Math-Net.Ru

Общероссийский математический портал

L. R. Ahmetzyanova, E. K. Alekseev, G. A. Karpunin, S. V. Smyshlyaev, О криптографических свойствах процедур генерации параметров CVV и PVV в платежных системах, *Матем. вопр. криптогр.*, 2018, том 9, выпуск 2, 23–46

DOI: <https://doi.org/10.4213/mvk250>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.93.64.190

26 апреля 2023 г., 13:51:56

On cryptographic properties of the *CVV* and *PVV* parameters generation procedures in payment systems

L. R. Ahmetzyanova, E. K. Alekseev, G. A. Karpunin, S. V. Smyshlyaev

CryptoPro LLC, Moscow

Получено 06.11.2017

Abstract. Two important mechanisms to provide security of payment systems are the checks made with parameters *CVV* and *PVV*. In the current paper the provable security approach is exploited to analyze the two-pass decimalization procedure used, for example, by VISA. It is shown that the standard method of upper estimating the insecurity does not allow to claim security of this procedure since it provides not very good bounds for probabilistic characteristics of practical parameters. Therefore this procedure is not recommended to be used in the MIR payment system. We propose a simple procedure as a replacement that turns out to be much more secure.

The work was supported by the Russian Foundation of Basic Research, the project 16-01-00226 A.

Key words: transaction security, payment systems, adversary models, provable security, security estimates

О криптографических свойствах процедур генерации параметров *CVV* и *PVV* в платежных системах

Л. Р. Ахметзянова, Е. К. Алексеев, Г. А. Карпунин, С. В. Смышляев

ООО «КриптоПро», Москва

Аннотация. Процедуры проверки значений параметров *CVV* и *PVV* являются двумя важными механизмами обеспечения безопасности платежных систем. В работе с использованием аппарата теории доказуемой стойкости показано, что безопасность таких процедур существенным образом зависит от близости распределения наборов десятичных чисел, получающихся в результате процедуры децимализации, к равновероятному распределению. Показано, что используемая в современных платежных системах (например, в системе VISA) двухэтапная процедура децимализации приводит к существенным статистическим отклонениям, поэтому ее использование в платежной системе МИР не рекомендуется. Предлагается альтернативная процедура, обладающая лучшими характеристиками.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 16-01-00226 А.

Ключевые слова: защита транзакций, платежные системы, модели злоумышленника, доказуемая стойкость, оценки стойкости

Citation: *Mathematical Aspects of Cryptography*, 2018, v. 9, № 2, pp. 23–46 (Russian)

© Академия криптографии Российской Федерации, 2018 г.

1. Introduction

Cryptographic mechanisms ensuring the transactions security in the Russian national payment system MIR are being developed. Such mechanisms include procedures for calculating the card check values and the *PIN*-codes check values (*CVV* and *PVV* respectively).

The most common payment systems VISA and Mastercard use a scheme that is based upon consistent application of well-known CBCMAC [1] scheme and so-called two-pass decimalization function, see, for example, [3], for procedures of calculating *CVV* and *PVV*. For the payment system MIR analogues of these procedures based on Russian national cryptographic standards, in particular block cipher Magma (GOST R 34.12–2015 [6]), are intended to be used.

The security of these procedures largely depend on the probability distribution on the set of strings of decimal digits that is induced by the two-pass decimalization function. The closer this distribution to the uniform one, the more secure the procedure. In the current paper we describe the probability distribution of the two-pass decimalization function. A simple analogue of such a function is described and it is shown that its probability distribution turns out to be much closer to the uniform one.

Below we define adversary models that are interesting from the practically-oriented point of view. We also define stronger adversary models that are traditionally used in theoretical cryptology. The security in practice-oriented models is evaluated with estimations obtained in these stronger models. We show that for the two-pass decimalization the adversary's capabilities for which non-trivial security bounds may be obtained are much more poor than for the proposed analogue. The results are obtained under the assumption that the best method of distinguishing the block cipher Magma from random permutation is key recovering by brute force.

2. Notation

Denote by V_n the set of n -component strings over $\mathbb{F}_2$. By $\mathbb{B}_n$ we denote the set of all byte strings of length n . By V_{n*} we denote the set of all strings with elements from $\mathbb{F}_2$ with non-zero length which is multiple of n .

Let $INT: V_n \rightarrow \mathbb{N}$ be the function such that

$$INT(X) = 2^{n-1} \cdot X_1 + 2^{n-2} \cdot X_2 + \dots + 2^0 \cdot X_n$$

if $X = (X_1, X_2, \dots, X_n) \in V_n$.

By $\mathbb{H}$ we denote the set $\{0, 1, \dots, 15\}$. By $\mathbb{D}$ we denote the set $\{0, 1, \dots, 9\}$. The elements $10, 11, \dots, 15$ of $\mathbb{H}$ are called *hex-symbols*, other elements of this set, which are members of $\mathbb{D}$, are called *decimal symbols*. The hex-symbols $10, 11, 12, 13, 14, 15$ we'll denote by A, B, C, D, E, F respectively. Strings from $\mathbb{B}_n$ are interpreted as strings from $\mathbb{H}^{2n}$ in a natural way (corresponding hex-symbols are obtained by transformation of leading and trailing 4-bit parts of bytes from left to right). Positive integers are also interpreted as strings from $\mathbb{H}^n$ or $\mathbb{D}^n$ for some n in a natural way ($123 \rightarrow 07B \in \mathbb{H}^3$ and $123 \rightarrow 0123 \in \mathbb{D}^4$).

For sets A, B by $Func(A, B)$ we'll denote all mappings from A to B . In the case $A = B$, by $Func(A)$ we'll denote $Func(A, A)$ for short.

By $Perm(A)$ we'll denote the family of all bijective transformations of A . A *block cipher* E is the family

$$\{E_K \mid K \in V_k\} \subseteq Perm(V_n),$$

the string K is called the *key*.

If there is some probability distribution $\mathcal{D}$ defined over the set Ω , then a random selection of element ω from Ω in accordance with the distribution $\mathcal{D}$ will be denoted as $\omega \xleftarrow{\mathcal{D}} \Omega$. The symbol $\mathcal{U}$ will denote a uniform distribution.

3. The procedure of computing check values

Definition 3.1. The *two-pass decimalization* function is the mapping

$$DEC_{m,r}^2: \mathbb{H}^m \rightarrow \mathbb{D}^r, \quad m \geq r,$$

that gives the output on an input $X \in \mathbb{H}^m$ by the following algorithm. If there are r decimal symbols in X , then the string $DEC_{m,r}(X)$ is the concatenation of the first r of them (from left to right). If the total number s of decimal symbols is less than r , then the string $DEC_{m,r}(X)$ is the concatenation of these s symbols and $r - s$ residues modulo 10 of the first $r - s$ hex-symbols of X .

For example,

$$DEC_{5,3}^2(0\|1\|C\|D\|E) = 0\|1\|(12 \bmod 10) = 0\|1\|2.$$

Definition 3.2. The *modular decimalization* function is the mapping $DEC_{m,r}^M: \mathbb{H}^m \rightarrow \mathbb{D}^r$, $m \geq r$, such that its value on the input $X \in \mathbb{H}^m$ is equal to $DEC_{m,r}^M(X) = INT(X) \bmod 10^r$.

For example,

$$DEC_{5,3}^M(0\|1\|C\|D\|E) = 0x1CDE \bmod 10^3 = 7390 \bmod 10^3 = 3\|9\|0.$$

Definition 3.3. For some block cipher $E = \{E_K \mid K \in V_k\} \subseteq \text{Perm}(V_n)$ by CBCMAC we denote the mapping $\text{CBCMAC}: V_k \times V_{n*} \rightarrow V_n$ defined by the formula

$$\text{CBCMAC}_K(M) = \text{CBCMAC}(K, M) = E_K(E_K(\dots(E_K(M_1) \oplus M_2) \oplus \dots) \oplus M_t),$$

where $M = M_1 \parallel \dots \parallel M_t$, $M_i \in V_n$.

From now on we will consider a block cipher E with block size $n = 64$ and key size $k = 256$.

Definition 3.4. By F^C we denote the family of functions $F_K^C: V_{128} \rightarrow \mathbb{D}^3$ defined in the following way:

$$F_K^C(M) = \text{DEC}_{16,3}(\text{CBCMAC}_K(M)),$$

where $\text{DEC}_{16,3}$ is one of the decimalization functions defined earlier. The choice of a particular function is clarified for concrete cases.

Definition 3.5. By F^P we denote the set of functions $F_K^P: V_{64} \rightarrow \mathbb{D}^4$ defined in the following way:

$$F_K^P(M) = \text{DEC}_{16,4}(\text{CBCMAC}_K(M)),$$

where $\text{DEC}_{16,4}$ is one of decimalization functions defined earlier. The choice of a particular function is clarified for specific cases.

In the notations described above the CVV and PVV parameters for the MIR payment system may be calculated as follows.

The check value CVV of a card is calculated according to a card number PAN (Personal Account Number, usually, 12–16 digits), an expiration date $ExpDate$ (4 digits in the form YYMM) and a service code SVC (3 digits, can take the only 6 values: 000, 999, 200, 201, 220, 221) using a key CVK (32 bytes). Before the main computation a 128-bit message M is generated:

$$M = (M_1 \parallel M_2) = (PAN \parallel \underbrace{0 \dots 0}_{pad} \parallel ExpDate \parallel SVC \parallel 0000000000),$$

where pad is the the minimum number of zero hex characters to make the length of $(PAN \parallel 0 \dots 0)$ equal to 16 hex characters.

The main CVV calculation uses F_{CVK}^C function:

$$CVV = F_{CVK}^C(M).$$

Similarly, the check value PVV of a card PIN-code is calculated according to a card number PAN , a key index $PVKI$ (number in range $0 \dots 6$), a PIN-code PIN (4 digits) using a key PVK (32 bytes). Before the main computation a 64-bit message M is generated:

$$M = (PAN|_{11} || PVKI || PIN),$$

where $PAN|_{11}$ means the first 11 digits of the card number PAN .

The main PVV calculation uses F_{PVK}^P function:

$$PVV = F_{PVK}^P(M).$$

4. Adversary models

Consider some adversary models, threats and attack scenarios defined from the practice-oriented point of view.

4.1. Models PRF, PRP-CPA and MAC-CPA

In the current section we provide the descriptions of standard adversary models that are used in mathematical cryptology for the analysis of various sets of functions (see [5]).

During the following considerations we assume that a certain computational model is chosen together with some encoding method in this model. By *adversary* we'll assume any probabilistic algorithm in that computational model. By *computational resources* we'll assume the sum of the number of steps made during his work in the worst case, and the encoding length of this adversary.

Let M be some adversary model and $\text{Adv}^M(\mathcal{A})$ is some characteristic of possibilities of $\mathcal{A}$ when implementing the threat in the model M . Let T be a tuple of limitations on computational resources of an adversary and any other parameters characterizing its work in the model M . For example, T may additionally contain the values limiting the total number of queries to the oracles considered by the model M . The set $\mathcal{A}(T)$ of the adversaries satisfying the limitations that are defined by T is finite. By $\text{Insec}^M(T)$ we'll denote the maximal value $\text{Adv}^M(\mathcal{A})$ for adversaries $\mathcal{A}$ in the set $\mathcal{A}(T)$.

From now on by $\text{Insec}^M(t, q, m)$ for a model M we'll denote a maximal value $\text{Adv}^M(\mathcal{A})$ for adversaries $\mathcal{A}$ in the set $\mathcal{A}(t, q, m)$, where t denotes the computational resources limitations, q is the maximal available number of queries to the exploited oracle, m is the maximal number of blocks in queries (optionally).

Definition 4.1. *The model MAC-CPA for a family $F = \{F_K: D \rightarrow R \mid K \in V_k\}$, where D, R are some arbitrary sets, is defined as follows. An adversary $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{MAC-CPA}}$ that chooses a key $K \xleftarrow{\mathcal{U}} V_k$ before starting his work. The adversary can make queries $M \in D$ to the oracle $\mathcal{O}^{\text{MAC-CPA}}$, and the oracle $\mathcal{O}^{\text{MAC-CPA}}$ responds with strings $MAC = F_K(M)$. As a result of computations the adversary $\mathcal{A}$ returns a pair (M', MAC') , where $M' \in D$ is not equal to any of queries made to $\mathcal{O}^{\text{MAC-CPA}}$. An advantage $\text{Adv}_F^{\text{MAC-CPA}}(\mathcal{A})$ of the adversary $\mathcal{A}$ in the model MAC-CPA is defined in the following way:*

$$\text{Adv}_F^{\text{MAC-CPA}}(\mathcal{A}) = \Pr[F_K(M') = MAC'].$$

The model MAC-CPA is used to obtain security estimations for functions in some family regarding to a forgery threat.

Definition 4.2. *The model PRF for a family $F = \{F_K: D \rightarrow R \mid K \in V_k\}$, where D, R are some arbitrary sets, is defined as follows. An adversary $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{PRF}}$ that chooses a bit $b \xleftarrow{\mathcal{U}} \{0, 1\}$ and if $b = 1$, then the oracle chooses a key $K \xleftarrow{\mathcal{U}} V_k$, assuming later $F' = F_K(\cdot)$. In the case $b = 0$ the oracle chooses $F' \xleftarrow{\mathcal{U}} \text{Func}(D, R)$. The adversary can make to $\mathcal{O}^{\text{PRF}}$ queries $M \in D$, and the oracle $\mathcal{O}^{\text{PRF}}$ responds with strings $RND = F'(M)$. As a result of computations based on pairs (M, RND) the adversary returns a bit $a \in \{0, 1\}$. An advantage $\text{Adv}_F^{\text{PRF}}(\mathcal{A})$ of the adversary $\mathcal{A}$ in the model PRF is defined in the following way:*

$$\text{Adv}_F^{\text{PRF}}(\mathcal{A}) = \Pr[a = 1 \mid b = 1] - \Pr[a = 1 \mid b = 0].$$

Informally, in the model PRF the adversary has to distinguish the target family from the random oracle.

Definition 4.3. *The model PRP-CPA for a cipher $E = \{E_K \mid K \in V_k\} \subseteq \text{Perm}(V_n)$ is defined as follows. An adversary $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{PRP-CPA}}$ that chooses a bit $b \xleftarrow{\mathcal{U}} \{0, 1\}$ and if $b = 1$, then the oracle chooses a key $K \xleftarrow{\mathcal{U}} V_k$, assuming later $F = E_K(\cdot)$. In the case $b = 0$ the oracle chooses $F \xleftarrow{\mathcal{U}} \text{Perm}(V_n)$. The adversary can make to $\mathcal{O}^{\text{PRP-CPA}}$ queries $M \in V_n$, and the oracle $\mathcal{O}^{\text{PRP-CPA}}$ responds with strings $C = F(M)$. As a result of computations based on pairs (M, C) the adversary returns a bit $a \in \{0, 1\}$. An advantage $\text{Adv}_E^{\text{PRP-CPA}}(\mathcal{A})$ of the adversary $\mathcal{A}$ in the model PRP-CPA is defined in the following way:*

$$\text{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}) = \Pr[a = 1 \mid b = 1] - \Pr[a = 1 \mid b = 0].$$

The model PRP-CPA is close to the model PRF, but is usually used to evaluate security of families of bijective mappings.

4.2. Search for the *CVV* value of a certain attacked card

The relevant adversary model for the task of searching for the *CVV* value of a certain attacked card is defined as follows. The adversary knows the parameters of q cards that have been issued by the issuer using the same key *CVK* unknown to the adversary, i. e., q sets $(PAN, ExpDate, SVC)$ and corresponding *CVV* values.

In practice, the adversary may obtain these values by stealing cards or spying on their parameters (including the *CVV*). Also some cards may be issued for the adversary with the actual key *CVK*.

The threat in this model is defined as follows: the adversary finds the correct value *CVV* for a certain (attacked) card with known parameters $(PAN, ExpDate, SVC)$, for which the corresponding value *CVV* was unknown.

This would give the adversary the possibility to make CNP (Card-Not-Present) transactions (e.g. for e-commerce).

4.3. Search for the (PIN, PVV) values of a certain attacked card

The relevant adversary model in the task of searching for the correct (PIN, PVV) pair of a certain attacked card is defined as follows. The adversary knows the parameters of q cards that have been issued by the issuer using the same key *PVK* unknown to the adversary, i. e., q sets $(PAN, PVKI)$ and corresponding (PIN, PVV) values.

In practice, the adversary may obtain these values by stealing cards together with PIN-codes. Also some cards may be issued for the adversary with the actual key value *PVK*. Also for an adversary to obtain the *PVV* values we assume that *PVV* are written on cards and the adversary can read them, and/or the *PVV* are stored in the issuer database and the adversary has access to it.

The threat in this model is defined as follows: the adversary finds the correct pair of values (PIN, PVV) for a certain (attacked) card with known parameters $(PAN, PVKI)$, for which the corresponding value of correct pair was unknown.

This would give the adversary the possibility to clone an existing card and make transactions with it in ATMs.

In the cryptographic analysis of the properties of the *CVV* and *PVV* computation functions the standard MAC-CPA model for the function families F^C and F^P is used. Capabilities of an adversary in this model are significantly stronger than ones that an adversary may have in practice.

The strengthening consists in allowing an adversary to obtain the values of CVV and (PIN, PVV) for any tuples of card parameters, and in case of PVV also to choose PIN values by himself. Also the strengthening of the model consists in replacing the selective forgery threat (where an adversary has to obtain a correct value of CVV and (PIN, PVV) for a certain parameter values) with the existential forgery threat. In the former case an adversary has to obtain a correct pair $((PAN, ExpDate, SV), CVV)$ (in case of CVV) or $((PAN, PVKI), (PIN, PVV))$ (in case of PVV).

4.4. Search for the PIN value of a certain attacked card with known PVV

The relevant adversary model for the task of searching for the correct PIN value of a certain attacked card in cases with known PVV value is defined as follows. The adversary knows the parameters of q different cards that have been issued by the issuer using the same key PVK unknown to the adversary, i.e., q different sets $(PAN_1, PVKI_1), \dots, (PAN_q, PVKI_q)$ and corresponding pairs of values $(PIN_1, PVV_1), \dots, (PIN_q, PVV_q)$. Moreover, for a certain attacked card with known parameters $(PAN, PVKI)$ the adversary knows only the correct PVV value.

In practice, the adversary may obtain these values by stealing cards together with PIN -codes. Also some cards can be issued for the adversary with the actual key value PVK . Also for the adversary to obtain the PVV_i values we assume that PVV_i are written on cards and the adversary can read them, and/or the PVV_i are stored in the issuer database and the adversary has access to it.

The threat in this model is defined in the following way: the adversary finds the correct value of PIN for a certain (attacked) card with known parameters $(PAN, PVKI)$ and known PVV .

This would give the adversary the possibility to clone an existing (possibly stolen) card and make transactions with it in ATMs.

Denote the considered model by **PR** (for **PIN Recovery**). To formalize the model we consider an oracle $\mathcal{O}_{FP}$ which implements a randomly selected function $F_K^P \xleftarrow{\mathcal{U}} F^P$. The values $(PAN, PVKI)$ are fed to the input of the oracle $\mathcal{O}_{FP}$ together with the value $res \in \{\text{FULL}, \text{REDUCED}\}$. The oracle $\mathcal{O}_{FP}$ chooses PIN randomly (according to the uniform distribution) and calculates the value $PVV = F_K^P(PAN \| PVKI \| PIN)$. If the flag $res = \text{FULL}$, then the oracle outputs the pair (PIN, PVV) , if $res = \text{REDUCED}$, then only PVV .

Assume that the adversary $\mathcal{A}^{\mathcal{O}_{FP}}$ makes q different queries $(PAN_1, PVKI_1), \dots, (PAN_q, PVKI_q)$ with the flag $res = \text{FULL}$ to this oracle and receives $(PIN_1, PVV_1), \dots, (PIN_q, PVV_q)$; and also makes one another query $(PAN, PVKI)$ with the flag $res = \text{REDUCED}$ and receives PVV as a result. Then the adversary tries to find a PIN' such that $PVV = F_K^P(PAN \| PVKI \| PIN')$.

Adversary success regarding this threat is estimated by the advantage

$$\text{Adv}_{FP}^{\text{PR}}(\mathcal{A}^{\mathcal{O}_{FP}}) = \Pr[\mathcal{A}^{\mathcal{O}_{FP}} \Rightarrow PIN' : PVV = F_K^P(PAN \| PVKI \| PIN')].$$

5. Distributions of the decimalization functions

In the current section we describe the distributions of the random variables $\text{DEC}_{m,r}^2(H)$ and $\text{DEC}_{m,r}^M(H)$ in case when H is chosen from $\mathbb{H}^m$ with the uniform distribution. We consider particular cases of (m, r) corresponding to the calculation of CVV and PVV . The distributions of these random variables define the resulting security bounds for the procedures analyzed in the current paper.

The output distribution of the function $\text{DEC}_{m,r}^2$ is computed in Appendix, Section 8.2, formula (7). For both cases CVV and PVV these distributions are presented in Table 1.

The output distribution of the function $\text{DEC}_{m,r}^M$ is computed in Appendix, Section 8.2, formulas (8) and (9). For both cases CVV and PVV these distributions are presented in Table 2.

For each case in Tables 1, 2 by “# values $V \in \mathbb{D}^r$ ” we mean a number of $V \in \mathbb{D}^r$ such that $\Pr[\text{DEC}_{m,r}(H) = V]$ equals a specified probability.

Table 1. Distributions of $\text{DEC}_{m,r}^2$ for CVV and PVV cases

CVV case ($m = 16, r = 3$)		PVV case ($m = 16, r = 4$)	
# values $V \in \mathbb{D}^3$	$\Pr[\text{DEC}_{16,3}^2(H) = V]$	# values $V \in \mathbb{D}^4$	$\Pr[\text{DEC}_{16,4}^2(H) = V]$
240	$\approx 10^{-3} + 2.975 \cdot 10^{-8}$	864	$\approx 10^{-4} + 3.696 \cdot 10^{-8}$
144	$\approx 10^{-3} + 4.108 \cdot 10^{-8}$	2400	$\approx 10^{-4} + 2.091 \cdot 10^{-8}$
216	$\approx 10^{-3} + 4.179 \cdot 10^{-8}$	1440	$\approx 10^{-4} + 3.507 \cdot 10^{-8}$
400	$\approx 10^{-3} - 5.520 \cdot 10^{-8}$	1296	$\approx 10^{-4} + 3.707 \cdot 10^{-8}$
		4000	$\approx 10^{-4} - 4.517 \cdot 10^{-8}$

Table 2. Distributions of $\text{DEC}_{m,r}^M$ for CVV and PVV cases

CVV case ($m = 16, r = 3$)		PVV case ($m = 16, r = 4$)	
# values $V \in \mathbb{D}^3$	$\Pr[\text{DEC}_{16,3}^M(H) = V]$	# values $V \in \mathbb{D}^4$	$\Pr[\text{DEC}_{16,4}^M(H) = V]$
616	$\approx 10^{-3} + 2.082 \cdot 10^{-20}$	1616	$\approx 10^{-4} + 4.545 \cdot 10^{-20}$
384	$\approx 10^{-3} - 3.339 \cdot 10^{-20}$	8384	$\approx 10^{-4} - 8.760 \cdot 10^{-21}$

6. Cryptographic analysis of the check values evaluation function

6.1. Security evaluation regarding to the forgery threat of check values

In the current section the security of F^C and F^P regarding to the forgery threat of check values (generated by them) is evaluated, i.e. in the model MAC-CPA.

Using the final estimates of Section 8.4 (formulas (18)–(21)) and estimate (13) we obtain the following relations which are considered as main results of the paper:

$$\text{Insec}_{F^C}^{\text{MAC-CPA}}(t, q) \leq 10^{-3} + \frac{2t}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{1}{2} - \delta, \quad (1)$$

where $\delta = \frac{1}{2} \cdot e^{-6q \cdot 10^{-5}}$ for $\text{DEC}_{16,3}^2$ and $\delta = \frac{1}{2} \cdot e^{-4q \cdot 10^{-17}}$ for $\text{DEC}_{16,3}^M$.

$$\text{Insec}_{F^P}^{\text{MAC-CPA}}(t, q) \leq 10^{-4} + \frac{2t}{2^k} + \frac{q^2}{2^{n-1}} + \frac{1}{2} - \delta, \quad (2)$$

where

$$\delta = \begin{cases} \frac{1}{2} \cdot e^{-5q \cdot 10^{-4}} & \text{for } \text{DEC}_{16,4}^2, \\ \frac{1}{2} \cdot e^{-5q \cdot 10^{-16}} & \text{for } \text{DEC}_{16,4}^M. \end{cases}$$

Пример 6.1. et us consider estimates (1), (2) for the payment system MIR. Since block cipher Magma is intended to be used in MIR, we have $n = 64$, $k = 256$. Also, suppose that adversary's computational resources t are bounded by 2^{128} ; the number of queries q is bounded by 10^7 . The last value is an upper bound for the number of payment cards emitted by a bank during the lifetime of the key CVK and/or PVK . Under these settings we obtain the following relations for the function DEC^M :

$$\text{Insec}_{F^C}^{\text{MAC-CPA}}(t, q) \leq 10^{-3} + 10^{-4.36} + 10^{-9.83}, \quad (3)$$

$$\text{Insec}_{F^P}^{\text{MAC-CPA}}(t, q) \leq 10^{-4} + 10^{-4.96} + 10^{-8.70}. \quad (4)$$

Note that for the case of DEC^2 similar bounds obtained via (1),(2) with the same parameter values is not small enough to claim security of DEC^2 : $\text{Insec}_{F^C}^{\text{MAC-CPA}}(t, q) \leq 0.51$ and $\text{Insec}_{F^P}^{\text{MAC-CPA}}(t, q) \leq 0.501$. Indeed, the probability near 0.5 is significantly greater than a priori probability 10^{-3} (10^{-4}) for random guessing CVV (PVV) value.

Remark 6.1. The obtained estimations for DEC^2 does not necessarily mean that the functions F^C and F^P with DEC^2 are insecure. It means that the used method to upper estimate insecurity does not allow to prove inexistence of adversaries with parameters bounded by t and q that are able to forge with probabilities significantly greater than a priori bound for guessing. It should be also noted that there are a lot of examples showing that the bounds obtained by this methods are close to tight (e.g., see [4]). By «tightness» in this case we mean the existence of a particular adversary with the limited (accordingly) resources who gains the obtained success probability bound.

6.2. Security evaluation regarding to the threat of obtaining PIN

In the current section the security of F^P regarding to the threat of obtaining PIN in the model PR is considered.

Combining the final estimates of Section 8.4 (formulas (18)–(21)) and Theorem 8.8, we obtain the following relations for the function F^P . These relations with estimates (1), (2) are considered as main results of the paper:

$$\text{Insec}_{F^P}^{\text{PR}}(t, q) \leq \frac{2t}{2^k} + \frac{(q+2)^2}{2^{n-1}} + \frac{2}{10^4} - \frac{1}{10^8} + \frac{1}{2} - \delta, \quad (5)$$

where

$$\delta = \begin{cases} \frac{1}{2} \cdot e^{-5q \cdot 10^{-4}} & \text{for } \text{DEC}_{16,4}^2, \\ \frac{1}{2} \cdot e^{-5q \cdot 10^{-16}} & \text{for } \text{DEC}_{16,4}^M. \end{cases}$$

Пример 6.2. Consider the obtained estimates for the settings of Example 6.1:

$$n = 64, \quad k = 256, \quad t = 2^{128}, \quad q \leq 10^7.$$

Under these settings we obtain the following bounds for the case of usage of DEC^M :

$$\text{Insec}_{F^P}^{\text{PR}}(t, q) \leq 2 \cdot 10^{-4} + 10^{-4.96} + 10^{-8.70}. \quad (6)$$

Here, similar to the model MAC-CPA, bound (5) for $\text{DEC}_{16,4}^2$ under the given settings is not small enough to claim security of DEC^2 :

$$\text{Insec}_{F^P}^{\text{RP}}(t, q) \leq 0.501.$$

7. Conclusion

In the current paper we obtain the security bounds for functions used to generate the check values CVV and PVV regarding to the forgery threat. We show that the usage of the function DEC^M leads to negligible difference between a priori guess probability and the probability of success of an adversary obtaining not more than 10^7 values CVV and PVV . It is also shown that the function F^P is secure regarding to the threat of obtaining PIN value when used with DEC^M ; i. e. its probability of success for an adversary with typically bounded resources is extremely close to the a priori one.

We show that for the same limitations, similar upper bounds for the functions using DEC^2 do not allow to claim security of this procedure.

The authors are very grateful to A. M. Zubkov and I. M. Goldovskii for fruitful discussions and to E. S. Smyshlyaeva for her valuable help, comments, and suggestions concerning the text of the paper.

References

- [1] Bellare M., Kilian J., Rogaway P., "The security of the cipher block chaining Message Authentication Code.", *J. Comput. Syst. Sci.*, **61**:3 (2000), 362–399
- [2] Nandi M., *A simple and unified method of proving unpredictability*, Cryptology eprint archive 2006/264, 2006
- [3] "Application Programmer's Guide, Appendix F. Cryptographic Algorithms and Processes, PIN Formats and Algorithms, VISA PIN Algorithms" (2011), http://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.csfb400/csfb4za2598.htm.
- [4] Iwata T., *Comments on "On the security of XCBC, TMAC and OMAC" by Mitchell*, 2003, <https://pdfs.semanticscholar.org/21b0/c40d3a08ffce60b11721b3fdd2516f37dce8.pdf>.
- [5] Bellare M., Rogaway P., *Introduction to Modern Cryptography: Lecture Notes*, 2005, 283 pp., <http://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf>.
- [6] Dolmatov V., *GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms*: RFC 5830, March 2010, <https://tools.ietf.org/html/rfc5830>.

8. Appendix

8.1. Calculating the distribution of $\text{DEC}_{m,r}^2$ outcomes

In this section we consider the following problem: calculate the probability distribution of the random variable $\text{DEC}_{m,r}^2(H)$, where H is taken from $\mathbb{H}^m$ according to the uniform distribution. To be definite we put $H = (h_1, \dots, h_m)$ and $\text{DEC}_{m,r}^2(H) = (d_1, \dots, d_r)$.

Let us fix some decimal digits $d_1, d_2, \dots, d_r \in \mathbb{D}$. Now we split the event

$$\{H \in \mathbb{H}^m : \text{DEC}_{m,r}^2(H) = (d_1, \dots, d_r)\}$$

into disjoint events of two types:

- A.** All decimal digits $d_1, d_2, \dots, d_r$ appear during the first pass of the two-pass decimalization. We denote this event by $\mathbf{A}(d_1, \dots, d_r)$.
- B.** The first s , $0 \leq s < r$, decimal digits $d_1, \dots, d_s$ appear during the first pass of the two-pass decimalization and the other $r - s$ digits $d_{s+1}, \dots, d_r$ appear during the second pass. Note that in this case the first s decimal digits $d_1, \dots, d_s$ may be arbitrary but digits $d_{s+1}, \dots, d_r$ should belong to $\{0, \dots, 5\}$. We denote this event by $\mathbf{B}(d_1, \dots, d_r; s)$.

Further, we calculate the probabilities of these events.

- A.** For convenience we split the event $\mathbf{A}(d_1, \dots, d_r)$ into events $\mathbf{A}_i$, where $\mathbf{A}_i = \{H \in \mathbf{A} : d_r = h_i\}$ is an event of occurring the last output digit d_r on the i th position of the input. Then

$$\Pr[\mathbf{A}(d_1, \dots, d_r)] = \frac{1}{16^m} \cdot \sum_{i=r}^m \binom{i-1}{r-1} \cdot (16-10)^{i-r} \cdot 16^{m-i}.$$

In this formula each summand is the cardinality of $\mathbf{A}_i$. The first factor $\binom{i-1}{r-1}$ of the summand is the number of ways to put other digits $d_1, \dots, d_{r-1}$ on the first $i-1$ input positions; the next factor $(16-10)^{i-r}$ is the number of ways to fill input positions $1, \dots, i$ by hexadecimal digits not belonging to $\{d_1, \dots, d_{r-1}\}$; finally, the factor 16^{m-i} is the number of ways to fill other input positions $i+1, \dots, m$ by arbitrary hexadecimal digits.

B. In this event only s digits of H are decimal digits, and they are equal to $d_1, d_2, \dots, d_s$. Other $m - s$ digits of H belong to $\{A, B, C, D, E, F\}$: the first $r - s$ digits of them are equal to $d_{s+1} + 10, \dots, d_r + 10$, and the other $m - r$ digits may be arbitrary. Then

$$\Pr[\mathbf{B}(d_1, \dots, d_r; s)] = \frac{1}{16^m} \binom{m}{s} \cdot (16 - 10)^{m-r}.$$

By definition, put $t = \max(\{0\} \cup \{i = 1, \dots, r \mid d_i \in \{6, 7, 8, 9\}\})$. It is easy to see that all digits $d_1, \dots, d_t$ appears during the first pass of the two-pass decimalization. Now, using the introduced notations, we obtain the final formula for calculating the probability of DEC^2 output:

$$\Pr[\text{DEC}_{m,r}^2(H) = (d_1, \dots, d_r)] = \Pr[\mathbf{A}(d_1, \dots, d_r)] + \sum_{s=t}^{r-1} \Pr[\mathbf{B}(d_1, \dots, d_r; s)]. \quad (7)$$

8.2. Calculating the distribution of $\text{DEC}_{m,r}^M$ outcomes

In this section we consider the following problem: calculate the probability distribution of the random variable $\text{DEC}_{m,r}^M(H)$, where H is taken from $\mathbb{H}^m$ according to the uniform distribution.

Let d and w be integers such that $16^m = d \cdot 10^r + w$, where $0 \leq w < 10^r$. Then for each of w values $V \in \{0, \dots, w - 1\}$ the cardinality of the preimage of V under $\text{DEC}_{m,r}^M$ is equal to

$$d + 1 = \frac{16^m + 10^r - w}{10^r}.$$

For each of all other $10^r - w$ values $V \in \{w, \dots, 10^r - 1\}$ the cardinality of the preimage of V under $\text{DEC}_{m,r}^M$ is equal to $d = (16^m - w)/10^r$. If H is taken from $\mathbb{H}^m$ according to the uniform distribution, then

$$\Pr[V] = \frac{16^m + 10^r - w}{10^r \cdot 16^m} = \frac{1}{10^r} + \frac{1 - w/10^r}{16^m} \quad \text{for } V \in \{0, \dots, w - 1\} \quad (8)$$

and

$$\Pr[V] = \frac{16^m - w}{10^r \cdot 16^m} = \frac{1}{10^r} - \frac{w/10^r}{16^m} \quad \text{for } V \in \{w, \dots, 10^r - 1\}. \quad (9)$$

8.3. Classical models and known results about their relationship

In the current section we provide known results on the security bounds in the models defined in Section 4.1.

8.3.1. On reductions and relation between Adv and Insec

In this section we briefly recall basic principles and goals of reductions. We also consider one subtlety that occurs during the transition from inequalities with Adv to inequalities with Insec. More precisely it is associated with the estimation of computational resources of one of the adversaries. The result of the following considerations is the quite natural Assumption 8.1 which allows to formulate rigorous statements on Insec.

Consider security models M and M' . Each of these models defines how an adversary may interact with target cryptosystem (e. g. they may define what oracles an adversary may interact with) and also defines the goal of an adversary — a threat. An adversary $\mathcal{A}'$ working in conditions defined by model M' (further shortly «working in M' ») may use any adversary $\mathcal{A}$ working in M as an interactive black box, i.e. $\mathcal{A}'$ intercepts $\mathcal{A}$'s oracle queries and provides $\mathcal{A}$ by corresponding answers formed by $\mathcal{A}'$. Also $\mathcal{A}'$ intercepts the result of $\mathcal{A}$'s work. We use the notation $\mathcal{A}'_{\mathcal{A}}$ to emphasize that $\mathcal{A}$ is an interactive black box of $\mathcal{A}'$. Since $\mathcal{A}'_{\mathcal{A}}$ uses $\mathcal{A}$ as a black box we can change $\mathcal{A}$ by another one, say $\mathcal{A}^*$ (working in M too), and consider the adversary $\mathcal{A}'_{\mathcal{A}^*}$. Here it is assumed that the difference between $\mathcal{A}'_{\mathcal{A}}$ and $\mathcal{A}'_{\mathcal{A}^*}$ is only in implementation of their black boxes (namely $\mathcal{A}$ and $\mathcal{A}^*$). When we need to talk on properties of $\mathcal{A}'_{\mathcal{A}}$ not considering its black box part we use the notation $\mathcal{A}'$ without specifying a particular adversary as a subscript.

The final goal of any cryptosystem $\mathcal{S}$ analysis is to estimate (as accurate as possible) the $\text{Insec}_{\mathcal{S}}^M(T)$ value for a given tuple of limitations T in the corresponding model M . This tuple is usually defined by the parameters of the practical systems where the target cryptosystem will be exploited. One way to upper estimate is to construct a reduction that allows to estimate $\text{Insec}_{\mathcal{S}}^M(T)$ using $\text{Insec}_{\mathcal{S}'}^{M'}(T')$ for some cryptosystem $\mathcal{S}'$ and tuple of limitations T' under the assumption that the value $\text{Insec}_{\mathcal{S}'}^{M'}(T')$ is known. The nature of T' will be discussed further. To obtain such an estimation we need to construct an adversary $\mathcal{A}'_{\mathcal{A}}$ working in M' which uses an arbitrary adversary $\mathcal{A} \in \mathcal{A}(T)$ working in M as a black box and to get lower estimation of $\text{Adv}_{\mathcal{S}'}^{M'}(\mathcal{A}'_{\mathcal{A}})$ as a function of $\text{Adv}_{\mathcal{S}}^M(\mathcal{A})$. Further we recall how the standard transition from obtained inequalities with Adv to inequalities with Insec is carried out.

Assume we construct an adversary $\mathcal{A}'$ which uses arbitrary adversary $\mathcal{A} \in \mathcal{A}(T)$ as a black box and we prove that the following inequality holds for any $\mathcal{A} \in \mathcal{A}(T)$:

$$\text{Adv}_{\mathcal{S}'}^{M'}(\mathcal{A}'_{\mathcal{A}}) \geq \text{Adv}_{\mathcal{S}}^M(\mathcal{A}) - \theta, \quad (10)$$

where $\theta = \theta(\mathcal{A}', T) \in [0, 1)$.

Since the inequality (10) holds for any $\mathcal{A} \in \mathcal{A}(T)$, then it holds for such an adversary $\mathcal{A}^*$ that $\text{Adv}_{\mathcal{S}}^M(\mathcal{A}^*) = \text{Insec}_{\mathcal{S}}^M(T)$. Therefore, the following inequality holds:

$$\text{Adv}_{\mathcal{S}'}^{M'}(\mathcal{A}'_{\mathcal{A}^*}) \geq \text{Insec}_{\mathcal{S}}^M(T) - \theta. \quad (11)$$

So now $\mathcal{A}'_{\mathcal{A}^*}$ is a fixed adversary working in M' and we can estimate its parameters such as computational resources and the total number of oracle queries. The result of such an estimation is the tuple of limitations T' such that $\mathcal{A}'_{\mathcal{A}^*}$ belongs to a set $\mathcal{A}'(T')$. So

$$\text{Insec}_{\mathcal{S}'}^{M'}(T') \geq \text{Adv}_{\mathcal{S}'}^{M'}(\mathcal{A}'_{\mathcal{A}^*}) \geq \text{Insec}_{\mathcal{S}}^M(T) - \theta,$$

and

$$\text{Insec}_{\mathcal{S}}^M(T) \leq \text{Insec}_{\mathcal{S}'}^{M'}(T') + \theta. \quad (12)$$

Remark 8.1. The goal of any reduction is to construct such an adversary $\mathcal{A}'$ for which the obtained final estimate (12) will be as accurate as possible. Ideally, the “tight” estimation should be obtained where the upper estimate is presented together with a particular adversary $\mathcal{A} \in \mathcal{A}(T)$ which advantage is equal to this estimate.

Described transition is standard. However, there is a subtlety which is traditionally paid a little attention to. This subtlety concerns $\mathcal{A}'$'s computational resources estimate which can be evaluated only if computational model (e. g. Turing machines, circuits) and encoding method are fixed. The significance of absolute accuracy in such estimates is questionable. This is because the results of this kind are closer to practical than to theoretical area — their main purpose is to define the “secure” parameters of the concrete cryptographic applications and protocols. In the vast majority of practical cases the complexity of the reduction, i. e. the resources of $\mathcal{A}'$, may be considered as negligible compared to the computational resources that are available to the adversary $\mathcal{A}$ in the model M . Exactly the structure of $\mathcal{A}'$ leads to the difference between the computational resources in T' and T . Note that other parameters in T' may often be evaluated accurately.

Assumption 8.1. *In the current paper we assume that the computational resources of any $\mathcal{A}'_{\mathcal{A}}$ is not greater than $2t$, where t is the limitation of computational resources of $\mathcal{A}$.*

8.3.2. The “PRF-as-a-MAC” principle

The following statement demonstrates the known principle “PRF-as-a-MAC paradigm” (see [5]).

Theorem 8.1 ([5]). *Let*

$$F = \{F_K : D \rightarrow R \mid K \in V_k\}.$$

If $\mathcal{A}$ is an adversary in the model MAC-CPA making at most q queries to the oracle $\mathcal{O}_F^{\text{MAC-CPA}}$, then there exists an adversary $\mathcal{B}$ that makes at most $q + 1$ queries to the oracle $\mathcal{O}_F^{\text{PRF}}$ and is such that

$$\text{Adv}_F^{\text{PRF}}(\mathcal{B}) \geq \text{Adv}_F^{\text{MAC-CPA}}(\mathcal{A}) - \frac{1}{|R|}.$$

The adversary $\mathcal{B}$ constructed in the proof of Theorem 8.1 makes some additional actions for intercepting queries of the adversary $\mathcal{A}$ to its oracle $\mathcal{O}_F^{\text{MAC-CPA}}$ and for retranslating responses of the $\mathcal{B}$ ’s oracle $\mathcal{O}_F^{\text{PRF}}$ to $\mathcal{A}$. Taking into account Assumption 8.1, we obtain the approximate inequality

$$\text{Insec}_F^{\text{MAC-CPA}}(t, q) \leq \text{Insec}_F^{\text{PRF}}(2t, q + 1) + \frac{1}{|R|}. \quad (13)$$

8.3.3. Security estimation of CBCMAC

It is known that CBCMAC is not secure in the model MAC-CPA in the case of variable-length queries. In [1] the following theorem was obtained.

Theorem 8.2 ([1]). *If $\mathcal{A}$ makes at most q queries of fixed length m to $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$, then there exists $\mathcal{B}$ making at most qm queries to $\mathcal{O}^{\text{PRP-CPA}}$ and is such that*

$$\text{Adv}_E^{\text{PRP-CPA}}(\mathcal{B}) \geq \text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{A}) - \frac{q^2 m^2}{2^{n-1}}.$$

Taking into account Assumption 8.1, we obtain the approximate inequality

$$\text{Insec}_{\text{CBCMAC}}^{\text{PRF}}(t, q, m) \leq \text{Insec}_E^{\text{PRP-CPA}}(2t, qm) + \frac{q^2 m^2}{2^{n-1}}. \quad (14)$$

8.3.4. Security estimation of a block cipher in the model of PRP-CPA

Assumption 8.2. *If for a block cipher no special methods of security decrease are known, then the value $\text{Insec}_E^{\text{PRP-CPA}}(t, q)$ is estimated by the general method that can solve the corresponding task.*

For the PRP-CPA model it is a brute force key search (or random guessing).

If the number of queries q exceeds the uniqueness distance of the block cipher E with block length n and key length k , i. e. $qn \geq k$, then, assuming that one key trial spends at least a unit of computational resource, we can suppose that for such a cipher the following inequality holds:

$$\text{Insec}_E^{\text{PRP-CPA}}(t, q) \leq \frac{t}{2^k}. \quad (15)$$

Thus for the cipher Magma with parameters $n = 64$ and $k = 256$ we can assume:

$$\text{Insec}_E^{\text{PRP-CPA}}(t, q) \leq \frac{t}{2^{256}}. \quad (16)$$

8.3.5. The problem of distinguishing two distributions

For a random variable X let $X[q]$ be a random vector of length q components of which are independent random variables distributed as X .

Let $d_{\text{stat}}(X, Y)$ be a statistical distance between random variables X and Y taking values in a finite set S :

$$d_{\text{stat}}(X, Y) = \frac{1}{2} \cdot \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|,$$

where $\Pr[X = s]$ and $\Pr[Y = s]$ are the probabilities of outcome $s \in S$ for random variables X and Y , respectively.

Now suppose that F and G are random variables and the adversary $\mathcal{A}$ is trying to distinguish the case when he observes the realizations of F from the case when he observes the realizations of G . Suppose $\mathcal{A}$ has an access to the oracle $\mathcal{O}$ that implements during one session either independent realizations of the random variable F or independent realizations of the random variable G .

We will denote by $\mathcal{O}_F$ ($\mathcal{O}_G$) the case when the oracle $\mathcal{O}$ implements the random variable F (G) during the session. Let $\mathcal{A}$ during one session gets q independent realizations $a_1, \dots, a_q$ of the random variable F or the random variable G from the oracle $\mathcal{O}$.

As a result of computations the adversary outputs 0 (an assumption that the oracle implements the random variable F) or 1 (an assumption that the oracle implements the random variable G). We will denote by

$$\mathcal{A} \Rightarrow 0 \quad \text{or} \quad \mathcal{A} \Rightarrow 1$$

the cases when the adversary outputs 0 or 1, respectively. *The advantage of the adversary* in this task is the following value:

$$\text{Adv}_{F,G}(\mathcal{A}) = |\Pr[\mathcal{A}^{\mathcal{O}_F} \Rightarrow 1] - \Pr[\mathcal{A}^{\mathcal{O}_G} \Rightarrow 1]|.$$

The following inequality was proved in [2]:

$$\text{Adv}_{F,G}(\mathcal{A}) \leq d_{\text{stat}}(F[q], G[q]). \quad (17)$$

8.4. Security evaluation regarding distinguishing against a random function

Let H be a random variable uniformly distributed over $\mathbb{H}^m$, D be a random variable uniformly distributed over $\mathbb{D}^r$. Consider a problem of distinguishing between $\text{DEC}_{m,r}(H)$ and D , where $\text{DEC}_{m,r}(H)$ is one of two decimalization functions defined in Section 3. From the inequality (17) we get immediately the following result.

Lemma 8.3. *Let $\mathcal{A}$ be an adversary that solves the distinguishing task between the random variable $\text{DEC}_{m,r}(H)$ and the random variable D . Suppose $\mathcal{A}$ gets q independent realizations of $\text{DEC}_{m,r}(H)$ or q independent realizations of D . Then*

$$\text{Adv}_{\text{DEC}_{m,r}(H),D}(\mathcal{A}) \leq d_{\text{stat}}(\text{DEC}_{m,r}(H)[q], D[q]).$$

Lemma 8.4. *Let $\mathcal{A}$ be an adversary that solves the distinguishing task between the random variable $\text{DEC}_{16,3}(H)$ and the random variable D . Suppose $\mathcal{A}$ gets q independent realizations of $\text{DEC}_{16,3}(H)$ or q independent realizations of D . Then*

$$\text{Adv}_{\text{DEC}_{16,3}^2(H),D}(\mathcal{A}) \leq \frac{1}{2} - \frac{1}{2} \cdot e^{-6q \cdot 10^{-5}},$$

$$\text{Adv}_{\text{DEC}_{16,3}^M(H),D}(\mathcal{A}) \leq \frac{1}{2} - \frac{1}{2} \cdot e^{-4q \cdot 10^{-17}}.$$

Proof. By Lemma 8.3, it is sufficient to estimate the values

$$d_{stat}(\text{DEC}_{16,3}^2(H)[q], D[q]) \quad \text{and} \quad d_{stat}(\text{DEC}_{16,3}^M(H)[q], D[q]).$$

Let us estimate the value

$$\frac{1}{2} \sum_{d \in \mathbb{D}^{3q}} |\Pr[\text{DEC}_{16,3}^2(H)[q] = d] - \Pr[D[q] = d]|.$$

From Table 1 we get

$$\begin{aligned} & \frac{1}{2} \sum_{d \in \mathbb{D}^{3q}} |\Pr[\text{DEC}_{16,3}^2(H)[q] = d] - \Pr[D[q] = d]| \leq \\ & \leq \frac{1}{2} \cdot 10^{3q} \cdot (10^{-3q} - (10^{-3} - 5.52 \cdot 10^{-8})^q) = \\ & = \frac{1}{2} \cdot (1 - (1 - 5.52 \cdot 10^{-5})^q) \leq \frac{1}{2} \cdot (1 - e^{-q \cdot 6 \cdot 10^{-5}}). \end{aligned}$$

The last inequality is due to the following relations:

$$(1 - \varepsilon)^q = e^{q \cdot \ln(1 - \varepsilon)} \geq e^{-q \cdot \varepsilon(1 + \varepsilon)}, \quad 0 < \varepsilon < \frac{1}{2}.$$

Similarly, from Table 2 we get the estimate in the case of $\text{DEC}_{16,3}^M$:

$$\begin{aligned} & \frac{1}{2} \sum_{d \in \mathbb{D}^{3q}} |\Pr[\text{DEC}_{16,3}^M(H)[q] = d] - \Pr[D[q] = d]| \leq \\ & \leq \frac{1}{2} \cdot (1 - e^{-q \cdot 4 \cdot 10^{-17}}). \end{aligned} \quad \square$$

Likewise, next Lemma 8.5 is proved in the case of $\text{DEC}_{16,4}$.

Lemma 8.5. *Let $\mathcal{A}$ be an adversary that solves the distinguishing task between the random variable $\text{DEC}_{16,4}(H)$ and the random variable D . Suppose $\mathcal{A}$ gets q independent realizations of $\text{DEC}_{16,4}(H)$ or q independent realizations of D . Then*

$$\begin{aligned} \text{Adv}_{\text{DEC}_{16,4}^2(H), D}(\mathcal{A}) & \leq \frac{1}{2} - \frac{1}{2} \cdot e^{-5q \cdot 10^{-4}}, \\ \text{Adv}_{\text{DEC}_{16,4}^M(H), D}(\mathcal{A}) & \leq \frac{1}{2} - \frac{1}{2} \cdot e^{-5q \cdot 10^{-16}}. \end{aligned}$$

Theorem 8.6. *Let $\mathcal{A}$ be an adversary in the model PRF for F^C making at most q queries to the oracle $\mathcal{O}_{FC}^{\text{PRF}}$. Then there exists another adversary $\mathcal{B}$ in the model PRF for CBCMAC making at most q queries to the oracle $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$ is such that:*

$$\text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{B}) \geq \text{Adv}_{FC}^{\text{PRF}}(\mathcal{A}) - \frac{1}{2} + \frac{1}{2} \cdot e^{-6q \cdot 10^{-5}} \quad \text{for } \text{DEC}_{16,3}^2,$$

$$\text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{B}) \geq \text{Adv}_{FC}^{\text{PRF}}(\mathcal{A}) - \frac{1}{2} + \frac{1}{2} \cdot e^{-4q \cdot 10^{-17}} \quad \text{for } \text{DEC}_{16,3}^M.$$

The length of each query equals two blocks of the cipher.

Proof. Let us construct the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ uses the adversary $\mathcal{A}$ as a black box (or a subprogram) and intercepts all queries of $\mathcal{A}$ to its oracle. Let M be a query of $\mathcal{A}$ to its oracle. Note that the length of M equals 128 bits or two blocks of the cipher. Then the adversary $\mathcal{B}$ treats this query in the following way: makes a query M to its own oracle $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$; gets the oracle response T , and returns the value $\text{DEC}_{16,3}(T)$ to $\mathcal{A}$. Obtaining $\text{DEC}_{16,3}(T)$ from $\mathcal{B}$, the adversary $\mathcal{A}$ does its own work and returns a bit. So now the adversary $\mathcal{B}$ outputs this bit.

Denote by b the bit that defines the behavior of the oracle $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$, by b' the bit that defines the behavior of the oracle $\mathcal{O}_{FC}^{\text{PRF}}$. Note that $\mathcal{B}$ simulates the oracle $\mathcal{O}_{FC}^{\text{PRF}}$ for $\mathcal{A}$.

Let us estimate the advantage of $\mathcal{B}$:

$$\begin{aligned} \text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{B}) &= \Pr[\mathcal{B} \Rightarrow 1b = 1] - \Pr[\mathcal{B} \Rightarrow 1b = 0] = \\ &= \Pr[\mathcal{A} \Rightarrow 1b' = 1] - \Pr[\mathcal{A} \Rightarrow 1b = 0] = \\ &= (\Pr[\mathcal{A} \Rightarrow 1b' = 1] - \Pr[\mathcal{A} \Rightarrow 1b' = 0]) - \\ &\quad - (\Pr[\mathcal{A} \Rightarrow 1b = 0] - \Pr[\mathcal{A} \Rightarrow 1b' = 0]) = \\ &= \text{Adv}_{FC}^{\text{PRF}}(\mathcal{A}) - (\Pr[\mathcal{A} \Rightarrow 1b = 0] - \Pr[\mathcal{A} \Rightarrow 1b' = 0]). \end{aligned}$$

Let us consider the difference $\Pr[\mathcal{A} \Rightarrow 1|b = 0] - \Pr[\mathcal{A} \Rightarrow 1|b' = 0]$. Note that if $b = 0$, then $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$ behaves as the random variable D uniformly distributed over $\mathbb{D}^3$. If $b' = 0$, then $\mathcal{O}_{FC}^{\text{PRF}}$ behaves as a random variable $\text{DEC}_{16,3}(H)$.

Therefore the following equality holds

$$|\Pr[\mathcal{A} \Rightarrow 1|b = 0] - \Pr[\mathcal{A} \Rightarrow 1|b' = 0]| = \text{Adv}_{\text{DEC}_{16,3}(H), D}(\mathcal{A}).$$

Finally, combining this equality and Lemma 8.4, we get the desired result. $\square$

In the same way, an analogous result for the function F^P may be proved.

Theorem 8.7. *Let $\mathcal{A}$ be an adversary in the model PRF for F^P making at most q queries to the oracle $\mathcal{O}_{F^P}^{\text{PRF}}$. Then there exists another adversary $\mathcal{B}$ in the model PRF for CBCMAC making at most q queries to the oracle $\mathcal{O}_{\text{CBCMAC}}^{\text{PRF}}$ is such that*

$$\text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{B}) \geq \text{Adv}_{F^P}^{\text{PRF}}(\mathcal{A}) - \frac{1}{2} + \frac{1}{2} \cdot e^{-5q \cdot 10^{-4}} \quad \text{for } \text{DEC}_{16,4}^2,$$

$$\text{Adv}_{\text{CBCMAC}}^{\text{PRF}}(\mathcal{B}) \geq \text{Adv}_{F^P}^{\text{PRF}}(\mathcal{A}) - \frac{1}{2} + \frac{1}{2} \cdot e^{-5q \cdot 10^{-16}} \quad \text{for } \text{DEC}_{16,4}^M.$$

The length of each query equals one block of the cipher.

Note that if $\mathcal{B}$ uses $\mathcal{A}$, then additional computational resources are needed for independent calculation of the functions $\text{DEC}_{m,r}$. the number of these additional resources may be estimated by q . Using bound (14) for $\text{Insec}_{\text{CBCMAC}}^{\text{PRF}}(2t, q, 2)$ and bound (15) for security of an exploited cipher in the model PRP-CPA, taking into account Assumption 8.1, we obtain the following inequalities for the function F^C :

$$\text{Insec}_{F^C}^{\text{PRF}}(t, q) \leq \frac{2t}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{1}{2} - \frac{1}{2} \cdot e^{-6q \cdot 10^{-5}} \quad \text{for } \text{DEC}_{16,3}^2, \quad (18)$$

$$\text{Insec}_{F^C}^{\text{PRF}}(t, q) \leq \frac{2t}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{1}{2} - \frac{1}{2} \cdot e^{-4q \cdot 10^{-17}} \quad \text{for } \text{DEC}_{16,3}^M. \quad (19)$$

For F^P we have the following inequalities:

$$\text{Insec}_{F^P}^{\text{PRF}}(t, q) \leq \frac{2t}{2^k} + \frac{q^2}{2^{n-1}} + \frac{1}{2} - \frac{1}{2} \cdot e^{-5q \cdot 10^{-4}} \quad \text{for } \text{DEC}_{16,4}^2, \quad (20)$$

$$\text{Insec}_{F^P}^{\text{PRF}}(t, q) \leq \frac{2t}{2^k} + \frac{q^2}{2^{n-1}} + \frac{1}{2} - \frac{1}{2} \cdot e^{-5q \cdot 10^{-16}} \quad \text{for } \text{DEC}_{16,4}^M. \quad (21)$$

8.5. Reduction of the PIN recovery problem to the PRF distinguishing problem

Theorem 8.8. *Let $\mathcal{A}$ be an adversary in the model PR for F^P making at most q queries to the oracle $\mathcal{O}_{F^P}^{\text{PR}}$. Then there exists another adversary $\mathcal{B}$ in the model PRF for F^P making at most $q + 2$ queries to the oracle $\mathcal{O}_{F^P}^{\text{PRF}}$ is such that*

$$\text{Adv}_{F^P}^{\text{PRF}}(\mathcal{B}) = \text{Adv}_{F^P}^{\text{PR}}(\mathcal{A}) - \frac{2}{10^4} + \frac{1}{10^8}.$$

Proof. Let us construct the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ uses the adversary $\mathcal{A}$ as an interactive black box and intercepts all queries of $\mathcal{A}$ to its oracle. Suppose $(PAN, PVKI)$ is a query of $\mathcal{A}$ to its oracle such that flag of this query equals FULL. Then the adversary $\mathcal{B}$ treats this query in the following way: choses $PIN \xleftarrow{\mathcal{U}} \mathbb{D}^4$, makes a query $M = (PAN, PVKI, PIN)$ to its own oracle $\mathcal{O}_{FP}^{\text{PRF}}$, gets the oracle response PVV , and returns the pair (PIN, PVV) to $\mathcal{A}$. Now suppose $(PAN', PVKI')$ is a unique special query of $\mathcal{A}$ to its oracle such that flag of this query equals REDUCED. Then the adversary $\mathcal{B}$ treats this query in the following way: choses $PIN' \xleftarrow{\mathcal{U}} \mathbb{D}^4$, makes a query $M = (PAN', PVKI', PIN')$ to its own oracle $\mathcal{O}_{FP}^{\text{PRF}}$, gets the oracle response PVV' , memorizes it, and returns only PVV' to $\mathcal{A}$. Obtaining PVV' from $\mathcal{B}$, the adversary $\mathcal{A}$ does its own work and returns a value PIN'' . So now the adversary $\mathcal{B}$ makes a query $M' = (PAN', PVKI', PIN'')$ to its oracle $\mathcal{O}_{FP}^{\text{PRF}}$, gets the response PVV'' , and outputs bit 1 if $PVV' = PVV''$, 0 otherwise.

Denote by b the bit that defines the behavior of the oracle $\mathcal{O}_{FP}^{\text{PRF}}$. Let us estimate the advantage of $\mathcal{B}$:

$$\begin{aligned} \text{Adv}_{FP}^{\text{PRF}}(\mathcal{B}) &= \Pr[\mathcal{B} \Rightarrow 1 \mid b = 1] - \Pr[\mathcal{B} \Rightarrow 1 \mid b = 0] = \\ &= \Pr[PVV' = PVV'' \mid b = 1] - \Pr[PVV' = PVV'' \mid b = 0] = \\ &= \text{Adv}_{FP}^{\text{PR}}(\mathcal{A}) - \Pr[PVV' = PVV'' \mid b = 0]. \end{aligned} \quad (22)$$

Let us consider the probability $\Pr[PVV' = PVV'' \mid b = 0]$. Note that if $b = 0$ then $\mathcal{O}_{FP}^{\text{PRF}}$ behaves as a random function $F \xleftarrow{\mathcal{U}} \text{Func}(V_{64}, \mathbb{D}^4)$.

For convenience, we introduce some notation. Let ω be the random variable representing the $\mathcal{A}$'s source of randomness. By $\mathcal{PIN}$ denote the random variable equal to the tuple $(PIN_1, \dots, PIN_q)$, by $\mathcal{PVV}$ denote the random variable equal to the tuple $(PVV_1, \dots, PVV_q, PVV')$. By $\tilde{\xi}$ we denote a specific value of a random variable ξ . Using this notation, we may represent the probability $\Pr[PVV' = PVV'' \mid b = 0]$ in the following form:

$$\begin{aligned} \Pr[PVV' = PVV'' \mid b = 0] &= \\ &= \sum_{\tilde{\omega}} \sum_{\widetilde{\mathcal{PIN}}} \sum_{\widetilde{\mathcal{PVV}}} \sum_{\widetilde{\mathcal{PIN}'}} \Pr \left[\omega = \tilde{\omega}, \left\{ \begin{array}{l} \mathcal{PIN} = \widetilde{\mathcal{PIN}} \\ \mathcal{PIN}' = \widetilde{\mathcal{PIN}'} \end{array} \right\}, \left\{ \begin{array}{l} \mathcal{PVV} = \widetilde{\mathcal{PVV}} \\ \mathcal{PVV}' = \widetilde{\mathcal{PVV}'} \end{array} \right\} \right] = \\ &= \sum_{\tilde{\omega}} \sum_{\widetilde{\mathcal{PIN}}} \sum_{\widetilde{\mathcal{PVV}}} \sum_{\widetilde{\mathcal{PIN}'}} \Pr \left[\omega = \tilde{\omega}, \left\{ \begin{array}{l} \mathcal{PIN} = \widetilde{\mathcal{PIN}} \\ \mathcal{PIN}' = \widetilde{\mathcal{PIN}'} \end{array} \right\}, \left\{ \begin{array}{l} F(PAN_i, PVKI_i, PIN_i) = \widetilde{PVV}_i \\ F(PAN', PVKI', PIN') = \widetilde{PVV}' \\ F(PAN', PVKI', PIN'') = \widetilde{PVV}' \\ (i=1, \dots, q) \end{array} \right\} \right]. \end{aligned}$$

Due to the definition of $\mathcal{A}$, fixing the random variables ω , $\widetilde{\mathcal{PIN}}$, and $\widetilde{\mathcal{PVV}}$ by specified values we find that the random variables $(\widetilde{PAN_i}, \widetilde{PVKI_i})$, $i = 1, \dots, q$, $(\widetilde{PAN'}, \widetilde{PVKI'})$, and $\widetilde{PIN''}$ take some fixed values $(\widetilde{PAN_i}, \widetilde{PVKI_i})$, $i = 1, \dots, q$, $(\widetilde{PAN'}, \widetilde{PVKI'})$, and $\widetilde{PIN''}$, correspondingly. Thus, we may transform the previous formula to the following one:

$$\begin{aligned} & \Pr[\widetilde{PVV'} = \widetilde{PVV''} \mid b = 0] = \\ & = \sum_{\tilde{\omega}} \sum_{\widetilde{\mathcal{PIN}}} \sum_{\widetilde{\mathcal{PVV}}} \sum_{\widetilde{PIN'}} \Pr \left[\omega = \tilde{\omega}, \left\{ \begin{array}{l} \widetilde{\mathcal{PIN}} = \widetilde{\mathcal{PIN}} \\ \widetilde{PIN'} = \widetilde{PIN'} \end{array} \right\}, \left\{ \begin{array}{l} F(\widetilde{PAN_i}, \widetilde{PVKI_i}, \widetilde{PIN_i}) = \widetilde{PVV_i} \\ F(\widetilde{PAN'}, \widetilde{PVKI'}, \widetilde{PIN'}) = \widetilde{PVV'} \\ F(\widetilde{PAN'}, \widetilde{PVKI'}, \widetilde{PIN''}) = \widetilde{PVV'} \\ (i=1, \dots, q) \end{array} \right\} \right]. \end{aligned}$$

Note that in the case $\widetilde{PIN'} = \widetilde{PIN''}$ there are $q + 1$ equations restricting F , otherwise there are $q + 2$ equations. Since the random variables ω , $\widetilde{\mathcal{PIN}}$, $\widetilde{PIN'}$, and F are independent, we have

$$\begin{aligned} & \Pr[\widetilde{PVV'} = \widetilde{PVV''} \mid b = 0] = \\ & = \sum_{\tilde{\omega}} \sum_{\widetilde{\mathcal{PIN}}} \sum_{\widetilde{\mathcal{PVV}}} \left(\left(\frac{1}{10^4} \right)^{q+1} \left(\frac{1}{10^4} \right)^{q+1} + (10^4 - 1) \left(\frac{1}{10^4} \right)^{q+2} \left(\frac{1}{10^4} \right)^{q+1} \right) \cdot \Pr[\omega = \tilde{\omega}] = \\ & = (10^{-4})^{2q+2} \cdot (2 - 10^{-4}) \cdot \sum_{\tilde{\omega}} \sum_{\widetilde{\mathcal{PIN}}} \sum_{\widetilde{\mathcal{PVV}}} \Pr[\omega = \tilde{\omega}] = \\ & = (10^{-4})^{2q+2} \cdot (2 - 10^{-4}) \cdot (10^4)^{2q+1} = 10^{-4} \cdot (2 - 10^{-4}). \end{aligned} \quad (23)$$

Combining (22), (23), we get the desired equality:

$$\text{Adv}_{FP}^{\text{PRF}}(\mathcal{B}) = \text{Adv}_{FP}^{\text{PR}}(\mathcal{A}) - \frac{2}{10^4} + \frac{1}{10^8}.$$

Note that additional computational resources when $\mathcal{B}$ uses $\mathcal{A}$ are needed to form and pass his queries. $\square$