Differential Analysis Using Phase-Substitution

C.Gebotys, B.A.White, Dept of E&CE, University of Waterloo, April 2007.

Abstract. Among several countermeasures suggested for thwarting differential analysis are the random ordering of operations, insertion of random operations, and random insertion of operations. This paper presents a phase-substitution technique which in combination with subsequent time-domain differential analysis is shown to be able to thwart these three countermeasures in several experiments. Unlike previous techniques for aligning traces, this approach makes use of the phase information. The proposed technique involves: fast fourier transform, phase-substitution, inverse fast fourier transform and time-based differential analysis. Results are demonstrated using electromagnetic traces acquired from a PDA device (representing a complex embedded system including cache misses, operating system events, etc). This research is important for future wireless embedded systems which will increasingly demand higher levels of security.

1 Introduction and Previous Research

Security is crucial for today's portable devices especially as more mobile code applications, such as Java-based security, migrate to wireless devices. However recent research has found that side channel analysis attacks may be powerful enough to extract secret keys from certain devices such as contact smart cards. In particular researchers showed that the power contact in the smart cards could be used in order to measure the power side channel. However often many devices are embedded and hence access to their power in order to utilize the power side channel may not be possible. Hence the electromagnetic side channel is of great interest for many embedded systems. For example researchers have already demonstrated that an EM attack is also viable[3,5] on an 8-bit processor running at 4MHz in a smartcard. Hence an attack may be successful in obtaining the secret keys stored in confidential memory in a wireless device. This attack may be possible through loss or theft of the device, or alternatively through temporary access to the device by monitoring the EM waves emanating from the device while performing cryptographic computations. In the latter case the attack may be able to extract the encryption keys, making future wireless communications insecure. Outside of smartcard research (which in the past has typically been limited to cheaper 8-bit or 16 bit processors) [15,6, 1], few researchers have examined secure implementations of cryptographic software (such as Rijndael[2] which has become a popular new standard) under the threat of EM attacks on 32-bit processors. The cryptographic algorithms which are essential for these applications are typically run by embedded processors in these wireless devices. Unfortunately cryptographic algorithms are already known to consume a significant amount of energy [8]. Even worse, cryptographic algorithms which are resistant to attacks are known to have latency overheads up to 1.96 times[12]. Although these attack resistant algorithms have been developed for smartcard applications (where energy dissipation is not critical), there is an important need to study EM attacks and energy optimized countermeasures on wireless portable devices, such as PDAs, cellphones, etc.

Typically EM analysis involves an attacker who captures the EM signals emanating from a device while it is executing a cryptographic process. The set of EM signals acquired over time is often referred to as a trace. Typically the attacker collects many EM traces (e.g. an EM trace for each plaintext input to the cipher). In symmetric encryption the plaintext and key are exclusive or'd together and then indexed into a table, known as the S-box table, as in the table method of the Rijndael advanced encryption standard [2]. The attacker may have control over the plaintext and by guessing each 8-bit secret key value (of the 128-bit key), can partition EM or power traces according to a bit in the guessed data at the output of the S-box table. The trace would be placed into group 0 if the expected value of the bit is zero or alternatively group 1 in the other case. By taking the mean of traces in each group and performing the difference of the two means, a differential trace can be created. By recording the height of the differential for each key guess, the attacker can determine the correct key (since it will have the highest differential value).

Although EM attacks on smart cards have been investigated [5,4], EM or power attacks on other embedded systems have not been widely researched, apart from far field EM emanations from a Palm-Pilot and SSL accelerator[10,11] and the recent attack on a RFID passive tag [16]. Previous research studied the correlation of EM variation with data values being manipulated (known as differential EM analysis, DEMA, or DPA for differential power analysis [1]) and instruction sequencing (known as simple EM analysis, SEMA, or simple power analysis, SPA). In the former case, DEMA, the DES encryption[5] was analyzed. Differential EM attacks on embedded low power processors have not been reported in the literature. In the later case of the RFID attack [16], the SPA of the receive antennae was used to extract the kill password, while the attacker sent data to the passive RFID tag. Higher order (n^{th} order) differential attacks[7] are an extension of the 1st order differential analysis which involve using joint statistics on multiple (n) points within power traces. Countermeasures, which are thwarted by higher order differential analyses, have been shown to provide more security, since a larger number of EM or power traces [7,5] are required in the analysis. For example, research with real EM measurements using an 8-bit processor running at 4MHz in a smart card demonstrated 2nd order DEMA [5] on a 2-way exclusive-or-based secret sharing scheme using 500 EM traces. In most cases, excellent EM or power trace alignment of the attack point is required since most previous differential analyses are performed in the time domain. The exception to this is [9] where a fast fourier transform is calculated, however transformation back into the time domain occurs before differential analysis is performed. More recently a phase-only correlation (POC) technique [14] describes a high-resolution waveform matching technique for aligning traces in a side-channel attack. They use a cross phase spectrum formed with a reference trace and another trace. The inverse fast fourier transform of this spectrum forms a distinct sharp peak at the location of translational displacement between the reference trace and other trace. Hence this approach can be used to estimate the displacement so that traces can be shifted in the time domain by this value. Their focus is on estimating and correcting small displacement errors (less than 1-2 sampling periods) between signal waveforms, with higher resolution than the sampling resolution. Higher resolution than the sampling resolution is obtained by fitting an analytical model of the correlation peak to the actual numerical data. The waveform is then shifted by the estimated (fractional) displacement using phase rotation of the waveform in the frequency domain and an inverse FFT to obtain the shifted waveform. This paper will assume DEMA or DPA denotes time domain differential analysis, unless otherwise specified.

Previously researched countermeasures have been suggested such as introducing random delays in the code, random sequencing of instructions (desynchronization), secret splitting[17], duplication method[18], multiplicative masking[19] and random masking[12]. Secret splitting involves splitting the secret data into smaller pieces and combining them with random data [17]. To attack the splitting method, a k^{th} order differential attack is required[17]. The duplication method[18] was used to support secure computations with multiple split variables for input to the S-box. These researchers also used table duplication such that one table contained a randomly-chosen secret transformation on x, A[x], and the alternate table contained A[x]+S[x], where + represents exclusive or operation. Multiplicative masking was also defeated by a DPA attack[19]. In the masking countermeasure, each secret piece of data is exclusive-or'd with a random data value (called a mask). To thwart a DPA attack the random data value must be changed periodically. However this involves remasking the tables (or exclusive-or the complete table data with a mask) within the algorithm. Some researchers have investigated storing a limited number of masked tables [18] (called the 'fixed-value' masking). Higher order analysis may be used to thwart these countermeasures, however perfect alignment and correlation to algorithm operations are required. In general results using real EM or power measurements were not performed.

Unlike previous research, this paper presents results of EM analysis on a real embedded system, a wireless Java-based PDA, using a new phase-substitution technique. The phase substitution is used to attempt to synchronize and restore the order of operations of many traces so that a differential analysis is successful. Furthermore it is also used to attempt to diffuse inserted random operations. Rijndael, a standardized cryptographic algorithm, is used to demonstrate the new attack. The next section will describe the proposed phase substitution technique and the following section will present the experimental results.

2 The Phase-substitution Technique

In general, a time sampled signal, *x[n],* can be represented in the frequency domain by *X(f)* which is composed of both the magnitude, *MagX(f)* and the phase, *PhaseX(f)*. It is well known that a shift of the signal in the time domain corresponds to a change only in the phase in the frequency domain. For example consider a trace which is a time-sampled signal *x[n]* which has *m* samples *n = (1,…m)* with magnitude and phase in the frequency domain equal to *MagX(f)* and *PhaseX(f)* respectively. A different trace, specifically $x[n+s]$, represents $x[n]$ except it is misaligned or shifted by s samples in the time domain, where $x[n+s]$ has a magnitude and phase of $MagX(f)$ and $PhaseX(f) + 2\pi s f$ (where f is a fraction of the sampling rate ranging from 0 to 0.5) respectively. Note that the adjustment to the phase component is a linear one.

Now consider a set of misaligned traces of an ideal noise-less signal, $x[n]$. These traces are represented as different time shifts of the signal $x[n]$, specifically $x[n+r1]$, $x[n+r2],\ldots x[n+rn]$ in the time domain where $r1,r2,\ldots rn$ represent small random time shifts. In the frequency domain the signals are represented by magnitudes and frequencies with the following relationship : $MagX1(f)=MagX2(f)=\ldots=MagXn(f)$, $PhaseX1(f) \neq PhaseX2(f) \neq \ldots \neq PhaseXn(f)$, (where $MagX1(f)$ is the magnitude of signal $x[n+r1]$ in the frequency domain, etc). In theory realignment of the traces may be possible if the set of phases for each trace were identical and represented a linear phase shift with respect to the current set of phases. Hence one could map each trace into the frequency domain and choose the phase of one trace to define a reference phase, such as $PhaseX1(f)= \theta 1$. The reference phase is then copied into the phases of all the remaining traces, specifically $PhaseX2(f) =\ldots= PhaseXn(f\,) = PhaseX1(f)$. This is referred to as phase substitution. Since only the relative phase difference between signals with the same frequency is of interest, one can use a phasor (or vector) representation as shown in figure 1a) to represent each misaligned trace (where the magnitude is represented by the vector length and the phase is the angle relative to the x-axis). The n phasors in figure 1a) represent the n misaligned traces $x[n+r1]$, $x[n+r2],\ldots x[n+rn]$. To the right of figure 1a), all phasors have the same phase $\theta 1$ after phase substitution. Hence for trace alignment the values $r1,\ldots rn$ do not need to be computed, unlike previous research such as [14]. This idea will next be discussed as a possible approach to resist the countermeasures which involve random ordering of operations and random insertion of operations.

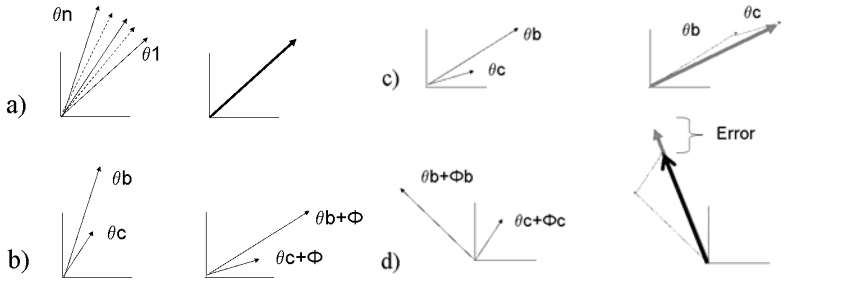


Figure 1. Phasor representation of misaligned traces a), linear phase shift b), and c), d) error for noncyclic operation ordering.

Consider traces capturing the side channel signals from a number of operations. Assume three operations a, b, c, have side channel signals represented by, $xa[n]$, $xb[n]$, $xc[n]$ each defined with m samples. Assume a trace, $x1[n]$, is captured from the side channel signals obtained from a device which executes operations in the order of a, b, c. The resulting trace could be represented as $x1[n]= xa[n+2m]+xb[n+m]+xc[n]$ composed of $3m$ samples. A different trace captured from the device which executes the operations in a different order such as (the non-cyclic order) *cba,* could be represented by $x2[n]= xa[n]+xb[n+m]+xc[n+2m]$. A non-cyclic ordering is one which cannot be obtained by a circular shifting of the operations. In all discussions it is assumed that signal $x2[n]$ is to be modified and $x1[n]$ is the reference signal (defining the reference phase). If one assumes that the set of frequencies defining each operation *abc* is distinct and non-overlapping then the phase substitution technique would reorder the operations without any error. For example since the FFT is a linear operator we have that $FFT(x1[n]) = FFT(xa[n+2m]) + FFT(xb[n+m]) + FFT(xc[n]),$ and similarly $FFT(x2[n]) = FFT(xa[n]) + FFT(xb[n+m]) + FFT(xc[n+2m].$ Furthermore we have that $PhaseXA\_of\_x2(f) = PhaseXA\_of\_x1(f) - 2\pi 2m\ f$ and $MagXA\_of\_x1(f) = MagXA\_of\_x2(f).$ Hence by setting the phases of $X2(f)$ to be equal to the phases of $X1(f)$, the order of operations in $x2[n]$ would be changed back to the same order of operations as used in $x1[n]$, specifically operations ordering *abc.* Hence the operations would be aligned. This would be possible for any random ordering of operations. However if more than one operation shared a particular frequency with another operation, which is a more realistic assumption, then only a cyclic reordering of operations, such as *cab,* could be modified using a linear phase adjustment or phase substitution to *abc.* Consider the phasors with θb and θc on the left in figure 1b). Using a linear phase shift of ϕ (achievable with phase substitution), the phasors are shifted as shown in the right of figure 1b), where the relative positions of components *b* and *c* remain the same in both diagrams, representing a cyclic ordering of operations. Now, consider the phasor in figure 1c) representing signal $x2[n]$ and in 1d) representing the $x1[n]$ signal. The angle between components *b* and *c* is $\theta b+\phi b-\theta c-\phi c$ in figure 1d), and is $\theta b - \theta c$ in figure 1c). This represents the case for a non-cyclic ordering of operations. Hence after phase substitution the resultant phasor in grey is shown on the right side in figure 1d) which has a different magnitude than the smaller black phasor (which represents the correct phasor). This phase substitution produces the magnitude error shown in the figure.

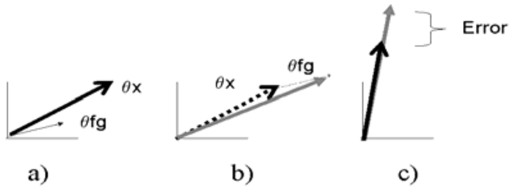


Figure 2. Phasor representation of operation *fg* insertion on signal *x* in a),b) and resulting error after phase substitution with new phase of *x* in c).

Next consider insertion of an operation *fg* in $x2[n]$ but not in $x1[n]$. Assume the $x1[n]$ is obtained from a set of operations *x* (such as *abc*) which does not include the random operation (*fg*). Figure 2a) shows the *x* and *fg* phasors and 2b) shows the resulting phasor for signal $x2[n]$ at a particular frequency. The phase substitution will move the resulting *xfg* phasor to the phase shown in figure 2c), shown as the longer gray vector. However the correct phasor would be represented by moving only phasor *x* to this new phase as shown by the black shorter phasor. Hence again magnitude error is introduced when a random operation is introduced into the traces and phase substitution technique is performed.

In all discussions it was assumed that signals were in fact noise free. In realistic cases each signal acquired will contain noise which is represented after the phase substitution technique by both magnitude and phase noise. For example the linear phase shift assumption is not possible since even the relative phases of two acquired traces which are shifted in time will not be linear due to acquisition noise. However if the noise introduced through phase substitution by the random operation reordering and random operation insertion is largely averaged out in the differential analysis (similar to the acquisition noise which is averaged out) then it may be possible for this approach to extract the secret key.

The proposed phase substitution technique is outlined in figure 3 as *PS_DEMA* (for phase-substitution DEMA). After the EM or power traces have been acquired, each trace is transformed into the frequency domain using the fast fourier transform (*FFT()*). One particular trace, referred to as the reference trace, is chosen randomly. This choice assumes that the attack point is within the trace (i.e. not cut off at the edge of the trace). This reference trace is referred to as trace *i* in the algorithm below. The phase of each trace (*PHASE()*) is replaced by the phase of trace *i.* After this replacement, an inverse fourier transform is performed in order to transform the trace back into the time domain. Then the normal DEMA is performed. An important part of the differential analysis is locating the significant peaks in a differential signal. The routine *SD_DOM* is the standard deviation of the difference of means. The differential peaks that exceed a constant multiple κ of *SD_DOM* are considered to be significant. We assume that κ is two or some other multiple of the standard deviation. The following terminology is used to describe the algorithms which follow, specifically *: j* ε *{0,...,n-1}* is the trace number; *b* ε *{0,1}* is the set number; T_j^b is the EM signal of set *b* and trace *j*.

```
PS_DEMA($T^0, T^1, \kappa, i$)
for each $j, j \in \{0,...,n-1\}$ $P_j \leftarrow FFT(T_j)$
$Phase \leftarrow PHASE(P_i)$,
for each $j, j \in \{0,...,n-1\}$ $PHASE(P_j) = Phase$
for each $j, j \in \{0,...,n-1\}$ $T_j \leftarrow Inverse\_FFT(P_j)$
$R \leftarrow SD\_DOM(T^0, T^1), D \leftarrow Mean(T^1) - Mean(T^0)$
$s \leftarrow Max(abs(D) - \kappa * R)$
return s
```

Figure 3. Phase-substitution DEMA algorithm

This approach is unlike previous research such as [13] which utilizes the magnitude of the signal in the frequency domain, not the phase, and research in [14][9] which perform correlations in the frequency domain. The next section describes the experimental results and comparison with the POC[14] technique.

3. Experimental Results

The experimental setup along with the results which illustrate the phase substitution technique and comparison with the POC approach[14] will be outlined in this section. Figure 4 illustrates the experimental setup used to acquire EM signals from the PDA device. A high sample rate oscilloscope, a 1-cm loop EM probe, wide band preamplifier, and a PDA (which was opened to expose the packaged chip over which the probe was placed) were used to acquire EM traces. The oscilloscope had the feature of being able to capture multiple traces, each activated by a trigger signal. In all cases 512 EM traces were captured in each scope acquisition. A trigger signal was generated from the PDA using the Java code to turn the light emitting diode (LED) on and off. The voltage across the terminals of the LED was used to trigger the scope. The Rijndael encryption algorithm (implemented using the table-based method of [2]) was used to illustrate the EM attack methodology. The Rijndael algorithm was written in Java, compiled into bytecode, and loaded onto the PDA device. Using the table method described in [2], the 32-bit outputs from the S-boxes are exclusive or'd together to create *t[0]* through *t[3]*. The plaintext was designed to allow a number of possible attack points such as the output of the first Sbox table or the value *t[0]* or the input to Sbox tables in round 2. For example the complete Rijndael encryption is executed in a loop on the PDA device for a finite number of iterations using different plaintext input. Specifically only the most significant byte of the 128-bit input plaintext is changed in order to allow only the first S-box table in round 1, to change. All other round one S-box table outputs remain constant, hence the noise created by these table accesses is minimized. This approach helped to maximize the probability of a successful attack (since initially it was not known exactly where the S-box load or *t[0]* store/load was located in the EM traces).

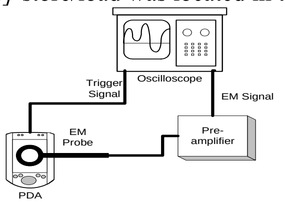


Figure 4. Experimental Setup

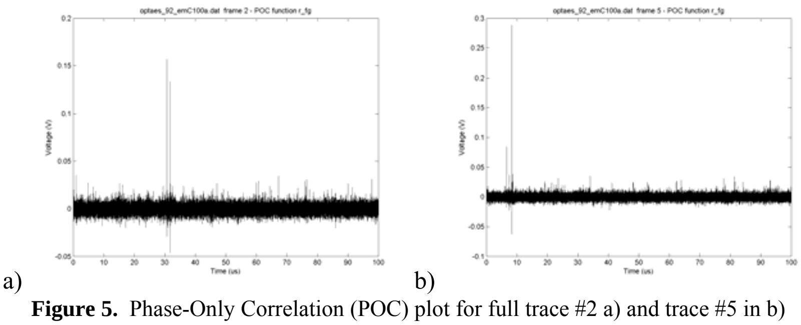
For trace alignment and random operation ordering problems, a comparison of the phase substitution technique is made to the POC technique[14]. The random ordering of operations countermeasure and random insertion of an operation countermeasure were simulated by manipulating traces in the time domain since the Java environment made it difficult to control the operation ordering and a non-deterministic processor was not available. The manipulation of traces in the time domain either reordered parts of the EM trace or inserted other real EM trace fragments representing other operations into the trace. All EM traces used in this section are from the PDA device and are not aligned due to cache misses and other interrupts. Initially each acquired EM trace had a length of 100us and contained 50,000 EM samples. The 15us traces were extracted using pattern recognition techniques from the 100us traces. However these 15us traces were not aligned (as verified by failed DEMAs). The 15us traces formed the basis of the experiments which follow and they each contained 7500 samples. Differential analysis was performed in each case using 512 EM traces.

The phase-substitution technique will be compared to a variation of the POC technique[14]. Specifically since resolution higher than one sample period is not required with our experimental setup, the POC technique is simplified by omitting the correlation peak function fitting. The POC function was defined [14] as:

$$r_{fg}(n) = Inverse\_FFT\left(\frac{F(k)\overline{G}(k)}{\left|F(k)\overline{G}(k)\right|}\right)$$

where *F(k)* is the FFT of the time domain signal *f(n)* and *G(k)* is the FFT of the reference time domain signal *g(n).* The location of the maximum of the POC function $r_{fg}(n)$, in terms of index n (integer number of sampling periods), can be used for the estimate of the displacement. The waveform shifting can then also be more efficiently implemented directly in the time domain as a single circular time shift of *f(n),* using the POC function peak estimate of the displacement, since this in now an integer number of sampling periods i.e. this is a simple array shift operation. Note that in [14] the displacement has a fractional part and cannot be implemented by a time domain circular time shift, and instead requires more complicated phase rotation of each frequency component, followed by an inverse FFT.

Figures 5 a) and b) show the POC plots for two different traces using the same reference trace. Figure 5a) has two peaks, where the highest peak indicates a displacement of 15,348 samples (30.696 us). However the correct displacement for the two traces is 15,820 samples (31.640 us) which is closer to the 2nd highest peak (31.636 us which is off by only 2 samples). Figure 5b) indicates a correct displacement of 4,230 samples (8.460 us). DEMA experiments verified that in 6 out of 8 cases the POC technique outlined above and the phase substitution were both able to align traces for successful DEMAs using the acquired EM traces with no countermeasures.



a)

b)

Figure 5. Phase-Only Correlation (POC) plot for full trace #2 a) and trace #5 in b)

3.1 Random Ordering of Operations: Comparison to POC

Countermeasures, such as random ordering of operations have been proposed previously to thwart DEMA or DPA. This countermeasure was examined with respect to the phase-substitution technique and the POC technique [14]. Experiments using cyclic and non-cyclic interchanges of operations were examined. For example a cyclic interchange of *abc* could be *cab* (by circularly shifting to the right) whereas a non-cyclic interchange could be *cba*. In the experiments to follow, each EM trace was divided into 3 parts, in order to represent 3 operations, (by time slicing the trace at areas of EM low activity) referred to as *abc*, see figure 6a). Part *'a'* was extracted from times 0-7.5us, part *'b'* from 7.5-10.5us and part *'c'* from 10.5-15us. Note that the region of attack is in time partition *'c'*. In all experiments the reference trace was trace 2, *abc*.

Half of the traces were transformed into a non-cyclic shift *cba* and the other half remained as *abc*. The non-cyclic shifts were created by moving the 3 parts of sampled signals in the time domain, as shown in figure 6b). Figure 6c) illustrates the final time-domain trace created after phase substitution where the *abc* order of operations is largely restored with additional noise. The POC technique was unable to find the correct key in the DEMA as shown in figure 7 b). Specifically the correct key had the 107th highest peak. For the first 256 traces (where a non-cyclic shift *cba* was utilized) the POC technique incorrectly creates operation ordering *acb*, but for the next 256 traces it maintains operation ordering *abc*. Since the region of attack is in time partition *'c'*, it is not aligned between the first and second half of traces, causing the DEMA to fail. The phase substitution technique which attempted to restore the *abc* order in all EM traces found the correct key as shown in the DEMA of figure 7 a).

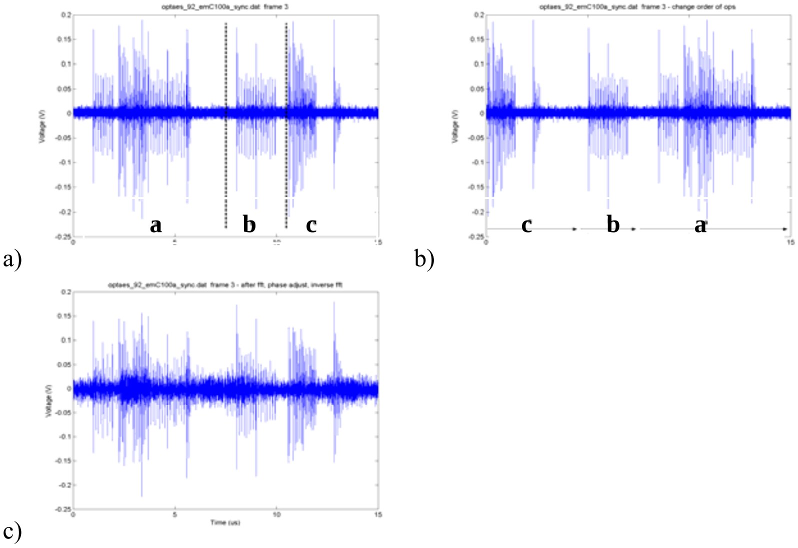


Figure 6. a) EM trace #3, showing sections *abc* in a) and reordered in b) to *cba* and in c) after the phase substitution using the phase from a reference trace with *abc*.

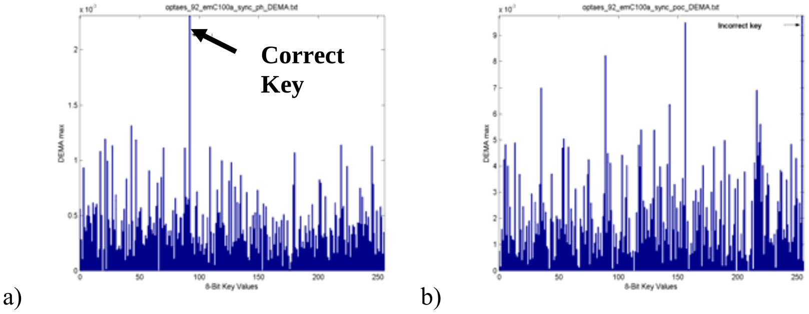
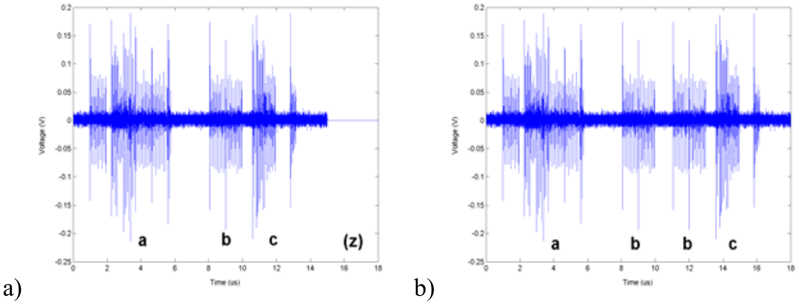


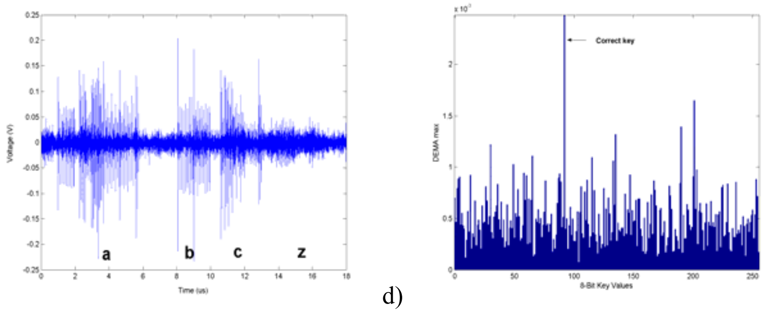
Figure 7. DEMA with Phase-Substitution results in a), with POC results in b)

3.2 Random Insertion of Operations

The remaining experiments investigated the simulation of the countermeasure which performs random insertion of operations. To simulate this countermeasure four experiments were performed. In some experiments, half of the traces had no changes to the *a,b,c* operations except appended zeros in order to have the same number of samples in each trace for the FFT, *abcz*, representing a 18us trace (3us of appended zeroes). The other half of the traces had different types of operations inserted. The first experiment took an existing operation from the 15us trace and repeated it. The second experiment utilized a new operation from the wider 100us trace containing the 15us trace, called operation *x*. The third experiment was similar to the second experiment however the reference trace was different. The final experiment took the new operation from the previous trace to ensure negligible data correlation.

The first experiment transformed all of the traces by duplicating operation *b*. For example in figure 8a), *abcz,* the operation *b* was extracted and copied twice into the same trace as shown in figure 8b), *abbc*. The reference trace was *abcz* similar to figure 8a). The trace shown in figure 8b) after phase substitution is shown in figure 8c). It appears that only one *b* operation is evident and the noise level is much higher than in figure 8b). The DEMA performed on this set of traces was successful and is shown in figure 8d).

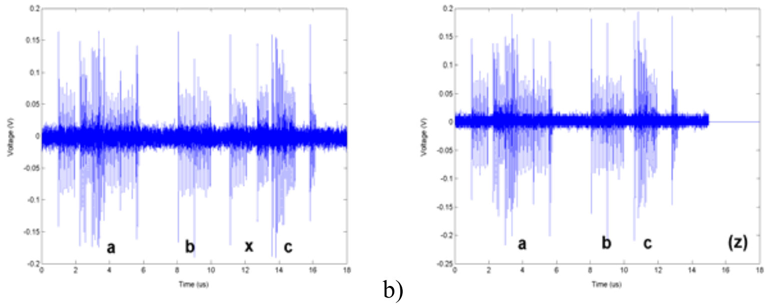




c) d)

Figure 8. Trace in a), modified with repeated operation *b* in b), after phase substitution in c) using reference phase from a trace *abcz* and successful DEMA in d).

Instead of duplicating an operation, a new operation *x* (obtained from outside of the 15us window of the 100us trace) was inserted in between operations *b* and *c,* to produce *abxc*. For example half the traces were of the form *abxc* and the other half were *abcz* and the reference trace was *abxc.* For example a trace with operation *x* inserted was transformed after phase substitution into the trace shown in figure 9a). The inserted operation *x* can be seen as well as some noise. Another trace which did not have the operation *x* inserted into it, shown in figure 9b) was transformed after phase substitution into the trace shown in figure 9c). It is very interesting to note that the region of operation *x* appears to have some noise shaped similar to operation *x.* The DEMA was not successful since the correct key was the 28th highest. However when the DEMA was run on a smaller 0.6us window, containing the attack point, the 2nd highest DEMA peak indicated the correct key as shown in figure 9d). Results were also rerun using a different reference trace of *abcz* without the operation *x* inserted*.* In this second case, the correct key had the 2nd highest DEMA peak without using a smaller time window.



a) b)

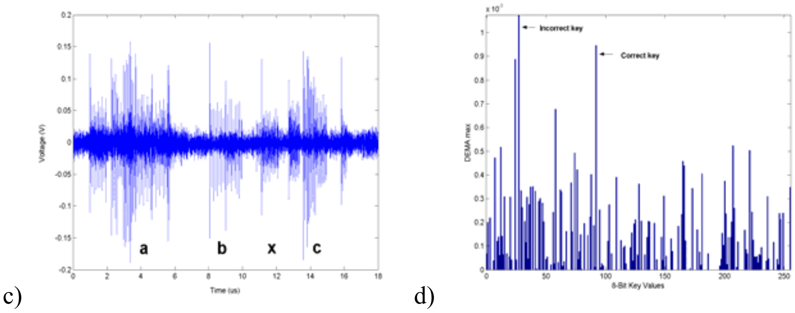


Figure 9 Traces with *x* in a), without *x* (in b)) in c) after phase substitution. DEMA in d).

Operation reordering and operation insertion was investigated in this third experiment. In half of the EM traces operations *abc* were changed to *cba* and additionally an operation *x*, was inserted to create traces of the form *cbxa*. The other half of the traces remained in their original form with appended zeros, *abcz* . For example the trace, see figure 8a), was modified in the time domain to the trace *cbxa* shown in figure 10a), where *x* is shown as *FG*. The reference trace was of the form *abcz.* The trace of figure 10a) was transformed into the trace in figure 10b) after phase substitution. It appears that the operations have been reordered correctly and operation *x* cannot be seen, however there is significant noise. Figure 10c) shows a trace after phase substitution

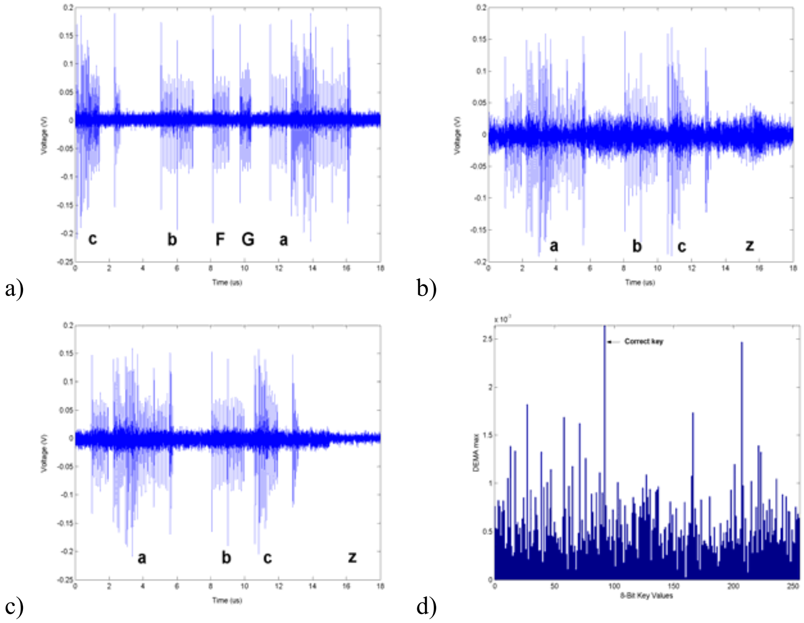


Figure 10 trace #3 with inserted operations *F,G* in a), and after phase substitution in b), c) trace #259 (originally *abcz)* after phase substitution and DEMA in d)

which originally was of the form *abcz.* It can be seen that figure 10b) and 10c) look very similar except 10b) has more noise. The DEMA is shown in figure 10d) where the correct key is obtained.

Several other experiments were also performed. For example one experiment took all traces and instead of substituting both *FG* only *F* was inserted, so that the traces were of the form *abfc*. The reference trace was of the form *abcz*. The DEMA found the correct key with the 2nd highest peak in this case. This experiment was repeated however the inserted operation *F* was taken from the previous trace in an attempt to minimize any possible data correlation effects. For example in trace 4 the inserted operation *F* was taken from trace 3, etc. The DEMA had similar results of finding the correct key with the 2nd highest peak.

4. Discussions and Conclusions

In summary this paper presents a new phase-substitution approach for aligning and resisting countermeasures which involve reordering of operations and insertion of operations. The traces used in the experimental section were real EM signals acquired from a Java based PDA device executing Rijndael. Unlike the phase-substitution technique, the POC technique[14] could not thwart the random operation ordering countermeasure. The phase-substitution technique was largely successful in thwarting the random operation ordering countermeasure experiment since it was able to reorder the operations according to the operation order in the reference trace. It is interesting to note that unlike [14] there is additional noise created using the phase-substitution approach. This noise can be seen in the time domain in comparison with the original EM trace. However this did not have any significant impact on the subsequent DEMA, likely since even the original EM trace has noise and the averaging in the DEMA reduces this effect. The operation insertion countermeasures were also in general quite successfully thwarted. However sometimes smaller time windows near the attack point were necessary. The results demonstrate that embedded mobile code must contain other countermeasures, such as masking [12], secret sharing[17], etc[18,19], in order to avoid attacks which extract secret cipher keys.

The proposed phase-substitution approach may also be applied to power signals, however these are more difficult to obtain from the chip of the PDA housing the respective processor. Power signals obtained from the battery of the mobile device may also be difficult since signals likely would be buried amidst the current drain of other components requiring power. For example on PDA devices, there are many different components including power management circuitry, radio circuitry, baseband processor, etc.

For the first time using real EM measurements from a PDA device executing Java-based cryptography, a phase-substitution approach is proposed and shown to be useful for resisting insertion of operations and operation reordering types of countermeasures as well as for alignment of EM traces. Unlike previous research [14], the phase-substitution approach is able to thwart the simulation of the random ordering of operations countermeasure. This new attack also thwarts previously suggested countermeasures which involve insertion of random delays to misalign the attack point. This research is crucial for supporting low energy security for embedded systems which will be prevalent in wireless embedded devices of the future. Authors would like to thank NSERC, OCE-CITO, and RIM for their support.

References

1. P.Kocher, J.Jaffe, B.Jun "Differential Power Analysis" Crypto'99, LNCS 1666 (1999)388-397
2. Dr.Brian Gladman, "A Specification for Rijndael, the AES Algorithm", at fp . gladman . plus . com / cryptography_ technology / rijndael / aes.spec.311.pdf (2003)
3. D.Agrawal et al. "The EM side-channel(s)" CHES 2002 (2002) 29-45
4. K.Gandolfi etal. "Electromagnetic Analysis: concrete results" CHES 2001, LNCS 2162, (2001) 251-261
5. D.Agrawal, etal. "The EM side-channel…methodologies" at http ://www.research.ibm.com/intsec/emf.html
6. K.Itoh etal. "DPA countermeasure based on the masking method", LNCS 2288 (2002) 440-456
7. T.Messerges "Using 2nd order power analysis to attack DPA resistant software", LNCS 1965 (2000) 238-251
8. S.Ravi, etal. "Securing Wireless Data: System architecture challenges", ISSS (2002) 195-200.
9. J.Waddle, D.Wagner "Towards efficient second-order power analysis" CHES 2004, LNCS 3156, (2004) 1-15
10. D.Agrawal, etal. "Advances in Side-Channel Cryptanalysis EM analysis and template attacks" RSA Cryptobytes, Vol6 No1 (2003) 20-32
11. D.Agrawal, etal "Power, EM and all that: is your crypto device really secure?" presentation ECC workshop http://www.cacr.math.uwaterloo.ca/conferences/2003/ecc2003/rohatgi.ppt (2003)
12. T.Messerges, "Securing the Rijndael finalists against power analysis attacks" LNCS 1978, (2001) 150-164
13. C.Gebotys, S.Ho, A.Tiu "EM Analysis of Rijndael and ECC on a Wireless Java-based PDA" Proceedings of CHES 2005, LNCS 3659, Springer-Verlag GmbH, pp.250-265.
14. N.Homma, etal. "High-resolution side-channel attack using phase-based waveform matching" Proceedings of CHES 2006, to appear in LNCS, Springer-Verlag GmbH
15. J-J. Quisquater, etal. "a new tool for non-intrusive analysis of smartcards based on EM emissions", Rump Session, Eurocrypt (2000)
16. Y.Oren, A.Shamir "Power analysis of RFID Tags" http://www.wisdom.weizmann.ac.il/~yossio/rfid 2006
17. S.Chari, etal. "Towards sound approaches to counteract power-analysis attacks", LNCS 1666 (1999) 398-412
18. L.Goubin, J.Patarin "DES and Differential power analysis- the duplication method" CHES (2001) 158-172
19. J.Golic "Multiplicative Masking and power analysis of Rijndael", CHES (2002) 1-10