Tek Kullanımlık Şifre Üretimi ve Uygulamaları

Muhammed YOZĞATLI

Özet- Günümüzde internetin hızla yayılmasıyla birlikte, dijital ortamda gerçekleşen çeşitli işlemler için güvenli kimlik doğrulama yöntemlerine olan ihtiyaç artmıştır. Elektronik devlet uygulamalarından e-ticarete kadar geniş bir yelpazede hizmet sunan platformlar, kullanıcı bilgilerini koruma amacıyla gelişmiş güvenlik önlemlerine ihtiyaç duymaktadır.
Bu bağlamda, projemizde kullanıcıların kimlik doğrulama sürecini güçlendirmek ve siber tehditlere karşı daha güvenli bir çevrimiçi deneyim sağlamak amacıyla Tek Kullanımlık Şifreler (OTP) üzerine odaklanmaktayız. Projemizde, özellikle e-posta yoluyla gönderilen doğrulama kodlarına odaklanan Zaman-tabanlı Bir Kerelik Şifre Algoritması (TOTP) kullanılmaktadır.
OTP, her giriş işlemi için tek seferlik olarak kullanılan şifrelerden oluşmaktadır. Projemizde, Zaman-tabanlı Bir Kerelik Şifre Algoritması (TOTP) ile kullanıcılara her oturum açma işleminde değişen güvenli şifreler sunulmaktadır. Bu algoritma, zaman senkronizasyonuna dayanarak güvenli bir kimlik doğrulama süreci sağlamaktadır.
Projemiz, siber saldırılara karşı dirençli, güvenilir ve kullanıcı dostu bir kimlik doğrulama mekanizması sunmayı hedeflemektedir. TOTP algoritması sayesinde kullanıcılar, her girişte güncellenen tek kullanımlık şifrelerle hesaplarını koruma altına alabilmektedir. Bu proje, çevrimiçi platformlarda güvenliği artırmak ve kullanıcı bilgilerini koruma odaklı bir yaklaşım sunmak adına önemli bir adımdır.

Anahtar Kelimeler- Tek Kullanımlık Şifreler, Çevrimiçi güvenlik, Güvenli kimlik doğrulama, Zaman-tabanlı Bir Kerelik Şifre Algoritması (TOTP)

I. GİRİŞ

İnsanlar, tarih boyunca bilgi güvenliği konusunda endişelerini dile getirmişlerdir. Günümüzde teknolojinin hızlı gelişimi ve birçok bilgi kaynağının paylaşılması, bilgi güvenliğinin önemini daha da artırmaktadır. Bu nedenle, askeri amaçlardan, banka uygulamalarına, kamu kurumlarındaki uygulamalardan kişisel e-posta iletimlerine kadar birçok alanda bilginin güvenliği hayati önem taşımaktadır. Bilgi güvenliği, bilginin izinsiz veya yetkisiz bir biçimde erişim, kullanım, değiştirilme, ifşa edilme, ortadan kaldırılma, el değiştirme ve hasar verilmesini önlemek olarak tanımlanır. Bilgi güvenliği, gizlilik, bütünlük ve erişilebilirlik olarak isimlendirilen üç temel unsurdan meydana gelir. Bu üç temel güvenlik öğesinden herhangi biri zarar görürse güvenlik zaafiyeti oluşur. Gizlilik, bilginin yetkisiz kişilerin eline geçmeme ve yetkisiz erişime karşı korunmasıdır. Bütünlük, bilginin yetkisiz kişiler tarafından değiştirilmemesidir. Erişilebilirlik, bilginin yetkili kişilerce ihtiyaç duyulduğunda ulaşılabilir ve kullanılabilir durumda olmasıdır[1].

Şifre bilimi olarak adlandırılan kriptoloji, şifreleme (kriptografi) ve şifre analiz (kriptoanaliz) olmak üzere iki ana başlıktan oluşmaktadır. Şifreleme bilgi güvenliği ile uğraşır yani bilginin gizlenmesini, saklanmasını amaçlar. Şifre Analizi ise şifrelemenin tam tersidir yani güvenli bilginin kırılması ve belirlenmesi için çalışılır [2]. Bilgi güvenliği, bilgileri izinsiz erişimlerden, kullanımından, ifşa edilmesinden, yok edilmesinden, değiştirilmesinden veya hasar verilmesinden koruma işlemidir [2]. Bilgi güvenliği ilk zamanlardan beri değişiklik uygulamaları olmakla beraber günümüzde genellikle matematiksel temellere ve işlemlere dayanan algoritmalarla sağlanmaktadır.
Günümüzde, banka hesaplarına ve kimlik doğrulama gerektiren diğer uygulamalara erişimde sabit şifre kullanımının yetersiz olduğu gözlemlenmektedir. Bu sistemler, artan güvenlik tehditlerine karşı etkisiz kalmaktadır, özellikle internet üzerinden yapılan alışveriş ve havale gibi işlemlerin güvenli olmayan ortamlarda gerçekleşmesi durumu, kimlik doğrulama yöntemlerini daha kritik hale getirmektedir. Akıllı kartlar, biyometrik sistemler ve şifreleme sistemleri olmak üzere üç ana kategoride değerlendirilen kimlik doğrulama yöntemleri, güvenlik açısından daha etkili alternatifler sunmaktadır. Bu nedenle, finansal işlemler ve kişisel bilgilerin korunması adına daha güçlü güvenlik önlemleri benimsenmelidir[3].
Kimlik doğrulama için kullanılan şifreler, genellikle statik ve dinamik olmak üzere iki ana kategoride yer almaktadır. Statik şifreler, her erişimde aynı kalır ve kullanıcıdan alınıp önceden saklanan şifre ile karşılaştırılır; eğer eşleşme sağlanırsa kimlik doğrulama gerçekleşir. Öte yandan, dinamik şifreler her erişimde değişir ve kullanıcı her seferinde farklı bir şifre kullanarak kimlik doğrulama işlemini tamamlar. Bu dinamik şifrelere genellikle Tek Kullanımlık Şifreler (OTP) veya kullan at şifreler denir. OTP yöntemi, kullanılan şifrelerin tekrar kullanılmamasını sağlar; bu nedenle, bir oturumda elde edilen şifre dahi olsa, bu bilgi ile önemli kaynaklara erişim mümkün olmaz[4].
OTP üretimi için çeşitli yöntemler mevcuttur. Bu çalışmada Senkron zamanlı OTP standardına bir örnek olarak “Zaman-tabanlı Bir Kerelik Şifre Algoritması (TOTP)” kullanılmıştır.

II. Zaman-Tabanlı Bir Kerelik Şifre Üretimi

Kullanıcıların kimliklerini doğrulama için mail yoluya iki faktörlü kimlik (2FA) temelli bir uygulamadır. Zaman tabanlı bir kerelik şifre (TOTP) ve HMAC tabanlı bir kerelik şifre (HOTP) kullanılır. Her 60 saniyede bir üretilen 6 haneli şifre ile desteklenen uygulamalara istemci taraflı şifre ile erişim sağlayamaya yarayan bir uygulamadır. TTOP ve HTOP kavramlarını açıklayacak olursak; TTOP (Time-Based One Time Password): Paylaşılan bir gizli anahtardan ve o anki zamandan tek seferlik bir parola hesaplayan bir algoritmadır ve aşağıdaki bileşenlere sahiptir:

- Paylaşılan Sır (Shared Secret)
- Geçerli zamandan türetilmiş bir girdi
- Bir imzalama fonksiyonu

HTOP (HMAG-based One Time Password): Tek seferlik parola oluşturmak için HMAC algoritmasını kullanan bir algoritmadır. HMAC (Hash-based Message Authentication Code) Algoritması, Hash tabanlı ileti kimlik doğrulama kodunu ifade eder ve bir değeri imzalamak için SHA1 kullanan algoritmadır. Orjinalliğin doğrulanmasında kullanılır ve yalnızca sırrı bilen insanlar aynı girdi için aynı çıktıyı üretir.

Google Authenticator parametreleri:

- SHA1 HMAC
- Base32 harici anahtar 34
- 80 Bit Gizli Anahtar
- 64 Bit zamana dayalı mesaj boyutu
- 60 Saniye periyodu
- 6 Digit kod uzunluğu

Bu parametrelerin kullanımını bir sözde kod (Pseudo Code) ile tanımlayacak olursak; Paylaşılan sır (Secret) manuel olarak ya da QR kod ile telefona girilmelidir. Bu kodu bir değişken ile gösterecek olursak:

Orjinal_Gizli_Key = xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx, bu değer 32 karakter yani 256 bittir.

Tüm byte değerleri görüntülenebilir olmadığı için küçük harfler ve boşluklar kaldırılarak Base32 dönüşümü yapılır;

Gizli_Key = Base32_decode (TO_UpperCse (Remove_spaces (Orjinal_Gizli_Key)

Bir sonraki aşama olarak input değerinin geçerli zamandan türetilmesi işlemidir. Bu işlemde unix zamanı kullanılmaktadır.60 saniye zaman aralığında sabit kalacak bir değer elde etmek için unix zamanı 60 saniyeye bölünmektedir.

Giriş = Current_Unix_Time ()/60

Sonraki aşamada HMAC-SHA1 imzalama fonksiyonu kullanılmaktadır;

Hmac = HMAC-SHA1(Gizli_Key, Giriş)

Hmac değeri standart uzunlukta bir SHA1 değeridir (20-byte, 40 hex karakteri). 20 Byte SHA1 ‘yı 6 basamaklı bir sayıya çevirmek için SHA1 ‘nın 4 bitini ve bu endeksten sonraki 4 baytı kullanıp 32 bitlik işaretsiz tam sayıya dönüştüreceğiz.

Buyuk_Sayi=INT (Hmac [Last_Byte (Hmac): Last_Byte (Hmac)+4]) bu değer 232-1 bir sayı olabilir dolayısıyla 7 haneli bir sayıya bölerek kalanı 6 haneli bir sayı garanti edilir**.**
6Haneli_Sayi = Buyuk_Sayi %1,000,000 // 6 Haneli sayı üretilmiş olur[5].

III. OTP E-Posta Doğrulama Uygulaması

Günümüzde dijitalleşme ve internet kullanımının artmasıyla birlikte, çeşitli online platformlarda güvenli kimlik doğrulama yöntemleri önem kazanmaktadır. Kullanıcı bilgilerinin güvenliğini sağlamak ve siber tehditlere karşı dirençli olmak adına birçok uygulama, geleneksel kimlik doğrulama yöntemlerini güçlendirmek amacıyla farklı stratejilere yönelmektedir[6]. Python ile tasarlanan bu uygulama, kullanıcıların güvenli online deneyimlerini artırmak amacıyla, özellikle e-posta üzerinden alınan doğrulama kodlarına odaklanarak Tek Kullanımlık Şifreler (OTP) ile kimlik doğrulama sürecini ele alacağız.

1. E-Posta ve Güvenli Kimlik Doğrulama:

Geleneksel kimlik doğrulama yöntemleri, siber tehditlere karşı zayıf kalabilmekte ve bu noktada e-posta tabanlı güvenlik önlemleri önem kazanmaktadır. E-posta, kullanıcıların çeşitli platformlara erişimini sağlayan kritik bir bağlantı noktasıdır. Şekil 1'deki kod parçacığı, kullanıcının e-posta adresini girmesi ve şifre talep etmesi için gerekli arayüzü sağlar.

```
20
21      # E-posta etiketi ve giriş alanı
22      self.email_label = tk.Label(self.root, text="E-posta Adresi", font=("Arial", 12))
23      self.email_label.grid(row=0, column=1, pady=10, padx=10)  # Yatay boşluk
24
25      self.email_entry = tk.Entry(self.root, font=("Arial", 12), width=35)  # width parametresi ile genişlik ayarlanıyor
26      self.email_entry.grid(row=1, column=1, pady=10, padx=10)
27
28      # Gönder butonu
29      self.send_button = tk.Button(self.root, text="Gönder", command=self.send_otp, font=("Arial", 12), bg="green", fg="white")
30      self.send_button.grid(row=2, column=1, pady=10, padx=10)
31
```

Şekil 1 E-posta Giriş Alanı Ve Şifre Talebi İçin Arayüz Oluşturma

2. Tek Kullanımlık Şifrelerin Rolü:

Uygulamamız, kullanıcıların e-posta aracılığıyla alacakları tek kullanımlık şifrelerle kimlik doğrulamasını sağlamaktadır. Bu, her oturum için benzersiz şifreler kullanarak güvenli bir giriş süreci sunmayı hedefler. Şekil 2.1 ve Şekil 2.2'de tek kullanımlık şifrelerin kullanıcılar için nasıl oluşturulduğunu gösterir. Tek kullanımlık şifreler, her oturumda değişen güvenli şifrelerdir. Bu şifreler, zaman tabanlı bir algoritma kullanılarak oluşturulur.

```
32      # OTP etiketi ve giriş alanı
33      self.otp_label = tk.Label(self.root, text="Tek Kullanımlık Şifre", font=("Arial", 12))
34      self.otp_label.grid(row=6, column=1, pady=10, padx=10)
35
```

Şekil 2.1 Tek Kullanımlık Şifre Giriş Yeri

Şekil 2.2'de kod parçacığı, TOTP (Time-Based One-Time Password) algoritmasını kullanarak tek kullanımlık şifre üretir. Bu işlev, her çağrıldığında geçerli bir tek kullanımlık şifre oluşturur. Secret_key, her kullanıcı için benzersiz ve gizli bir anahtardır[7].

```
54      def generate_otp(self, secret_key):
55
56          totp = pyotp.TOTP(secret_key, interval=60)
57          otp = totp.now()
58          return otp
59
```

Şekil 2.2 Tek Kullanımlık Şifre Oluşturma

3. Python Tabanlı OTP Uygulaması:

Python programlama dili ve Tkinter grafiksel kullanıcı arayüzü kullanılarak geliştirilen uygulama, kullanıcının e-posta adresini kullanarak tek kullanımlık şifre talep etmesini ve güvenli bir şekilde oturum açmasını sağlar. Şekil 3.1'da E-posta gönderme işlemi, kullanıcının talep ettiği tek kullanımlık şifreyi güvenli bir şekilde iletmeyi amaçlar. Aşağıda yer alan Python kod parçacığı, SMTP (Simple Mail Transfer Protocol) protokolü aracılığıyla e-posta gönderme işlemini gerçekleştirir. Bu işlem, uygulamanın temel iletişim mekanizmasını oluşturur.
Bu kod parçacığı, uygulamanın e-posta gönderme işlemini gerçekleştiren işlevselliği içerir. Kullanıcıya ait e-posta adresi, gönderici e-posta adresi ve şifre ile birlikte iletilen şifreleme işlemleri, güvenli bir iletişim sağlar.

```
60      def send_email(self, sender_email, app_password, receiver_email, subject, body):
61          # E-posta gönderme işlemi
62          try:
63              # E-posta başlığı ve içeriği oluştur
64              message = MIMEMultipart()
65              message["From"] = sender_email
66              message["To"] = receiver_email
67              message["Subject"] = subject
68              message.attach(MIMEText(body, "plain"))
69
70              # SMTP bağlantısı oluştur
71              server = smtplib.SMTP("smtp.gmail.com", 587)
72              server.starttls()
73
74              # E-posta hesabıyla giriş yap
75              server.login(sender_email, app_password)
76
77              # E-postayı gönder
78              server.sendmail(sender_email, receiver_email, message.as_string())
79
80              # SMTP bağlantısını kapat
81              server.quit()
82
83              messagebox.showinfo("Bilgi", "E-posta başarıyla gönderildi.")
84
85          except Exception as e:
86              messagebox.showerror("Hata", f"E-posta gönderme hatası:\n{str(e)}")
87
```

Şekil 3.1 SMTP Protokolü İle E-posta Gönderme İşlemi

Şekil 3.2'de kod parçacığı "send_otp" fonksiyonunu daha açıklayıcı hale getirir. Bu fonksiyon, kullanıcının girdiği e-posta adresine tek kullanımlık bir şifre göndermek için gerekli adımları içerir. Gönderen e-posta adresi, şifresi, alıcı e-posta adresi ve oluşturulan tek kullanımlık şifre bilgileri bu fonksiyon içinde yer almaktadır.

```
88      def send_otp(self):
89          sender_email = "ates.[redacted]@gmail.com"
90          app_password = "[redacted]"
91          receiver_email = self.email_entry.get()
92
93          # E-posta başlığı ve içeriği
94          otp_secret_key = pyotp.random_base32()
95          self.otp = self.generate_otp(otp_secret_key)
96          subject = "Tek Kullanımlık Şifre"
97          body = f"Giriş Şifreniz {self.otp} olarak oluşturulmuştur."
98
99          # E-postayı gönder
100         self.send_email(sender_email, app_password, receiver_email, subject, body)
101
102         # Geri sayımı başlat
103         self.start_countdown()
```

Şekil 3.2 Kullanıcının Girdiği E-posta Adresine Tek Kullanımlık Bir Şifre Göndermek İçin Gerekli Adımları İçerir

4. Güvenli Yönlendirme ve Zaman-tabanlı Şifreler:

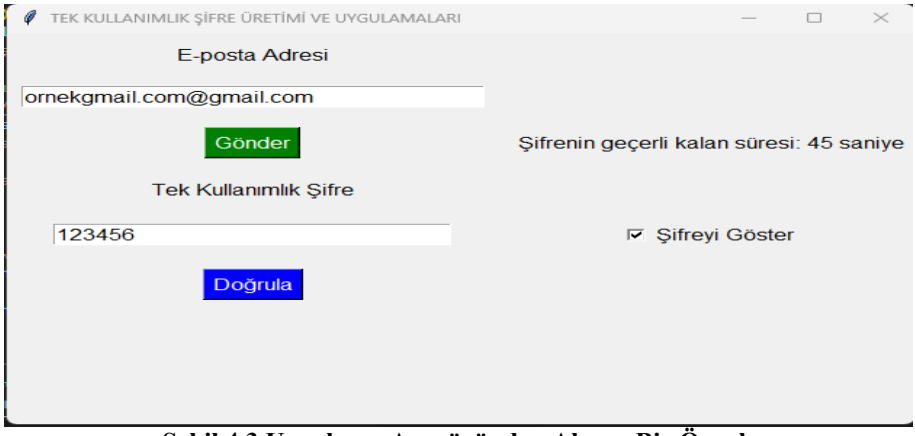
Kullanıcı, doğru şifreyi girdiği takdirde başarılı bir şekilde oturum açma işlemini tamamlar ve belirli bir siteye güvenli bir şekilde yönlendirilir. Zaman-tabanlı şifreler, her oturumda değişen güvenli şifreleri temsil eder. Şekil 4.1 ve Şekil 4.2'deki kod parçacıkları, kullanıcının şifresini doğrulama ve zaman tabanlı bir geri sayımı güncelleme işlemlerini içerir.

```
105     def verify_otp(self):
106         girdi = self.otp_entry.get()
107         if girdi == self.otp and self.remaining_time > 0:
108             messagebox.showinfo("Başarılı", "SAYFAYA YONLENDIRILIYORSUNUZ...")
109
110             # Başarılı oturum açıldıktan sonra belirli bir siteye yönlendir
111             webbrowser.open("https://ue.beun.edu.tr/Account/LoginBefore")
112         elif girdi == self.otp and self.remaining_time <= 0:
113             messagebox.showinfo("Uyarı", "Üzgünüz, süreniz doldu. Yeni şifre alınız.")
114         else:
115             messagebox.showerror("Hata", "Doğrulama başarısız. Erişim Reddedildi...")
116
```

Şekil 4.1 Şifre Doğrulama

```
    def start_countdown(self):
        # Geri sayım işlemi için threading kullanılıyor
        countdown_thread = threading.Thread(target=self.update_countdown)
        countdown_thread.start()

    def update_countdown(self):
        while self.remaining_time > 0:
            time.sleep(1)  # 1 saniye bekle
            self.remaining_time -= 1

            # Geri sayım etiketini güncelle
            self.countdown_label.config(text="Şifrenin geçerli kalan süresi: {} saniye".format(self.remaining_time))

        # Geri sayım tamamlandığında etiketi güncelle
        self.countdown_label.config(text="Süre bitti yeni şifre alın!")

    def run(self):
        self.root.mainloop()
```

Şekil 4.2 Şifrenin Kalan Geçerli Süresi



Şekil 4.3 Uygulama Arayüzünden Alınan Bir Örnek

SONUÇ

Günümüzdeki hızlı dijitalleşme ve teknolojik ilerleme, online platformlarda gerçekleşen birçok işlemi daha da karmaşık hale getirirken, güvenli kimlik doğrulama yöntemlerine olan ihtiyacı da artırmaktadır. Elektronik devlet uygulamalarından e-ticarete kadar geniş bir yelpazede hizmet sunan bu platformlar, kullanıcı bilgilerini korumak adına gelişmiş güvenlik önlemlerine ihtiyaç duymaktadır.

Bu bağlamda, Tek Kullanımlık Şifreler (OTP) odaklı projemiz, çevrimiçi kimlik doğrulama süreçlerini güçlendirmeyi ve siber tehditlere karşı daha dirençli bir çözüm sunmayı amaçlamaktadır. Projemizde, özellikle e-posta aracılığıyla alınan doğrulama kodlarına odaklanan Zaman-tabanlı Bir Kerelik Şifre Algoritması (TOTP) kullanılmaktadır.

Kimlik doğrulama süreçlerinde güvenlik açısından kritik olan bu yöntem, her oturum için farklı, tek seferlik şifreler oluşturarak kullanıcı hesaplarını ekstra bir koruma katmanıyla donatmaktadır. Bu, geleneksel şifreleme yöntemlerine kıyasla daha güçlü bir güvenlik sağlar.Projemizin ana bileşenleri arasında kullanıcı arayüzü, TOTP algoritması, e-posta gönderme işlemi ve güvenli oturum açma süreci bulunmaktadır. Kullanıcılar, kullanıcı dostu bir arayüz üzerinden e-posta adreslerini girebilir ve talep ettikleri tek kullanımlık şifreyi e-posta yoluyla alabilirler.

Geliştirilen TOTP algoritması, her oturumda güncellenen ve belirli bir süre içinde geçerli olan şifreleri üretir. E-posta gönderme işlemi, SMTP protokolü aracılığıyla güvenli bir şekilde gerçekleştirilir, böylece doğrulama kodları güvenli bir iletişim kanalıyla kullanıcılara iletilir.

Kullanıcılar doğru şifreyi girdiklerinde başarılı bir oturum açma gerçekleştirilir ve belirli bir siteye yönlendirilirler. Bu süreç, kullanıcıların çevrimiçi platformlarda güvenli bir deneyim yaşamalarını sağlar.

Sonuç olarak, Tek Kullanımlık Şifreler (OTP) üzerine odaklanan bu proje, çevrimiçi güvenliği artırmak ve kullanıcı bilgilerini daha etkili bir şekilde korumak adına önemli bir adımdır. Sürdürülebilir ve güçlü bir kimlik doğrulama yöntemi olarak OTP, gelecekteki dijital güvenlik çözümlerinde önemli bir rol oynamaya devam edecektir.

REFERANSLAR

[1] Bhole A. T., Chaudhari S., 2013. Web Based Security using Online Password Authentication in Mobile Application, International Journal of Science and Research (IJSR), vol. 2319-7064, no. 2, pp. 74-77, Haziran.
[2] (Kasım, 2013) TÜBİTAK UEKAE Açık Anahtar Altyapısı Eğitim Kitabı, Şifreleme. [Online]. http://www.kamusm.gov.tr/dosyalar/kitaplar/aaa/
[3] Tsai C.S., Lee C.C., Hwang M.S., 2005. Password Authentication Scheme: Current Status and Key Issues, International Journal of Network Security, Vol. 3, No 2, Sayfa 101-115, Eylül.
[4] Yinxiang L., Li X., Zhong L., Jing Y., 2010. One-time Password Authentication Scheme Authentication System and Apllication in Banking Financial System OneTime Password Authentication Scheme Authentication System and Apllication in Banking Financial System, Networked Computing and Advanced Information
[5]Management (NCM), 2010 Sixth International Conference on, Seoul, pp. 172 - 175.
Comparison Of Cryptographic Hash Functions.(2019).11 Aralık 2019 tarihinde https://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions
[6]Stallings W. , Cryptography and Network Security Principles and Practices, Chapter 51 11 Cryptographic Hash Functions, Fifth Edition ed.: Prentice Hall, 2011.