Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
https://doi.org/10.53070/bbd.990915

Journal of Computer Science
ISSN: 2548-1304
Volume: IDAP-2021, Issue: Special, pp: 61-69
@Anatolian Science

Nesnelerin İnternetinde Rassal ve Kontrollü Bekleme Süresi İle Zamanlama Analizi Saldırılarının Önlenmesi

Preventing Timing Analysis Attacks with Random and Controlled Waiting Times

Muhammed Saadetdin KAYA [*1], Kenan İNCE [2]

[1]Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, TÜRKİYE

[2]Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, TÜRKİYE

(saadetdin.kaya@gmail.com, kenanince@gmail.com)

Received: Sep.3, 2021 Accepted: Sep.16, 2021 Published: Oct.20, 2021

Özetçe— Yaygınlaşan Nesnelerin Interneti konsepti ile birlikte çeşitli güvenlik zaafiyetlerinin önlenmesi daha fazla önem arz etmektedir. Bu zaafiyetler doğrudan veya dolaylı olarak ortaya çıkabilmektedir. Her türlü istenmeyen veri sızıntısı bütün sistem için tehlike teşkil etmektedir. Bir sistemden elde edilen zamanlama bilgisinden faydalanılarak yapılan Zamanlama Analizi Saldırıları; bir işlemin veya algoritmanın değişken şartlara verdiği tepkinin yorumlanmasıyla, sistem hakkında bilgi edinmeyi amaçlar. Bu çalışmada, bir haberleşme fonksiyonunun en iyi ve en kötü durumdaki işleme sürelerine bakılarak rassal olarak geciktirilmesi ile gözlemlenen işlem süresi bilgisinin anlamsız hale getirilmesi amaçlanmıştır. Yapılan deneysel çalışma sonucunda zamanlama bilgisi ile anahtar eşleşme oranında doğrusal ilişki ve sunulan yöntemin bu doğrusal ilişkinin gizlenmesi için önemli bir alternatif olabileceği görülmüştür.

Anahtar Kelimeler : *Yan-kanal Saldırıları, Nesnelerin İnterneti, Zamanlama Analizi Saldırıları, Kriptografik Analiz, Rassal Sayı Üreteci*

Abstract— Along with the widespread concept of the Internet of things, it became more important to prevent various security weaknesses. These weaknesses can occur directly or indirectly. Any unintended information leakage is a danger to whole system. Timing analysis attacks with the use of timing information obtained from a system; they aim to obtain information about the system by interpreting the response of a process or algorithm to variable conditions. In this study, it was aimed to make the information about the observed processing time meaningless by randomly delaying the communication function by looking at the processing times in the best and worst state. As a result of the experimental study, it was observed that the linear relationship between timing information and the key match ratio and the presented method can be an important alternative to hiding this linear relationship.

Keywords : *Side-channel Attacks, Internet of Things, Timing Analysis Attacks, Cryptography, Cryptographic Analysis, Random Number Generator*

1.Giriş

Günümüzde kahve makinesi, buzdolabı, televizyon gibi gündelik eşyalardan araba ve tren gibi ulaşım araçlarına kadar birbirinden bağımsız birçok alanda internete bağlı cihazlara rastlamak mümkün olmaktadır. Bu cihazların oluşturduğu ekosistem Nesnelerin İnterneti (IoT – Internet of Things) olarak adlandırılmaktadır. İnterneti kullanan nesnelerin çeşitliliği ve bu nesnelerin sürekli iletişim halinde olmaları çeşitli riskleri de beraberinde getirmektedir (Samani vd., 2015). Genellikle fiziksel olarak bu cihazlara erişimin zor olmasının yanında gerekli güvenlik önlemlerinin alınmaması durumunda bu nesnelerin tamamen ele geçirilmeleri veya haberleşmeleri esnasında araya girilerek hassas bilgilerin sızdırılması gibi istenmeyen durumlarla karşılaşılması oldukça olasıdır (Birkel ve Hartmann, 2020; Abbas vd., 2019)

Fiziksel olarak bu cihazlara erişim olmasa bile bir işlemin ne kadar sürede yapıldığı, yapılırken harcanan güç, ortaya çıkan elektromanyetik yayınım, işlem süresince çıkan sesin şiddeti gibi sistem dışına istemsiz çıkışlar kullanılarak çeşitli analizler yapılabilmektedir (Zhao ve Ge, 2013). Bu istemsiz çıkışlar sistemin çözümlenmesinde kullanılabilecek nitelikte ise bu bilgiler yan-kanal bilgisi, bu bilgiler kullanılarak yapılan analizler ise yan-kanal analizi olarak adlandırılmaktadırlar. Yan-kanal analiz saldırıları (YAS), bu analizler vasıtasıyla sistem hakkında bilgi edinmeyi veya çözümlenme sağlanmasını hedefleyen saldırılardır (Joy Persial vd., 2011).

Yan-kanal analizi saldırıları, genel olarak aktif saldırılar ve pasif saldırılar olmak üzere iki grupta incelenebilir. Aktif saldırılar ya da diğer adıyla kurcalama saldırıları hedef sistemin içindeki devrelere ulaşılmasını gerektirirler. Pasif saldırılar ise, sistemin çalışmasına doğrudan müdahale etmeden üretmiş olduğu yan–kanal bilgilerinden faydalanırlar (Anderson ve Kuhn, 1996; Ordu ve Yalçın, 2016). Genellikle analiz sırasında kullandıkları bilgiye göre adlandırılan pasif yan-kanal saldırıları; Güç Analizi, Elektromanyetik Analizi ve Zamanlama Analizi Saldırılarıdır (Kocher, 1996)

Analiz işleminin ve uygulanmasının nispeten daha kolay olduğu Zamanlama Analizi Saldırıları (ZAS) bir işlemin veya algoritmanın çeşitli şartlar altında değişen işleme süresinin yorumlanmasıyla sistem hakkında bilgi edinmek amacıyla yapılan bir saldırı türüdür. Çeşitli çalışmalar, bu saldırı türü ile sistem hakkındaki en kritik bilgilerin ortaya çıkarılabileceğini göstermiştir (Kocher, 1996; Janke ve Laackmann, 2002)

IoT'de cihazlar kendi aralarında ve kullanıcılarla sıklıkla ortak internet ağları üzerinden iletişim kurarlar. Genellikle düşük bellek, düşük güç ve düşük işlem yeteneklerine sahip olan bu cihazlar için daha az kaynak kullanımı ile güçlü şifreleme çözümleri tasarlanması gerekmektedir (Mukherjee, 2015). Düşük kaynak kapasitesi sebebiyle bilgisayar tabanlı konvansiyonel kriptografik çözümlerin birçoğunun tam olarak doğrudan uygulanamaması, hafif-siklet algoritmaların bu alanda sıklıkla tercih edilmesi sonucunu doğurmuştur (Kim, 2017).

Hafif siklet kriptografik şifreleme, RIFD etiketleri, sensörler, temassız akıllı kartlar vb. kısıtlı ortamlarda uygulamalar için uyarlanmış, özel senaryolar için özel çözümler öneren kriptografik algoritmalar ve protokollerdir (Katagi ve Moriai, 2008). Genellikle bilgisayar tabanlı kriptografik çözümlerin hafifletilmeleriyle IoT sistemlerine uygulanan bu algoritmalar, hem yapıları gereği hem de hafifletilmeleri sebebiyle bazı zaafiyetleri bünyelerinde bulundurmaktadırlar (Kaps, 2008; Kim ve Yoon, 2014). Bu zafiyetler ve yan-kanal saldırıları kullanılarak şifrelemede kullanılan gizli anahtara ulaşılması veya sistemin çözülmesi mümkün olmaktadır (Kim ve Yoon, 2014; Williams, 2008; Zhao vd., 2009).

Bu çalışmada, aynı ağa bağlı iki farklı cihazın gizli anahtar kullanarak iletişim sağlama durumu için anahtar eşleştirilmesi senaryosunda; değişken girdiler kullanılarak eşleşme işlemi sırasında geçen süreler analiz edilmiştir. Elde edilen zamanlama bilgisinin, eşleşme işleminin kontrollü olarak bekletilmesiyle gizlenmesi hedeflenmiştir. Bu alanda yapılan diğer çalışmaların aksine kaynak kullanımı düşük seviyede tutularak yalnızca ZAS'a karşı koruma sağlayacak bir hafif sıklet koruma yöntemi sunulmuştur.

2. ZAMANLAMA ANALİZİ

Zamanlama analizi (ZA), yapılan işlemlerin değişken şartlarda sızdırdıkları zamanlama bilgilerini kullanarak yapılan işlemler veya sistem hakkında bilgi edinmek amacıyla yapılan analiz türüdür (Kocher, 1996).

2.1. Zamanlama Analizi Saldırıları:

ZAS, bir kriptografik sistemin çeşitli koşullarda gerçekleştirilirken geçen sürelerin farklılıkları kullanılarak yapılan bir yan kanal saldırısı türüdür. Girdiye bağlı olarak değişen işleme süresine sahip algoritmaların sızdırdığı zamanlama yan-kanal bilgisinden faydalanılır (Kocher, 1996; Janke ve Laackmann 2002; Popp vd., 2007). Farklı işlemci komutları uygulanırken, toplama, üs alımı veya bölme gibi matematiksel işlemler gerçekleştirilirken farklı işleme süreleri ortaya çıkmaktadır. Örneğin; kullanıcıdan girdi alınarak yapılan bir üs alma işlemi büyüyen girdiler için daha uzun işleme sahip olacaktır. Bu durum algoritma çözümlenmese bile gizlenmek istenilen veri hakkında bilgi sahibi olunmasını sağlar. Genellikle fiziksel müdahaleye ihtiyaç duyulmadan gerçekleştirilebilmeleri sebebiyle ZAS, IoT alanında uygulanması nispeten daha kolay olan bir saldırı türüdür.

Bu alanda yapılan çeşitli çalışmalar işlemlerin yürütülme sürelerindeki farklardan faydalanarak elde edilen zamanlama yan-kanal bilgisi ile gizlenmek istenen bilgiye ulaşmayı başarmışlardır (Kocher, 1996; Janke ve Laackmann 2002; Perianin vd., 2020; Lerman vd., 2011; Won vd., 2021).

Kocher (1996), yapmış olduğu çalışmada algoritma adımlarının birinde gizli anahtara bağlı olarak üs alma işlemi sırasında işlem süresinin değişkenliğinden faydalanılarak yapılan bir ZAS örneği görülmüştür. Yine aynı şekilde Janke ve Laackmann (2002), yapmış oldukları çalışmada ortaya koyulan senaryoda anahtara bağlı olarak ortaya çıkan dallanma işlemleri sonucunda yan-kanal zamanlama bilgisi kullanılarak anahtara ulaşılabileceği ortaya koyulmuştur.

ZAS tek başına uygulanabileceği gibi, diğer yan-kanal saldırıları ile birlikte de uygulanabilir. Walter ve Thompson 2001 yılında yapmış oldukları çalışmada zamanlama analizi ve güç analizi saldırıları birlikte kullanılarak modüler üs alıcılarla ilgili gizli bilgilere ulaşılmıştır (Walter ve Thompson, 2001).

2.2. Zamanlama Analiziz Saldırılarından Korunma Yöntemleri:

Sistem hakkında bilgi sahibi olunabilmesinin yanında doğrudan sistem çözümlenmesine kadar varabilen ZAS'dan korunmak veya etkilerini azaltmak için çeşitli yöntemler geliştirilmiştir (Käsper ve Schwabe, 2009; Ambrose vd., 2008; Dhem, 1998; Walter, 1999; Walter 2002; Hachez ve Quisquater, 2000). Alınabilecek önlemler donanımsal ve yazılımsal olarak iki şekilde sınıflandırılabilirler.

Donanımsal önlemler daha çok zamanlama ölçümü yapılmasını veya gürültüyü arttırarak ya da sızan zamanlama bilgisini sınırlandırarak yapılan saldırıyı zorlaştırmayı amaçlamaktadırlar. Örneğin; bir tümleşik devre içerisine yerleştirilen fiziksel bir rassal sayı üreteci ile gürültünün arttırılması donanımsal önlem olarak nitelendirilebilir (Kocher vd., 1999).

Yine bu saldırılara karşı alınabilecek temel önlemlerden biri de güç tüketimi yapılan işleme göre değişmeyen donanımsal elemanların kullanılmasıdır (Tiri ve Verhauwhede, 2003). Ancak bu çözümler genellikle asgari boyutta alan ve kaynak kullanımının çok önemli olduğu IoT sistemlerinde yaygın olarak kullanılamamaktadırlar. Zamanlama analizinin her senaryo özelinde uygulanması gerektiği gibi çözümlerinin de yine her senaryoda değişebileceği unutulmamalıdır (Schindler, 2015).

Yazılımsal olarak alınabilecek önlemlerden biri yine donanımsal önlemlerdeki gibi işlem süresinin rastgele hale getirilmesidir (Coron, 1999). Bu korunma yönteminde, algoritma içerisine işlem sonucunu değiştirmeyecek çeşitli işlemler eklenir. Bu işlemlerin ortaya çıkaracağı gecikmenin rassal olarak oluşturulması sağlanır ve bu sayede saldırganın zamanlama bilgisini anlamlandırması zorlaştırılır. Bu yöntem tamamen rassal olması sebebiyle yapılan işlemlerin sisteme daha fazla yük bindirmesi ve genellikle hızlı tepki süresine ihtiyaç duyan IoT sistemleri için ağır kalabilmektedir.

En çok kullanılan yazılımsal korunma yöntemlerinden biri de kullanılan algoritmanın sabit zamanlı çalışacak şekilde tasarlanmasıdır (Ordu ve Yalçın, 2016). Bu yöntem ZAS'a karşı en güçlü korunma yöntemi olsa da, bir kriptografik şifreleme algoritmasının sabit zamanlı çalışmasının tek yolu her

koşulda en kötü durumdaymış gibi tepki vermesiyle mümkün olmaktadır. Bu da kriptografik şifreleme algoritmalarının bile hafif-siklet olarak kullanılmasını gerektiren IoT alanı için istenmeyen bir durumdur.

Algoritmayı oluşturan işlemlerin bloklar halinde yerlerinin değiştirilmesi ve buna bağlı olarak algoritmaya takdim edilen kaynak miktarının değiştirilmesi de yazılımsal olarak alınabilecek önlemlerdendir (Goubin ve Patarin, 1999). Ancak bu önlemler de yine diğer önlemler gibi IoT alanında doğrudan uygulanamamaktadır.

3. UYGULAMA

IoT sistemlerindeki düşük sistem özellikleri sebebiyle konvanvsiyonel kriptografik şifreleme algoritmalarının tam olarak uygulanamaması sebebiyle bu algoritmalar hafifletilerek uygulanmak zorunda kalmaktadır (Kim, 2017). Bu kısıtlayıcı durum, şifreleme algoritmaları gibi saldırılara karşı alınacak olan korunma yöntemlerini de etkilemektedir. Bu çalışmada, daha önce bahsedilen yazılımsal korunma yöntemlerinden biri olan işlem süresinin rastgele hale getirilmesi yönteminin hafifletilerek daha az kaynak kullanımı ile daha hızlı tepki vermesi hedeflenmiştir.

Bir adet Raspberry Pi 3 ve iki adet Arduino MEGA kullanılarak bir IoT sistemi oluşturularak gizli anahtar eşleşme senaryosu kullanıcı tarafından anahtar girdisi alınarak simüle edilmiştir. Kullanıcı tarafından alınan farklı boyuta ve eşleşme değerlerine sahip girdiler karşısında sistemin sızdırmış olduğu zamanlama bilgisi analiz edilmiştir.

Yapılan analiz sonucunda sızan zamanlama bilgisi N, algoritma işleme zamanı K, en iyi durum Ω (%0 eşleşme oranı) ve en kötü durum O (%100 eşleşme oranı) için;

$$\Omega \leq N \leq 2 \times O \; ve \; 0 \leq m \leq O \;\; için; \; N = K + m \tag{1}$$

modeli oluşturulmuştur. N ve m değerlerinin, belirtilmiş aralıklarda olmaları sağlanacak şekilde, rassal sayı üreteci kullanılarak m değerinin oluşturulması ve algoritmanın hiçbir şey yapmadan bekletilmesi sağlanmıştır. Yapılan bu işlem sonucunda; sisteme ekstra yük bindirmeden, tepki süresinin minimumda tutularak sızan zamanlama yan-kanal bilgisinin en anlamsız hale getirilmesi amaçlanmıştır.

Sınır değerleri belirlendikten ve model oluşturulduktan sonra algoritma sabit değerli zamanlama bilgisi verecek şekilde düzenlenerek ZA uygulanmıştır. Önerilen yöntem ve sabit zamanlı algoritmanın ZA sonuçları karşılaştırılmıştır. Alınan sonuçlar sistem yükü değişiklikleri göz ardı edilmek amacıyla her aşama için yaklaşık 55.000 değişken sistem yükleri altında test edilerek kaydedilmiştir.

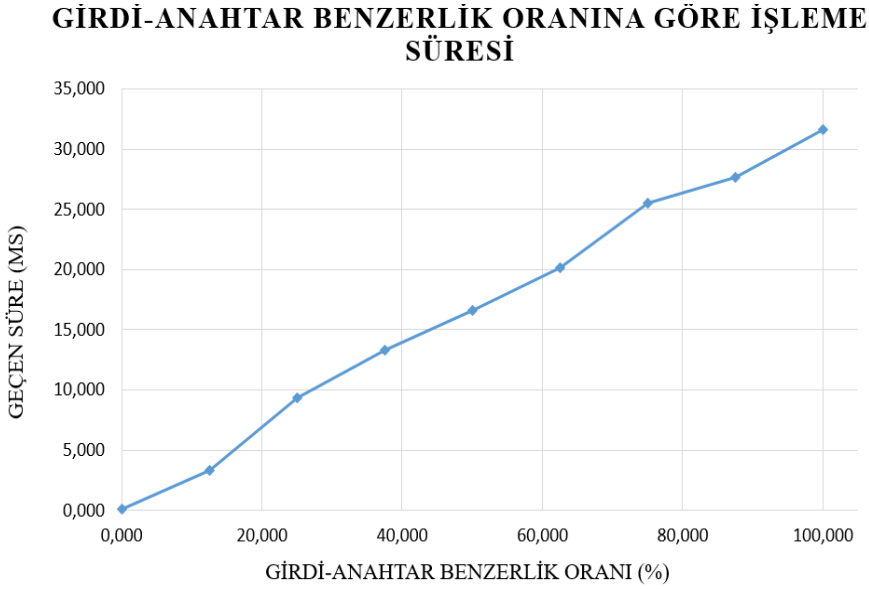
4. SONUÇLAR

Uygulamanın ilk aşaması olan yalın haldeki algoritmaya ZA, sistem üzerinde gizli anahtarla farklı eşleşme oranlarına sahip girdilerle test edilerek uygulanmış olup Tablo 1'deki sonuçlara ulaşılmıştır.

Tablo 1. Ω ve O Durumlarının Yalın Algoritmadaki Zamanlama Bilgileri

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	0,054	0,101	0,342
O	29,332	31,628	46,321

Herhangi bir önlem alınmadan eşleştirme işleminin yapılması sonucunda ortaya çıkan analiz sonuçları Şekil 1.'de görülmektedir. Girdi-anahtar benzerliği arttıkça işleme süresinin de arttığı, benzerlik ile işleme süresi arasında doğru orantı olduğu görülmektedir. Bu durum, farklı gerçekleşimlerin yalnızca sürelerine bakılarak o gerçekleşimde kullanılan girdinin anahtar ile benzerliğinin yorumlanabilmesini hatta gizli anahtarın tahmin edilebilmesini mümkün kılmaktadır.



Şekil 1. Eşleşme Fonksiyonunun Zamanlama Analizi Sonuçları

Algoritmanın yalın haline uygulanan ZA sonrasında algoritma girdi boyutundan bağımsız olarak cevap verecek şekilde düzenlenmiştir. Kaydedilen zamanlama bilgileri Tablo 2.'de görülmektedir.

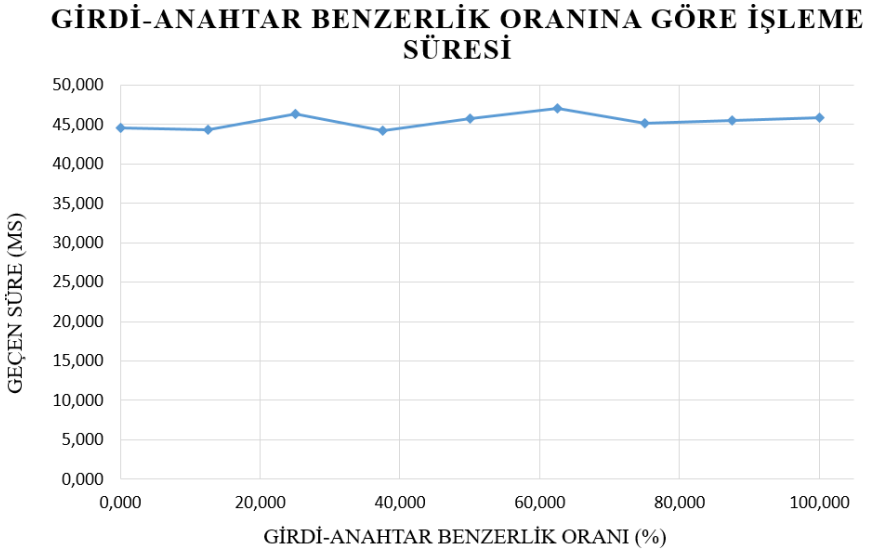
Tablo 2. Ω ve O Durumlarının Sabit Zamanlı Algoritmadaki Zamanlama Bilgileri

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	43,867	45,441	47,967
O	44,389	45,789	47,943

Şekil 2.'de görülen sonuçlar sabit zamanlı işleme süresine sahip olacak şekilde eşleştirme işleminin düzenlenmesi sonucunda ortaya çıkan analiz sonuçlarıdır. Fonksiyon girdi boyutundan ve girdi-anahtar benzerliğinden bağımsız olarak en kötü durumda cevap vermiştir. Bu çözüm zamanlama analizini engellemeseine karşın karmaşık işlemlerin olduğu fonksiyonlarda veya anahtar boytunun çok büyük olduğu durumlarda sistem üzerindeki yükü arttıracaktır. Bu da düşük kaynaklara sahip olan IoT uygulamalarında istenmeyen bir durumdur.

Yapılan ZA sonucunda, (1)'deki model Tablo 1.'deki sonuçlar kullanılarak uygulanmış ve bu yeni uygulamaya ait ZA sonuçları Tablo 3.'te olduğu şekilde kaydedilmiştir.

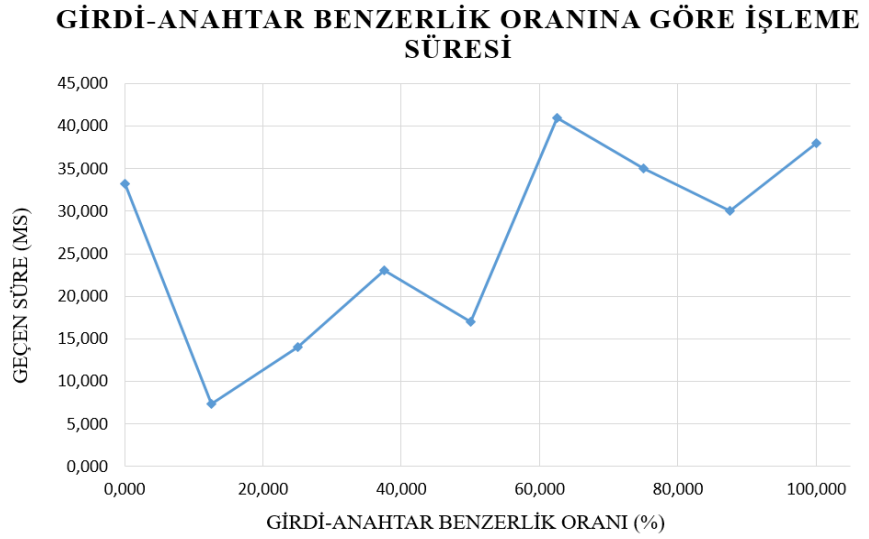
Şekil 3.'te, rastgele bekleme yapacak şekilde değiştirilmiş eşleştirme işleminin analiz sonuçları bulunmaktadır. Girdi-anahtar benzerliği ile işleme süresi arasında herhangi bir doğrudan ilişki grafiğe bakılarak görülememektedir.



Şekil 2. Sabit Zamanlı Eşleşme Fonksiyonunun Zamanlama Analizi Sonuçları

Tablo 3. Ω ve O Durumlarının Önerilen Yöntemin Uygulandığı Algoritmadaki Zamanlama Bilgileri

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	0,095	21,421	29,967
O	30,153	44,311	83,121



Şekil 3. Önerilen Çözüme Ait Eşleşme Fonksiyonunun Zamanlama Analizi Sonuçları

Son olarak girdi-anahtar eşleştirme algoritması yalın, sabit zamanlı ve önerilen yöntemli olmak üzere üç durumda da yeniden uygulanarak girdi-anahtar benzerliği oranına göre vermiş oldukları sonuçlar ve ortalama işleme süreleri karşılaştırılmıştır.

Tablo 4.'te görüldüğü üzere önerilen rassal ve kontrollü bekleme yöntemi ortalama işleme süresi olarak yalın haldeki algoritmadan daha yavaş kalmakta ancak sabit zamanlı algoritmaya göre zamanlama bilgisini maskelemesine rağmen neredeyse yarı yarıya daha hızlı tepki vermektedir.

Tablo 4. Girdi-Gizli Anahtar Benzerlik Oranına Göre Algoritmaların Zamanlama Analizi Sonuçları

	Zamanlama Değeri (ms)		
Benzerlik (%)	Yalın	Sabit Zamanlı	Rassal ve Kontrollü Bekleme
0	0,101	44,621	33,220
12,5	3,301	44,331	7,330
25,0	9,366	46,321	14,000
37,5	13,287	44,200	23,000
50,0	16,599	45,698	17,000
62,5	20,110	47,000	41,000
75,0	25,467	45,146	35,000
87,5	27,626	45,542	30,000
100,0	31,628	45,889	38,000
ORTALAMA	**16,387**	**45,416**	**26,505**

5. TARTIŞMA

Bu çalışmada, bilgisayar uygulamalarında kullanılan bir ZAS korunma yönteminin hafifletilerek IoT sistemlerine uyarlanıp geliştirilerek uygulanması görülmektedir. Önerilen yöntemin uygulanması ile birlikte yaklaşık %50'lik bir gecikme kazancı çok daha az kaynak kullanımı ile gerçekleştirilmiştir.

Eşleştirme işlemi esnasında geçen sürenin, girdi boyutundan bağımsız olması ve girdi-anahtar benzerliğini doğrudan yansıtmaması analizi zorlaştırmış olup en iyi ve en kötü durum arasında rassal bir değere sahip olması sistem üzerindeki genel yükü diğer çözüm önerilerine göre azaltmıştır. Bu kazanç çok daha kompleks işlemler göz önüne alındığında, işlem gücü düşük olan IoT cihazları için önem arz etmektedir.

Aynı ağa bağlı birden fazla IoT cihazlarının oluşturduğu ve aynı kanal üzerinden yönetildiği bir IoT ekosisteminde, bilgisayar tabanlı kriptografik algoritmaların hafifletilerek uyarlandığı gibi kabul görmüş mevcut yan-kanal saldırılarından korunma yöntemlerin de hafifletilerek ve geliştirilerek uygulanması ihtiyacı vardır.

Yapılan çalışmanın ve modelin sistem özelinde belirlenmiş olan senaryoya uygun olarak oluşturulduğu, uygulanmak istenmesi halinde uygulanacak sistem veya algoritma özelinde tekrar ZA yapılarak yeni bir model belirlenmesinin gerekliliği unutulmamalıdır. Sistem analizinin yapay zekâ kullanılarak veya daha karmaşık analiz yöntemleri ile önerilen yöntemin daha optimize bir halde uygulanabilmesi mümkün olacaktır.

Teşekkür

Bu çalışma, İnönü Üniversitesi Bilimsel Araştırma Projeleri Bölümü'nün (BAPB) FBG-2020-2143 sayılı projesi ile desteklenmiştir. Yazar, değerli geri bildirimleri için İnönü Üniversitesi BAPB'ye teşekkür eder.

Kaynaklar

Samani, A., Ghenniwa, H. H., & Wahaishi, A. (2015). Privacy in Internet of Things: A model and protection framework. Procedia Computer Science, 52, 606-613.

Kocher, P. C. (1996, August). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Annual International Cryptology Conference (pp. 104-113). Springer, Berlin, Heidelberg.

Janke, M., & Laackmann, P. (2002). Power and timing analysis attacks against security controllers. Infineon Technologies AG, Technology Update, Smart Cards.

Anderson, R., & Kuhn, M. (1996, November). Tamper resistance-a cautionary note. In Proceedings of the second Usenix workshop on electronic commerce (Vol. 2, pp. 1-11).

Ordu, L., & Yalçın, S. B. Ö. (2016, December) Yan-Kanal Analizi Saldırılarına Genel Bakış.

Popp, T., Mangard, S., & Oswald, E. (2007). Power analysis attacks and countermeasures. IEEE Design & test of Computers, 24(6), 535-543.

Birkel, H. S., & Hartmann, E. (2020). Internet of Things–the future of managing supply chain risks. Supply Chain Management: An International Journal.

Abbass, W., Bakraouy, Z., Baina, A., & Bellafkih, M. (2019). Assessing the Internet of Things Security Risks. J. Commun., 14(10), 958-964.

Zhao, K., & Ge, L. (2013, December). A survey on the internet of things security. In 2013 Ninth international conference on computational intelligence and security (pp. 663-667). IEEE.

Joy Persial, G., Prabhu, M., & Shanmugalakshmi, R. (2011). Side channel attack-survey. Int J Adva Sci Res Rev, 1(4), 54-57.

Perianin, T., Carré, S., Dyseryn, V., Facon, A., & Guilley, S. (2020). End-to-end automated cache-timing attack driven by machine learning. Journal of Cryptographic Engineering, 1-12.

Lerman, L., Bontempi, G., & Markowitch, O. (2011). Side channel attack: an approach based on machine learning. Center for Advanced Security Research Darmstadt, 29.

Won, Y. S., Chatterjee, S., Jap, D., Bhasin, S., & Basu, A. (2021). Time to Leak: Cross-Device Timing Attack On Edge Deep Learning Accelerator. In 2021 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

Käsper, E., & Schwabe, P. (2009, September). Faster and timing-attack resistant AES-GCM. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 1-17). Springer, Berlin, Heidelberg.

Ambrose, J. A., Parameswaran, S., & Ignjatovic, A. (2008, November). MUTE-AES: A multiprocessor architecture to prevent power analysis based side channel attack of the AES algorithm. In 2008 IEEE/ACM International Conference on Computer-Aided Design (pp. 678-684). IEEE.

Dhem, J. F. (1998). Design of an efficient public-key cryptographic library for RISC-based smart cards (Doctoral dissertation, UCL-Université Catholique de Louvain).

Walter, C. D. (1999). Montgomery exponentiation needs no final subtractions. Electronics letters, 35(21), 1831-1832.

Walter, C. D. (2002, February). MIST: An efficient, randomized exponentiation algorithm for resisting power analysis. In Cryptographers' Track at the RSA Conference (pp. 53-66). Springer, Berlin, Heidelberg.

Hachez, G., & Quisquater, J. J. (2000, August). Montgomery exponentiation with no final subtractions: Improved results. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 293-301). Springer, Berlin, Heidelberg.

Schindler, W. (2015, September). Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 229-247). Springer, Berlin, Heidelberg.

Mukherjee, A. (2015). Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints. Proceedings of the IEEE, 103(10), 1747-1761.

Kim, J. T. (2017, May). Analyses of secure authentication scheme for smart home system based on internet on things. In 2017 International Conference on Applied System Innovation (ICASI) (pp. 335-336). IEEE.

Katagi, M., & Moriai, S. (2008). Lightweight cryptography for the internet of things. Sony Corporation, 2008, 7-10.

Kaps, J. P. (2008, December). Chai-tea, cryptographic hardware implementations of xtea. In International Conference on Cryptology in India (pp. 363-375). Springer, Berlin, Heidelberg.

Kim, Y., & Yoon, H. (2014). First Experimental Result of Power Analysis Attacks on a FPGA Implementation of LEA. IACR Cryptol. ePrint Arch., 2014, 999.

Williams, D. (2008). The tiny encryption algorithm (tea). Network Security, 1-14.

Zhao, X. J., Wang, T., & Zheng, Y. (2009). Cache Timing Attacks on Camellia Block Cipher. IACR Cryptol. ePrint Arch., 2009, 354.

Walter, C. D., & Thompson, S. (2001, April). Distinguishing exponent digits by observing modular subtractions. In Cryptographers' Track at the RSA Conference (pp. 192-207). Springer, Berlin, Heidelberg.

Kocher, P., Jaffe, J., & Jun, B. (1999, August). Differential power analysis. In Annual international cryptology conference (pp. 388-397). Springer, Berlin, Heidelberg.

Tiri, K., & Verbauwhede, I. (2003, September). Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 125-136). Springer, Berlin, Heidelberg.

Coron, J. S. (1999, August). Resistance against differential power analysis for elliptic curve cryptosystems. In International workshop on cryptographic hardware and embedded systems (pp. 292-302). Springer, Berlin, Heidelberg.

Goubin, L., & Patarin, J. (1999, August). DES and differential power analysis the "Duplication" method. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 158-172). Springer, Berlin, Heidelberg.