Capacity of EM Side Channel Created by Instruction Executions in a Processor

Baki Berkay Yilmaz and Alenka Zajic
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Milos Prvulovic
School of Computer Science
Georgia Institute of Technology
Atlanta, GA 30332, USA

***Abstract*—This paper proposes a methodology to estimate leakage capacity of electromagnetic (EM) side channels created by execution of instruction sequences (e.g. a function, a procedure, or a program) in a processor. We propose to use Markov Source model to include the dependencies that exist in instruction sequence since each program code is written systematically to serve a specific task. The channel input sources are considered as the emitted EM signals while executing an instruction. We derive a mathematical relationship between the emanated instruction power (IP) and total emanated signal power while running a microbenchmark to obtain the channel input powers. The results demonstrate that leakages could be severe enough for a dedicated attacker to obtain some prominent information.**

I. Introduction

Covert/side channels are unintentional channels, which could be a back door for severe attacks to steal some sensitive information [1]. For example, some electromagnetic (EM) signals, which carry prominent information, can be emanated while executing a legitimate program in computer systems [2]. By exploiting these vulnerabilities, dedicated attackers can monitor some of the activities done by their victims. These attacks could be related to power variation [3], [4], temperature analysis [5], [6], cache-based analysis [7], [8], etc. Fortunately, detection probability of these attacks are high because all these attacks require some degree of direct access to the monitored systems. However, attacks based on emanated EM signals only require close proximity, hence, detecting these attacks is highly unlikely [9]. Therefore, a methodology to quantify possible information leakage can help designers to improve their computer systems to be more secure. In that respect, Millen links Shannon's capacity to define an upper bound for side-channel leakage [10]. In [11], [12], this work is further extended to account for different length of instructions, and it was shown that Shannon's capacity overestimates side-channel leakage capacity. However, work in [10], [11], [12] do not take into account that individual instructions in the program are not independent, but connected by program functionality. This paper addresses these problems by proposing a methodology which quantifies the information leakage by taking into account instruction dependencies. To model such a system, we propose a Markov source model where the states are instructions, and emanated signal powers of instructions are the outputs at each state. Furthermore, to calculate the emanated instruction signal power, we derive a mathematical relationship between instruction power and the total signal power while running a program. The rest of the paper is organized as follows: Section II reviews capacity of Markov Sources over noisy channels, and introduces the model to calculate the leakage capacity, Section II-C introduces Emanated Signal Power (ESP), and derives the relation between ESP and the total emanated power while executing a program, Section III provides experimental results of various devices, and Section IV concludes the discussion.

II. Information Leakage Based on Markov Source Model Over a Noisy Channel

In this section, we describe Markov Source model used to model instructions and their dependencies in the sequence of instructions. In the model, we consider instructions as states, the emanated instruction signal as the signal emitted at each state, and the noise corrupted version of emitted signal as the channel output.

A. Brief Overview of Markov Model Capacity

We start with a brief overview of the capacity definition for Markov Source models. The capacity of a Markov model over noisy channels can be written as [13]

$$C = \max_{P_{i,j}} \sum_{i,j:(i,j)\in\mathcal{T}} \mu_i P_{i,j} \left[\log \frac{1}{P_{i,j}} + T_{ij} \right]. \quad (1)$$

where $\mathcal{T}$ is a set of valid state transitions, μ_i is the stationary probability of state i, which satisfies $\mu_i = \sum_{k\in\mathcal{S}} \mu_k P_{k,i}, \forall i \in \mathcal{S}$, $\mathcal{S}$ is the set of states, and

$$T_{ij} = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \left[\log \frac{P_t(i,j|Y_1^n)^{\frac{P_t(i,j|Y_1^n)}{\mu_i P_{i,j}}}}{P_t(i|Y_1^n)^{\frac{P_t(i|Y_1^n)}{\mu_i}}} \right]. \quad (2)$$

In this equation, Y_1^n represents the received sequence between $t = 1$ to $t = n$, $P_t(i|Y_1^n)$ is the probability that S_{t-1} is i, and $P_t(i,j|Y_1^n)$ is the probability that the states are $S_{t-1} = i$ and $S_t = j$. There is no closed form solution to the optimization problem given in (1) due to the calculation of T_{ij}. However, an expectation-maximization algorithm is presented in [13] to calculate this capacity. We will later utilize this algorithm to obtain the leakage capacities of computer systems.

B. Modeling Sequence of Program Instructions as a Markov Source

To model sequence of instructions, we propose Markov Source model where states represent the executed instructions in a processor, and each state emits an EM signal with a specific power. The proposed model allows us to account for the dependencies among instructions because the probability of executing instruction i after instruction j changes with respect to the task that the program serves. Moreover, each instruction can be executed after any instruction, which makes the model indecomposable. An illustration of the model is given in Fig. 1 when the cardinality of instruction set is three. Here, we consider three instructions, division DIV, multiplication MUL, and subtraction SUB and the probabilities $P_{i,j}$ which represent the probabilities that instruction j is executed after instruction i, e.g. $P_{DIV,SUB}$.

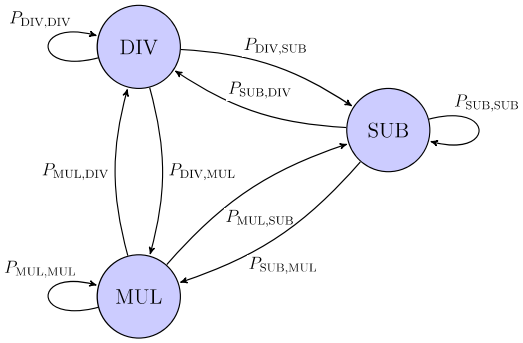

Fig. 1. Modeling the sequence of instructions using Markov model.

In the model, we assume that channel inputs are the powers of EM signals emanated while executing instructions in a processor, and channel outputs are the corrupted versions of these EM signals. Because the model is indecomposable, we can utilize the algorithm in [13] to quantify the information leakage. However, we do not have any knowledge about channel input powers in advance since emanated instruction power varies depending on the computer system. Hence, to quantify the leakage capacity, we have to obtain the channel inputs for individual instructions.

C. Measuring Emanated Instruction Power (IP)

One of the key steps in estimating leakage capacity is estimation of input powers. Here, we start by using a program that is designed to eliminate all unrelated system activities and to capture only emanated signal, as described in [14]. The program is a simple microbenchmark which contains a for-loop with a period of T_{alt} and two nested for-loops. The first for-loop executes Activity A, and the second for-loop repeats the execution of Activity B. It is shown that if these activities have different characteristics, a spectral component appears at the alternation frequency, $f_{\mathrm{alt}} = 1/T_{\mathrm{alt}}$. The power at the alternation frequency corresponds to total emanated signal power due to difference between two instructions executed in the inner loops. Please note that this is not equivalent to emanated power of single instruction so additional signal manipulation is needed to obtain input powers for the proposed Markov model. To achieve that, we use this microbenchmark when the second for-loop is filled with no-instruction (NOP).

To proceed further, we first need to define the instruction power (IP) which is given in discrete time as

$$\mathrm{IP[A]} = \frac{\mathrm{T_s}}{\mathrm{R}} \sum_{\mathrm{m}=0}^{\mathrm{N_I}-1} |a_{\mathrm{A}}[\mathrm{m}]|^2, \tag{3}$$

where N_I is the number of samples taken only when A is executed, T_s is the sampling time, and a_{A} is the signal emanated only when A is executed. We need to note that a_{A} is assumed to be the ideal signal, which does not contain any other signal components. In Appendix A, we show that $\mathrm{IP[A]}$ can be obtained by the following theorem:

Theorem. *Let $\boldsymbol{\mathcal{P}}_{\mathrm{A}}(f_{\mathrm{alt}})$ be the normalized emanated power which is defined as*

$$\boldsymbol{\mathcal{P}}_{\mathrm{A}}(f_{\mathrm{alt}}) = \mathcal{P}_{\mathrm{A}}(f_{\mathrm{alt}}) - \mathcal{P}_{\mathrm{NOP}}(f_{\mathrm{alt}}), \tag{4}$$

where $\mathcal{P}_{\mathrm{NOP}}(f_{\mathrm{alt}})$ is the measured emanated power when both for-loops of the microbenchmark execute NOP *instructions. Then, the mathematical relationship between* $\mathrm{IP[A]}$ *and $\mathcal{P}_{\mathrm{A}}(f_{\mathrm{alt}})$ while running the activity* A *in the first for-loop can be written as:*

$$\mathrm{IP[A]} \quad = \quad \left(\frac{\pi}{2}\right)^2 \frac{\boldsymbol{\mathcal{P}}_{\mathrm{A}}(f_{\mathrm{alt}}) \cdot N_L}{N_I \cdot f_{\mathrm{alt}} \cdot n_{inst}}, \tag{5}$$

where N_L is the number of samples taken when the outer loop of the microbenchmark runs only one time, n_{inst} is the number of repetitions of the nested for-loops to obtain a spectral component at f_{alt}, and $\mathcal{P}_{\mathrm{A}}(f_{\mathrm{alt}})$ is the power measured when the first nested for-loop executes instruction A.

Proof. Please see Appendix A. ■

Since we have a method to obtain the channel input power and the corresponding model, in the next section, we demonstrate the severity of these leakages on different platforms.

III. Experimental Results and Capacity Leakage Analysis

In this section, we provide the experimental results for the emanated signal power of each instruction, and leakage capacity of some platforms. We investigate the activities given in [14], and the experimental setups are shown in Fig. 2. We used a spectrum analyzer (Agilent MXA N9020A), and a magnetic loop probe (AAronia PBS-H3) for the FPGA board, and a magnetic loop antenna (AOR LA400) for the laptop. The alternation frequency, f_{alt}, is set to 80 kHz. The distance is kept as close as possible to the processor since our goal is to reveal the channel input powers.

To analyze the behavior of the leakage capacity for various noise regimes, we define the signal to noise ratio (SNR) as:

$$\mathrm{SNR} = \left[\sum_{i \in \mathcal{S}} \left(\mathrm{IP}[i]\right)^2\right] / \left(|\mathcal{S}| \times N_0/2\right) \tag{6}$$

where $|\mathcal{S}|$ is the cardinality of instruction set $\mathcal{S}$.

Fig. 2. Measurement setups used in the experiments.

We first consider the instruction powers for NIOS Processor on DE1 FPGA board. The instruction set we consider for the experiment is $\mathcal{S}_{\text{FPGA}}$ = {LDM, LDL1, DIV, ADD, SUB, MUL}. After performing the experiment introduced in Section II-C, and exploiting the equation given in (5), we obtain IPs for the set $\mathcal{S}_{\text{FPGA}}$ as **IP** = [4.6, 1.68, 0.17, 0.002, 0.03, 0.11] in aJ, respectively. As the second example, we provide the instruction powers for AMD Turion X2 laptop which has 64 KB 2 way L1 Cache and 1024 KB 16 way L2 Cache. The instruction set considered for this experiment is $\mathcal{S}_{\text{AMD}}$ = {LDL2, LDM, STM, STL2, MUL, DIV}. Applying the same procedure for FPGA, we obtain the IPs as [1.80, 0.17, 0.13, 2.26, 0.003, 0.33] in aJ, respectively.

Since we have the channel input powers, i.e., IPs, the next step is to calculate the information leakage capacity of the model given in (1). To utilize the algorithm given in [13], we assume the channel outputs are noise corrupted versions of the instruction signal where noise is additive white Gaussian with power calculated based on the SNR given in (6). The

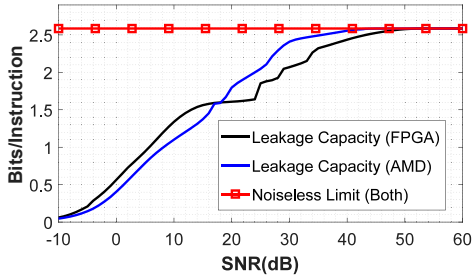

Fig. 3. Leakage Capacity of FPGA and AMD Turion X2 Laptop.

results are given in Fig. 3. We observe that both devices show similar characteristic for different SNR regimes. However, the increase in the capacity as SNR increases could be different since the distributions of the channel inputs are different. We also observe the effect of non-uniform distribution of channel input powers thanks to the zigzag pattern exists in the plots. Moreover, the figure illustrates the severity of the information leakage since the leakage is high even for low SNR regimes. Since each instruction takes a couple of clock cycles to execute on average, this result means thousands of bits of information can be leaked through software activities considering the clock frequency of modern computer systems.

IV. Conclusion

This paper has proposed a methodology to estimate leakage capacity of EM side channels created by execution of series of instructions (e.g. a function, a procedure, or a program) in a processor. We proposed to use Markov Source model to include the dependencies that exist in instruction sequence since each program code is written systematically to serve a specific task. Channel input sources are considered as emitted EM signals during instruction executions. We have also derived a mathematical relationship between the emanated instruction power (IP) and total emanated signal power while running a microbenchmark to obtain channel inputs. The results demonstrate that leakages could be severe enough for a dedicated attacker to obtain some prominent information.

Appendix A
Mathematical Derivation of IP

This section provides how IP is related to the alternation power at the considered frequency. For measurement and derivation purposes of IP, the microbenchmark given in [14], and the derivations given in [11], [12], [15] are heavily exploited. The framework given in this paper provides much simpler analysis than [15] since we assume instructions takes equal time to execute.

Let assume $i[m]$ be the ideal emanated signal sample and $w[m]$ be additive white noise with zero mean and variance σ_w^2. We can write the received signal samples as $s[m] = i[m] + w[m]$ under the assumption that the noise term captures all disruptive effects of the environment.

Let the signal for the first embedded for-loop be $s_1^{F_1}[m]$, whose length is N_L. We can decompose $s_1^{F_1}[m]$ into three different sequences. These sequences are:

* The samples regarding the execution of the instruction A:

$$a_A[m] = [0, \ \cdots, \ 0, \ a_A[0], \ \cdots, \ a_A[N_I - 1], 0, \cdots, 0] \, .$$

* The samples of other activities rather the instruction A:

$$o_{F_1}[m] = [o[0], \ o[1], \ \cdots, \ o[N_L - 2], \ o[N_L - 1]] \, .$$

  These samples belong to other activities in the microbenchmark to make the program more practical.
* Finally, the last sequence contains the noise components:

$$w_{F_1}[m] = [w[0], \ w[1], \ \cdots, \ w[N_L - 1]] \, .$$

Therefore, we have $s_1^{F_1}[m] = a_A[m] + o_{F_1}[m] + w_{F_1}[m]$. Similarly, the second for-loop signal, denoted as $s_2^{F_2}[n]$, can be written as $s_2^{F_2}[m] = o_{F_2}[m] + w_{F_2}[m]$. Here, we assume that the signal power produced by NOP is relatively zero. We also observe that the for-loops are almost identical except the part where A is inserted. Therefore, we assume $o_{F_1}[m]$ and $o_{F_2}[m]$ are identical and referred as $\mathrm{o}[m]$.

Let $p[m]$ be a square wave whose period and duty cycle are $2N_L n_{inst}$ samples and 50%, respectively. Moreover, let $\mathbf{s}[m]$ be the one period signal of the outer for-loop, $\mathbf{a}_A[m]$ and $\mathbf{o}[m]$ be generated by concatenating $a_A[m]$ and $\mathrm{o}[m]$ by $2n_{inst}$ times, respectively. Under the assumption that the noise components are i.i.d., we can write

$$\mathbf{s}[m] = p[m]\mathbf{a}_A[m] + \mathbf{o}[m] + \mathbf{w}[m].$$

The first harmonic of $\mathbf{s}[m]$ can be written as

$$\mathbf{S}[1] = \frac{\sum_{\gamma=0}^{2N_L n_{inst}-1} P[1-\gamma]\mathbf{A}_A[\gamma]}{2N_L n_{inst}} + \mathbf{O}[1] + \mathbf{W}[1]. \quad (7)$$

After taking the magnitude square of both sides, we can write the first harmonic of $\mathbf{s}[m]$ as

$$|\mathbf{S}[1]|^2 \approx \left|\frac{P[1]}{2N_L n_{inst}}\mathbf{A}_A[0]\right|^2 + |\mathbf{W}[1]|^2 \quad (8)$$

since $\mathbf{O}[k]$ and $\mathbf{A}_A[k]$ have nonzero frequency components only if $k = 2\cdot n_{inst}\cdot l$, where $\forall l \in \{0,\cdots,N_L-1\}$, $|P[1]| \gg |P[1-2n_{inst}]|$, and under the assumption that

$$\left|\frac{P[1]}{2N_L n_{inst}}\mathbf{A}_A[0]\right| \gg \Re e\left\{P[1]\mathbf{A}_A[0]\mathbf{W}^*[1]\right\}.$$

The next step is to obtain an expression for $\mathbf{A}_A[0]$. By utilizing Discrete Fourier Series analysis, we have

$$\mathbf{A}_A[0] = \sum_{\gamma=0}^{2N_L n_{inst}} \mathbf{a}_A[\gamma] \overset{(\alpha_1)}{=} 2n_{inst}\sum_{\gamma=0}^{N_L}\mathbf{a}_A[\gamma] \quad (9)$$

where (α_1) follows the fact that $\mathbf{a}_A[m]$ is periodic with N_L samples. Since only N_I entries of $\mathbf{a}_A[m]$ have nonzero values, (9) can be written as

$$\mathbf{A}_A[0] = 2N_I n_{inst}\mathbb{E}\left[\mathbf{a}_A[m]\right] \overset{(\alpha_2)}{=} 2N_I n_{inst}\mu_A \quad (10)$$

where (α_2) follows the assumption that N_I is large. Utilizing (3), we can write $\mathrm{IP}[A]$ as

$$\mathrm{IP}[A] = \frac{T_s N_I}{R}\mathbb{E}\left[|\mathbf{a}_A[m]|^2\right] = \frac{T_s N_I}{R}\left(\mu_A^2+\sigma_A^2\right) \quad (11)$$

where σ_A is the standard deviation of the samples while executing an instruction. Combining (10) with (11), we have

$$\mathrm{IP}[A] \overset{(\alpha_3)}{\approx} \frac{T_s N_I}{R}\mu_A^2 \approx \frac{T_s}{4RN_I n_{inst}^2}|\mathbf{A}_A[0]|^2 \quad (12)$$

where (α_3) follows the assumption that the variation in an instruction signal is much smaller than its mean. The final step is to show the link between $\mathrm{IP}[A]$ and the alternation power $\mathcal{P}(f_{\mathrm{alt}})$. The relation between the first harmonic of the signal and the power measure through the spectrum analyzer is given as [16], [17]

$$\mathcal{P}(f_{\mathrm{alt}}) = \frac{2}{R}\left(\frac{|\mathbf{S}[1]|}{2\cdot N_L\cdot n_{inst}}\right)^2. \quad (13)$$

Let $\mathcal{P}_A(f_{\mathrm{alt}})$ be the measured alternation power when the first and second embedded for-loops of the microbenchmark are filled with A and NOP, respectively. Moreover, let $\mathcal{P}_0(f_{\mathrm{alt}})$ be the measured power when both for-loops are filled with NOP. Here, we need to note that the number of inserted NOP into the microbenchmark needs to be chosen carefully so that the loops takes same amount of time with the instruction A inserted version of the microbenchmark.

The critical observation here is that the term related to A in (8) is zero when both for-loops are filled with NOP. Assume $\mathbf{S}_A[1]$ and $\mathbf{S}_0[1]$ denote the first harmonics of the signal when 1) the instruction A is inserted, and 2) both loops are filled with NOP, respectively. Considering this setup, we can write

$$\pi^2\left|\mathbf{S}_A[1]\right|^2 - \left|\mathbf{S}_0[1]\right|^2 \approx \left|\mathbf{A}_A[0]\right|^2 \quad (14)$$

where we utilize the approximation that $\pi|P[1]| \approx 2N_L n_{inst}$. Let $\boldsymbol{\mathcal{P}}_A(f_{\mathrm{alt}})$ be the normalized alternation power for the instruction A which is defined as $\boldsymbol{\mathcal{P}}_A(f_{\mathrm{alt}}) = \mathcal{P}_A(f_{\mathrm{alt}}) - \mathcal{P}_0(f_{\mathrm{alt}})$. Using the equations given in (12), (13), (14), and $2\cdot f_{\mathrm{alt}}\cdot n_{inst}\cdot N_L\cdot T_s = 1$, $IP[A]$ can be written as

$$\mathrm{IP}[A] = \left(\frac{\pi}{2}\right)^2\frac{\boldsymbol{\mathcal{P}}_A(f_{\mathrm{alt}})\cdot N_L}{N_I\cdot f_{\mathrm{alt}}\cdot n_{inst}} \quad (15)$$

which concludes the proof.

REFERENCES

[1] J. Millen, "20 years of covert channel modeling and analysis," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*. IEEE, 1999, pp. 113–114.

[2] A. Zajic and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," in *IEEE Transactions on Electromagnetic Compatibility, Volume: 56, Issue: 4,*, 2014, p. 885893.

[3] S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi, "Towards sound countermeasures to counteract power-analysis attacks," in *Proceedings of CRYPTO'99, Springer, Lecture Notes in computer science*, 1999, pp. 398–412.

[4] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis: leaking secrets," in *Proceedings of CRYPTO'99, Springer, Lecture notes in computer science*, 1999, pp. 388–397.

[5] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *Smart Card Research and Advanced Applications*, ser. Lecture Notes in Computer Science, A. Francillon and P. Rohatgi, Eds. Springer International Publishing, 2014, vol. 8419, pp. 219–235. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08302-5_15

[6] J. Brouchier, T. Kean, C. Marsh, and D. Naccache, "Temperature attacks," *Security Privacy, IEEE*, vol. 7, no. 2, pp. 79–82, March 2009.

[7] D. Gullasch, E. Bangerter, and S. Krenn, "Cache games–bringing access-based cache attacks on aes to practice," in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 490–505.

[8] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 494–505.

[9] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, 2002, pp. 29–45.

[10] J. K. Millen, "Covert channel capacity," in *Security and Privacy, 1987 IEEE Symposium on*, April 1987, pp. 60–60.

[11] B. B. Yilmaz, R. Callan, M. Prvulovic, and A. Zajić, "Quantifying information leakage in a processor caused by the execution of instructions," in *Military Communications Conference (MILCOM), MILCOM 2017-2017 IEEE*. IEEE, 2017, pp. 255–260.

[12] B. B. Yilmaz, R. Callan, A. Zajic, and M. Prvulovic, "Capacity of the em covert/side-channel created by the execution of instructions in a processor," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 605–620, 2018.

[13] A. Kavcic, "On the capacity of Markov sources over noisy channels," in *2009 IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, 2001, pp. 2997–3001.

[14] R. Callan, A. Zajic, and M. Prvulovic, "A Practical Methodology for Measuring the Side-Channel Signal Available to the Attacker for Instruction-Level Events," in *Proceedings of the 47th International Symposium on Microarchitecture (MICRO)*, 2014.

[15] B. B. Yilmaz, M. Prvulovic, and A. Zajić, "Electromagnetic side channel information leakage created by execution of series of instructions in a computer processor," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in c," *Cambridge University Press*, vol. 1, p. 3, 1988.

[17] G. Heinzel, A. Rüdiger, and R. Schilling, "Spectrum and spectral density estimation by the discrete fourier transform (dft), including a comprehensive list of window functions and some new at-top windows," 2002.