The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators

A. Theodore Markettos and Simon W. Moore

Computer Laboratory, University of Cambridge, UK
theo.markettos@cl.cam.ac.uk

Abstract. We have devised a frequency injection attack which is able to destroy the source of entropy in ring-oscillator-based true random number generators (TRNGs). A TRNG will lock to frequencies injected into the power supply, eliminating the source of random jitter on which it relies. We are able to reduce the keyspace of a secure microcontroller based on a TRNG from 2^{64} to 3300, and successfully attack a 2004 EMV ('Chip and PIN') payment card. We outline a realistic covert attack on the EMV payment system that requires only 13 attempts at guessing a random number that should require 2^{32}. The theory, three implementations of the attack, and methods of optimisation are described.

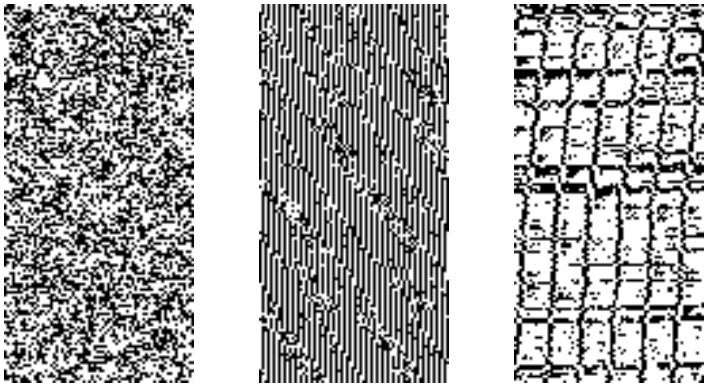
1 Introduction

Random numbers are a vital part of many cryptographic protocols. Without randomness, transactions are deterministic and may be cloned or modified. In this paper we outline an attack on the random number generators used in secure hardware. By injecting frequencies into the power supply of a device we can severely reduce the range of random numbers used in cryptography. Fig. 1 illustrates the patterns from our attack on a secure microcontroller.

Consider an example in the EMV banking protocol (initiated by Europay, MasterCard and Visa, marketed as 'Chip and PIN' in the UK) [1]. For cash withdrawal an automatic telling machine (ATM) picks an unpredictable number from four billion possibilities. Imagine if an insider can make a small covert modification to an ATM to reduce this to a small number, R.

He could then install a modified EMV terminal in a crooked shop. A customer enters and pays for goods on their card. While the modified terminal is doing the customer's EMV transaction with their secret PIN, it simulates an ATM by performing and recording $\sqrt{R}$ ATM transactions. The customer leaves, unaware that extra transactions have been recorded.

The crooked merchant then takes a fake card to the modified ATM. The ATM will challenge the card with one of R random numbers. If the shop recorded a transaction with that number, he can withdraw cash. If not, the fake card terminates the transaction (as might happen with dirty card contacts) and starts again. By the Birthday Paradox we only need roughly $\sqrt{R}$ attempts at the ATM



(a) No injection, 70×140 bits

(b) 1.822880 MHz injection, 70×140 bits

(c) 1.929629 MHz injection, 73×146 bits

Fig. 1. TRNG bitstream from secure microcontroller with frequency injection, raster scanning left-to-right then top-to-bottom. Bit-widths chosen to illustrate sequences found. Recording into SRAM of 28KB of sequential random bytes at maximum rate, later replayed through the serial port as a hexadecimal string.

before, at least, a 50% chance of success. The customer has no defence: both their card and PIN were used in the transaction, just not at the time they expected.

In this attack we have reduced the ability of a microcontroller known to be used in ATMs to produce 4 billion (2^{32}) random numbers, to just 225 ($< 2^8$) (illustrated in Fig. 1). For more than a 50% chance of a successful attack we only need to record 13 transactions in the shop and try 13 transactions at the ATM. Our attack is based on injecting signals into the power supply of a smartcard or secure microcontroller. It is made possible by adding a small number of extra components costing tens of dollars.

2 Random Number Generation

A true random number generator (TRNG) must satisfy two properties: *uniform statistics* and contain a source of *entropy*. Non-uniform statistics might enable the attacker to guess common values or sequences. Entropy comprises a source of uncertainty in a normally predictable digital system. Failure of these properties in even subtle ways leads to weaknesses in cryptographic systems ([2], [3]).

A common implementation of a TRNG is provided by comparing free-running oscillators. These are designed to be sensitive to thermal, shot or other types of random noise, and present it as timing variations. Such timing variations can be measured by a digital system, and the entropy collected. An oscillator that is easy to fabricate on a CMOS digital integrated circuit is the ring oscillator (see Fig. 2), which is used in many TRNG designs.

Practical TRNG sources are typically *whitened* by post-processing before cryptographic use, to ensure uniform statistics. Typically whitening functions include calculating the remainder of a polynomial division using a linear-feedback shift register (LFSR), or hash functions. If the entropy source is removed, TRNG outputs revert to a repeating sequence from the whitening function.

In this paper we examine the operation of the ring oscillator, and explain how the principle of *injection locking* may be used by an attacker to take control of this entropy source.

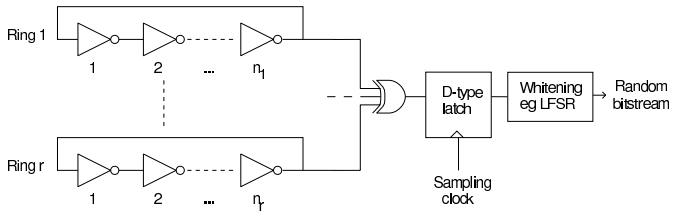


Fig. 2. Outline of the basic ring oscillator TRNG.

3 Theory

3.1 Ring Oscillator TRNG Operation

Hajimiri et al.[4] give the frequency of a single-ended[1] CMOS ring oscillator formed from N inverters with equal-length NMOS and PMOS transistors to be:

$$f_0 \equiv \frac{\omega_0}{2\pi} \approx \frac{\mu_{\text{eff}} W_{\text{eff}} C_{\text{ox}} (\frac{V_{\text{DD}}}{2} - V_{\text{T}})}{8\eta N L q_{\max}} \tag{1}$$

This relates the fundamental frequency f_0 to the gate-oxide capacitance per unit area C_{ox}, transistor length L, power supply voltage V_{DD}, gate threshold voltage V_T and proportionality constant $\eta \approx 1$. $q_{\max}$ is the amount of charge a node receives during one switching period. We consider both NMOS and PMOS transistors together, giving effective permeability μ and transistor width W:

$$W_{\text{eff}} = W_{\text{n}} + W_{\text{p}} \tag{2}$$

$$\mu_{\text{eff}} = \frac{\mu_{\text{n}} W_{\text{n}} + \mu_{\text{p}} W_{\text{p}}}{W_{\text{n}} + W_{\text{p}}} \tag{3}$$

These are all physical constants determined in the construction of the oscillator. A ring oscillator with no other effects would be completely predictable.

[1] In a *single-ended* ring the connection between each node is a single unbalanced signal, as opposed to a *differential* ring, in which each connection is a balanced pair.

Oscillators do not have a perfectly stable output. In the time domain, random noise means they sometimes transition before or after the expected switching time. In the frequency domain this implies small random fluctuations in the phase of the wave, slightly spreading its spectrum. This same effect is referred to as *jitter* in the time domain and as *phase noise* in the frequency domain. These are both cumulative over time (seen in Fig. 3).

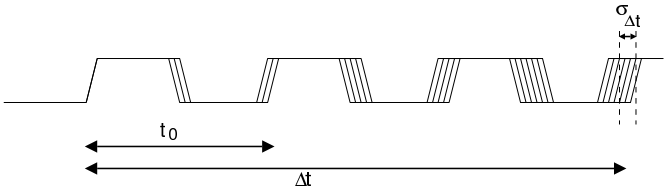


Fig. 3. Jitter in the time domain causes increasing uncertainty in the timing of transitions.

In a single-ended ring oscillator, a time Δt after the starting, Hajimiri derives that the jitter due to thermal noise will have a standard deviation:

$$\sigma_{\Delta t} \approx \sqrt{\frac{8}{3\eta}} \sqrt{\frac{kT}{P} \frac{V_{\mathrm{DD}}}{V_{\mathrm{char}}}} \tag{4}$$

where P is the power consumption and kT the Boltzmann constant multiplied by temperature. V_{char} is the characteristic voltage across a MOSFET – in the long-channel mode it is $\frac{2}{3}((V_{\mathrm{DD}}/2) - V_{\mathrm{T}})$.

This is equivalently written as a phase noise spectrum:

$$L\{\omega\} \approx \frac{8}{3\eta} \frac{kT}{P} \frac{V_{\mathrm{DD}}}{V_{\mathrm{char}}} \frac{\omega_0^2}{\omega^2} \tag{5}$$

where ω_0 is the natural angular frequency of the oscillator and variable ω is some deviation from it (ie $\omega = 0$ at ω_0).

In a TRNG based on ring oscillators jitter is converted into entropy by measuring the timing of transitions: jitter causes the exact timing to be unpredictable. There are two main ways to construct such a TRNG: relatively prime ring lengths ([5] and several patents) and identical ring lengths [6]. Both employ a topology based on that of Fig. 2. The combined signals from the rings are sampled at some uncorrelated frequency, producing a stream of bits, which is then whitened before cryptographic use.

In the identical rings context, we have two or more rings running at the same frequency. Entropy is wasted when jitter from one transition overlaps jitter from another since only one transition is measured. Sunar et al.[6] extends this to tens or hundreds of rings to increase the probability that at time t there will be a ring R that does not transition. Cumulative jitter is measured as the phase drift between each ring.

With relatively prime rings, the outputs slide past each other, minimising the likelihood of two rings transitioning together. Transition timing is based on a prime factor and the integral of past jitter. Sunar points out that fabrication of relatively prime rings to produce more concentrated entropy is expensive. In our experimental work we concentrate on relatively prime rings, since, we suggest, these are more difficult to lock to an input frequency (or frequencies). For identical rings it should be much simpler.

3.2 Frequency Injection Attacks

Bak [7] describes how a dynamical system will, at certain frequencies, resonate, and, at others, be chaotic. A resonator, such as a pendulum, with natural frequency m, will lock when driven by any frequency n forming a rational m/n. Adler [8] describes the conditions for lock as applied to a vacuum tube LC electronic oscillator. This effect is known as *injection locking*.

Our attack constitutes injecting a signal of frequency $f_i \equiv \omega_i/2\pi$ and magnitude V_i into the ring oscillators, causing them to lock to the injected frequency. Locking is a steady state: at lock the relative phase ϕ between the two oscillators is constant, so $d\phi/dt = 0$. Once lock has been achieved, the ring's natural frequency is irrelevant; jitter in the injecting signal will be received equally by all the rings, impairing any TRNG that compares jitter between oscillators.

Mesgarzadeh and Alvandpour [9] analyse this for a three-ring CMOS oscillator deliberately made asymmetric by the forcing input being an additional gate overdriving one signal. They prove Adler's work also applies to their ring oscillator. Rearranging their condition for lock in Adler's form, we have:

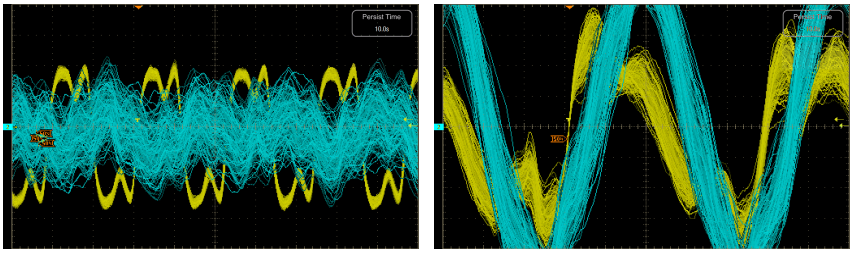
$$2Q \left| \left(\frac{\omega_i}{\omega_0} - 1 \right) \right| < \frac{V_i}{V_0} \tag{6}$$

where V_0 is the amplitude of the oscillator at its natural frequency and Q is its quality factor, a measure of the damping of the oscillator. From our experiments, Fig. 4 shows the difference between rings sliding past each other and in lock.

To achieve injection locking, we must ensure our interference can reach the ring oscillators in a secure circuit. In this paper we achieve it by coupling the injection frequency on to the power supply of the device.

The difficulty in proceeding with an analytic solution is determining Q. Adler originally derived the formulation in the context of an LC tank that has a natural sinusoidal operating frequency. Such an oscillator converts energy between two forms, voltage and current in this case. It synchronises to an external signal of suitable frequency to maximize the energy extracted from this driving source. For instance, a pendulum will adjust phase so an external periodic displacement will accelerate its own velocity.

A ring oscillator lacks a clear system-wide change between two alternating states, being just a circle where a rising and a falling edge chase each other, without any natural point defining where a new cycle starts. An idealised 3-element ring consists of three identical inverters, connected via three identical



(a) No injected signal, rings slide past each other

(b) Strong injection, traces lock phase

Fig. 4. Injection locking effect shown on oscilloscope persistent display (discrete inverter experiment from Sec. 4). View dimensions 8 V × 200 ns; 5 V power supply. Triggering on 3-element ring, with 5-element-ring trace in front. Note in particular how resonances are set up in the ring oscillators that increase the amplitude above the power rails from 5 Vp-p to 10 Vp-p.

transmission lines. All three inverters and transmission lines oscillate in exactly the same way, but 120° out of phase. A waveform applied via the power supply or an EM field is a *global stimulus* that affects all three inverters equally. It will, therefore, encourage the ring oscilator to synchronise simultaneously with three versions of the stimulus, all 120° apart in phase. Their synchronising effect is thus largely cancelled out.

A global stimulus can only be effective if the three parts are not exactly identical. In a real-world ring oscillator layout asymmetries, device variations, and loading due to the output tap all break this 120° symmetry and will allow one of the 120° alternatives to win over the other two. How quickly the ring will lock on to a global stimulus will largely depend on the size of this asymmetry.

Unlike pendula or LC tanks, ring oscillators are also non-linear. In short rings, such as $N = 3$, each gate is in a constant state of transition, so operates linearly, and the output more clearly resembles a sinusoid. But in longer rings, where $N \gg 10$, each gate spends a small fraction of the time in transition, so the ring output is more like a square wave. Adler's model fits this case less well.

3.3 Effect of Injection on Jitter

Mesgarzadeh and Alvandpour indicate their injection can operate on the ring signals as a first-order low-pass filter with a single pole located at:

$$p = 2\pi\omega_i \ln \frac{1}{1+S} \tag{7}$$

where S is the injection ratio V_i/V_0 or, in power terms, $\sqrt{P_i/P_0}$. In other words, the function in the domain of the Laplace transform is:

$$H(j\omega) = \frac{1}{1 + (2\pi\omega_i \ln \frac{1}{1+S})j\omega} \tag{8}$$

where $j = \sqrt{-1}$. It is analogous to a series R-C filter with $RC = p$.

If we can locate pole p close enough to the origin we can filter out high frequency and large cycle-to-cycle jitter. Increased injection power S reduces this filtering effect.

A successful attack is the intersection of two regions. From (6), if the injection power is too low the system will not lock. From (8), if the power is too high jitter will not be filtered out. For the TRNG system a weaker condition is required: if the jitter of the rings mirrors that of the driving oscillator, the attack is a success. The TRNG measures differences in jitter between rings, so will not measure jitter common to all rings.

We analyse the attack on relatively prime rings but demonstrate our attack in 'black box' experiments with no knowledge of device construction. Therefore we assume that we are reducing jitter by the effect of equalising jitter between rings, rather than a reduction of jitter in the whole system.

Yoo et al.[10] describes locking effects due to poor layout but generally not in an adversarial manner. They investigate changing the DC supply voltage, but not its AC components. Sunar [6] considers active glitch attacks and concludes these are only able to attack a finite number of bits due to their limited duration. The frequency injection attack is much more powerful, since it can attack all bits simultaneously for as long as desired.

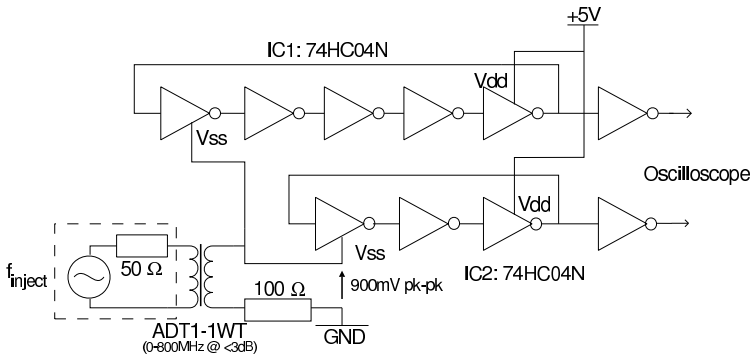
4 Discrete Logic Measurements

We set out to measure phase differences in two relatively prime rings. Given their primality, the ring outputs should drift past each other, based on a combination of cumulative jitter and the underlying ring frequency differences. For non-locked rings, we expect phase lag to be uniformly distributed. When locked, phase lag will be concentrated on one value.

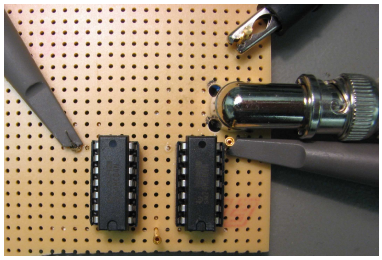
Injection locking is very difficult to simulate in a transient analysis tool such as SPICE [11]. It requires very small timesteps for each oscillation cycle, and a high Q oscillator may require thousands of cycles to lock. When close to the natural frequency the beat frequency may be a few tens of Hertz. To measure this accurately in a simulation with picosecond-scale steps requires an infeasibly long simulation time. In addition, the asymmetries of a real system will not be modelled in a simulated ideal design.

Due to I/O buffering, it is difficult to measure such behaviour of fast analogue signals inside an FPGA or ASIC. We first measured the effect in discrete logic. With limited complexity possible, we investigated the simplest ring oscillators: the outputs from three- and five- element rings, with and without frequency injection in the power supply. We used the 74HC04 inverter chip to construct the two mutually-prime rings seen in Fig. 5(a). Phase lag was measured by triggering an oscilloscope on the rising edge of the three-element ring, and measuring the time up to the rising edge of the five-element ring. Such short rings are used in real TRNGs – though they may have a greater region of linear operation than longer rings.

We set up a Tektronix AFG3252 function generator to inject a sine wave at 900 mV pk-pk into the 5 V power rails and by sweeping frequency we observed locking at 24 MHz. A Tektronix TDS7254B oscilloscope measured the phase lag between the two rings when injecting and the resulting histograms are plotted in Fig. 6. A very clear clustering around 10 ns can be seen, indicating a lock. This effect is visible in the time domain traces seen in Fig. 4, which show a marked reduction in the variability of the 5-element output. The slight clustering seen in Fig. 6(a) is, we believe, due to slightly non-uniform oscilloscope triggering.



(a) Schematic



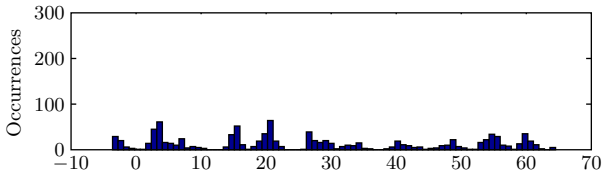
(b) Test board

Fig. 5. Measurement experiment using 74HC04 inverter ICs.

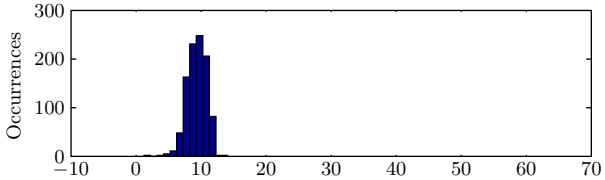
5 Secure Microcontroller

We tested an 8051-compatible secure microcontroller which has been used in ATMs and other security products. It includes features such as an anti-probing coating and tamper detection and at release claimed to be the most secure product on the market. Our example had a date code of 1995 but the device is still recommended for new payment applications by the manufacturer.

It provides a hardware TRNG based on frequency differences between two ring oscillators and timing from the user's crystal (11.059 MHz here), and pro-



(a) No injection: phase lag/ns



(b) 24MHz injection: phase lag/ns

Fig. 6. Phase delay between 74HC04 3- and 5- element rings. (a) with no injection, (b) with 24 MHz at 900 mV pk-pk injected into power supply. (25000 samples).

duces 8 bits every 160 μs. 64 bits from the TRNG may be used as the internal encryption key. No further operation details are documented.

We programmed the device to deliver the random bitstream as hexadecimal digits through the serial port and displayed it in realtime as a two dimensional black and white image. We adjusted the function generator to inject a sinusoid at 500 mV peak-peak into the chip's power supply as shown in Fig. 7.

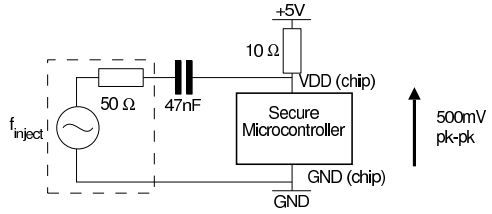


Fig. 7. Frequency injection to secure microcontroller.

By sweeping the frequency we spotted changes in the patterns produced by the TRNG. The most interesting f_{inject} was at about 1.8 MHz. Obviously periodic sequences were visible: see Fig. 1(a)–1(c). In particular the sequence length of the TRNG was controlled by the injected frequency. With another choice of f_{inject} we could also prevent the TRNG returning any values. At no time during any of these tests did the microcontroller otherwise misbehave or detect a fault condition. The device is designed to operate at 5 V with a minimum operating voltage of 4.25 V so it is running within specification.

Fig. 1(b) indicates a distinct 15-bit long texture, on top of a longer 70-bit sequence. Uncertainty is only present at the overlap between these textures. In a 420-bit sample, we estimate 18 bits of noise. In a 15-bit sequence that means 0.65 bits may be flipped. The attacker knows the base key is the sequence 010101010..., but not where the 15-bit sequence starts (15 possibilities) or the noise. In most cases noise is constrained to 3 bits at the start of the 15 bit sequence. In the full 64-bit sequence, the bit flips are $0.65 \times 4 = 2.6$. 3 bits flipped over the whole 64-bit sequence in one of 12 positions gives $\binom{12}{3} = 220$ combinations. Thus we estimate that the total keyspace is smaller than $220 \times 15 = 3300$. In a key length of 32 bits there are 1.3 bits of noise; the equivalent calculation with 2 bits gives a keyspace of less than $\binom{6}{2} \times 15 = 225 \approx 2^8$.

6 EMV Smartcard Attack

6.1 Methodology

We applied this methodology to an EMV payment card issued in 2004 by a British High Street bank. We picked the first available; with no knowledge of the internals we treated it as a 'black box'. This is a non-invasive attack where no modifications are required to the card under attack.

First we needed to deduce the operating frequency of the card's TRNG. We assumed that such a card would have power analysis protection, so we performed an electromagnetic assessment. An electric field antenna was constructed on a test card. Copper foil was attached beneath the chip as shown in Figs. 8 and 9, with foil traces between the backside foil patch and the ground pad of the chip. The card was inserted into a Chipdrive Micro 100 card reader, and standard ISO7816-4 `GET CHALLENGE` commands used to read the RNG.

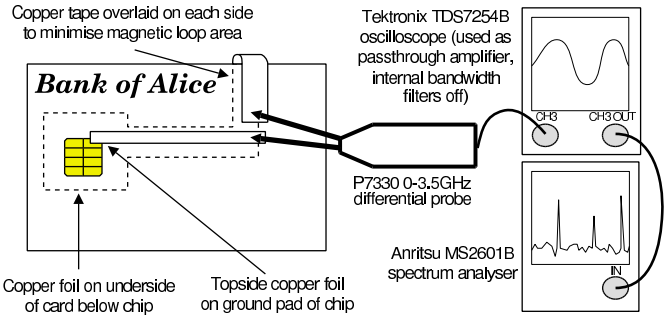


Fig. 8. Electric field characterisation of EMV smartcard.

We measured three different spectra: (a) not powered or attached to reader (background interference); (b) powered, attached to reader but held in reset and not clocked; and (c) when reading random numbers.

Since a ring oscillator is likely to remain powered even when the card is not initialised, we looked for frequencies that exist when the card is inserted into the reader and unclocked, but not present when the card is removed. We found four such frequencies in the range 0–500 MHz: we chose f_{inject} to be 24.04 MHz, the only frequency below 100 MHz. As this is a black-box system, we do not know if this is optimal; it is merely the first frequency we tried.

We modified the reader to inject a signal as shown in Fig. 10, and observed the random number statistics. Sweeping the injected frequency and graphically viewing random bits, we saw no obvious pattern changes. However statistical analysis of the random data revealed injecting f_{inject} at 1 V pk-pk across the card powered at 5 V caused the random function to skew. At all times during measurement the card continued to respond correctly to ISO7816-4 commands and would perform EMV transactions while under attack.

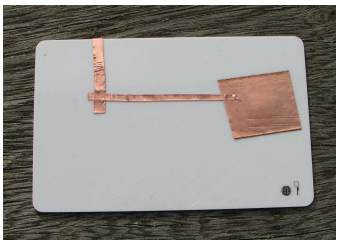


Fig. 9. Electric field antenna on underside of example smartcard.

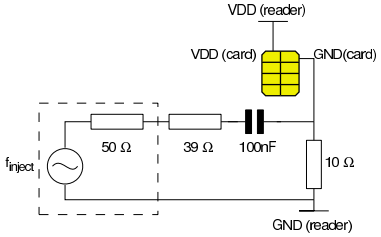


Fig. 10. Smartcard frequency injection circuit.

The statistics were analysed using all the tests in the NIST [12] and Dieharder [13][2] test suites using 1.56×10^9 bits. An outline of the results are shown in Table 1, with tabulated NIST results in the Appendix. By failing most of the tests we can see that the sequence has become non-random. The FFT test reveals periodicities of around 2000 and 15000 bits. The Rank test, where a 32×32 matrix of random bits should have a rank > 28 (true for our control data), fails with many ranks as low as 19 implying rows or columns are not independent.

Table 1. Statistical test results from injection into EMV card.

NIST	**Pass**			**Fail**
No injection	187			1
Injection	28			160
Dieharder	**Pass**	**Poor**	**Weak**	**Fail**
No injection	86	6	6	9
Injection	28	16	5	58

[2] Dieharder version 2.28.1, a superset of the DIEHARD suite

7 Recommendations and Further Work

7.1 Optimisation of the Attack

In the Introduction we outlined an attack on the EMV payment system, which works whether the smartcard uses Static or Dynamic Data Authentication protocols.

An ATM is a computer in a safe, comprising either a PC or custom circuit boards. At least one ATM uses the secure microcontroller we tested as its cryptoprocessor. ATM physical security focuses on preventing access to the money held inside; this attack needs no access to the cash compartment. Adding injection hardware involves adding a tap to one wire on the PCB – this could be done by an insider or simply by picking the mechanical locks. June 2009 reports [14] uncovered malware in ATMs installed by insiders, while in another case [15] an attacker bought up 'white-label' ATMs (normally found in shops and bars), fitted internal recording devices and resold them.

The required number of transactions is small and unlikely to raise alerts at the bank, which is afraid of false alarms. Customers complain about false positives, so there is commercial pressure to be lenient. If the cash withdrawal is performed before the card is used again by the customer, the bank has no way of knowing the transaction was recorded earlier. ATMs are typically only serviced when they go wrong. Even if our proposed frequency injector could be spotted by a technician, it may be many months before they are on site.

While we developed our attack with laboratory equipment, the cost of characterising each smartcard, card reader or ATM processor can be made very low. An electric field antenna may be fitted inside a commercial reader, so that nothing is fixed to the card surface. A commercial tunable radio receiver may be attached to the antenna to scan for frequencies of interest, while the frequency synthesiser in a similar receiver may be modified as a cheap way to generate injection frequencies. Given a quantity of identical cards (cheaply acquired on the black market, having little value once expired or cancelled by the bank) the search is easy to parallelise.

Attacking the TRNG on a card can be optimised by listening to other commands it performs. Each card provides an Answer To Reset – a response from the card software which can also be used to fingerprint its manufacturer [16]. We found cards with the same ATR emitted the same frequencies, most likely if they were built on the same operating system/hardware combination. After characterisation, the attacker can decide which frequencies to inject to a live card based on the ATR. This logic can be built into a terminal or ATM tap; interception of the card serial data line will reveal the ATR.

Due to electromagnetic interference (EMI) regulations, devices are designed to operate in the presence of interference. Neither of the commercial devices tested failed to operate at any point during these attacks. It is difficult to see how TRNGs could actively detect such attacks without compromising their EMI immunity.

7.2 Defences

We have demonstrated this attack allows the keyspace to be reduced to a size easily brute-forced. As soon as the attacker knows some plaintext, the key may be easily found. The simplest defence is to prevent a brute-force attack. Therefore the best system allows few permitted guesses, which raises the risks for the attacker. Preventing the attacker gaining large quantities of random material would also prevent device characterisation.

To prevent interference a device can filter injected frequencies. Voltage regulation or merely extra power supply smoothing may prevent locking, or shielding may be required for electromagnetic attacks. Devices could refuse to operate at their known-vulnerable frequencies. While this may render them less EMI-immune, it may be acceptable in high security applications. TRNG designs where the feedback loop is combined with logical or register elements [17] may be sufficient to break the locking effect.

Designers can work towards preventing locking by reducing asymmetries in the rings. Carefully balanced transistors may be used, as may equal tapping points on each node. Also, the differential ring oscillator is less affected by supply and substrate noise [18]. It may be feasible to use this instead of the single-ended ring commonly used. Careful design is required to ensure reducing susceptibility does not destroy the source of entropy – Hajimiri [4] indicates the differential oscillator increases the phase noise, which may be beneficial.

7.3 Further work

We have outlined the principles of this attack but there are many ways in which it could be refined.

Further analysis of the effect of power supply injection is necessary. In the literature, injection locking has mostly been analysed through direct coupling to the signal undergoing oscillation, while, here, we use a different mode of coupling, by co-ordinated biasing of the gates it passes through. It would be instructive to determine the minimum power required for this attack and, in particular, how much it can be reduced by on-chip filtering. There are some well-known defences against passive power analysis; it would be interesting to evaluate these for protecting against frequency injection attacks.

In addition, it may also be feasible to perform this attack via high-powered electromagnetic radiation, which is more easily focused and more difficult to mitigate than a power attack. This could be done using magnetic loops to induce currents into the device at the appropriate frequencies, or using the device itself to demodulate the injected frequency (such as a 3 GHz carrier amplitude-modulated by 1.8 MHz); the carrier will more readily propagate, but be filtered away by parasitic capacitance on the chip leaving the 1.8 MHz harmonic.

Systems with identical ring lengths may be particularly vulnerable due to their shared resonant frequencies. There is further scope for directing this attack if the ring geometries are known. Fig. 1 shows some texture of our TRNG; it may be interesting to use this approach to reverse engineer a TRNG's design from the bitstream.

8 Conclusion

In this paper we have outlined an attack on ring-oscillator based random number generators. We have described the effect, and measured its consequences on a security microcontroller used in the EMV system, and in an EMV card. We believe this is an important effect, which all designers of random number generators should test.

Acknowledgements Markus Kuhn suggested the experiments with the secure microcontroller and provided many components of the experimental setup plus valuable feedback on the paper. Steven Murdoch provided the Python EMV protocol library used to drive communication with the smartcard.

References

1. EMVCo, LLC: EMV 4.2 specification. (June 2008) http://www.emvco.com/.
2. Bellare, M., et al.: "Pseudo-Random" number generation within cryptographic algorithms: The DSS case. In Kaliski, Jr., B.S., ed.: CRYPTO. Volume 1294 of Lecture Notes in Computer Science., Springer (1997) 277–291
3. Bello, L.: DSA-1571-1 openssl – predictable random number generator. Debian Security Advisory (2008) http://www.debian.org/security/2008/dsa-1571.
4. Hajimiri, A., Limotyrakis, S., Lee, T.H.: Jitter and phase noise in ring oscillators. IEEE J. Solid-State Circuits **34**(6) (1999) 790–804
5. Eastlake, D., Schiller, J., Crocker, S.: Best Common Practice 106: Randomness requirements for security. Technical report, IETF (2005)
6. Sunar, B., Martin, W.J., Stinson, D.R.: A provably secure true random number generator with built-in tolerance to active attacks. IEEE Trans. Computers **56**(1) (2007) 109–119
7. Bak, P.: The Devil's staircase. Physics Today **39**(12) (1986) 38–45
8. Adler, R.: A study of locking phenomena in oscillators. Proc. IRE and Waves and Electrons **34** (1946) 351–357
9. Mesgarzadeh, B., Alvandpour, A.: A study of injection locking in ring oscillators. Proc. IEEE International Symposium on Circuits and Systems **6** (2005) 5465–5468
10. Yoo, S.K., Karakoyunlu, D., Birand, B., Sunar, B.: Improving the robustness of ring oscillator TRNGs. http://ece.wpi.edu/~sunar/preprints/rings.pdf.
11. Lai, X., Roychowdhury, J.: Analytical equations for predicting injection locking in LC and ring oscillators. In: IEEE 2005 Custom Integrated Circuits Conference. (2005) 461–464
12. Rukhin, A., et al.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical Report SP800-22, National Institute of Standards and Technology, USA (2008)
13. Brown, R.G., Eddelbuettel, D.: Dieharder: A random number test suite http://www.phy.duke.edu/~rgb/General/dieharder.php. Accessed 2009-03-03.
14. Mills, E.: Hacked ATMs let criminals steal cash, PINs. ZDNet UK (June 2009) http://news.zdnet.co.uk/security/0,1000000189,39660339,00.htm.
15. Bogdanich, W.: Stealing the code: Con men and cash machines; criminals focus on A.T.M.'s, weak link in banking system. The New York Times (August 2003) http://query.nytimes.com/gst/fullpage.html?res=9803E6DD103EF930A3575BC0A9659C8B63.

16. Rousseau, L.: pcsc_tools package: ATR table `http://ludovic.rousseau.free.fr/softwares/pcsc-tools/smartcard_list.txt`. Accessed 2009-03-03.
17. Sunar, B.: True random number generators for cryptography. In Koç, Ç.K., ed.: Cryptographic Engineering. Springer (2009) 55–74
18. Herzel, F., Razavi, B.: A study of oscillator jitter due to supply and substrate noise. IEEE Trans. Circuits and Systems II **46**(1) (1999) 56–42

Appendix
Tabulated NIST Test Results from EMV Smartcard

Table 2. NIST results from EMV smartcard.

	No injection			**Apply $f_{\mathbf{inject}}$**		
	χ^2 P-value	Passes	Overall	χ^2 P-value	Passes	Overall
Frequency	0.3215	97.44%	PASS	0.0000	21.54%	FAIL
Block Frequency	0.6262	98.97%	PASS	0.0000	0.51%	FAIL
Cumulative Sums	0.2904	97.95%	PASS	0.0000	22.05%	FAIL
Cumulative Sums	0.3902	97.95%	PASS	0.0000	21.54%	FAIL
Runs	0.3811	99.49%	PASS	0.0000	40.00%	FAIL
Longest Run	0.3548	98.97%	PASS	0.0000	73.85%	FAIL
Rank	0.5501	100.00%	PASS	0.0000	0.00%	FAIL
FFT	0.0001	100.00%	PASS	0.0000	0.51%	FAIL
Non-Overlapping Template[a]	0.4523	99.00%	PASS	0.0000	90.89%	FAIL
Overlapping Template	0.4470	98.97%	PASS	0.0000	9.23%	FAIL
Universal	0.0211	98.97%	PASS	0.1488	98.46%	PASS
Approximate Entropy	0.1879	98.97%	PASS	0.0000	1.54%	FAIL
Random Excursions[b]	0.3050	99.26%	PASS	0.2836	99.50%	PASS
Random Excursions Variant[c]	0.4922	99.39%	PASS	0.3053	99.56%	PASS
Serial	0.1168	100.00%	PASS	0.0000	0.00%	FAIL
Serial	0.5501	98.97%	PASS	0.0000	0.00%	FAIL
Linear Complexity	0.0358	99.49%	PASS	0.9554	98.46%	PASS

Dataset of 195×10^6 random bytes. NIST performed 195 runs each using fresh 10^6 bytes. Minimum pass rate 96.86% except Random Excursions (96.25% no injection, 93.03% injected)

[a] Mean over 148 tests
[b] Mean over 8 tests
[c] Mean over 18 tests