EM Analysis of Rijndael and ECC on a PDA

Catherine Gebotys, Simon Ho, Agnes Tiu
Dept of Electrical and Computer Engineering, University of Waterloo
Waterloo, Canada
cgebotys@uwaterloo.ca

Abstract

Although many wireless portable devices offer more resistance to bus probing and power analysis attacks due to their compact size, susceptibility to electromagnetic (EM) attacks must be analyzed. This paper demonstrates, for the first time, a real EM-based attack of a PDA running Rijndael and elliptic curve cryptography. A new frequency-based differential EM analysis, which computes the power spectral density and spectrogram, is presented. Additionally a low energy countermeasure for symmetric key cryptography is presented which avoids large overheads of table regeneration or excessive storage. Unlike previous research the new differential analysis does not require perfect alignment of EM traces, thus supporting attacks on real embedded systems. This research is important for future wireless embedded systems which will increasingly demand higher levels of security.

1. Introduction

As more security applications migrate to the wireless device, resistance to attacks on the PDA or cellphone will become a necessity. These attacks may not only arise from device theft or loss but also during everyday use where unintentional electromagnetic (EM) waves radiated from the wireless device during cryptographic computations may leak confidential data to a nearby attacker. Researchers have already demonstrated that this EM attack is viable[7,10] on a 8-bit processor running at 4MHz in a smartcard. For example 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. Nevertheless large overheads in energy to achieve resistance to these attacks may not be practical for wireless embedded systems. Outside of smartcard research (which in the past has typically been limited to cheaper 8-bit or 16 bit processors) [5,12,1,4], few researchers have examined secure implementations of cryptographic software (such as Rijndael[6] 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 significant amount of energy [2]. Even worse, cryptographic algorithms which are resistant to attacks are known to have latency overheads up to 1.96 times[3]. Although these attack resistant algorithms have been developed for smartcard applications, (where energy dissipation is not viewed as important), there is an important need to study EM attacks and energy optimized countermeasures on wireless portable devices, such as PDAs, cellphones, etc.

2. Previous Research

Typically in symmetric encryption the plaintext and key are exclusive or'd together and then indexed into a table, as in the table method of the Rijndael advanced encryption standard[6]. The attacker has control over the plaintext and by guessing the 8-bit key value, can partition EM or power traces according to a bit in the data at the output of the table. By taking the difference of the average of the partitioned traces and 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). In elliptic curve cryptography (ECC) the data at the output of a double operation can be similarly partitioned according to the guess of the scalar key bit as described in [19]. Although EM attacks on smart cards have been investigated [10,8], EM attacks on other embedded systems have not been researched. 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 or SEMA). In the former case, DEMA, the DES encryption[10] was analyzed. Differential EM attacks of embedded low power processors have not been reported in the literature. Higher order (n^{th} order) differential attacks[16] are an extension of the 1st order differential analysis which involve using joint statistics on multiple (n) points within power traces. These higher order differential analyses have been shown to provide more security, since they require more EM or power traces[16,10]. For example research with real EM measurements using a 8-bit processor running at 4MHz in a smart card, demonstrated 2nd order DEMA [10] on a 2-way XOR-based secret sharing scheme using 500 EM traces. However no known results using real EM measurements have been obtained for an attack on ECC. In most cases, good EM or power trace alignment of the attack point is required since most previous differential analyses is performed in the time domain. The exception to this is [17] where the fast fourier transform is calculated, however transformation back into the time domain occurs before differential analysis is performed. This paper will use the terms DEMA or DPA to denote time domain differential analysis.

Previously researched countermeasures have been suggested such as random sequencing of instructions (desynchronization), secret splitting[9], duplication method[14], multiplicative masking[15] and random masking[3]. Secret splitting involves splitting the secret data into smaller pieces and combining them with random data [9]. To attack the splitting method, a k^{th} order differential attack is required[9]. The duplication method[14] 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[15]. 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 [14] (called the 'fixed-value' masking). However results using real EM or power measurements were not performed. Countermeasures for power or EM analysis of ECC include i) indistinguishable formulas for point operations, ii) identical operation sequences regardless of key bits, and iii) random addition chains. In general the first approach, i), restricts the ECC to specifically chosen curves [24,18] and is generally vulnerable to differential attacks [22]. The second approach, ii), includes algorithms like the double-and-always add algorithm [19] and others [18, 21], which thwarts simple analysis attacks but can be attacked using differential analysis. The random addition chains approach, iii), utilizes sequences of additions, subtractions, and doublings that can mutate randomly[26]. Other general approaches to resisting differential analysis of ECC include randomizing the base point (such as point blinding [19]) and randomizing the scalar [19,25]. Countermeasures designed for thwarting differential analysis of sliding window implementations of ECC include [23]. In general there are few publications reporting differential analysis of ECC using real power or EM.

Unlike previous research, this paper presents results of EM analysis on a real embedded system, a wireless Java-based PDA. Rijndael and elliptic curve scalar multiplication are used to demonstrate a new differential attack based solely upon analysis in the frequency domain. Furthermore analysis is performed in the presence of a countermeasure for table-based symmetric key cryptography suitable for

PDA like devices. Comparison to previously researched attacks show that frequency based analysis is crucial for real embedded systems, since previous time domain analysis were not successful in finding the correct key in all cases. A most-significant-bit differential attack is found to be stronger for ECC than attacking any bit as previously described in [19]. The EM analysis attack is demonstrated for the first time on ECC running on a PDA. The next section will describe the proposed frequency based differential analyses techniques and the following section will present the experimental results.

3. Differential Analysis in the Frequency Domain

This paper proposes an extension of the existing differential side channel attack, where instead of performing analysis in the time domain, the frequency domain is used. Two approaches are described, one is called differential frequency analysis (DFA), where the power spectral density (PSD) is used for analysis, and the other will be called differential spectrogram analysis (DSA), since a spectrogram (SPECGRAM) is created. In general instead of computing the differential signals in the time domain (as in [4] and in almost all previous research), the computation is performed in the frequency domain. Analyzing signals captured in the frequency domain solves the problem of misalignment (or time-shifts) in traces since fast fourier transform (FFT) analysis is time-shift invariant. Both the DFA and DSA are important for attacking real embedded systems where uncorrelated temporal misalignment (or time-shifting) of traces (typically caused by the triggering signals or Java operating system) is a big concern. Additionally, frequency analysis may reveal loops and other repeating structures in an algorithm that is not possible with time domain analysis. However there are two problems with using frequency domain signals in differential analysis. First, it reveals no information of when data-dependant operations occur. This timing information is very useful as it helps an adversary focus the signal analysis on these data-dependant operations. Secondly, any peaks in frequency domain due to an event that occurs in a short duration may be discernable if the acquisition duration is a lot longer. The solution of these problems is to use spectrogram, which is a time dependant frequency analysis.

There are two main components of creating a spectrogram (see DSA) for each window in a time domain trace. The first component is taking the FFT, which results in a frequency domain signal. Again, from Nyquist criterion, the size of this frequency signal is half of the size of the time window. The second component is taking a dot product between the frequency signal and a Hamming window. The application of Hamming function suppresses the Gibbs' phenomena in spectral windowing. The creation of spectrogram is detailed in the algorithm below. When only one window whose width is equivalent to the duration of the time domain trace is used, the power spectral density (see DFA) can be used to perform differential analysis in the frequency domain.

Each signal trace is measured over an interval of m time points. A spectrogram applied on p time windows with w time points would have $w/2$ frequency points in each window (assuming w is an even number) and $wp/2$ points in total, according to Nyquist criterion. The overlap is the difference of window size from interval between windows. The following terminology is used to describe the algorithms which follow, specifically *PSD, SPECGRAM, SD_DOM, DSA,* and *DFA.* :

$i =$ trace number, $i \in \{0,\ldots,n-1\}$; $b =$ set number, $b \in \{0,1\}$; T_i^b = EM signal of set b and trace i;
V_i^b = spectrogram of set b and trace i; P_i^b = power spectral density of set b and trace i; $t =$ time, $t \in \{0,\ldots,m-1\}$;
$s =$ frame number in spectrogram, $s \in \{0,\ldots,p-1\}$; $w =$ window size; $f =$ frequency, $f \in \{0,\ldots,\frac{wp}{2}-1\} = \{0,\ldots,\frac{m}{2}-1\}$.

```
PSD(T)
for each $b, b \in \{0,1\}$ and
        for each $i, i \in \{0,\ldots,n-1\}$:
    $P_i^b \leftarrow \left|FFT(T_i^b)\right|^2$
return $P$
```

```
SPECGRAM(T)
for each $b, b \in \{0,1\}$ and for each $i, i \in \{0,\ldots,n-1\}$:
    for each $s, s \in \{0,\ldots,p\}$:
        $F \leftarrow \mathrm{FFT}(T_i^b(s*w:(s+1)*w-1))$
        $V_i^b(s*\frac{w}{2}:(s+1)*\frac{w}{2}-1) \leftarrow F \bullet \mathrm{HAMMING}(\frac{w}{2})$
return $V$
```

An important part of 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. The DSA and DFA algorithms are detailed below and are identical except DSA computes the differential of the spectrogram signals whereas the DFA computes the differential of power spectral density signals.

```
DSA($T^0, T^1, \kappa$)
return $DA(T^0, T^1, \kappa, SPECGRAM)$
```

```
DFA($T^0, T^1, \kappa$)
return $DA(T^0, T^1, \kappa, PSD)$
```

```
DA($T^0, T^1, \kappa, type$)
if type = PSD then, $P \leftarrow PSD(T)$, else $P \leftarrow SPECGRAM(T)$
$R \leftarrow$ SD_DOM$(P^0, P^1)$, $D \leftarrow \mathrm{Mean}(P^1) - \mathrm{Mean}(P^0)$, $s \leftarrow 0$
for each $f, f \in \{0,\ldots,\frac{wp}{2}-1\}$:
    if (abs$(D(f)) > \kappa * R(f)$)
        $s \leftarrow s + (\mathrm{abs}(D(f)) - \kappa * R(f))$
return s
```

The analysis methodology involves first using DSA on a large window to locate possible areas of attack. Next the attacker can focus in on smaller windows which show interest. Finally the DFA can be computed on specific areas for more detail. For ECC attacks, the most-significant-bit is chosen to perform the partitioning into the two sets (V^b, P^b, T^b). The theory behind choosing the most-significant-bit is given in appendix A. The DFA is analogous to using the DSA but computed over the entire time interval using only one hamming window. Mathematically there are differences, for example the power spectral density is calculated and no hamming window is used in the DFA, unlike the DSA. However for our analysis purposes, similar results were obtained for the DFA and the DSA using one hamming window. The DFA was chosen since it made more sense in the case of analysis of a single time period.

The next section will present the results of the DFA, DSA, proposed countermeasure (see Appendix B), and previous attack techniques on Rijndael and ECC running on a wireless PDA.

4. Experimental Results

A high sample rate oscilloscope, a 1-cm loop EM probe, wide band amplifier, and a PDA (which was opened to expose the packaged chip over which the probe was placed) were used to acquire EM traces. Figure 1 is a photograph of a single loop EM probe over the chip in the PDA. The Rijndael (de)encryption algorithm (implemented using the table-based method of [6]) was used to illustrate the EM attack and countermeasure verification. Additionally analysis of an Elliptic Curve point multiplication (using 192bit prime field with Jacobi projective coordinates as standardized in FIPS 186-2[27]) is performed using the new differential analysis techniques, DFA and DSA. Both the Rijndael and ECC code was written in Java and loaded onto the PDA device. 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.

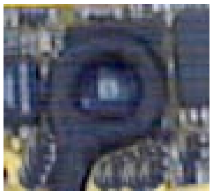

Figure 1. EM Probe over chip in PDA device.

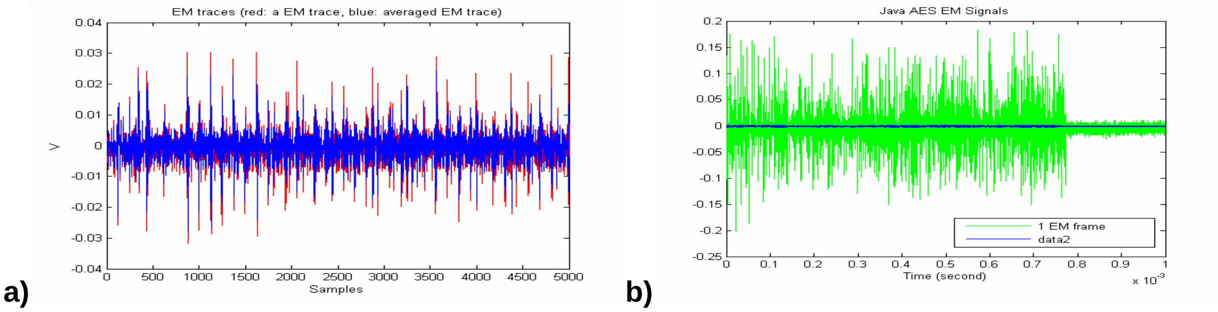

Figure 2. Averaged (blue) versus raw EM signal (red/green) from single processor Evaluation Board in a) and Java-based PDA device in b)

Figure 2 illustrates the difference between capturing EM signals from a single processor evaluation board (where assembly code can be written and executed directly on the processor) in a) and EM signals from a real embedded device with operating system and executing Java code in b). In the PDA experiment, Rijndael was run with the same input data for capturing of over 1000 EM traces. The PDA captured the end of the Rijndael algorithm, whereas the evaluation board captured a different part of the algorithm. It is clear that there is little difference between the average signal and the raw EM signal in figure 2a). However in figure 2b) it is clear that there is significant difference between EM signals running on a Java-based PDA device.

Figure 3a) presents a single EM trace of 128 bit Rijndael captured by the scope from the EM signals emanating from the chip in the PDA device. Each of the 10 rounds can be seen in the figure, thus illustrating a SEMA attack on the device. The EM signals at the end are created by the timer interrupt. A thread is created to call the AES encryption algorithm. After AES's execution has completed, the thread is programmed to sleep. Therefore, there are minimum EM signals as shown in the graph when the thread is in the sleep mode. However the timer interrupt occurs every few milliseconds to check if a thread needs to be activated (evident from the last EM burst, which is not part of the AES algorithm). Figure 3b) illustrates scope capture of EM signals from PDA running a 192 bit Rijndael, where 12 rounds are evident.

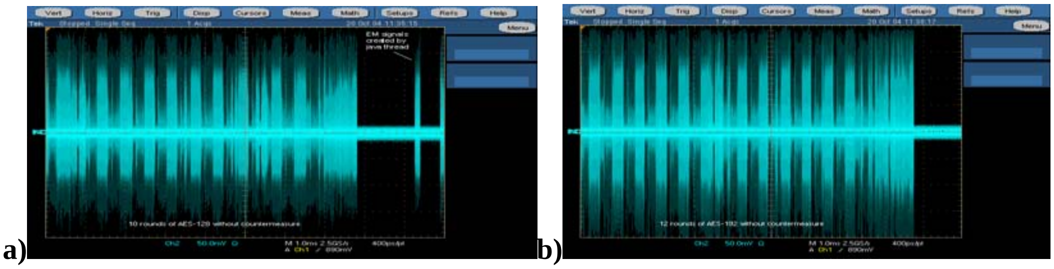
Vert
Horiz
Trig
Disp
Cursors
Meas
Math
Setups
Refs
Help
Menu
Tek Stopped Single Seq 1 Acqs 20 Oct 04 11:35:15
EM signals created by java thread
10 rounds of AES-128 without countermeasure
Ch2 50.0mV Ω
M 1.0ms 2.5GS/s 400ps/pt
A Ch1 ∕ 890mV
a)
Tek Stopped Single Seq 1 Acqs 20 Oct 04 11:38:17
12 rounds of AES-192 without countermeasure
b)

Figure 3. SEMA attack on Rijndael in PDA device with 10 rounds a), and 12 rounds in b).

The EM trace and EM average of elliptic curve cryptography running on the PDA is shown in figure 4. For example different iterations of the same point double routine in the ECC algorithm are shown in figure 4a) and figure 4b). Clearly there are significant differences in the EM traces.

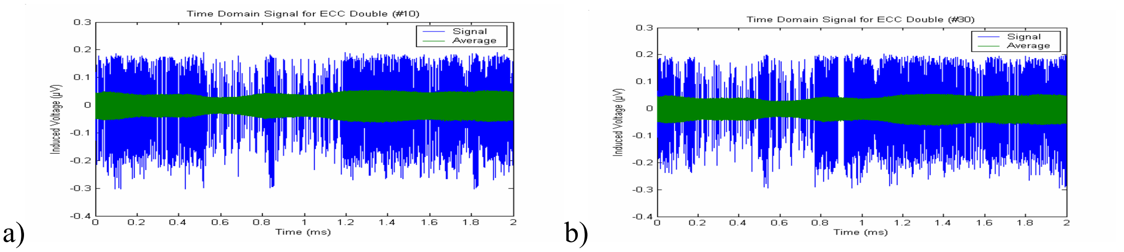

Figure 4. EM Signal of ECC Double on the PDA over Iteration #10 and #30 in a) b) respectively.

4.1 Differential EM Results for Rijndael

The EM traces were acquired from the PDA while it was executing a Java-based implementation of the table-based method of the Rijndael algorithm[6]. Results are presented for DSA, DFA and the previously researched DEMA. Additionally the proposed split mask countermeasure (detailed in Appendix B) is also analyzed. Differential traces and plots of all keys versus peak height are provided.

According to section 3's DSA algorithm, the spectrogram is calculated for each EM trace. Then the set of spectrograms is partitioned into two groups based upon a key guess. The mean of each group of spectrograms is calculated. Finally the difference of these two means will be called the differential spectrogram trace. The differential spectrogram trace, is illustrated in figure 5a) for a correct key guess partitioning of EM traces (where partitioning was based upon the least significant bit at the output of the first Sbox table). Time intervals of 0.1ms were used to create this spectrogram. In between the 0.1 time intervals is the plot of the differential signal over a range of frequencies. For example a more detailed look at the frequency plot for the time period 0.8 to 0.9ms of figure 5a) is analogous to the plot shown in figure 6a) (except in figure 6a) the PSD is utilized rather than the spectrogram, see section 3 for mathematical differences). A total of 1030 EM traces were acquired with the scope set at 25M samples/sec. Each EM trace had 25,000 sample points. Figure 5b) illustrates the analysis of the same set of EM traces, however partitioned for an incorrect key guess. Plus or minus two standard deviations are show as red in the figures and the actual differential spectrogram trace is shown in blue. Clearly Figure 5a) indicates significant differentials over the region of 0.7 to 0.9 ms.

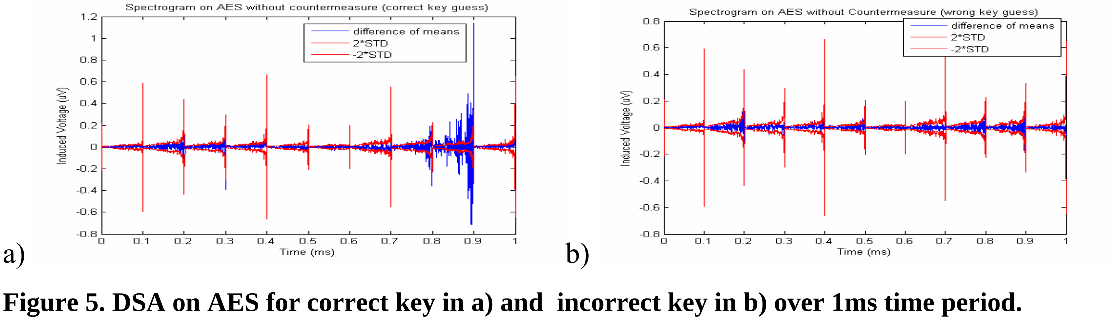

Figure 5. DSA on AES for correct key in a) and incorrect key in b) over 1ms time period.

The differential PSD traces, or the difference of means of the power spectral density of each EM trace, are shown in figure 6. Each EM trace was a acquired over a smaller window corresponding to the 0.8 to 0.9ms time period of figure 5. The differential PSD trace using a correct key guess for partitioning is shown in figure 6a), and an incorrect key guess differential is shown in figure 6b). Again the correct key guess is evident from the significant differential trace (in blue) appearing outside of the two standard deviations (in red).

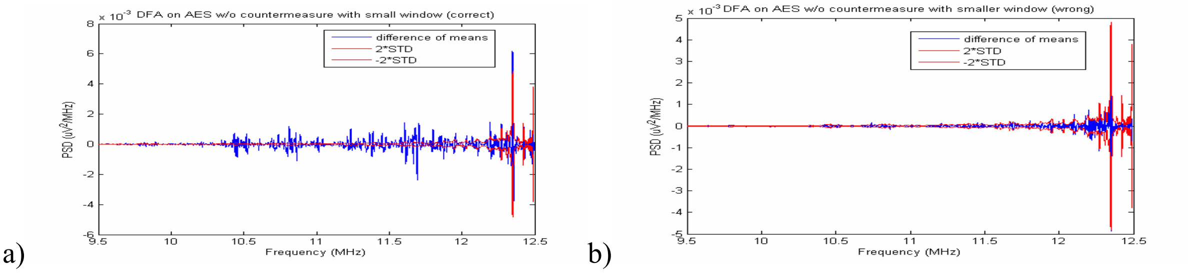

Figure 6. DFA on AES for correct key in a) and incorrect key in b) for 0.8 to 0.9ms time period.

Figure 7 illustrates the results of the previously researched DEMA applied to the same set of acquired EM traces. This differential trace, which is computed solely in the time domain, does not show any significant spike outside of the two standard deviations. These insignificant results were further confirmed by computing the DEMA for all possible key guesses, and plotting the maximum absolute peak value of the differentials for each key, as shown in figure 9c).

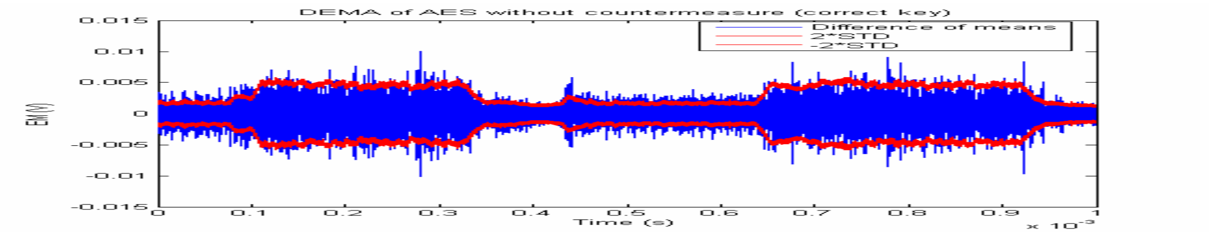

Figure 7. DEMA of AES for correct key over 1ms.

The DSA algorithm as detailed in section 3, returns the sum of the absolute part of the differential trace which was outside of the two standard deviations across the frequency spectrum (in each time region) of the spectrogram. The plot of this value versus key value is shown in figure 8a). The correct key is clearly evident since it has the highest value. Figure 8b) provides the same plot but obtained from analyzing the EM traces acquired from Rijndael with the proposed split mask countermeasure (see appendix B). Results show that the correct key is not even close to highest peaks in figure 8b), hence the countermeasure is effective against this differential spectrogram analysis. The previously researched second order analysis techniques in [17], which are also known to be time-shift invariant, were used to attack the countermeasure. However they were not successful in attacking the countermeasure even though over 2000 EM traces were used. The key plot for the 2nd order attack in [17] on the proposed countermeasure is shown in figure 8c), which uses double the number of EM traces (over 2000), and does not reveal the correct key.

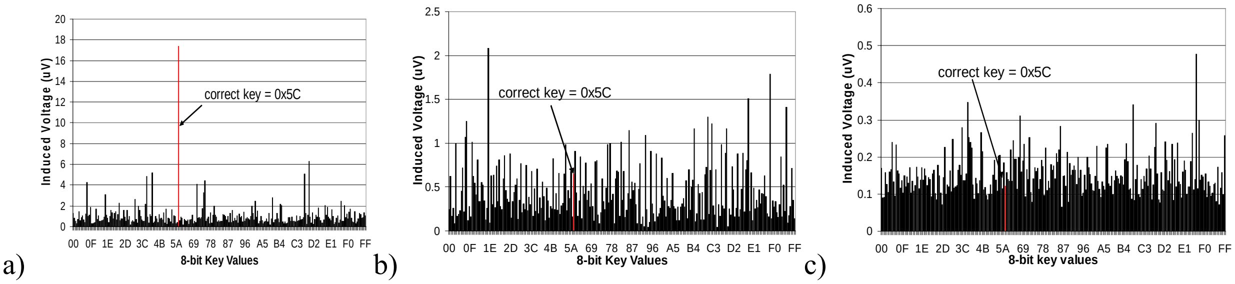

Figure 8. All keys vs Differential for Rijndael, using DSA in a), with the split mask countermeasure using DSA in b) and using 2nd order attack from [17] on same EM traces with countermeasure c).

A plot of key guess versus the value returned from the DFA was also performed on the same set of EM traces. The DFA algorithm returns the sum of the absolute value of the differential trace which appears outside of the two standard deviations and summed across the entire set of frequencies. This analysis was done for both a large window (1ms) and a small window (0.8-0.9ms area), as shown in figure 9 a) and b) respectively. Results are more pronounced for the smaller window size. The previously researched DEMA was also analyzed with respect to all key guesses, and the maximum absolute peak value of the differential outside of the two standard deviations was also recorded in each case in figure 9c). Similar results were also obtained by taking the maximum absolute peak value (and ignoring the standard deviations). The correct key is not evident from this DEMA, since the correct key has a lower peak than several other keys. Likely the misalignment of traces cancels out the spike that would normally occur in DEMA. However, both differential spectrogram analysis and differential frequency analysis (DFA) could determine the correct key and are less affected by the time shift problem.

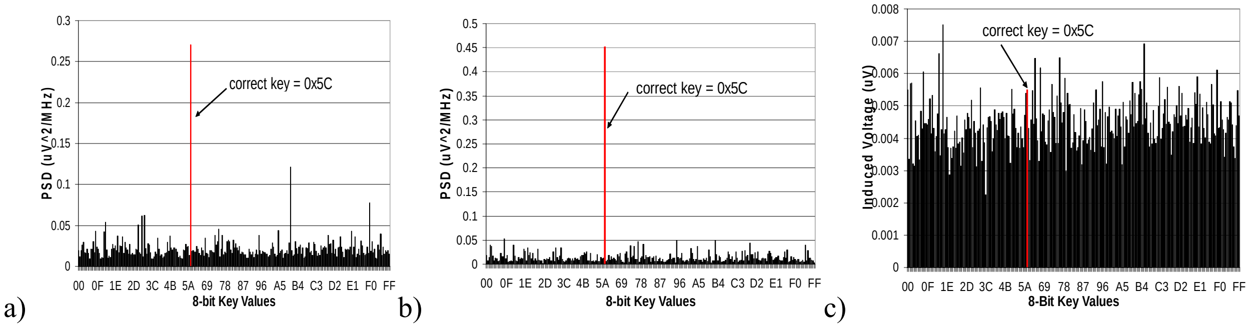

Figure 9. All keys vs Differential for Rijndael, using DFA with large window in a), DFA with small window in b) and using previously researched DEMA in c).

It was not possible to accurately measure the energy of the PDA device while it was computing the cryptographic algorithm because the pins of the chip were too small to access, and the fraction of battery energy was also too small to measure. Hence to obtain the energy measurements, a separate processor evaluation board was used which contained a 32-bit ARM7TDMI RISC processor core on one chip separate from the memory. The proposed countermeasure, implemented with one extra mask table, provided 1.7 times increase in energy over Rijndael with no countermeasure (AES) and 5.2 times less energy than the countermeasure in [3] where table regeneration is required when a new mask is applied. When two mask tables were used in the proposed split mask countermeasure, the energy dissipation was 2.4 times more than the Rijndael without a countermeasure, and 3.7 times less energy than the countermeasure in [3]. Since these results did not represent the energy dissipation of the memory, and since it is well known that memory energy dissipation is significant and often dominates, an analysis was performed with static RAM models from [11]. The proposed countermeasure, with one or two extra mask tables, requires significantly less memory when supporting the same number of masks. For example with $(1/5)^{th}$ the number of masks, [13] dissipates up to 10.4 times more energy than the proposed countermeasure.

4.2 Differential EM Results for Elliptic Curve Cryptography

Over 1300 EM traces of the 192bit prime field (projective coordinates) elliptic curve point double operation were captured. The DSA, DFA and DEMA differential traces are presented for both correct and incorrect scalar key bit guesses. All differential results presented in this section were obtained from partitioning based upon the most significant bit of the x-coordinate of the input point of the point double operation, unless otherwise stated. Each EM trace of the elliptic curve point double routine contained 25K sample points over 2ms.

Figure 10 shows the previously researched DEMA generated with our EM traces when a correct and incorrect scalar bit is chosen in a) and b) respectively. The three standard deviations (SD) were chosen to encompass the incorrect scalar key bit (shown with green for 3 positive standard deviations and red for the negative 3 standard deviations). The differential signals (in blue) above and below the 3 standard deviations (in green and red) were considered to be significant. Figure 10a), which shows the DEMA for a correct scalar key bit, features multiple significant peaks. The peaks are likely corresponding to the time of finite field computations on the x-coordinate of the input point.

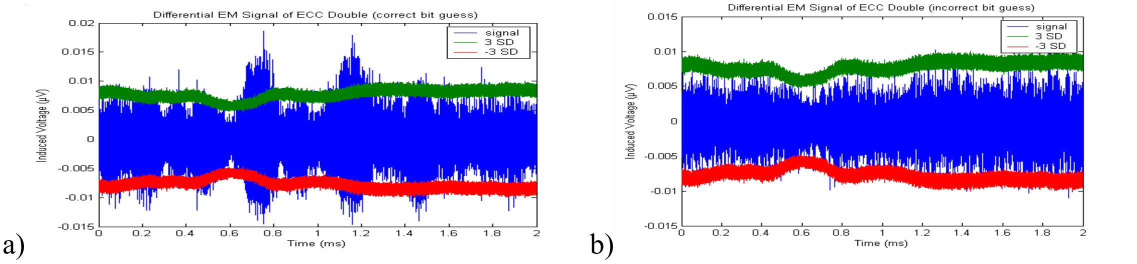

Figure 10. DEMA of ECC double with correct in a) and incorrect scalar key bit guess in b).

Figure 11 is the differential EM spectrogram for ECC double operation with a correct and incorrect scalar bit guess in a) and b) respectively. Again the 3 standard deviations are chosen to encompass the incorrect scalar key bit differential in figure 11b). The standard deviation of difference of means always peaked at 0 frequency, indicating that there is considerable fluctuation of EM signals over different traces. This is indicated in the figures by the long vertical green and red standard deviation lines at the 0.1ms time intervals (0,0.1,0.2, ..etc). Clearly, figure 11a) features many significant peaks above and below the 3 standard deviation curves. Furthermore, peaks in figure 11a) correlate with differential EM peaks in figure 10a), such as those that appear at 0.7ms and 1.1ms. This is expected as the differential EM signal and differential EM spectrogram are simply two different perspectives of looking at the same events unfolding on the PDA device.

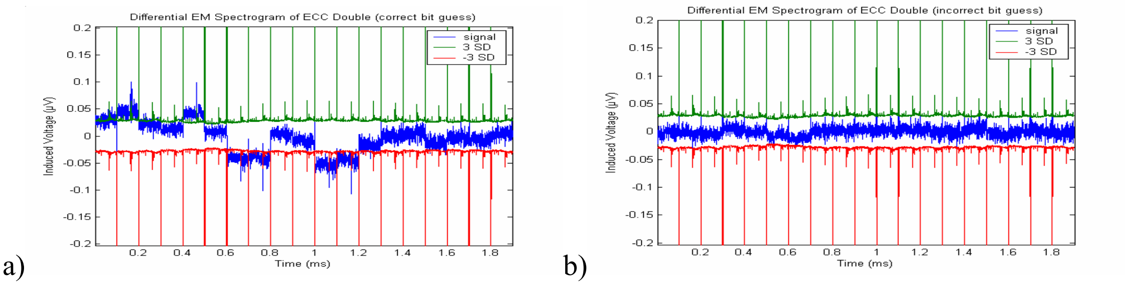

Figure 11. DSA of ECC double with correct in a) and incorrect scalar key bit guess in b).

Figure 12 shows the differential PSD analysis of a short segment of the ECC double routine between 0.6 and 1.2 ms. In Figure 12a) significant differential signals are observed in the spectrogram analysis with the correct bit guess. For example significant differential signals are indicated by the blue peak which extends below the red standard deviation curve at 4MHz and other blue signals also extending below the standard deviation curve at other frequencies. In figure 12b) signals are derived from an incorrect scalar key bit guess and as expected the graph does not feature any significant peaks, since the blue differential signals are encompassed within the green and red standard deviation curves.

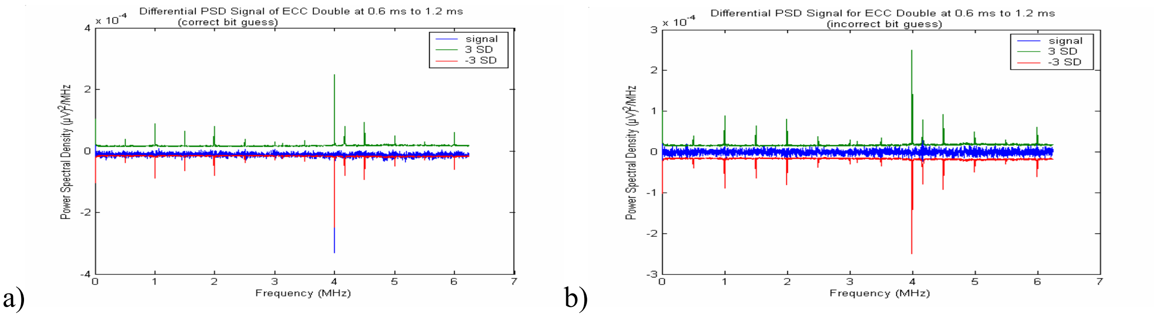

Figure 12. DFA of ECC double with correct in a) and incorrect key bit guess in b) for time period 0.6 to 1.2ms.

Figure 13a) shows the results of performing the differential EM analysis with correct bit partitioning using second most significant bit (MSB), instead of the first most significant bit as used with previous results. Clearly, the amplitude of the peaks in the differential signals diminishes significantly (compared to results using the most significant bit in figure 10a). Results using the third most significant bit with correct scalar key bit partitioning did not reveal any significant differential signals at all. As described in appendix A, this is because the chance of overflow is not as high if the 2nd or 3rd MSB is one, hence the probability of overflow does not correlate as closely to bits other than MSB. This demonstrates the impact of overflow in sub-operations within point doubling on the resulting EM signals. Clearly, MSB of the input coordinates works better than other bits. Figure 13b) shows a similar situation using the spectrogram. The 2nd MSB with correct scalar key bit partitioning in figure 13b) indicates some diminished significance at 0.7 ms and 1.1 ms.

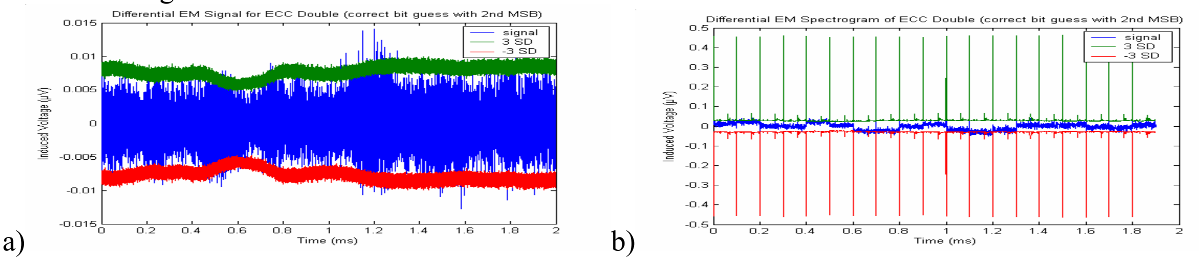

Figure 13. DEMA in a) and DSA in b) for correct scalar key bit partitioning on 2nd MSB

5. Discussions and Conclusions

This study presents for the first time new differential frequency analysis approaches using real EM measurements of a real embedded system, a Java-based wireless PDA. Additionally a countermeasure was also presented suitable for embedded devices. In the Rijndael analysis, the previously researched DEMA was not successful in obtaining the correct key on the PDA, likely since DEMA requires good EM trace alignment for DEMA to work (which is not present on the PDA since it is a larger embedded device running Java with an operating system). If differential spikes are slightly out of alignment in time, they will cancel out rather than reinforce when averaging. The spikes analyzed in DPA or DEMA can be as small as 5 sample points wide, so a misalignment of 1 or 2 sample points can already cause significant loss of information when traces are averaged together. Both DFA and DSA, which are less vulnerable to the effects of time-shift, were successful in obtaining the correct key from the PDA device. A countermeasure is shown to defeat the first order DEMA, DFA and DSA attacks in Rijndael. Similar to [3], the proposed countermeasure has the potential for attack through a 2nd order DEMA, however unlike [3], the proposed countermeasure can increase the security by increasing the number of extra tables. The split mask countermeasure can trade off memory for security, thus requiring a higher *n*th order DPA for (*n-1*) extra tables (*n >=3*). A previously researched 2^{nd} order analysis technique [17], which is also known to be time-shift invariant, was not successful in attacking the countermeasure after using over 2000 EM traces. In this case it is possible that a larger number of EM traces would be required. For the first time an attack of an elliptic curve algorithm was presented using real EM traces. Both the DEMA and DSA attacks were successful on the elliptic curve algorithm only when the 1^{st} and 2^{nd} most significant bits of the elliptic point data were used for partitioning. This attack worked since the MSB's were correlated with EM activity from the overflow or underflow computations (ie. those of modular reduction, etc). This is unlike previous research such as [19] where time alignment of traces was required to correlate any bit of the elliptic point data to the simulated power traces [19]. It is likely that the time misalignment was too large or the number of EM traces too small to successfully apply DSA to attack the data value (or correlate the EM trace with any bit of the elliptic point data as in [19]). It is interesting to note that a possible countermeasure for the MSB differential analysis attack of ECC is to use the same algorithm for finite field computations regardless of whether an overflow occurs or not. As discussed earlier, spectrogram could pinpoint a time segment where there are more significant spikes which is important for determining where the activity of interest is in the traces. Differential analysis using spectrogram may be unsuccessful if the window size and the fraction of window overlap are unsuitably chosen and should be determined experimentally. Similar to previous research [17], the DFA and DSA require a Fast Fourier Transform, however unlike [17], this paper proposes a 1^{st} order analysis where all computations are done in the frequency domain. The extension of DFA and DSA to higher order analysis is the subject of future work, however the 2^{nd} order technique in [17] with over 2000 EM traces was not successful, hence supporting the belief that higher order analysis attacks are very difficult to launch. In summary the new proposed analysis techniques were successful in obtaining the correct key from both Rijndael (a symmetric key encryption standard) as well as elliptic curve cryptography (a public key cryptographic standard). They are general and applicable to other cryptographic algorithms, power as well as EM, and other embedded systems.

Using real EM measurements from a PDA device executing Java-based cryptography, a new frequency-based (time-shift invariant) differential analysis was demonstrated. Previous differential analysis techniques requiring alignment of traces in the time domain were not successful in correlating EM signals to bits of the data. Results show that a low energy countermeasure for Rijndael supporting scalable security without large overheads of table regeneration or excessive storage was able to thwart the new differential techniques, but could not successfully be attacked by higher order techniques [17] with over 2000 EM traces. This research is crucial for supporting low energy security for embedded systems which will be prevalent in wireless embedded devices of the future.

6. References

[1] P.Kocher "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", LNCS 1109, 1996.

[2] S.Ravi, etal. "Securing Wireless Data: System architecture challenges", ISSS, 2002, pp.195-200.

[3] T.Messerges, "Securing the Rijndael finalists against power analysis attacks" LNCS 1978, 2001, pp.150-164.

[4] P.Kocher, J.Jaffe, B.Jun "Differential Power Analysis" Crypto'99, LNCS 1666, 1999

[5] J-J. Quisquater, etal. "a new tool for non-intrusive analysis of smartcards based on EM emissions", Rump Session, Eurocrypt 2000.

[6] Dr.Brian Gladman, "A Specification for Rijndael, the AES Algorithm", at fp.gladman.plus.com/ cryptography_technology/ rijndael/aes.spec.311.pdf,2003.

[7] D.Agrawal et al. "The EM side-channel(s)" CHES 2002, 2002, pp.29-45.

[8] K.Gandolfi etal. "Electromagnetic Analysis: concrete results" CHES 2001, LNCS 2162, pp.251-261.

[9] S.Chari, etal. “Towards sound approaches to counteract power-analysis attacks”, LNCS 1666, 1999, pp.398-412.

[10] D.Agrawal, etal. “The EM side-channel…methodologies” at http ://www.research.ibm.com/intsec/emf.html

[11] W.Liao etal. “leakage power modeling and reduction with data retention”, IEEE ICCAD, 2002, pp. 714-719.

[12] M.Akkar, etal. “Power analysis, what is now possible…”, LNCS 1976, 2000, pp489-502.

[13] K.Itoh etal. “DPA countermeasure based on the masking method”, LNCS 2288, 2002,pp.440-456.

[14] L.Goubin, J.Patarin “DES and Differential power analysis- the duplication method” CHES 2001.

[15] J.Golic “Multiplicative Masking and power analysis of Rijndael”, CHES 2002.

[16] T.Messerges “Using 2nd order power analysis to attack DPA resistant software”, LNCS 1965, 2000, pp.238-251.

[17] J.Waddle, D.Wagner “Towards efficient second-order power analysis” CHES 2004, LNCS 3156, pp.1-15.

[18] E. Brier and M. Joye, “Weierstraβ Elliptic Curves and Side-Channel Attacks", *PKC 2002*, LNCS 2274, pp. 335-345, Springer-Verlag, 2002.

[19]J. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems”, *CHES 1999*, LNCS 1717, pp. 292-302, 1999.

[20]T. Izu, B. Moller, and T. Takagi, “Improved Elliptic Curve Multiplication Methods Resistant against Side Channel Attacks”, *Indocrypt 2002*, LNCS 2551, pp. 296–313, Springer-Verlag, 2002.

[21] T. Izu and T. Takagi, “A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks", Technical Report CORR 2002-03, University of Waterloo, 2002. Available from http://www.cacr.math.uwaterloo.ca/.

[22] T. Izu and T. Takagi, “On the Security of Brier-Joye's Addition Formula for Weierstrass-form Elliptic Curves", TR No. TI-3/02, Technische University Darmstadt, 2002. Available from http://www.informatik.tu-darmstadt.de/TI/.

[23] K. Itoh, J. Yajima, M. Takenaka, and N. Torii, \DPA Countermeasures by improving the Window Method", *CHES 2002*, LNCS 2523, p. 303 ff, 2002.

[24] M. Joye and J. Quisquater, “Hessian elliptic curves and side-channel attacks”, *CHES 2001*, LNCS 2162, pp. 402-410, 2001.

[25] M. Joye and C. Tymen, “Protections against differential analysis for elliptic curve cryptography”, *CHES 2001*, LNCS 2162, pp. 377-390, 2001.

[26] E. Oswald, M. Aigner, “Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks”, CHES 2001, LNCS 2162, 2001

[27] National Institute of Standards and Technology, Digital Signature Standard, FIPS Publication 186-2, February 2000.

Appendix A. Analysis of Correlation between MSB and Overflow

An execution of the finite field operation varies when arithmetic overflow occurs. For example, when overflow occurs during a modular addition, the result must then be subtracted by the field defining polynomial *p*. The extra field subtraction operation would produce distinguishable signal in the EM side channel. Similarly, when underflow occurs during a modular subtraction, the result must be added by the polynomial *p*. Again, the extra addition operation would produce distinguishable EM signals. The coordinates of the ECC points are finite field elements. In the EC point operation, the *x*-coordinate of the input is directly fed as input to three finite field operations: scalar multiplication by 4, modular addition, and modular subtraction. Scalar multiplication is implemented by bitwise shift to the left. It is simple to see that in scalar multiplication by 4, overflow occurs when either one of the two most significant bits of the field element is one. The overflow can be observed in the EM side channel. This partially explains why when the EM traces are correctly partitioned by either the MSB or the 2nd MSB, significant peaks are observed in the differential signal.

The MSB of the *x*-coordinate also changes the probability of overflow in the addition operation. When MSB is 1, the probability of overflow in addition operation between the *x*-coordinate and a random finite field element is approximately ¾. However, when MSB is 0, the probability becomes approximately ¼. To show that this is true, one can observe that there are *n* different values of the random field elements that can cause overflow when the *x*-coordinate is equal to *n*. In the case when MSB is set to 0, since the range of the *x*-coordinate is from 2^{191}-1 to 0, the number of ways the addition operation can overflow is the sum of the arithmetic series from 2^{191}-1 to 0. In the case when MSB is set to 1, since the range of the *x*-coordinate is from 2^{192}-1 to 2^{191}, the number of ways the addition operation would overflow is the sum of the arithmetic series from 2^{192}-1 to *p*. The total number of different combinations of *x*-coordinates (with MSB sets to a value) and random finite field elements is $p^2/2$. For simplicity, the calculations below assume *p* is approximately equaled to 2^{192}.

when MSB = 0

$$P(overflow) \approx \frac{1}{2^{383}} \times \sum_{n=0}^{2^{191}-1} n$$

$$P(overflow) \approx \frac{1}{2^{383}} \times \frac{(2^{191}-1)(2^{191})}{2}$$

$$P(overflow) \approx \frac{1}{2^{383}} \times \frac{(2^{191})(2^{191})}{2}$$

$$P(overflow) \approx \frac{1}{4}$$

when MSB = 1

$$P(overflow) \approx \frac{1}{2^{383}} \times \sum_{n=2^{191}}^{2^{192}-1} n$$

$$P(overflow) \approx \frac{1}{2^{383}} \times \frac{(2^{192}-1+2^{191})(2^{192}-1-2^{191}+1)}{2}$$

$$P(overflow) \approx \frac{1}{2^{383}} \times \frac{(2^{192}+2^{191})(2^{192}-2^{191})}{2}$$

$$P(overflow) \approx \frac{1}{2^{383}} \times \frac{2^{382} \times (2+1)(2-1)}{2}$$

$$P(overflow) \approx \frac{3}{4}$$

Using a similar analysis, one can show that MSB of the *x* coordinate changes the probability of underflow in subtraction operation. When MSB is 1, the probability of underflow in addition operation between the *x*-coordinate and a random finite field element is ¼. However, when MSB is 0, the probability becomes ¾.

In summary, since the MSB value can both exactly and probabilistically cause modulo reduction in finite field operations, it correlates with the signals in the EM side channel.

Appendix B. Split Mask Countermeasure

This section will describe a proposed countermeasure which will thwart the first order analysis attack of the previous section on some symmetric key encryption algorithms, such as Rijndael, DES, CAST. The proposed split mask countermeasure stores randomly masked data in the S-box (masked S-box table, $S'[x]$). Unlike previous research [13,14], each addressed data in the table uses a different random mask ($S'[x] = S[x]+r[x]$, where + is exclusive or operation , $r[x]$= random mask, which is different for each table address, x). A second corresponding table (called the mask table, $M[x]=m+r[x]$, m is a fixed random value) is used to store a corresponding mask for each address. Since tables are generated only once, this value m along with mask of the round keys is used to precompute tables before the cryptographic algorithm is downloaded to the device. To avoid a first order DPA, the exclusive or of the S-box masked table and mask table, $S'[x] +M[x]=m+S[x]$, is never computed during the execution of the encryption algorithm. Figure A.1 illustrates the computations performed on the mask tables before they are merged with the masked substitution tables to avoid a 1st order DPA, DEMA or DFA.

For example, in the table method of Rijndael (described in [6] for fast implementation on 32bit processors), all masked S-box tables are accessed and their results are xor'd together (as would normally be done for unmasked S-box tables in the original algorithm). Next all corresponding mask table outputs are xor'd together. Finally the xor result from the S-box masked tables is exclusive or'd with the xor result of the mask tables. Figure A.1 illustrates the computations required in the scheme for Rijndael, where $S1', S2', S3', S4'$ and M are the masked S-box tables and the mask table, respectively. Note that there is only one mask table, M, which is accessed four times (so it is shown four times) in figure A.1. In the table method of Rijndael one would normally compute : $t0 = S1(\{s0\}_{b3}) + S2(\{s1\}_{b2}) + S3(\{s2\}_{b1}) + S4(\{s3\}_{b0}) + rki$. However with this countermeasure one would compute : $t0a = S1'(\{s0\}_{b3}) + S2'(\{s1\}_{b2}) + S3'(\{s2\}_{b1}) + S4'(\{s3\}_{b0})$ and $t0b = M(\{s0\}_{b3}) + M(\{s1\}_{b2}) + M(\{s2\}_{b1}) + M(\{s3\}_{b0})$ (where $\{w\}_b$ refers to byte b of the 32bit word w). Then one would merge them as : $t0 = t0a+ t0b$ *and* $t0=t0 + rki$ (where $t0$ value is then combined with the masked round key for input to the next set of tables, which may also have masked inputs). A similar implementation for DES and other cryptographic algorithms is also possible.

Similar to previously researched countermeasures [3,16], a 2nd order DEMA attack on this countermeasure may be possible. However a 1st order DEMA is thwarted since the data values output from the S-box tables have been decorrelated through random masking. The 2nd order DEMA could use statistical processing of EM samples of both the output of the S-box masked table and the output of the mask table to launch an attack. However unlike previous countermeasures[3,16], by increasing the number of tables, an increase in the order of the required DEMA attack occurs, hence the security of this countermeasure scales with the number of tables. For n mask tables ($M1, M2\ldots , Mn$), and one S-box masked table, (S') a $(n+1)^{th}$ order DEMA attack is required (where $m= r [x]+M1[x] +M2[x]\ldots +Mn[x]$, for all x , thus splitting mask m into $n+1$ masks). For example with 2 extra mask tables ($M1,M2$ where $m=r[x]+M1[x]+M2[x]$, for all x), a 3rd order DEMA may possibly be launched. The higher order attack typically will require many more EM traces and thus provides an increase in difficulty of launching the attack. Note that once again, even with n tables, to avoid a first order DEMA, the exclusive or of the S-box masked table and mask table, specifically $S'[x]+ M1[x] +M2[x]\ldots +Mn[x]=m+S[x]$ is never computed.

The proposed split mask countermeasure is similar to the duplication method [14] however unlike [14] the second table does not hold the random mask of the S-box entry, $r[x]$. The duplication method [14] also used two tables whereas this countermeasure can increase the number of tables supporting higher order DPA. Additionally these masked tables and set of mask tables would produce a masked value unlike [14] which would produce an unmasked value. Previous research [3,13] used a constant mask, for all data in the table and when a new random masking is required, the complete set of S-box tables were regenerated[3] to utilize the next different random value mask.

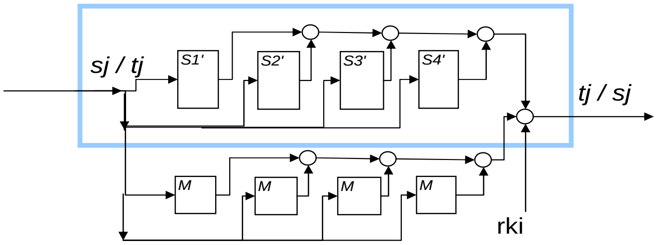

Figure B.1. Partial Rijndael implementation of proposed countermeasure

Although this countermeasure is applicable to symmetric key cryptography, it has too large an overhead for application to ECC. In ECC this countermeasure which would replace the sliding window technique with randomized window look up tables, does not require table regeneration unlike point randomization. However it does have a larger overhead of requiring extra point doubles and point summations on the mask table output. Hence it is not directly suitable for ECC in embedded systems, even if t-adic formulations are used.