Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi: 21.10.2022
Makale Kabul Tarihi: 09.12.2022

S-kutusu Büyüklüğünün Korelasyon Güç Analizi Sonuçlarına Etkisi
Impact of S-box Size on Results of Correlation Power Analysis

Selçuk Kavut[1], *Yasin Reşit Yargıcı*[2]
[1]Bilgisayar Mühendisliği Bölümü, Balıkesir Üniversitesi, Balıkesir, Türkiye
skavut@balikesir.edu.tr
[2]İşbir Sentetik Dokuma Sanayi A.Ş., Balıkesir, Türkiye
ysn.yargici@gmail.com

Öz

Simetrik bir kripto-sistemde küçük S-kutularının kullanımı, gömüldüğü donanımın güç tüketimini azaltmaktadır. Bu durumun yan kanal analizi (YKA) sonuçlarında gürültünün bozucu etkisini arttırdığı bilinmektedir. Bu çalışmamızda, bahsedilen etkiyi deneysel olarak doğrulamak için, 4×4 S-kutularına sahip hafif sıklet blok şifreleme algoritması PRESENT, SAKURA-X kriptografik donanımı üzerinde gerçeklenmiş ve en etkili YKA yöntemlerinden olan korelasyon güç analizi (KGA) yapılmıştır. Bunun sonucunda, ölçüm düzeneğimiz vasıtasıyla alınan güç ölçümlerinde oluşan gürültünün doğru anahtar tespitini zorlaştırdığı görülmüştür. Ayrıca, PRESENT için gürültülü güç ölçümlerinin benzetimi ile KGA yürütüldüğünde, doğru anahtarın başarılı bir şekilde elde edildiği gözlenmiştir. Diğer taraftan, S-kutuları 8×8 AES S-kutusu ile değiştirilmiş PRESENT için aynı ölçüm düzeneğiyle KGA uyguladığımızda, güç tüketimi artışına paralel olarak, gürültü etkisinin daha az olduğu ve doğru anahtarın daha kolay elde edildiği gösterilmiştir.
Anahtar Kelimeler: AES, PRESENT, SAKURA-X, Korelasyon Güç Analizi (KGA)

Abstract

Use of small S-boxes in a symmetric crypto-system reduces power consumption of its embedded hardware. It is known that this increases the adverse effect of noise on the results of side channel analysis (SCA). Here, to verify the mentioned effect experimentally, the lightweight block cipher PRESENT having 4×4 S-boxes is implemented on the cryptographic hardware SAKURA-X and correlation power analysis (CPA), one of the most powerful methods of SCA, is realized. Consequently, we find that the noise occurring within the power traces obtained by our measurement setup makes it difficult to identify the correct key. Further, when we apply CPA to PRESENT by simulating the noisy power traces, we get the correct key successfully. On the other hand, applying CPA, with the same measurement setup, to PRESENT in which the S-boxes are replaced with the AES S-box, we show that, parallel to the increase in power consumption, the noise effect is lesser and it is easier to find the correct key.
Key Words: AES, PRESENT, SAKURA-X, Correlation Power Analysis (CPA)

1. Giriş

Günümüzde, oldukça kısıtlı donanım kapasitesine sahip olan radyo frekansı ile tanımlama (RFID) etiketleri, sensör ağları, temassız akıllı kartlar ve nesnelerin interneti (IoT) cihazları gibi elektronik sistemlerde verinin korunmasını amaçlayan hafif sıklet kriptografi, simetrik kriptografinin artan bir ilgi odağı haline gelmiştir. Simetrik kripto-sistem tasarımında, n biti m bite gönderen bir fonksiyon olarak tanımlanan S-kutusu (yerleştirme kutusu), kullanıldığı kripto-sistemin güvenliği ve verimliliği açısından önemli rol oynayan temel yapı taşıdır. S-kutularının büyüklüğü, genellikle kullanılan mikroçip alanı ve güç tüketimini etkileyen en önemli faktördür. Çalışmamızda ele aldığımız PRESENT algoritması [1], hafif sıklet şifreleme gerektiren uygulamalar için ISO/IEC 29192-2:2012 standardına [2] göre önerilen bir blok şifredir. Donanımsal açıdan eniyilenmiş bir algoritma olan PRESENT, 4×4 büyüklüğünde S-kutuları kullanmaktadır ve örneğin, 8×8 S-kutularına sahip ileri şifreleme standardı AES ile karşılaştırıldığında 2.5 kat daha küçük mikroçip alanında gerçeklenebilmektedir [1].

1996'da Kocher'in değişen şifreleme zamanlarının gizli anahtar hakkında bilgi sızdırdığını gösteren çalışması [3] ve 1999'da Kocher vd.'nin kriptografik cihazın güç tüketimine bağlı olarak gerçekleştirdikleri kriptoanaliz [4], yan kanal analizi (YKA) literatürünün gelişimine katkı sağlayan öncü çalışmalardır. Genel olarak, kriptografik donanımların kullanılan şifreleme algoritması ve anahtara bağlı olarak dışarıya sızdırdıkları zamanlama, ses, güç tüketimi ve elektromanyetik radyasyon gibi yan kanal bilgilerinden faydalanılarak yapılan kriptoanalize YKA denilmektedir. Örneğin, kriptografik cihazda yürütülen bir blok şifrenin tur işlemi esnasında tükettiği gücün, S-kutusu çıkışındaki bitlerin Hamming ağırlığına (veya giriş ve çıkış bitleri arasındaki Hamming uzaklığına) bağlı olduğunu varsayarsak, ortalamaların uzaklığı, karşılıklı bilgi veya Pearson korelasyonu gibi bir istatistiksel ayırıcı vasıtasıyla doğru korelasyon bulunarak gizli anahtar ortaya çıkarılabilir. Bahsedilen ayırıcıların kullanıldığı YKA yöntemleri (veya saldırı türleri) literatürde sırasıyla, farksal güç analizi (FGA) [4], karşılıklı bilgi analizi (KBA) [5] ve korelasyon güç analizi (KGA) [6] olarak bilinmektedir. Güç analizi sınıfına giren bu saldırı türleri, kriptografik donanıma zarar vermeden ve az maliyetle gerçekleştirilebilmektedirler. FGA saldırısı, Lo vd. [7] tarafından AES algoritması için gerçekleştirilen güç analizinin sonuçları ile deneysel olarak gösterildiği gibi, KGA

ile karşılaştırıldığında gürültü ve girişimden daha fazla etkilenmekte ve yanlış anahtar tahminlerinin artmasına neden olmaktadır. Bununla birlikte, çoklu bitin kullanıldığı FGA saldırısının, Hamming ağırlığı modelinin uygulandığı KGA saldırısı ile eşdeğer olduğu Dogett vd. [8] tarafından ıspatlanmıştır. Moradi vd.'nin [9], KBA saldırısını KGA ile karşılaştırdığı çalışmasında ise, KGA'nın daha verimli olduğu teorik olarak ıspatlanmış ve deneysel olarak doğrulanmıştır. Diğer taraftan, donanımsal açıdan maliyet artışına neden olmakla birlikte, güç analizi saldırılarına karşı koymak için kriptografik algoritmanın hesapladığı ara değerlerin rastgeleleştirilmesine dayanan maskeleme (gizlilik paylaşımı) [10, 11], gürültü eklenerek sinyal-gürültü (SNR) oranının azaltılması [12] veya yürütülen işlem sırasının/program akışının rastgeleleştirilmesi [13] gibi önlemler literatürde bilinen yöntemlerdendir.

Aynı algoritma kullanılsa bile kriptografik donanıma bağlı olarak farklı biçim ve değerlerde yan kanal bilgisi ortaya çıkabildiğinden, YKA gerçekleştirildiği donanıma özeldir. Bununla birlikte, gerektirdiği veri karmaşıklığı ve işlemci gücü açısından pratikte yürütülmesi genellikle mümkün olmayan farksal [14] ve doğrusal [15] kriptoanaliz gibi şifreleme algoritmasının sadece matematiksel tanımından faydalanan kriptoanaliz türleri ile karşılaştırıldığında, YKA çok daha verimli şekilde kullanılan anahtarı ortaya çıkarabilmektedir.

Aşağıdaki bölümde literatürde yapılan önceki çalışmalar özetlenmekte ve bir sonraki bölümde PRESENT algoritması tanıtılmaktadır. Bölüm 4'te sunulan ölçüm düzeneği ile, Bölüm 5'te açıklanan KGA için gerekli güç ölçümlerinin nasıl elde edildiği açıklanmaktadır. Bölüm 6'da gerçekleştirdiğimiz KGA sonucu ortaya çıkan bulgular tartışılmakta ve bir sonraki sonuç bölümü ile makalemiz sonlanmaktadır.

2. Önceki çalışmalar

Yan kanal bilgisi vasıtasıyla hedeflenen şifreleme algoritmasının aşırı tanımlanmış bir denklem sistemi biçiminde temsil edilmesine dayanan bir saldırı türü olan cebirsel YKA [16], literatürde bilinen güçlü kriptoanaliz yöntemlerinden birisidir. PRESENT algoritması için gerçekleştirilen ilk YKA saldırıları [17, 18], kriptografik algoritmanın belirli çıktılarının düşük dereceli bir polinom biçiminde temsil edilmesiyle gerçekleştirilen ve küp saldırıları olarak bilinen bu türden saldırılardır. Daha yakın zamanda, Duan vd. [19] PRESENT algoritmasının ATmega 16 geliştirme kartı üzerinde gerçeklemesine FGA saldırısı uygulayarak, rastgele değerlerle maskelemeye göre daha hızlı olan ve daha az bellek gerektiren sabit değerle maskelemenin, algoritmayı FGA saldırısına karşı dayanıklı hale getirilebildiğini deneysel olarak göstermişlerdir. Sadhukhan vd. [20], PRESENT dâhil olmak üzere bazı hafif sıklet blok şifrelerin hem maskesiz hem de (eşik gerçeklemesi olarak bilinen) maskeli versiyonlarını performans ve verimlilik açısından uygulamaya özgü tümleşik devre (ASIC) ve alanda programlanabilir kapı dizileri (FPGA) platformlarında AES ile karşılaştırmışlardır.

Bir blok şifrenin YKA saldırılarına karşı en hassas bileşeninin S-kutuları olduğu bilinmektedir. Goubin vd. [21], 8×8 büyüklüğündeki AES S-kutularını 4×4 ve 16×16 büyüklüğündeki S-kutuları ile değiştirerek, S-kutusu büyüklüğünün maskeleme durumunda yan kanal analizine etkisini çalışmışlar ve 16×16 büyüklüğündeki S-kutularının daha iyi güvenlik sağladığını göstermişlerdir. Diğer taraftan, Carlet vd. [22] tarafından bir S-kutusunun YKA karşısındaki (maskeleme yapılmaksızın) içsel dayanıklılığı bilgi teorisi açısından çalışılmış ve öz-korelasyon spektrumu ile ilişkisi ortaya konulmuştur. Elde edilen bu ilişki, aynı varyansa sahip bağımsız toplanır Gauss gürültüsü varsayımı altında genel olarak küçük S-kutularının YKA karşısında daha güçlü olduğunu teorik olarak göstermektedir. Heuser vd.'nin [23] 4×4 S-kutularına sahip hafif sıklet blok şifrelerin dayanıklılığını 8×8 S-kutuları kullanan blok şifreler ile (güç ölçümlerinin benzetimi vasıtasıyla) başarı oranı metriğine göre karşılaştırdığı çalışması, [22] çalışmasında gösterilen teorik sonucu doğrular niteliktedir. Fiziksel donanım üzerinde gerçekleştirme göz önüne alındığında, PRESENT algoritmasının KGA saldırısı karşısında dayanıklılığı ilk olarak Zhang vd. [24] tarafından ASIC üzerinde donanım gerçeklemesi hedef alınarak çalışılmıştır. Bu çalışmanın daha pratik hale getirildiği Wang vd.'nin [25] çalışmasında, Hamming uzaklığı modeline göre güç ölçümü tahmini yapmak için kullanılan bit sayısının artırılması ile, daha az sayıda güç ölçümünün anahtarı ortaya çıkarabildiği gösterilmiştir. PRESENT algoritmasının Arduino Uno üzerinde yazılım gerçeklemesine KGA saldırısı uygulayan Lo vd. [26], aynı çalışmada güç ölçümü tahmininde S-kutusu kullanımının tur anahtarı ekleme fonksiyonuna göre daha iyi sonuç verdiğini göstermişlerdir. Yakın zamanda, PRESENT algoritmasının ASIC gerçeklemesini hedef alan Fang ve Alioto [27], son tura kapalı metinin kullanılmasına dayanan bir KGA saldırısı gerçekleştirerek, hem ilk hem de son turun korelasyon matrislerinin kullanıldığı birleştirilmiş bir KGA saldırısı ile anahtarın daha verimli şekilde tespit edilebildiğini gözlemlemişlerdir. Ayrıca, Gunathilake vd. [28] tarafından PRESENT algoritmasının Arduino Uno üzerinde yazılım gerçeklemesine ilk defa korelasyon elektromanyetik analizi uygulanmıştır. İlgili literatür ile karşılaştırıldığında, bu çalışmada ele aldığımız PRESENT algoritmasının büyüklüğü değiştirilmiş S-kutularına sahip bir versiyonunun herhangi bir donanım gerçeklemesi ve bunun KGA sonuçlarına etkisi bildiğimiz kadarıyla literatürde daha önce incelenmemiştir. Elde ettiğimiz sonuçlar, Carlet vd.'nin [22] teorik çıkarımını ile Heuser vd.'nin [23] benzetim sonuçlarını doğrulamakta ve bununla birlikte nicemleme hatasından dolayı PRESENT algoritmasında kullanılan anahtarın tespit edilemeyebileceğini göstermektedir.

3. PRESENT Algoritması

PRESENT algoritması 31 turluk bir blok şifredir. Bu algoritmada 64-bit uzunluktaki açık veriler, 80 veya 128 bitlik anahtar ile şifrelenerek kapalı veriler elde edilmektedir. Çalışmamızda, 80 bitlik anahtar ile gerçekleme yapılmıştır. PRESENT algoritmasının sözde kodu Şekil 1'de gösterilmiştir.

```
for (i = 1; i <= 31; i ++)
{
    addRoundKey(Blok, K_i);
    sBoxLayer(Blok);          } i. tur
    pLayer(Blok);
}
addRoundKey(Blok, K_32);
```

Şekil 1. PRESENT algoritmasının sözde kodu.

Yukardaki sözde koddan görüldüğü gibi, şifrelemenin her bir turu üç aşamada gerçekleştirilmektedir. Bu aşamalar sırasıyla tur anahtarı ekleme (addRoundKey), yer değiştirme (sBoxLayer) ve permütasyon (pLayer) aşamalarıdır. İlk aşamada tur girdisi olan 64-bit uzunluğundaki blok, PRESENT'in anahtar üretme algoritması tarafından üretilen aynı uzunluktaki tur anahtarı K_i (i = 1, 2, …, 32) ile XOR işlemine tabi tutulur. Şekil 1'de *Blok* ile gösterilen tur girişi veya son tur çıkışı $(b_0, b_1, \dots, b_{63})$, üretilen i. tur anahtarı $1 \leq i$

≤ 32 için $K_i = (k_0^i, k_1^i, \dots, k_{63}^i)$ olmak üzere, addRoundKey aşağıdaki işlemi gerçekleştirir:

$$(b_0, b_1, \dots, b_{63}) \leftarrow (b_0, b_1, \dots, b_{63}) \oplus (k_0^i, k_1^i, \dots, k_{63}^i). \quad (1)$$

Bir sonraki sBoxLayer aşamasında XOR işleminden çıkan 64 bitlik blok, her biri 4-bit uzunluğunda olan 16 alt bloğa bölünür ve bölünen bu alt bloklardaki her 4-bit (Tablo 1'de onaltılık tabanda verilen) S-kutusundan geçirilerek sBoxLayer aşamasının çıktısı elde edilir. Diğer bir ifadeyle, $0 \leq i \leq 15$ için $\omega_i = (b_{4i}, b_{4i+1}, b_{4i+2}, b_{4i+3})$ olarak tanımlanırsa, 16 alt blok $(\omega_0, \omega_1, \dots, \omega_{15})$ ile gösterilebilir; bu durumda, sBoxLayer aşağıdaki işlemi gerçekleştirir:

$$(b_0, b_1, \dots, b_{63}) \leftarrow \left(S^b(\omega_0^h), S^b(\omega_1^h), \dots, S^b(\omega_{15}^h)\right), \quad (2)$$

burada, en soldaki bit en değerli bit olmak üzere, ω_i^h ve $S^b(\omega_i^h)$ sırasıyla ω_i'nin onaltılık tabandaki ve $S(\omega_i^h)$'nin ikili tabandaki karşılıklarını temsil etmektedirler.

Tablo 1. PRESENT algoritmasının 4-bit S-kutusu.

x	0	1	2	3	4	5	6	7
$S(x)$	C	5	6	B	9	0	A	D

x	8	9	A	B	C	D	E	F
$S(x)$	3	E	F	8	4	7	1	2

pLayer aşamasında ise, her $0 \leq i \leq 15$ ve $0 \leq j \leq 3$ değerleri için, permütasyon sonrasında $i + 16j$ pozisyonundaki bit, permütasyon öncesi $4i + j$ pozisyonundaki bit ile aynı olacak şekilde, eşitlik (2) ile elde edilen bloğun bitleri permütasyona uğrar.

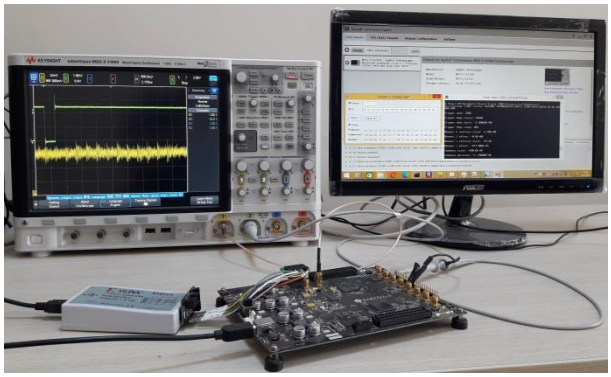
Bahsedilen üç aşamanın oluşturduğu tur, 31 kere döngüsel biçimde gerçekleştirildikten sonra, döngü çıkışı Şekil 1'de görüldüğü gibi anahtar algoritmasının ürettiği 32. tur anahtarı (K_{32}) ile XOR edilir ve kapalı metin elde edilmiş olur.

4. Ölçüm düzeneği

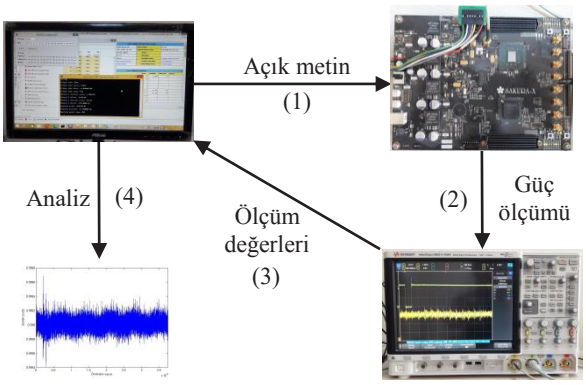
Yan kanal analizini yürütmek için gereken güç ölçümlerini almak amacıyla kullandığımız SASEBO-GIII (yan kanal analizi standart değerlendirme kartı) olarak da bilinen SAKURA-X [29], Keysight MSO-X 4104A osiloskop ve bilgisayardan oluşan ölçüm düzeneğimizin görseli, kullanılan cihazların döngüsel olarak tekrar eden çalışma sırası ile birlikte Şekil 2'de gösterilmiştir.

Çalışma döngüsüne girmeden önce, bilgisayar tarafından ölçüm parametreleri (kanal ayarları, tetikleme modu gibi) osiloskopa ve şifrelemede kullanılacak olan gizli anahtar da SAKURA-X kriptografik kartına gönderilir. Gönderilen bu başlangıç parametrelerinden sonra, Şekil 2'de görüldüğü gibi, öncelikle rastgele üretilen bir açık metin bilgisayardan SAKURA-X kartına gönderilir ve şifreleme işlemi hem bilgisayarda hem de kriptografik kartta gerçekleştirilir. SAKURA-X kartında şifreleme gerçekleştiği esnada oluşan güç tüketimi, osiloskop vasıtasıyla ölçülür ve ölçüm değerleri bilgisayara kaydedilir. SAKURA-X kartında şifreleme sonucu elde edilen kapalı metin aynı zamanda bilgisayara gönderilerek, bağımsız olarak bilgisayarın da hesapladığı kapalı metin ile karşılaştırılır. Bu işlem döngüsel olarak kaç adet güç ölçümü isteniyorsa o kadar tekrar edilir.

Ölçüm düzeneğinde kullanılan SAKURA-X kartı Spartan-6 (kontrol FPGA'i) ve Kintex-7 (kriptografik FPGA) olmak üzere iki FPGA içermektedir. Kintex-7 kriptografik algoritmayı gerçeklemek için kullanılırken, Spartan-6 kriptografik FPGA ile haberleşerek konfigürasyon kontrolünü sağlamaktadır [30]. Deney düzeneğinde, SAKURA-X kartından osiloskop ile ölçüm almak için 50 Ω değerindeki SMA-BNC kablo kullanılmaktadır. Bu ölçüm değerleri osiloskop ile 2.5GS/s örnekleme hızında örneklenmekte ve bilgisayara da bu şekilde kaydedilmektedir. Döngüsel olarak çalışan ölçüm düzeneğinden elde edilen güç ölçümlerine, (aşağıdaki bölümde ele alınan) Hamming uzaklığına dayalı KGA uygulanmakta ve (bir sonraki bölümde) bulunan sonuçlar saldırının başarısını ölçen bir metrik olan tahmin entropisi [31] ile değerlendirilmektedir.



(a)

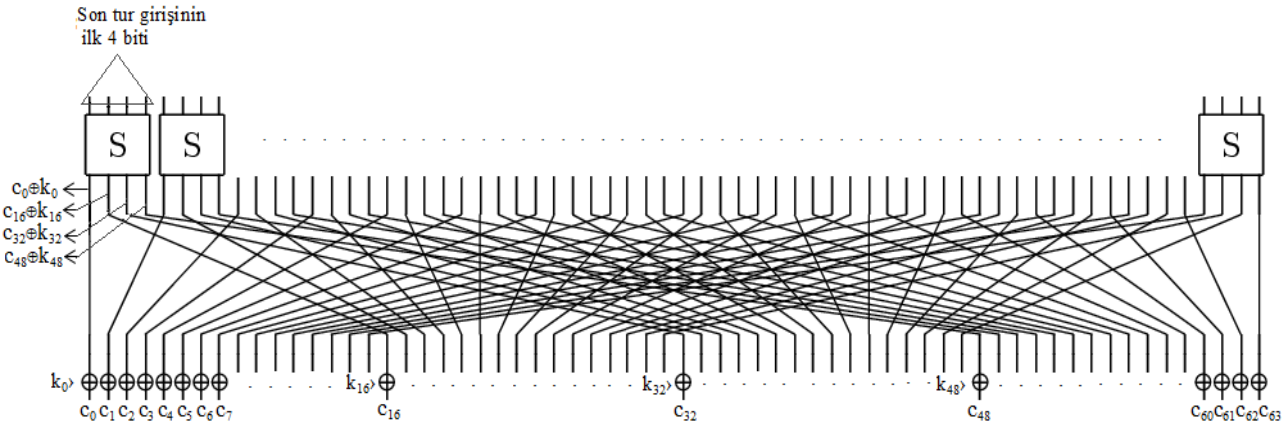


(b)

Şekil 2. Ölçüm düzeneği (a) görseli, (b) çalışma döngüsü.

5. Korelasyon güç analizi

Kriptografik bir cihaz, şifreleme algoritmasını çalıştırırken algoritmanın kullandığı veriye bağlı olarak güç tüketir. Güç analizi ile yürütülen bir saldırıda, bu güç tüketimleri ölçülerek cihazın kullandığı gizli veri (anahtar) elde edilmeye çalışılır. Özel olarak, en güçlü YKA yöntemlerinden birisi olan KGA saldırısında, güç ölçüm değerleri Hamming ağırlığı veya Hamming uzaklığı gibi güç modelleri kullanılarak tahmin edilir ve bu tahminin gerçek güç ölçümü ile istatistiksel ilişkisine bakılarak kriptografik donanımda kullanılan anahtar ortaya çıkarılmaya çalışılır. Çalışmamızda, Hamming ağırlığı modeline göre daha verimli olan Hamming uzaklığı modeline dayalı KGA yürütülmüştür. Hamming uzaklığı modeli genel olarak herhangi bir şifreleme algoritmasının bir turu gerçekleştiğinde harcanan gücün, o turun girişindeki ve çıkışındaki blokların Hamming uzaklığı ile orantılı olduğu varsayımına dayanır. KGA saldırısı uygulanırken, kriptografik



Şekil 3. Son tur anahtarı için Hamming uzaklık modelinin uygulanması.

donanımdan elde edilen gerçek güç ölçümleri ile bu donanımda işlenen veriden elde edilen güç tüketim tahminleri arasındaki ilişkinin doğrusal olduğu kabul edilir. Bu nedenle, eğer güç tüketim tahminini elde etmek için kullanılan aday anahtar doğru anahtar ise bahsedilen korelasyon en yüksek olur.

PRESENT blok şifreleme algoritmasının son turda kullanılan tur anahtarını elde etmeye çalıştığımızı düşünelim. Hamming uzaklığı ile güç ölçümü tahminini gerçekleştirebilmek için, tur girişini bilmemiz gerekir. Ancak tur girişi kullanılan tur anahtarına bağlı olduğu için, bunu sadece tur anahtarını tahmin ederek yapabiliriz. Bu tahminler arasında doğru tur anahtarı, yukarıda bahsedildiği gibi, gerçek güç ölçümleri ile kullanılan model arasındaki istatiksel ilişkiye bakılarak bulunmaktadır.

Şekil 3'te gösterildiği gibi, son tur çıkışı olan kapalı metnin ilk dört bitini kullanarak tur anahtarının dört bitini bulmayı hedeflediğimizi varsayalım (saldırının gerçekleşebilmesi için kapalı metini biliyor olmamız gerekmektedir). Kapalı metini $C = (c_0, c_1, c_2, \dots, c_{63})$ ile ve son tur anahtarını $K = (k_0, k_1, k_2, \dots, k_{63})$ ile gösterelim. Kapalı metine pLayer permütasyonunun tersi uygulanarak elde edilen bloğun ilk dört biti $(c_0, c_{16}, c_{32}, c_{48})$ olur. Bu dört bite, karşılık gelen tur anahtarının dört biti $(k_0, k_{16}, k_{32}, k_{48})$ XOR işlemi ile eklendiğinde, elde edilen $(c_0 \oplus k_0, c_{16} \oplus k_{16}, c_{32} \oplus k_{32}, c_{48} \oplus k_{48})$ sonucunu veren S-kutusu girişi (S-kutusu dönüşümünün tersi uygulanarak elde edilebilir) son tur girişinin ilk dört bitini verir. Burada bahsedilen dört bitlik tur anahtarını bilmediğimiz için olası bütün adaylar (2^4 = 16 tane) tahmin edilir. Bu ise, son tur girişinin ilk dört biti için 16 tane tahmin yaptığımız anlamına gelmektedir. Son tur çıkışını (kapalı metini) bildiğimizi varsaydığımızdan, yapılan tahminlerin her birinin kapalı metnin ilk dört biti olan (c_0, c_1, c_2, c_3) vektörüne Hamming uzaklığı gerçek güç tüketimi için yapılan bir tahmin olmaktadır. Uyguladığımız Hamming uzaklığına dayalı KGA saldırısına göre, bu tahminlerden gerçek güç tüketimleri ile korelasyonu en yüksek olan aday, kullanılan gerçek dört bitlik tur anahtarı olarak elde edilir.

Yan kanal analizinin uygulanacağı kriptografik donanımdan N tane güç ölçümü alındığını ve her bir güç ölçümü için donanımın ürettiği kapalı metinleri bildiğimizi varsayalım. Korelasyon analizini yürütmek için öncelikle kapalı metinlerin her birine karşılık gelen olası bütün güç ölçümü tahminleri yapılır. Örneğin, yukarıda bahsedildiği gibi son tur anahtarının 4 biti elde edilmek istendiğinde, son tur girişinin 4 biti için 16 aday olacağından her bir kapalı metinden 16 güç ölçümü tahmini bulunur. Genel olarak aday sayısına A dersek, $N \times A$ büyüklüğünde bir tahmin matrisi oluşturulur. Bu matris H olsun. Şifreleme süresi boyunca alınan güç ölçümünün B örneklemden oluştuğunu varsayarsak, donanımdan elde edilen gerçek güç ölçümleri ile de $N \times B$ büyüklüğünde bir başka matris oluşturulur. Bu matrise de T diyelim. Korelasyon analizinde, H matrisin her bir kolonu h_i (i = 1, 2, …, A) için T matrisinin her bir kolonu t_j (j = 1, 2, …, B) ile olan korelasyonu aşağıdaki formül ile hesaplanır:

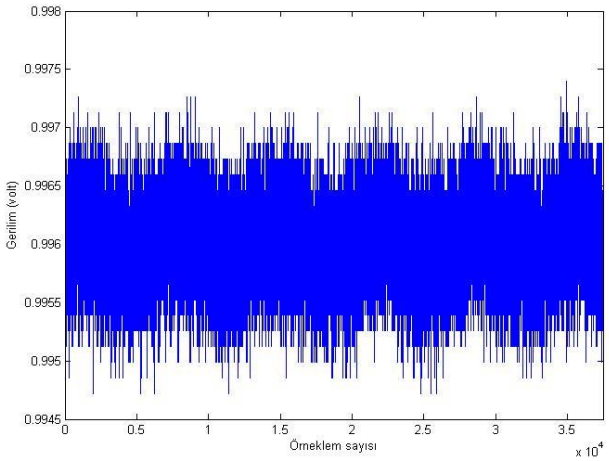
$$R_{i,j} = \frac{\sum_{k=1}^{N}(h_{k,i} - \bar{h}_i)(t_{k,j} - \bar{t}_j)}{\sqrt{\sum_{k=1}^{N}(h_{k,i} - \bar{h}_i)^2 \sum_{k=1}^{N}(t_{k,j} - \bar{t}_j)^2}}, \quad (3)$$

burada $\bar{h}_i$ ve $\bar{t}_j$, sırasıyla H ve T matrislerin i. ve j. kolonlarının ortalama değerleridir. Verilen denklem ile elde edilen $A \times B$ büyüklüğündeki R matrisi gerçek güç ölçümleri ve Hamming uzaklığı modeli ile tahmin edilen güç ölçümleri arasındaki istatiksel ilişkiyi veren bir korelasyon matrisidir. R matrisinin her bir satırı tahmin edilen bir anahtar (örneğin, daha önce bahsedilen PRESENT algoritması durumunda son tur anahtarının 4 biti) için korelasyon profili oluşturur. Doğru tahmin edilen anahtar için bu profil yüksek tepe değerlere sahip olacaktır. Böylelikle hangi tahminin doğru anahtar olduğuna karar verilir.

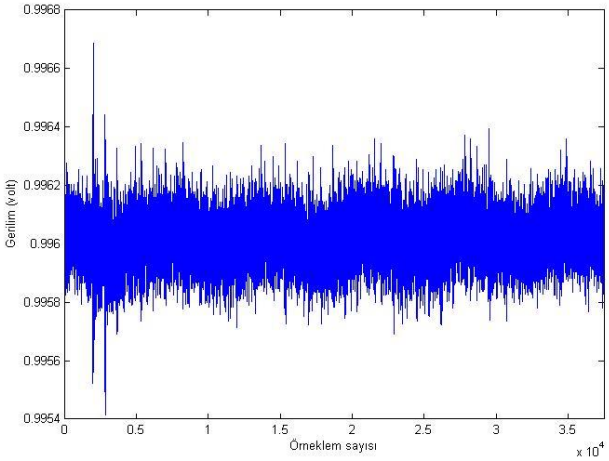
Bu çalışmada, S-kutuları başvuru tablosu olarak gerçeklenmiş ve saldırı başarısını değerlendirmek için YKA karşısındaki dayanıklılığı ölçen bir metrik olan tahmin entropisi kullanılmıştır. Tahmin entropisi, belirli bir sayıda güç ölçümü ile doğru anahtarın elde edilebilmesi için denenmesi gereken aday anahtarlarının ortalama sayısını gösteren bir ölçüttür. KGA için bu ölçüt, R matrisinin belirlediği korelasyon profillerine göre denenmesi gereken anahtarlar arasında doğru anahtarın kaçıncı sırada olduğu bulunarak hesaplanır.

6. Bulgular

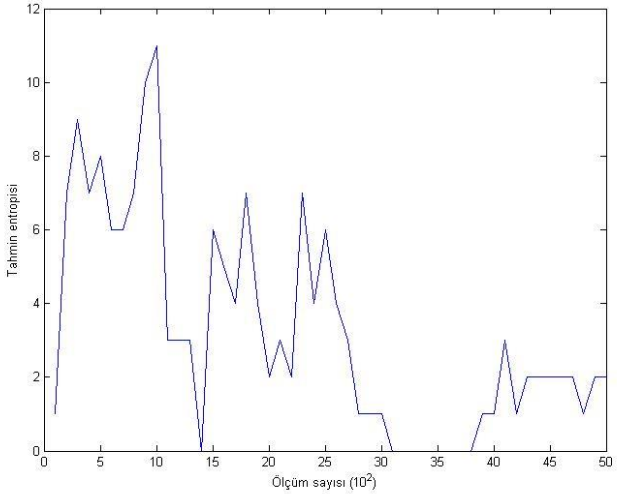
PRESENT blok şifreleme algoritmasının SAKURA-X tarafından icra edilmesi esnasında bir şifreleme süresince osiloskop ile alınan tipik bir ölçüm sonucu aşağıdaki Şekil 4'te gösterilmektedir. Ölçümlerdeki gürültü miktarını azaltmak ve böylelikle daha sağlıklı sonuçlar elde etmek amacıyla, aynı açık metin ve anahtar için SAKURA-X kartının birden fazla sayıda şifreleme yapması sağlanmış ve her bir şifrelemede tekrar ölçüm alınarak elde edilen ölçümlerinin ortalaması KGA saldırısını yürütmek için kullanılmıştır. Şekil 5'te bahsedildiği gibi elde edilen 30 ölçümünün ortalaması gösterilmektedir.



Şekil 4. PRESENT algoritması için tipik bir ölçüm sonucu.



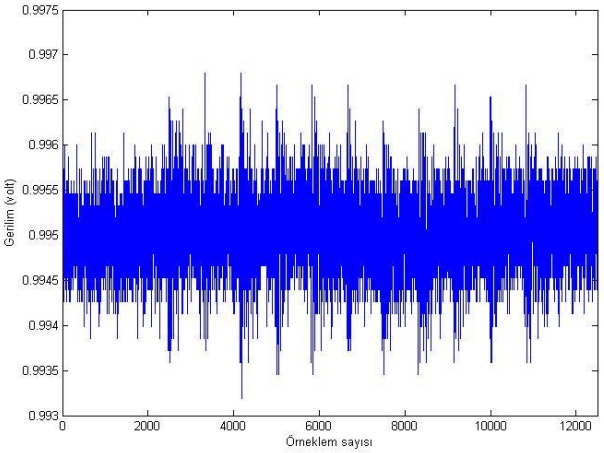
Şekil 5. PRESENT algoritması için 30 ölçüm ortalaması.



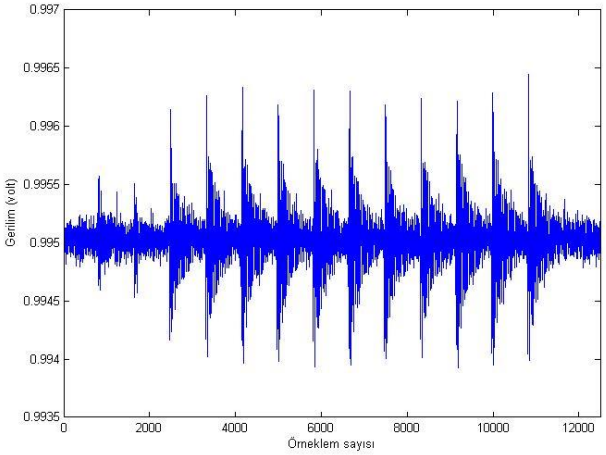
Şekil 6. PRESENT için elde edilen tahmin entropisi.

Şekil 5'ten, ortalama gürültü değerlerinin daha düşük olduğu, bununla birlikte tur zamanlarının halen belirlenemediği ve gürültü etkisinin devam ettiği gözlenmektedir. Her bir güç ölçümü (aynı açık metin ve anahtar için) 30 ölçümün ortalamasından elde edilen 5000 güç ölçümü ile yürüttüğümüz KGA sonucunda elde edilen doğru anahtarın tahmin entropisi grafiği Şekil 6'da verilmiştir. Şekil 6'dan görüldüğü üzere, 5000 güç ölçümü sonucunda, doğru anahtarı bulmak için denenmesi gereken anahtar sayısı azalmış olmakla birlikte, doğru anahtar elde edilememiştir. Ortalaması alınan ölçümlerinin sayısını 200'e çıkardığımızda bile doğru anahtarın tespit edilemediği gözlenmiştir. Gerçekte bu sonuç, az güç tüketimi nedeniyle, ölçüm sayısını artırarak azaltamadığımız gürültünün, osiloskopun nicemleme hatasından (veya dikey çözünürlüğünün yetersiz kalmasından) kaynaklandığını ve bu hatanın KGA başarısını etkilediğini göstermektedir. Bununla birlikte, nicemleme hatasının olmadığı durumlarda 30 yerine çok daha yüksek sayıda ölçümlerin ortalamaları kullanılırsa, tahmin entropisi grafiğinde daha hızlı bir düşüş beklenebilir.

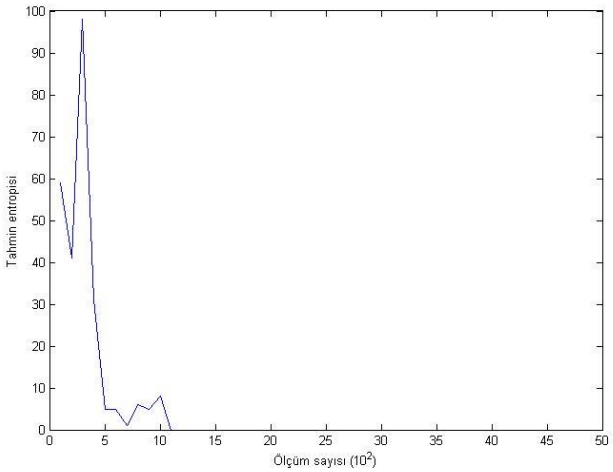
Bir blok şifreleme algoritmasında kullanılan S-kutusunun büyüklüğünün kriptografik donanımın harcadığı gücü etkileyen bir faktör olduğu bilinmektedir. Bunu test etmek için, S-kutusu büyüklüğü 8-bit olan bir başka şifreleme algoritması AES için de KGA gerçekleştirilmiş ve sonuçlar karşılaştırılmıştır. Şekil 7'de AES algoritmasının bir şifreleme süresince osiloskop ile alınan tipik bir ölçüm sonucu ve Şekil 8'de PRESENT algoritmasında olduğu gibi aynı açık metin ve anahtar için elde edilen 30 ölçümün ortalaması gösterilmektedir. PRESENT için elde edilen ölçümler (Şekil 4 ve Şekil 5) ile karşılaştırıldığında, bu iki şekildeki ölçümlerin nispeten daha az gürültülü oldukları (bir başka deyişle sinyal-gürültü oranının daha yüksek olduğu) ve her iki şekilde de AES algoritmasındaki turların gerçekleşme zamanlarının görülebildiği gözlenmektedir.



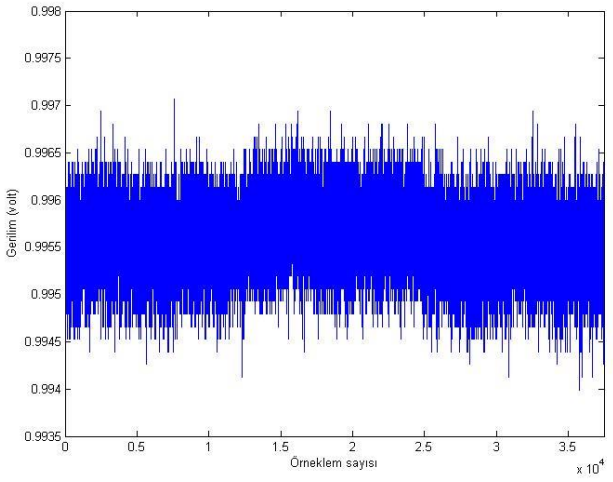
Şekil 7. AES algoritması için tipik bir ölçüm sonucu.



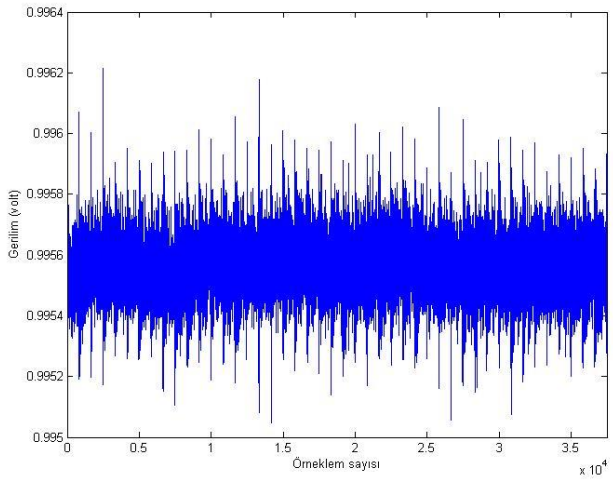
Şekil 8. AES algoritması için 30 ölçüm ortalaması.



Şekil 9. AES için elde edilen tahmin entropisi.

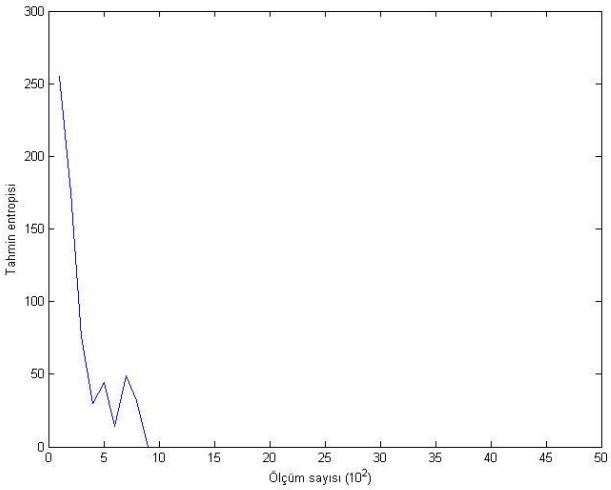


Şekil 10. Değiştirilmiş PRESENT için bir ölçüm.



Şekil 11. Değiştirilmiş PRESENT için 30 ölçüm ortalaması.

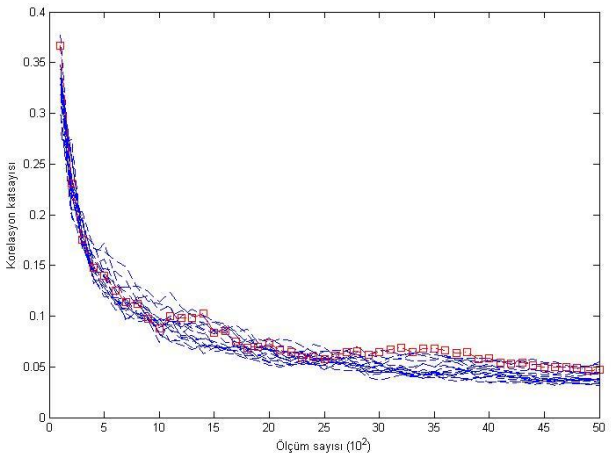
AES algoritması için de, her bir güç ölçümü (aynı açık metin ve anahtar için) 30 ölçümünün ortalamasında elde edilen 5000 güç ölçümü ile yürüttüğümüz KGA sonucunda bulunan doğru anahtarın tahmin entropisi grafiği Şekil 9'da verilmiştir. Şekil 9'da görüldüğü gibi 1100 güç ölçümünden sonra doğru anahtar elde edilmektedir. Bu sonuç, S-kutusu büyüklüğü arttıkça elde edilen güç ölçümlerinin gürültüden daha az etkilenebileceğini doğrulamaktadır. Ayrıca, çalışmamızda bu gerçek, PRESENT algoritmasında kullanılan 16 adet 4×4 büyüklüğündeki S-kutusu yerine 8 adet 8×8 büyüklüğündeki AES S-kutusu kullanılarak da doğrulanmıştır. Şekil 10 ve Şekil 11'de S-kutuları değiştirilmiş PRESENT için sırasıyla tipik bir ölçüm sonucu ve 30 ölçüm ortalaması verilmektedir. Bu ölçümler, orijinal PRESENT algoritması için verilen Şekil 4 ve Şekil 5 ile karşılaştırıldığında, özellikle Şekil 11'de görülen ortalama ölçümde tur işlemini yansıtan tepe değerlerin çok daha net bir şekilde seçilebildiği gözlenmektedir. Gerçekte bu gözlem, devrenin harcadığı gücün arttığının bir göstergesidir. Bahsedilen S-kutusu değişikliği gerçekleştirildiğinde elde ettiğimiz doğru anahtar için tahmin entropisi Şekil 12'de gösterilmektedir. Şekil 12'den görüldüğü gibi, 900 ölçümden sonra doğru anahtar bulunabilmektedir.



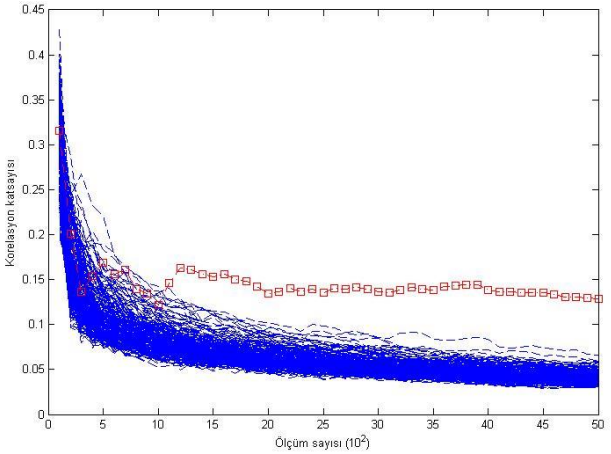
Şekil 12. Değiştirilmiş PRESENT için tahmin entropisi.

Şekil 7 ve Şekil 8'de AES için elde edilen güç ölçümleri ile karşılaştırıldığında ise, değiştirilmiş PRESENT algoritmasının her tur için kullandığı S-kutusu sayısının daha az olması nedeniyle, Şekil 10 ve Şekil 11'e göre tur geçişleri AES için daha belirgin görülmektedir. Bununla birlikte, KGA saldırısında 64 bitlik şifreleme yapan PRESENT algoritmasında tur anahtarının 1/8'i kullanılırken, 128 bitlik şifreleme yapan AES algoritmasında tur anahtarının 1/16'sı kullanılmaktadır. Bu nedenle, Hamming uzaklığı modeline göre yapılan güç tahmininin PRESENT için güç ölçümüne daha iyi bir yaklaşıklama yaptığı ve böylelikle daha az sayıda güç ölçümü kullanılarak doğru anahtarın tespit edilebildiği sonucuna varılabilir.

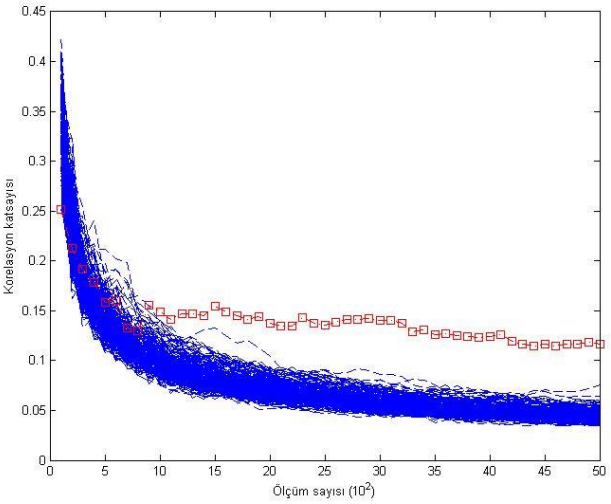
Doğru anahtar, aday anahtarların tahmin entropilerine bakılarak belirlenebileceği gibi elde edilen korelasyon profillerine bakılarak da belirlenebilir. Bir önceki bölümde (3) eşitliği ile verilen korelasyon matrisi *R*, PRESENT, AES ve S-kutuları değiştirilmiş PRESENT algoritmaları için hesaplanarak, tüm aday anahtarların korelasyon profilleri sırasıyla Şekil 13, Şekil 14 ve Şekil 15'te sunulmuştur. Bu üç şekilde de doğru anahtarın korelasyon değerleri kırmızı renkte gösterilmiştir. AES ve S-kutuları değiştirilmiş PRESENT için verilen Şekil 14 ve Şekil 15'te doğru anahtarın korelasyon değerleri en yüksektir ve kolaylıkla belirlenebilmektedir. Buna karşın PRESENT için verilen Şekil 13'te bu durumun gerçekleşmediği ve dolayısıyla doğru anahtarın tespitinin zorlaştığı görülmektedir.



Şekil 13. PRESENT için korelasyon profilleri.

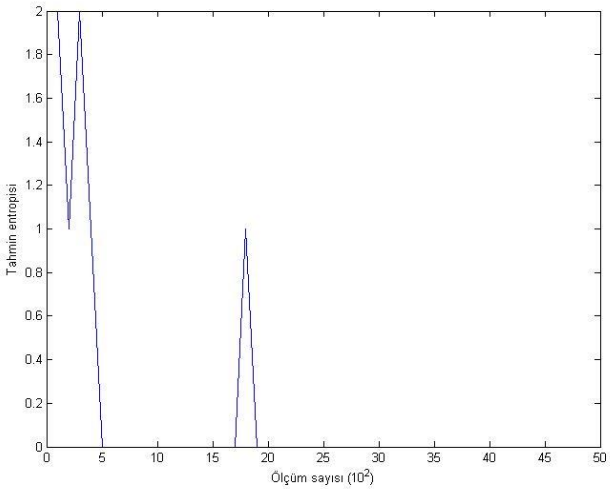


Şekil 14. AES için korelasyon profilleri.

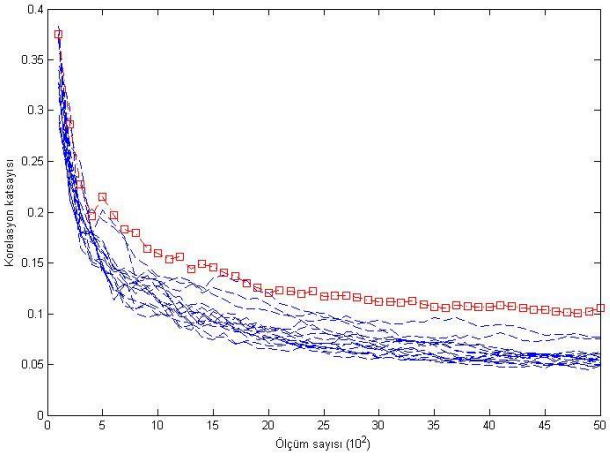


Şekil 15. Değiştirilmiş PRESENT için korelasyon profilleri.

Son olarak, PRESENT algoritması için gerçek güç ölçümleri kullanılmaksızın, MATLAB ortamında güç ölçümlerinin benzetimi ile KGA uygulanmıştır. Kriptografik donanımda oluşan gürültü, benzetimde toplanır Gauss gürültüsü olarak modellenmiştir. Bu benzetimi gerçekleyen MATLAB kodu [32]'de verilmektedir. Örneğin gürültü varyansı 4 olarak alındığında, kodun çalıştırılması ile elde edilen tahmin entropisi grafiği ve korelasyon profilleri sırasıyla Şekil 16 ve Şekil 17'te verilmektedir. Bu grafiklerin her ikisinden de doğru anahtarının tespit edilebildiği açıkça görülmektedir.



Şekil 16. PRESENT için benzetim sonucu tahmin entropisi.



Şekil 17. PRESENT için benzetim sonucu korelasyon profilleri.

7. Sonuç

Bu çalışmada, PRESENT blok şifreleme algoritmasının SAKURA-X kriptografik kartı üzerinde FPGA gerçeklemesi yapılarak, en etkili YKA yöntemlerinden birisi olan KGA bu donanımsal gerçeklemeye uygulanmıştır. Bunun sonucunda, az güç tüketiminin neden olduğu sinyal-gürültü oranının azalmasına bağlı olarak, doğru anahtarın tespitinin zorlaştığı görülmüştür. S-kutusu büyüklüğünün elde edilen sonuçlara etkisini gözlemlemek için, PRESENT algoritmasının 4×4 büyüklüğündeki S-kutusu yerine AES algoritmasının 8×8 büyüklüğündeki S-kutusu kullanılmış ve bu durumda doğru anahtarın tespit edilebildiği gösterilmiştir. Ayrıca bulunan sonuçlar, PRESENT için MATLAB benzetimi ve AES için SAKURA-X kartı üzerinde gerçeklemesi ile elde edilen güç ölçümlerinin kullanıldığı KGA sonuçları ile karşılaştırılmıştır.

Kaynakça

[1] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems - CHES 2007, LNCS, vol. 4727, pp. 450-466, Springer, 2007.

[2] ISO. ISO/IEC 29192-2:2012 Information technology - Security techniques - Lightweight cryptography - Part 2: Block ciphers. URL: https://www.iso.org/standard/56552.html (Erişim tarihi: 10, 19, 2022)

[3] P. Kocher. Timing attacks on implementations of Diffie-Hellmann, RSA, DSS, and other systems. In: Advances in Cryptology - CRYPTO'96, LNCS, vol. 1109, pp.104-113, Springer, 1996.

[4] P. Kocher, J. Jaffe, B. Jun. Differential power analysis. In: Advances in Cryptology - CRYPTO'99, LNCS, vol. 1666, pp. 388-397, Springer, 1999.

[5] B. Gierlichs, L. Batina, P. Tuyls, B. Preneel. Mutual information analysis. In: Cryptographic Hardware and Embedded Systems - CHES 2008, LNCS, vol. 5154, pp. 426-442, Springer, 2008.

[6] E. Brier, C. Clavier, F. Olivier. Correlation power analysis with a leakage model. In: Cryptographic Hardware and Embedded Systems - CHES 2004, LNCS, vol. 3156, pp. 16-29 Springer, 2004.

[7] O. Lo, W. J. Buchanan, D. Carson. 2016. Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). Journal of Cyber Security Technology, 1(2):88-107, 2016.

[8] J. Doget, E. Prouff, M. Rivain, F.-X. Standaert. Univariate side channel attacks and leakage modeling. Journal of Cryptographic Engineering, 1(2):123-144, 2011.

[9] A. Moradi, N. Mousavi, C. Paar, M. Salmasizadeh. A comparative study of mutual information analysis under a Gaussian assumption. In: Information Security Applications - WISA 2009, LNCS, vol. 5932, Springer, 2009.

[10] S. Nikova, C. Rechberger, V. Rijmen. Threshold implementations against side-channel attacks and glitches. In: Information and Communications Security - ICICS 2006, LNCS, vol. 4307, pp. 529-545 Springer, 2006.

[11] T. De Cnudde, O. Reparaz, B. Bilgin, S. Nikova, V. Nikov, V. Rijmen. Masking AES with d+1 shares in hardware. In: Cryptographic Hardware and Embedded Systems - CHES 2016, LNCS, vol. 9813, pp. 194-212 Springer, 2016.

[12] T. Güneysu, A. Moradi. Generic side-channel countermeasures for reconfigurable devices. In: Cryptographic Hardware and Embedded Systems - CHES 2011, LNCS, vol. 6917, pp. 33-48, Springer, 2011.

[13] S. Mangard, E. Oswald, T. Popp. Power analysis attacks: revealing the secrets of smart cards, Springer, US, 2007.

[14] E. Biham, A. Shamir. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3-72, 1991.

[15] M. Matsui. M. Linear cryptanalysis method for DES cipher. In: EUROCRYPT'93, LNCS, vol. 765, pp. 386-397, Springer, 1994.

[16] M. Renauld, F.-X. Standaert. Algebraic side-channel attacks. In: Information Security and Cryptology - Inscrypt 2009. LNCS, vol. 6151, pp. 393-410, Springer, 2010.

[17] L. Yang, M. Wang, S. Qiao. Side channel cube attack on PRESENT. Cryptology and Network Security - CANS 2009, LNCS, vol. 5888, pp. 379-391, Springer, 2009.

[18] X.-J. Zhao, T. Wang, S.-Z. Guo. Improved side channel cube attacks on PRESENT. Cryptology ePrint Archive, URL: https://eprint.iacr.org/2011/165 (Erişim tarihi: 10, 19, 2022).

[19] X. Duan, Q. Cui, S. Wang, H. Fang, G. She. Differential power analysis attack and efficient countermeasures on PRESENT. In: Proceedings of the 8th IEEE International Conference on Communication Software and Networks - ICCSN 2016, pp. 8-12, IEEE, 2016.

[20] R. Sadhukhan, S. Patranabis, A. Ghoshal, et al. An evaluation of lightweight block ciphers for resource-constrained applications: area, performance, and security. Journal of Hardware and Systems Security, 1:203-218, 2017.

[21] L. Goubin, A. Martinelli, M. Walle. Impact of Sboxes size upon side channel resistance and block cipher design. In: Progress in Cryptology - AFRICACRYPT 2013, LNCS, vol. 7918, pp 240-259, Springer, 2013.

[22] C. Carlet, E. de Chérisey, S. Guilley, S. Kavut, D. Tang. Intrinsic resiliency of s-boxes against side-channel attacks best and worst scenarios. IEEE Transactions on Information Forensics and Security, 16:203-218, 2021.

[23] A. Heuser, S. Picek, S. Guilley, N. Mentens. Lightweight ciphers and their side-channel resilience. IEEE Transactions on Computers, 69(10):1434-1448, 2020.

[24] J. Zhang, D. Gu, Z. Guo, L. Zhang. Differential power cryptanalysis attacks against PRESENT implementation. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering - ICACTE 2010, vol. 6, pp. 61-65, IEEE, 2010.

[25] C. Wang, M. Yu, J. Wang, P. Jiang, X. Tang. A more practical CPA attack against PRESENT hardware implementation. In: Proceedings of the 2nd International Conference on Cloud Computing and Intelligence Systems - CCIS 2012, pp. 1248-1253, IEEE, 2012.

[26] O. Lo, W. J. Buchanan, D. Carson. Correlation power analysis on the PRESENT block cipher on an embedded device. In: Proceedings of the 13th International Conference on Availability Reliability and Security-ARES 2018, pp. 6-11, ACM, 2018.

[27] Q. Fang, M. Alioto. Last-round and joint first/last-round power analysis attacks on PRESENT. In: Proceedings of Asian Hardware Oriented Security and Trust Symposium - AsianHOST 2021, pp. 1-6, IEEE, 2021.

[28] N. A. Gunathilake, A. Al-Dubai, W. J. Buchanan, O. Lo. Electromagnetic side-channel attack resilience against PRESENT lightweight block cipher. In: 6th International Conference on Cryptography, Security and Privacy (CSP), pp. 51-55, IEEE, 2022.

[29] SAKURA (SASEBO-GIII). URL: https://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-X.html (Erişim tarihi: 10, 19, 2022).

[30] Y. Hori, T. Katashita, A. Sasaki, A. Satoh. SASEBO-GIII: A hardware security evaluation board equipped with a 28-nm FPGA. In: Proceedings of the 1st IEEE Global Conference on Consumer Electronics, pp. 657-660, IEEE, 2012.

[31] F.-X. Standaert, T. G. Malkin, M. Yung. A unified framework for the analysis of side-channel key recovery attacks. In: Advances in Cryptology - EUROCRYPT 2009, LNCS, vol. 5479, pp 443-461, Springer, 2009.

[32] GitHub. URL: https://github.com/Selcuk-kripto/cpa_present (Erişim tarihi: 10, 19, 2022).

Özgeçmişler

Selçuk Kavut, Ankara Üniversitesi Elektronik Mühendisliği Bölümü'nden lisans derecesini 1998 yılında, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik ve Elektronik Mühendisliği Anabilim Dalı'ndan yüksek lisans ve doktora derecelerini sırasıyla 2002 ve 2008 yıllarında almıştır. 2009-2014 yılları arasında Gebze Yüksek Teknoloji Enstitüsü'nde Öğr. Gör. Dr. olarak çalıştıktan sonra Balıkesir Üniversitesi'ne geçmiş olup, halen Bilgisayar Mühendisliği Bölümü'nde Doç. Dr. olarak çalışmaktadır. Çalışma alanları kriptoloji ve kodlama teorisi üzerinedir.

Yasin Reşit Yargıcı, lisans ve yüksek lisans derecelerini Balıkesir Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü'nden 2014 ve 2019 yıllarında almıştır. Özel sektörde ARGE uzmanı olarak çalışmaktadır. Çalışma alanları otomasyon, güç elektroniği ve kriptoloji alanlarıdır.