{10}} \mathcal{P}(N_i = n_i \mid \forall p\colon\ O_{p,i} = o_{p,i}) \\ &= \sum_{(o_{1,i},\ldots,o_{s,i}) \in (\mathbb{Z}_{10})^s} \max_{n_i \in \mathbb{Z}_{10}} \mathcal{P}(N_i = n_i \ \wedge\ \forall p\colon\ O_{p,i} = o_{p,i}). \end{aligned}$$

Auszüge aus Beispielen für die Ausgabe dieses Programmes finden sich in den Tabellen 1 und 2.

Die von dem Programm für die verschiedenen Ausgangslagen (1, 2 oder 3 Offsets; 1. PIN-Ziffer oder 2. bis 4. PIN-Ziffer) ermittelten Erfolgswahrscheinlichkeiten sind Tabelle 3 zusammengestellt (die Zahlen in der Tabelle sind gerundet). Die Wahrscheinlichkeit, eine ganze PIN (mit einem Versuch) zu erraten, ist ebenfalls angegeben. Wir nennen sie $E(s)$; sie ist das Produkt der Einzelwahrscheinlichkeiten für die vier PIN-Stellen:

$$E(s) = \prod_{i \in \underline{4}} E_i(s)$$

In der letzten Tabellenzeile steht der Kehrwert dieser Wahrscheinlichkeit; er gibt an, auf wieviele zufällig ausgewählte Karten man einen Erfolg, die ganze PIN mit einem Versuch richtig zu erraten, zu erwarten hat. (Bei PIN-Verfahren mit optimaler Sicherheit sollte dieser Wert gleich der Anzahl der möglichen PINs sein. Wenn PINs wie bei dem hier untersuchten Verfahren aus vier Dezimalziffern bestehen sollen, deren erste nicht 0 sein darf, wäre das also 9000.)

Außerdem relevant ist die Wahrscheinlichkeit für das Erraten einer PIN, wenn zwei oder drei Versuche erlaubt werden[8]. Um diese zu bestimmen, kann man jedoch nicht einfach $E(s)$ – die Wahrscheinlichkeit bei einem erlaubten Versuch – mit zwei oder drei multiplizieren: Denn nicht in jedem Fall ergibt die auf dem Magnetstreifen vorgefundene Offsetkombination zwei bzw. drei verschiedene PINs, die die gleiche Wahrscheinlichkeit haben. $2E(s)$ und $3E(s)$ sind lediglich obere Abschätzungen der gesuchten Wahrscheinlichkeiten.

Die genaue Bestimmung der Erfolgswahrscheinlichkeit gestaltet sich in diesen Fällen recht umständlich. Statt dessen begnügen wir uns hier damit, untere Abschätzungen der Trefferwahrscheinlichkeiten für zwei und drei erlaubte Rateversuchen anzugeben.

Hat man zwei Versuche, so kann man beim ersten Versuch für $n_1n_2n_3n_4$ wieder anhand der Offsets die „beste PIN“ (oder einer der „besten PINs“, wenn es mehrere davon gibt) wählen und für den zweiten Versuch eine PIN aussuchen, die sich nur an einer Stelle hiervon unterscheidet. Man kann noch zusätzlich festlegen, daß der Unterschied in der zweiten Stelle liegen soll; die beim zweiten Versuch geratene PIN hat also die Form $n_1n_2'n_3n_4$. (Das ist nicht in jedem Fall eine optimale Wahl, hilft aber beim Abschätzen der erreichbaren Trefferwahrscheinlichkeit.) Wählt man die Ziffer $n_2' \neq n_2$ möglichst gut, so ergibt sich dafür, daß $n_1n_2n_3n_4$ oder $n_1n_2'n_3n_4$ die richtige

[6] „Nicht mehr gültige offsets“ sind auf der Karte vom Automaten „bei der nächsten Benutzung auf ‚0‘ zu setzen“ [3]. Diesen Fall kann man anhand des Magnetstreifeninhaltes zunächst nicht davon unterscheiden, daß ein gültiger Offset mit dem Wert 0000 auftritt. Durch Auslesen der Magnetstreifen mehrerer ec-Karten – die, sofern die Vermutung in Abschnitt 2.2 über ein „Rotieren“ des Vorrates an Poolschlüsseln zutrifft, etwa zur gleichen Zeit ausgestellt worden sein sollten – wird sich aber problemlos feststellen lassen, welche der drei Offset-Felder gültige Werte enthalten und welche nicht.

[7] Es ist nämlich

$$\begin{aligned} & \mathcal{P}(\forall i\colon\ N_i = n_i \mid \forall p, i\colon\ O_{p,i} = o_{p,i}) \\ &= \frac{\mathcal{P}(\forall i\colon\ N_i = n_i \ \wedge\ \forall p, i\colon\ O_{p,i} = o_{p,i})}{\mathcal{P}(\forall p, i\colon\ O_{p,i} = o_{p,i})} \\ &= \frac{\prod_i \mathcal{P}(N_i = n_i \ \wedge\ \forall p\colon\ O_{p,i} = o_{p,i})}{\prod_i \mathcal{P}(\forall p\colon\ O_{p,i} = o_{p,i})} \\ &= \prod_i \mathcal{P}(N_i = n_i \mid \forall p\colon\ O_{p,i} = o_{p,i}). \end{aligned}$$

[8] ec-Karten sollen nämlich hinsichtlich der Benutzung an Automaten ungültig gemacht werden, wenn dreimal eine falsche PIN eingegeben wurde (der Fehlversuchszähler wird jedesmal auf null gesetzt, wenn nach ein oder zwei Fehlversuchen die korrekte PIN eingegeben wird). Nach dem dritten Fehlversuch wird die Karte vom Geldausgabeautomaten einbehalten, sofern es sich um einen Automaten bei der Bank des Kunden handelt. Bei Automaten anderer Banken gilt dies nicht, damit der Kunde jedenfalls noch seine Scheckkarte zur Verfügung hat. In der Regel können also bei einer ec-Karte drei verschiedene PINs am Automaten ausprobiert werden; nur zwei, wenn die „Entwertung“ oder das Einbehalten der Karte vermieden werden soll.

Tabelle 1: Wahrscheinlichkeiten für die erste PIN-Ziffer bei Kenntnis eines Offsets *(Auszug)*

```
Offset 0  (Wahrscheinlichkeit: 2.80000)
=======================================
                                      PIN-Ziffer 1:    8 /  28 = 0.28571
PIN-Ziffer 2:    4 /  28 = 0.14286    PIN-Ziffer 3:    4 /  28 = 0.14286
PIN-Ziffer 4:    4 /  28 = 0.14286    PIN-Ziffer 5:    4 /  28 = 0.14286
PIN-Ziffer 6:    1 /  28 = 0.03571    PIN-Ziffer 7:    1 /  28 = 0.03571
PIN-Ziffer 8:    1 /  28 = 0.03571    PIN-Ziffer 9:    1 /  28 = 0.03571

Beste PIN-Ziffer (Wahrscheinlichkeit 0.28571):  1


Offset 1  (Wahrscheinlichkeit: 2.90000)
=======================================
                                      PIN-Ziffer 1:    8 /  29 = 0.27586
PIN-Ziffer 2:    4 /  29 = 0.13793    PIN-Ziffer 3:    4 /  29 = 0.13793
PIN-Ziffer 4:    4 /  29 = 0.13793    PIN-Ziffer 5:    4 /  29 = 0.13793
PIN-Ziffer 6:    2 /  29 = 0.06897    PIN-Ziffer 7:    1 /  29 = 0.03448
PIN-Ziffer 8:    1 /  29 = 0.03448    PIN-Ziffer 9:    1 /  29 = 0.03448

Beste PIN-Ziffer (Wahrscheinlichkeit 0.27586):  1

...

Insgesamt würde man bei 64 von 256 Fällen richtig raten
(Erfolgswahrscheinlichkeit: 0.2500000000).
```

Tabelle 2: Wahrscheinlichkeiten für die i-te PIN-Ziffer ($i > 1$) bei Kenntnis dreier Offsets *(Auszug)*

```
...

Offsetkombination 6 7 4  (Wahrscheinlichkeit: 0.05800)
=====================================================

PIN-Ziffer 0:    8 /  58 = 0.13793    PIN-Ziffer 1:    8 /  58 = 0.13793
PIN-Ziffer 2:    4 /  58 = 0.06897    PIN-Ziffer 3:    2 /  58 = 0.03448
PIN-Ziffer 4:    4 /  58 = 0.06897    PIN-Ziffer 5:    4 /  58 = 0.06897
PIN-Ziffer 6:    4 /  58 = 0.06897    PIN-Ziffer 7:    8 /  58 = 0.13793
PIN-Ziffer 8:    8 /  58 = 0.13793    PIN-Ziffer 9:    8 /  58 = 0.13793

Beste PIN-Ziffern (Wahrscheinlichkeit jeweils 0.13793):  0 1 7 8 9


Offsetkombination 6 7 5  (Wahrscheinlichkeit: 0.06400)
=====================================================

PIN-Ziffer 0:   16 /  64 = 0.25000    PIN-Ziffer 1:    8 /  64 = 0.12500
PIN-Ziffer 2:    4 /  64 = 0.06250    PIN-Ziffer 3:    2 /  64 = 0.03125
PIN-Ziffer 4:    2 /  64 = 0.03125    PIN-Ziffer 5:    4 /  64 = 0.06250
PIN-Ziffer 6:    4 /  64 = 0.06250    PIN-Ziffer 7:    8 /  64 = 0.12500
PIN-Ziffer 8:    8 /  64 = 0.12500    PIN-Ziffer 9:    8 /  64 = 0.12500

Beste PIN-Ziffer (Wahrscheinlichkeit 0.25000):  0

...

Insgesamt würde man bei 13312 von 65536 Fällen richtig raten
(Erfolgswahrscheinlichkeit: 0.2031250000).
```

Tabelle 3: Erfolgsaussichten beim Raten der PIN (ein Rateversuch pro Karte)

s	Anzahl Offsets	**1**	**2**	**3**
$E_1(s)$	Erfolgswahrscheinlichkeit bei der 1. Ziffer	0,250 000	0,265 625	0,278 564
$E_i(s), i > 1$	Erfolgswahrscheinlichkeit bei der 2. – 4. Ziffer	0,156 250	0,183 594	0,203 125
$E(s)$	Erfolgswahrscheinlichkeit für die ganze PIN	0,000 953 674	0,001 643 776	0,002 334 618
$1/E(s)$	Kartenanzahl pro Erfolg	**1048,6**	**608,4**	**428,3**

PIN ist, die Wahrscheinlichkeit

$$E_1(s)\,(E_2(s) + E_2'(s))\,E_3(s)E_4(s)$$

$$= \left(1 + \frac{E_2'(s)}{E_2(s)}\right) E(s);$$

hierbei sei $E_i'(s)$ die Wahrscheinlichkeit dafür, daß die gemäß der jeweiligen Offsetkombination für die i-te Stelle „zweitbeste“ PIN-Ziffer die eigentlich richtige ist. (Falls es mehrere „beste“ PIN-Ziffern gibt, nimmt man als „zweitbeste“ natürlich auch eine davon.)[9]

Entsprechend kann man bei drei erlaubten Versuchen noch zusätzlich einen Tip $n_1n_2n_3'n_4$ angeben. Die Erfolgswahrscheinlichkeit, mit dieser Methode als $n_1n_2n_3n_3$, $n_1n_2'n_3n_3$ oder $n_1n_2n_3'n_3$ die richtige PIN zu erhalten, beträgt dann

$$\begin{aligned} & E_1(s)E_2(s)E_3(s)E_4(s) \\ & +E_1(s)E_2'(s)E_3(s)E_4(s) \\ & +E_1(s)E_2(s)E_3'(s)E_4(s) \end{aligned}$$

$$= \left(1 + \frac{E_2'(s)}{E_2(s)} + \frac{E_3'(s)}{E_3(s)}\right) E(s).$$

Die Werte $E_i'(s)$ werden durch das Programm `offsets` bestimmt. Die Quotienten

$$q(s) := \frac{E_i'(s)}{E_i(s)} \qquad (i > 1)$$

sind für $s = 1, 2, 3$ Offsets in Tabelle 4 angegeben. Hierbei ist $q(1) = 1$ (das heißt, bei Kenntnis eines Offsets gibt es *immer* mehrere „beste“ PIN-Ziffern für die Stellen zwei bis vier); alle anderen Werte in der Tabelle sind gerundet. Außerdem angegeben sind die Abschätzungen für die Wahrscheinlichkeiten, mit zwei und drei Versuchen die PIN zu erraten, und deren Kehrwerte. Die mit optimaler Auswahl der geratenen PINs erreichbare Erfolgswahrscheinlichkeit bei r Versuchen bezeichnen wir als $E(r, s)$. Es ist also

$$(1 + q(s))E(s) \le E(2, s) \le 2E(s)$$

und

$$(1 + 2q(s))E(s) \le E(3, s) \le 3E(s).$$

(Bei $s = 1$ gilt wegen $q(s) = 1$ sogar Gleichheit.)

[9] Anders gesagt ist $E_i'(s)$ die Zahl mit

$$E_i(s) + E_i'(s)$$

$$= \sum_{(o_{p,i})_{p\in\underline{s}}\in(\mathbb{Z}_{10})^s} \mathcal{P}(\forall p\colon O_{p,i} = o_{p,i}) \cdot \max_{\{n_i, n_i'\}\subset\mathbb{Z}_{10}} \mathcal{P}(N_i \in \{n_i, n_i'\} \mid \forall p\colon O_{p,i} = o_{p,i}).$$

3.1 Verbesserungsmöglichkeiten

Die Methode der „Offsets“, wie sie in [3] festgelegt ist, führt zu einer unnötigen Verschlechterung der Sicherheit. Die Kenntnis der Offsets würde beim Erraten der PIN keinen Vorteil bringen, wenn anstelle von (3), also

$$\mathit{offset}_p = \mathit{PIN} - (\mathrm{dec} \circ \Pi \circ \mathrm{DES}_{K_p})(D),$$

etwa die Formel

$$\mathit{offset}_p' := \mathit{PIN} \oplus (\Pi \circ \mathrm{DES}_{K_p})(D) \tag{5}$$

benutzt würde; $\oplus$ steht hier für bitweises Exklusiv-Oder (XOR). offset_p' läge so in $\mathbb{X}^4$ (und i. a. nicht in $\mathbb{Z}_{10}{}^4$). Der Berechnungsaufwand für (5) und die zugehörige von institutsfremden Automaten zur PIN-Prüfung zu verwendende Rechenvorschrift

$$\mathit{PIN} = (\Pi \circ \mathrm{DES}_{K_p})(D) \oplus \mathit{offset}_p'$$

wäre sogar geringer als bei dem ec-PIN-Verfahren. Die Entscheidung für (3) ist also als ein grober Entwurfsfehler anzusehen.

Außerdem sollten zur Verbesserung der Sicherheit die PINs möglichst gleichverteilt aus dem erlaubten Bereich gewählt werden. Anders als bei obigem Kritikpunkt wäre diese Verbesserung allerdings nur mit einem deutlich erhöhten Realisierungsaufwand möglich, was diesen Mangel erklären könnte. Um beinahe gleichverteilte PINs von 1000 bis 9999 zu erhalten, könnte man zum Beispiel die Ausgabe des Verschlüsselungsalgorithmus als eine 64 Bit lange natürliche Zahl auffassen und ihren modulo 9000 reduzierten Wert zu 1000 addieren:

$$\mathit{PIN}' := 1000 + (\mathrm{DES}_{K'}(D) \bmod 9000)$$

Schließlich sollte der Verschlüsselungsalgorithmus DES durch ein Verfahren mit einem größeren Schlüsselbereich ersetzt werden (dazu mehr im folgenden Abschnitt 4); auch bei den obigen Verbesserungsvorschlägen ist die Bezeichnung DES lediglich als Platzhalter für einen sichereren Algorithmus zu betrachten.

4 Schlüsselsuche

Wer einen der im ec-PIN-Verfahren benutzten Institutsschlüssel K' oder Poolschlüssel K_p kennt, kann damit aus den Daten vom Magnetstreifen einer ec-Karte sofort deren PIN bestimmen, sofern der betreffende Schlüssel beim Ausstellen dieser Karte für die PIN- bzw. Offset-Berechnung benutzt wurde. Hierbei ist die gleiche Berechnung durchzuführen, die auch ein Geldausgabeautomat beim Prüfen der PIN vornimmt (siehe Abschnitt 2.2).

Tabelle 4: Abschätzungen für die Erfolgsaussichten bei zwei/drei Versuchen

s	Anzahl Offsets	**1**	**2**	**3**
$q(s)$		1	0,872 340	0,826 923

Erfolgsaussichten beim Raten der PIN (zwei Rateversuche pro Karte)

$2E(s)$			0,003 287 552	0,004 669 237
			$\geq$	$\geq$
$E(2,s)$	Erfolgswahrscheinlichkeit	0,001 907 349	$E(2,2)$	$E(2,3)$
			$\geq$	$\geq$
$(1+q(s))E(s)$			0,003 077 708	0,004 265 167
$\frac{1}{2E(s)}$			304,1	214,1
			$\leq$	$\leq$
$\frac{1}{E(2,s)}$	Kartenanzahl pro Erfolg	**524,3**	$1/E(2,2)$	$1/E(2,3)$
			$\leq$	$\leq$
$\frac{1}{(1+q(s))E(s)}$			**325,0**	**234,5**

Erfolgsaussichten beim Raten der PIN (drei Rateversuche pro Karte)

$3E(s)$			0,004 931 328	0,007 003 855
			$\geq$	$\geq$
$E(3,s)$	Erfolgswahrscheinlichkeit	0,002 861 023	$E(3,2)$	$E(3,3)$
			$\geq$	$\geq$
$(1+2q(s))E(s)$			0,004 511 640	0,006 195 717
$\frac{1}{3E(s)}$			202,7	142,7
			$\leq$	$\leq$
$\frac{1}{E(3,s)}$	Kartenanzahl pro Erfolg	**349,5**	$1/E(3,2)$	$1/E(3,3)$
			$\leq$	$\leq$
$\frac{1}{(1+2q(s))E(s)}$			**221,7**	**161,5**

Wir wollen den Aufwand dafür abschätzen, durch erschöpfende Suche einen der DES-Schlüssel zu ermitteln. Als Voraussetzung nehmen wir an, daß man die Datensätze von fünf ec-Karten zur Verfügung hat (d. h., die jeweiligen Werte für D und die Offsets sind bekannt) und deren korrekte PINs kennt[10]. Diese Karten sollten vom gleichen Institut ausgegeben worden sein, damit nicht nur Poolschlüssel gesucht werden können, sondern auch der Institutsschlüssel K'. Falls die Vermutung in Abschnitt 2.2 richtig ist, daß von Zeit zu Zeit ein alter, ungültig gewordener Poolschlüssel durch einen neu erzeugten Schlüssel ersetzt wird, sollten außerdem alle Karten entsprechend neu sein, damit die Schlüsselsuche möglichst viele Ziele hat.

Für jede Karte und jedes p gilt wegen (3):

$$\mathit{EDP}_p = \mathit{PIN} - \mathit{offset}_p \tag{6}$$

Die Werte EDP_p lassen sich also ohne weiteres berechnen. Außerdem weiß man, daß es einen Schlüssel K_p gibt, so daß (bei allen Karten)

$$\mathit{EDP}_p = (\mathrm{dec} \circ \Pi \circ \mathrm{DES}_{K_p})(D)$$

gilt. Die Grundidee der Schlüsselsuche ist es, in beliebiger Reihenfolge alle möglichen Schlüssel K auszuprobieren, bis der korrekte gefunden ist. (Dabei ist eine beliebig starke Parallelisierung möglich, indem vielen identischen Einheiten verschiedene Teile des Schlüsselraumes zugeteilt werden.) Der Aufwand für die Schlüsselsuche wird nur wenig höher, wenn man nicht nur für ein bestimmtes p den Poolschlüssel K_p sucht, sondern nach allen Poolschlüsseln (zu denen Offsets auf den Karten stehen) gleichzeitig sucht: Denn $(\mathrm{dec} \circ \Pi \circ \mathrm{DES}_K)(D)$ muß nur einmal berechnet werden, und das Vergleichen mit mehreren verschiedenen Werten EDP_p kostet nur einen geringen Aufwand. Entsprechend kann man auch den Institutsschlüssel K' mitsuchen: Hierfür muß man zunächst ggf. eine 0 an der ersten Stelle durch eine 1 ersetzen, um $(\mathrm{dec}' \circ \Pi \circ \mathrm{DES}_K)(D)$ zu erhalten; dieses Ergebnis vergleicht man mit der PIN der betreffenden Karte.

Im folgenden soll der ungefähre Aufwand für das Durchsuchen des *kompletten* Schlüsselraumes (es gibt bei DES 2^{56} verschiedene Schlüssel) bestimmt werden; $T_{\max}$ sei der Erwartungswert für die Zeit, die für dieses vollständige Absuchen gebraucht wird. Der Erwartungswert für die Zeit, nach der ein *bestimmter* Schlüssel gefunden wird, ist dann $\frac{1}{2}T_{\max}$. Bei der Suche nach n Schlüsseln (dem Institutsschlüssel und ein bis drei Poolschlüsseln) beträgt der Erwartungswert für die Zeit bis zum Auffinden *irgendeines* dieser Schlüssel $\frac{1}{n+1}T_{\max}$, bis zum Auffinden *sämtlicher* Schlüssel $\left(1 - \frac{1}{n+1}\right) T_{\max}$.[11]

[10] Solche Ausgangsdatensätze lassen sich zum Beispiel auch dadurch beschaffen, daß man eine ec-Karte „verliert“: Die Ersatzkarte mit einer neuen Kartenfolgenummer – also einem anderen D – und einer neuen PIN ist genauso geeignet wie eine ec-Karte für ein anderes Konto.

[11] Die Zeit bis zum Auffinden des i-ten der n Schlüssels kann man (mit unbedeutendem Fehler) als $T_{\max} \cdot T_i$ ausdrücken, wobei T_i eine auf dem Intervall $[0,1]$ gleichverteilte Zufallsvariable ist. Bei stochastisch unabhängigen $T_1, \ldots, T_n$ gilt dann für den Erwartungswert von

Bei der Suche ist zu beachten, daß die Gültigkeit der Gleichung

$$EDP_p = (\mathrm{dec} \circ \Pi \circ \mathrm{DES}_K)(D) \tag{7}$$

bzw.

$$PIN = (\mathrm{dec}' \circ \Pi \circ \mathrm{DES}_K)(D) \tag{8}$$

für ein bestimmtes K noch nicht heißen muß, daß dieses K auch tatsächlich der gesuchte Schlüssel K_p bzw. K' ist. (Wenn die Gleichung *nicht* gilt, ist allerdings klar, daß $K \neq K_p$ bzw. $K \neq K'$ ist.) Deshalb wurde verlangt, daß fünf verschiedene Datensätze $(D, PIN, (offset_p))$ – und damit nach (6) auch $(D, (EDP_p))$ – vorliegen.

Wir wählen folgende Strategie: Eine der Karten wird als „erste“ Karte ausgewählt. Für jeden möglichen DES-Schlüssel K wird für diese „erste“ Karte überprüft, ob (7) für ein oder mehrere p gilt und/oder ob (8) gilt. Für diejenigen Paare (K, p), die diese erste Filterung durch (7) überstehen, wird auch anhand der Daten (D, EDP_p) der anderen vier Karten überprüft, ob (7) für diese ebenfalls gilt. Ist dies der Fall, so wird (K, p) als Suchergebnis ausgegeben. Entsprechend wird bei denjenigen K, die den Test mit (8) bestanden haben, die Gültigkeit von (8) für die Daten (D, PIN) der anderen Karten geprüft; bei Erfolg wird K ausgegeben.[12]

Für ein „falsches“ K – also einen Schlüssel, der keinem der gesuchten Schlüssel entspricht –, ist der Wert $(\Pi \circ \mathrm{DES}_K)(D)$ quasi zufällig mit Gleichverteilung in $\mathbb{X}^4$. Die Wahrscheinlichkeit, daß (7) für ein bestimmtes p gilt, ist dabei schlimmstenfalls 2^{-12} (denn im ungünstigsten Fall gibt es nach der Definition von dec bei jeder der vier Stellen von $(\Pi \circ \mathrm{DES}_K)(D)$ zwei Möglichkeiten, für die $(\mathrm{dec} \circ \Pi \circ \mathrm{DES}_K)(D)$ den vorgegebenen Wert hat; dann gilt (7) also in 2^4 von $16^4 = 2^{16}$ Fällen). Entsprechend ist die Wahrscheinlichkeit, daß (8) gilt, schlimmstenfalls 2^{-11}.

In der ersten Suchebene werden alle 2^{56} DES-Schlüssel ausprobiert. Für jedes p kommen voraussichtlich höchstens etwa $2^{56} \cdot 2^{-12}$ davon als mögliche Poolschlüssel in die engere Auswahl, außerdem voraussichtlich höchstens etwa $2^{56} \cdot 2^{-11}$ als mögliche Institutsschlüssel. Es sind also zusammen weniger als etwa $2^{56} \cdot (3 \cdot 2^{-12} + 2^{-11})$ genauere Überprüfungen zu erwarten. Der Erwartungswert für die Anzahl der dabei jeweils vorzunehmenden Verschlüsselungen beträgt weniger als $1 + 2^{-11} \cdot (1 + 2^{-11} \cdot (\ldots))$, wenn man die Schlüsselkandidaten nacheinander anhand der verschiedenen Kartendaten mit (7) bzw. (8) testet und nach einem Mißerfolg abbricht.

Diese Suchstrategie benötigt also voraussichtlich insgesamt weniger als $2^{56} + 2^{56} \cdot (3 \cdot 2^{-12} + 2^{-11}) \cdot (1 + 2^{-11}(\ldots))$, also weniger als $2^{56} \cdot (1 + 2^{-9})$, Verschlüsselungen für das Absuchen des gesamten Schlüsselraums. Fünf Karten werden meistens ausreichen, um jedes K_p eindeutig zu bestimmen: Denn die Wahrscheinlichkeit, daß (7) für (K, p) bei allen fünf Karten gilt, wenn tatsächlich $K \neq K_p$ ist, beträgt weniger als $(2^{-12})^5 = 2^{-60}$. Es sind also (pro p) weniger als etwa $2^{-60} \cdot 2^{56} = 2^{-4}$ falsche Suchergebnisse zu erwarten. Bei K' ist eher mit einem nicht eindeutigen Ergebnis zu rechnen. Falls der Suchvorgang für ein K_p oder für K' kein eindeutiges Ergebnis erzielt, kann man leicht anhand einer weiteren Karte (oder notfalls an mehreren) ausprobieren, welche Antwort richtig ist.

Gegenüber einer Schlüsselsuche, bei der ein vollständiges Plaintext/Ciphertext-Paar bekannt ist (also $P, C \in \mathbb{X}^{16}$, so daß für den zu bestimmenden Schlüssel K die Beziehung $C = \mathrm{DES}_K(P)$ gilt), steigt der Zeitaufwand also nur geringfügig. (Der erste Erfolg – das Auffinden irgendeines von mehreren Schlüsseln – tritt außerdem voraussichtlich eher ein, nämlich bei n gesuchten Schlüsseln nach der Zeit $\frac{1}{n+1} T_{\max}$.) Daher kann man sich hier an der Schätzung aus [8] orientieren:

> “At present, it would take a year and a half for someone using $ 10,000 worth of FPGA technology to search out a DES key. [...] A serious effort — on the order of $ 300,000 — by a legitimate or illegitimate business could find a DES key in an average of 19 days using off-the-shelf technology and in only 3 hours using a custom developed chip.”

Der Zeitaufwand zum Absuchen des gesamten Schlüsselraumes ist jeweils das Doppelte der im Zitat angegebenen Durchschnittszeit. Die Systemkosten wird man etwas höher ansetzen müssen, um die „mehrstufige Suche“ anhand der fünf Datensätze zu realisieren.

Angesichts der Geldbeträge, die nach einer erfolgreichen Schlüsselsuche mit gestohlenen ec-Karten an Geldausgabeautomaten illegal beschafft werden könnten[13], ist klar, daß die Sicherheit von (einfachem) DES für diese Anwendung bei weitem nicht ausreichend ist. Die Schlüssellänge muß deutlich höher liegen, es könnte zum Beispiel Triple-DES mit 112-Bit-Schlüsseln benutzt werden.

$\min(T_{\max} \cdot T_1, \ldots, T_{\max} \cdot T_n)$ folgendes, wobei

$$f(s, t_1, \ldots, t_n) := \begin{cases} 1 & \text{falls } \forall i \in \underline{n}\colon\ s < t_i \\ 0 & \text{sonst} \end{cases}$$

sei:

$$\begin{aligned} & \mathrm{E}[\min(T_{\max} \cdot T_1, \ldots, T_{\max} \cdot T_n)] \\ = \ & T_{\max} \cdot \int_{[0,1]^n} \min(t_1, \ldots, t_n)\, \mathrm{d}(t_1, \ldots, t_n) \\ = \ & T_{\max} \cdot \int_{[0,1]^n} \int_{[0,1]} f(s, t_1, \ldots, t_n)\, \mathrm{d}s\, \mathrm{d}(t_1, \ldots, t_n) \\ = \ & T_{\max} \cdot \int_{[0,1]} \int_{[0,1]^n} f(s, t_1, \ldots, t_n)\, \mathrm{d}(t_1, \ldots, t_n)\, \mathrm{d}s \\ = \ & T_{\max} \cdot \int_{[0,1]} (1-s)^n\, \mathrm{d}s \ = \ T_{\max} \cdot \frac{1}{n+1} \end{aligned}$$

Analog ist $\mathrm{E}[\max(T_{\max} \cdot T_1, \ldots, T_{\max} \cdot T_n)] = T_{\max} \cdot \left(1 - \frac{1}{n+1}\right)$.

[12] Es werden sehr viele Paare (K, p) den ersten Test mit (7) und sehr viele K den ersten Test mit (8) bestehen. Alle diese zwischenzuspeichern, bis die erste Sucheben vollständig abgearbeitet worden ist, wäre nicht sinnvoll. Andererseits kann es je nach Implementierung der Suche – insbesondere je nach benutzter DES-Implementierung – von Vorteil sein, erst eine bestimmte Anzahl von Paaren (K, p) bzw. von Institutsschlüssel-Kandidaten K zu sammeln, bevor diese anhand der Daten der weiteren Karten genauer geprüft werden.

[13] Solange eine ec-Karte noch nicht gesperrt ist, können DM 400,00 pro Tag an beliebigen ec-Geldausgabeautomaten abgehoben werden; bei der kontoführenden Bank gilt meistens eine höheres (vierstelliges) Limit.

Literatur

[1] *Die Bank,* Ausgabe 10/81, S. 407 und Ausgabe 2/83, S. 55

[2] Aufbau und Feldbelegung der 3. Spur von Magnetstreifen auf eurocheque-Karten zur Benutzung im institutsübergreifenden Geldausgabeautomaten-System; Anhang 1.1 zu den *Richtlinien für das institutsübergreifende Geldausgabeautomaten-System*

[3] Verschlüsselungsverfahren innerhalb des institutsübergreifenden Geldausgabeautomaten-Systems; Anhang 3 zu den *Richtlinien für das institutsübergreifende Geldausgabeautomaten-System*

[4] DIN ISO 4909 (Febr. 1989), *Bankkarten; Magnetkarten. Aufbau und Inhalt der Spur 3,* deutsche Fassung von ISO 4909:1987. Ersatz für Ausgabe 07.80 (ISO 4909:1978)

[5] International Organization for Standardization, <URL:http://www.iso.ch/cate/withdraw.html>

[6] H. S. Schmidt (IBM Informationssysteme Deutschland GmbH), *Sicherheitstechnologie für Transaktionssysteme der Kreditwirtschaft am Beispiel „electronic cash“,* Version 2.0, 23. Juni 1993

[7] US National Bureau of Standards, *Data Encryption Standard,* Federal Information Processing Standard (FIPS) Publication 46, 1977

[8] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, M. Wiener. *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security,* A report by an ad hoc group of cryptographers and computer scientists, January 1996

A Das Perl-Programm `offsets`

```
#!/usr/local/bin/perl -w

# Aufruf: offsets 1|2|3 0|1 [s]
#   Erstes Argument: Anzahl der Offsets
#   Zweites Argument: 0 für zweite bis
#                       vierte PIN-Ziffer,
#                     1 für die erste
#                       PIN-Ziffer.
#   Drittes Argument: "s" fuer Bestimmung
#                     der Wahrscheinlichk.
#                     für 2. Rateversuch

use English;

($s, $i, $second) = @ARGV;
$s = 3 unless defined $s;
die "1, 2 oder 3 Offsets\n"
    unless ($s==1 || $s==2 || $s==3);
$i = 0 unless $i == 1;
if (defined($second) && ($second eq "s")) {
  $second = 1;
} else {
  $second = 0;
}

$XYcases = 16 ** (1 + $s);
$offsetcases = 10 ** $s;


##########################################

# Aufstellen der Statistik


$OUTPUT_AUTOFLUSH = 1;

$X = 0;
for (1 .. $s) { $Y[$_] = 0; }

XY: while (1) {
  $M = $X % 10;
  $M = 1 if $M == 0 && $i == 1;

  for (1 .. $s) {
    $N[$_] = $Y[$_] % 10;
    $O[$_] = ($M - $N[$_]) % 10;
  }

  $offsets = 0;
  for (1 .. $s) {
    $offsets *= 10;
    $offsets += $O[$_];
  }

  # Ergebnisse registrieren
  $hits[$M][$offsets]++;
  $offsets[$offsets]++;

  # "Zufallsvariablen" für nächsten
  # Schleifendurchlauf vorbereiten
  $X++;
  if ($X == 16) {
    $X = 0;

    $p = 1;
    $Y[$p]++;
    while ($Y[$p] == 16) {
      # Ausführungsfortschritt anzeigen
      print "." if $p == 1;
      print "\n" if $p == 2;

      # Schon alle Fälle bearbeitet?
      last XY if $p == $s;

      $Y[$p] = 0;
      $Y[++$p]++;
    }
  }
}

$OUTPUT_AUTOFLUSH = 0;


##########################################

# Auswertung

$erfolge = 0;
```

```
$scnd_erfolge = 0;

for (1 .. $s) { $o[$_] = 0; }

OFFSETS: while (1) {

  unless ($second) {
    print "\n\nOffset";
    print "kombination" if $s > 1;
    for (1 .. $s) {
      print " " . $o[(1+$s) - $_];
    }
  }

  $offsets = 0;
  for (1 .. $s) {
    $offsets *= 10;
    $offsets += $o[$_];
  }

  $o_richtig = $offsets[$offsets];

  unless ($second) {
    printf "  (Wahrscheinlichkeit: %6.5f" .
      ")\n", $o_richtig/$offsetcases;
    print "=" x (($s > 1 ? 48 : 37) + 2*$s)
      . "\n";
    print "\n" unless $i == 1;
    if ($i == 1) {
      print " " x 40;
    }
  }


  ($best, @best) = (0);
  $scnd_best = 0;

  for ($i .. 9) {
    $pin = $_;

    $pin_und_o_richtig =
        $hits[$pin][$offsets];
    $sum_pin_und_offsets +=
        $pin_und_o_richtig;

    unless ($second) {
      printf "PIN-Ziffer $pin:  " .
        "%3d / %3d = %6.5f",
        $pin_und_o_richtig, $o_richtig,
        ($pin_und_o_richtig / $o_richtig);

      print $pin % 2 == 0 ? " "x6 : "\n";
    }

    if ($pin_und_o_richtig == $best) {
      @best = (@best, $pin);
    } elsif ($pin_und_o_richtig > $best) {
      if ($second) {
        if ($best > $scnd_best) {
          $scnd_best = $best;
        }
      }

      $best = $pin_und_o_richtig;
      @best = ($pin);
    }
    if ($second) {
      if ($pin_und_o_richtig > $scnd_best
          && $pin_und_o_richtig < $best) {
        $scnd_best = $pin_und_o_richtig;
      }
    }
  }

  unless ($second) {
    printf "\nBeste PIN-Ziffer" .
      ( @best > 1 ? "n" : "") .
      " (Wahrscheinlichkeit" .
      ( @best > 1 ? " jeweils" : "") .
      " %6.5f):", ($best / $o_richtig);
    print "  @best\n";
  }

  $erfolge += $best;

  if ($second) {
    $scnd_best = $best if (@best > 1);
    $scnd_erfolge += $scnd_best;
  }

  # Offsets für nächsten
  # Schleifendurchlauf bestimmen
  $p = 1;
  $o[$p]++;
  while ($o[$p] == 10) {
    # alle Offsetkombinationen bearbeitet?
    last OFFSETS if $p == $s;

    $o[$p] = 0;
    $o[++$p]++;
  }
}


print "\n\nInsgesamt würde man ";
if ($second) {
  print "im ersten Versuch ";
}
print "bei $erfolge" .
  " von $XYcases Fällen richtig raten";
printf "\n(Erfolgswahrscheinlichkeit: " .
    "%11.10f).\n\n", ($erfolge / $XYcases);

if ($second) {
  print "Erfolgswahrscheinlichkeit beim " .
    "zweiten Versuch: " .
    "$scnd_erfolge / $XYcases = ";
  printf "%11.10f\n\n",
    ($scnd_erfolge / $XYcases);
}
```