Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

SSLChain: Blokzincir Yöntemiyle Sunucu-İstemci Arası Güvenli Web Haberleşmesi

Durdu ÖZDEN ONAR [a,*], Resul KARA [b]

[a] *Bilgisayar Mühendisliği ABD, Fen Bilimleri Enstitüsü, Düzce Üniversitesi, Düzce, TÜRKİYE*
[b] *Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE*
** Sorumlu yazarın e-posta adresi: durduozden42@gmail.com*
DOI: 10.29130/dubited.866319

Öz

Blokzincir teknolojisi adını bitcoin ile duyurmuş olsa da yapabilecekleri sadece kripto para ile sınırlı kalmayıp birçok teknolojide kullanılabilir. Blokzinciri, sürekli büyüyen bir veri yapısı olup dağıtık bir mimariye sahiptir. İşlemler değiştirilemez, yapılan işlemleri ve bir önceki bloğun adresi bloklara yazılır. Blokzinciri merkezi bir sistem olmayıp ağa katılan tüm katılımcıların zincirde bir halka olarak yer almasını sağlar. Bu çalışmada; günümüzde web sayfalarında istemci ve sunucu arasında iletişim kurmada kullanılan SSL/TLS işlevini yerine getirecek yeni bir yöntem olan blokzincir kullanılarak güvenli bağlantı önerilmiştir. Sunucu-istemci arasındaki anahtar paylaşımının yönetilmesinde blokzincirinin görev alacağı şekilde tasarım yapılmıştır. Bu durumda zincire yeni bir halka dahil edilerek güvenliğin artırılması kolaylaşmaktadır.

Anahtar Kelimeler: *Blokzinciri, SSL, Güvenlik, Dağıtık mimari, SSLChain*

SSLChain: Client-Server Secure Web Communication With Blockchain Method

Abstract

Although the term blockchain has been heard with bitcoin, it can be used in many technologies, not just crypto money. Blockchain has a distributed architecture and is a constantly growing data structure. Data blocks cannot be changed, each block has its own data and the previous block's address data. Blockchain does not have a centralized system, it ensures that all participants participating in the network are included in the chain as a link. In this paper; Secure connection has been proposed using a new method, blockchain, instead of SSL / TLS, which is used to communicate between client and server on web pages. It is designed to be used as a blockchain in the management of key sharing between server and client. In this case, it becomes easier to increase security by including a new link in the chain.

Keywords: *Blockchain, SSL, Security, Distributed architecture, SSLChain*

I. GİRİŞ

İletişim kurma eğilimi insanın temel ihtiyaçlar arasında yer alır. Kayda alınan tarihsel olaylardan da anlaşılacağı üzere insanlığın bir süre sesli iletişim kuramadan sadece el kol ve yüz ifadeleri ile iletişim kurduklarını belirlense de bir süre sesli iletişim ardından sesli ve yazılı iletişimi kullandığı anlaşılmıştır. İnsan yapısında iletişim kurma içgüdüsünü taşıdığı gibi mahremiyeti de birlikte taşımıştır. Mahremiyet kelimesi farklı zamanlarda farklı anlamları üzerine yüklese de temelinde gizliliği betimler. "Mahremiyet" kelimesi kendisine akademik metinlerde "kişinin kendisine ilişkin bir alanın olması, kendi başına kalabilmesi, yalnız kendisinin bildiği ve sadece kendi istediği kişilerle paylaştığı özel bilgi ya da niteliklerine ilişkin doğal, insani bir hak" gibi tanımlar bulmuştur [1]. İnsanların en temel manada iletişimlerini mahremiyet altına alma konusunu değerlendirecek olursak sözle iletişim kuran iki kişinin iletişimlerini 3. kişilerden gizlemek için 3. kişilerin duyamayacağı bir şekilde konuşmaları gerekir. Bununla ilgili iletişimdeki kişiler çeşitli yöntemler geliştirebilir. Temel amaç iletişim kuran kişiler arasında kesintisiz bir aktarım gerçekleştirmesini sağlamalarıdır. Aktarılan bilgilerin yetkisiz kişilerce erişilmesini engellemek aktarılan verilerin tamamının doğru şekilde anlaşılmasını sağlamak ve aktarılanların değiştirilmeden karşıya ulaştırılmasıdır [2]. Yöntemi nasıl olursa olsun iletişim sırasında gönderilen bilgi gizli, bütünlüğünü yitirmemiş ve erişilebilir olmalıdır. En ilkel yöntemden en gelişmiş yönteme kadar hepsinde bu kurallar geçerlidir.

1960'lı yıllarda ortaya çıkan ve 1990'lı yılların sonunda günümüze kadar aktif bir şekilde kullanılan internet ile birlikte iletişim teknikleri de dijitalleşmiş ve internet bir iletişim aracı haline gelmiştir [3]. İnternette iletişim ise belirli kurallara bağlanmış ve protokoller geliştirilmiştir [4]. Günümüzde bu protokolleri bilinçli ya da bilinçsiz olarak çok aktif bir şekilde kullanıyoruz. Bunların en başında gelen ise Hyper Text Transfer Protocol (HTTP) protokolüdür. HTTP, internette sunucular ve son kullanıcılar arasında bilgilerin nasıl aktarılacağına dair kurallar ve yöntemleri düzenleyen uygulama katmanında çalışan bir iletişim protokolüdür. Web sitesi görüntülemek ve üzerinde çeşitli işlemler yapmak için kullanılır [5]. İstemci sunucu arası metin transferi protokolüdür. İletişim sırasında herhangi bir şifreleme söz konusu değildir. Oluşan bu açığın kapatılması için Hyper Text Transfer Protocol Secure (HTTPS) protokolü geliştirilmiş iletişim şifreli hale getirilmiştir.

2018 yılında Uluslararası IEEE veri madenciliği konferansında yayınlanan 'Certificate Transparency Using Blockchain' makalesinde IBM'in Hyperledger Fabric blockchain platformunda herhangi bir sertifika otoritesinden sertifika onayı almaksızın blockzincirinde sertifikayı üretmiştir [13].

Bu çalışmada HTTP'nin güvenli versiyonu olan HTTPS'e alternatif bir sistem önerilmiştir. Ağda sadece istemci sunucunun paydaş olmadığı genel ağdaki paydaşların da iletişimdeki şifrelemeye katıldığı blokzinciri teknolojisi kullanılarak güvenli HTTPS bağlantısına alternatif güvenli bir HTTP + blokzinciri güvenliği önerilmiştir.

II. MATERYAL VE METOT

A. SSL/TLS YÖNTEMİ

Netscape tarafından 1994 yılında çıkarılan Secure Sockets Layer (SSL) diğer ismi ile Transport Layer Security (TLS) güvensiz ortamda verinin sunucu ve istemci arasında şifreli/güvenli bir şekilde haberleşmesini sağlamak için geliştirilen bir protokoldür [6]. Önceden bir şifreleme algoritması ile şifrelenerek ve sadece gönderilen alıcının çözebileceği, uygulama katmanı ile taşıma katmanı arasında bulunan verinin kriptografik işlemlerden geçerek istemciye ulaşmasını sağlar. Veriyi şifrelemek için genel anahtar veya diğer adıyla açık anahtar (public key), şifrelenen veriyi çözmek için özel anahtar veya diğer adıyla gizli anahtar (private key) kullanılır.

HTTP protokolündeki iletişimi SSL/TLS ile şifrelendiği için protokolün yeni adı HTTPS olmuştur. İstemci bir sunucuya erişmek istediğinde ilk olarak alan adının sertifikasını alır ve ulaşmak istediği sunucunun asıl gitmek istediği yer olduğu kesinleşmiş olur. İnternet üzerindeki iletişim bir mektuplaşmaya benzetilirse; gelen bir postanın doğru göndericiden geldiğinin bir kesinliği olamaz ya da gönderilen mektubun yolda üçüncü kişiler tarafından okunmayacağının bir garantisi olamaz. Bunun sebebi, içeriğinin herkesin anlayacağı bir şekilde gönderilmiş olmasıdır. İnternet ortamındaki web sayfalarının transferi için de aynı durum söz konusudur.

SSL/TLS'in ortaya çıkış motivasyonunda karşılıklı güven eksikliği vardır. İstemci bir istekte bulunmak istediğinde karşındaki kişinin ulaşmak istediği kişi olup olmaması güvensizliği bir sertifika ile aşılmaya çalışılmıştır. Sunucunun isteğe cevap olarak ortak anahtarı ile birlikte sertifika sunar. Fakat sertifikanın da doğrulanması gereklidir. Sertifikanın ispatlanması için bir veya daha fazla sertifika yetkilisinden oluşan güvenilir kuruluşlara ihtiyaç duyulur [7].

SSL sertifikaları, çok sayıda güvenli kuruluş tarafından sağlanmaktadır. SSL 'de anahtar uzunluğu en az 256 bit kullanılarak güvenli istemci-sunucu haberleşmesi gerçekleştirilebilir [8].

İstemci bağlanmak istediği sunucuya Transmission Control Protocol (TCP) 443 portu üzerinden istek gönderir. Sunucunun yapılandırılmasına göre bu port değiştirilebilir ancak varsayılan port numarası 443'tür. Sunucu, istemcinin bu isteğine karşılık alan adına tanımlı sertifikasını istemciye gönderir. İstemci sertifikayı üreten kurumunun sağladığı servisler ile sertifikayı doğrular. Sunucudan gelen sertifika ile birlikte açık anahtar da alınmış olur. Sertifikanın geçerlilik kontrolleri tamamlandıktan sonra, istemci, oturum için ürettiği iletişimde kullanılacak anahtarı sunucudan gelen açık anahtar ile şifreleyerek sunucuya gönderir. İstemci tarafından oluşturulan bu anahtar artık oturum boyunca iletişimde kullanılır. Sunucu istemciden gelen şifrelenmiş metni kendi gizli anahtarı ile açarak oturumdaki anahtarı elde etmiş olur. Bu aşamadan sonra istemci ve sunucu karşılıklı iletişimlerinde bu anahtarı kullanırlar.

B. BLOKZİNCİR YÖNTEMİ

Blokzincir; kayıtların tutulduğu ve kayıtların herkes tarafından izlendiği merkezi olmayan dağıtık veritabanı yapısıdır. Literatürdeki adı "Blockcahin" olup dilimize blokzincir olarak yerleşmiştir. Ekonomide meydana gelen 2008 yılındaki finans krizi sonucu dijital para olarak bilinen Bitcoin ortaya çıkmıştır. Blokzincir ilk kimliği tam olarak belli olmayan Satoshi Nakamoto isimli biri tarafından "Bitcoin: A Peer-to-Peer Electronic Cash System" başlıklı makalede ortaya çıkmıştır [9]. Bitcoin hiçbir merkezi otoriteye bağlı olmadan arada aracı bir banka olmadan katılan herkese açık olan ve tamamen ağ üzerinden yönetilen dağıtık bir yapıdır. Bu da tam olarak Blokzinciri yapısını oluşturmaktadır. Ancak yapabilecekleri bununla sınırlı olmayan, günümüz teknolojileri uygulamalarının geliştirilmesi ve daha önce hiç kullanılmayan alanlara uygulanmasını sağlamıştır. Blokzinciri tamamen güvensiz ortamda meydana gelen kişiler arasında güveni sağlar. Blokzincir kullanımı için ortamın bu ihtiyacı hissetmesi gerekir. Blokzinciri sanal para yanında, sağlık, eğitim, oy kullanma, tedarik zinciri gibi pek çok alanda kullanılmaktadır. Kullanım alanı şeffaf ve merkezi olmayan yapısının anlaşılmasıyla farklı sektörlerde kullanılarak geniş bir yelpazeye sahip olmuştur [10]. Teknoloji, dağıtılmış bir defter yapısına ve konsensüs sürecine dayanmaktadır.

Blokzinciri temel olarak kayıtlar arasında mutlak bir bağın olduğu ve kopyalarının ağdaki tüm katılımcılara dağıtıldığı bir veri tabanıdır. Merkezi otoriteye ihtiyaç yoktur, tüm katılımcıların doğrulama yapabildiği bir dağıtık defter teknolojisi ortaya çıkar. Bir merkeze bağlı olmayan ağların, ağa saldırı olması durumunda sadece ilgili düğümü etkisiz hale getirip sistemin çökmesi engellenir [11].

İnternette haberleşmede güvenli bağlantı oluşturmanın bilinen en yaygın yöntemi sertifika kullanarak bağlantı oluşturmaktır. Bir otoriteden sertifika almanın belirli maliyetleri vardır. Dahası sertifikalar her ne kadar güvenli olursa olsun çeşitli saldırılara maruz kalmaktadır. Bu çalışmada herhangi bir

elektronik sertifika kullanmadan güvenli bir HTTP bağlantısı oluşturmak için blokzinciri gücünden faydalanılmıştır.

Literatürde blokzincir aracılığı ile güvenli bağlantıyı oluşturacak bir tarayıcı yazılımı ya da eklentisi mevcut değildir. Blokzinciri ile güvenli HTTP bağlantısını yapacak bir tarayıcının olmayışı dışarıdan harici bir uygulama ile yapılmasını zorunlu kılmıştır. Bu uygulama ileride tarayıcılar için ilham oluşturup blokzinciri için uygun tarayıcıların geliştirilmesine olanak sağlayacağı düşünülmektedir.

Önerilen yöntem SSL teknolojisine alternatif oluşturduğu için adına SSLChain denilmiştir.

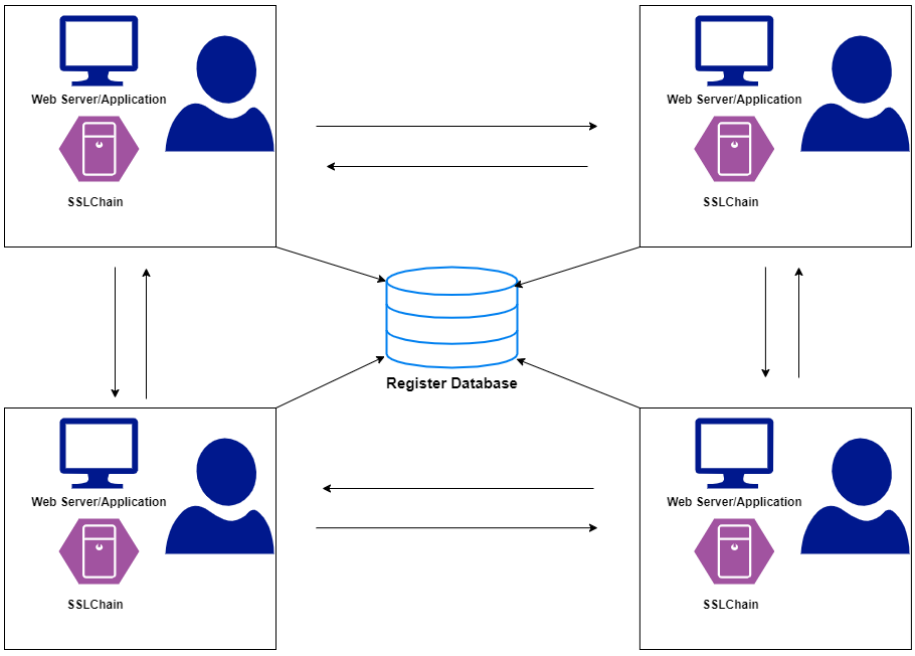
III. BLOKZİNCİRİ İLE SUNUCU-İSTEMCİ İLETİŞİMİ

Bu kısımda sunucu ile istemci arasında SSL/TLS ile güven oluşturma yöntemine alternatif olarak önerilen, blokzincir yönteminin kullanıldığı ve adına SSLChain dediğimiz yöntemin detayları verilmiştir.

A. SSLCHAİN ÇALIŞMA ADIMLARI

SSLChain, blokzinciri kullanılarak oluşturulacak güvenli bağlantıda blokların yönetiminden sorumlu bir çekirdek uygulamadır. Bu uygulama güvenli bağlantıyı sağlayacak bir P2P ağı ile merkezi olmadan çalışacak bir uygulamadır. HTTP protokolü üzerinden güvenli bir iletişim için önerilen yöntemi kullanan uygulama ağa katılacak tüm düğümlerde yüklü olması gerekir. Uygulama, mevcut tarayıcıların blokzinciri varsayılan olarak desteklememesi nedeniyle geliştirilmiştir. Güvenli bir iletişim için sunucuda çalışacak uygulamanın da blokzincir ile güvenli iletişime uyarlanması gerekmektedir. Mevcut sunucu sistemlerinde çalışan bir uygulamada uygulamayı geliştiren taraf, iletişimin güvenliğinden kendini sorumlu hissetmez. Bunun yerine bu görevi sunucuda geliştirilmiş uygulamalar ve tarayıcılar üstlenir. Bu çalışmaya konu olan güvenli iletişimi gerçekleştirecek uygun tarayıcı olmadığı gibi güvenli iletişimde sunucuda bu iletişim için bir aksiyon gerçekleştirmeyecektir. Bu görev uygulama geliştiriciye düşmektedir. İletişimin çekirdeğinde yer alacak SSLChain uygulaması yine HTTP üzerinde hizmet vereceğinden bir web servis olarak geliştirilmiştir.

Önerilen SSLChain yönteminin çalışma yapısı Şekil 1'de verilmiştir.



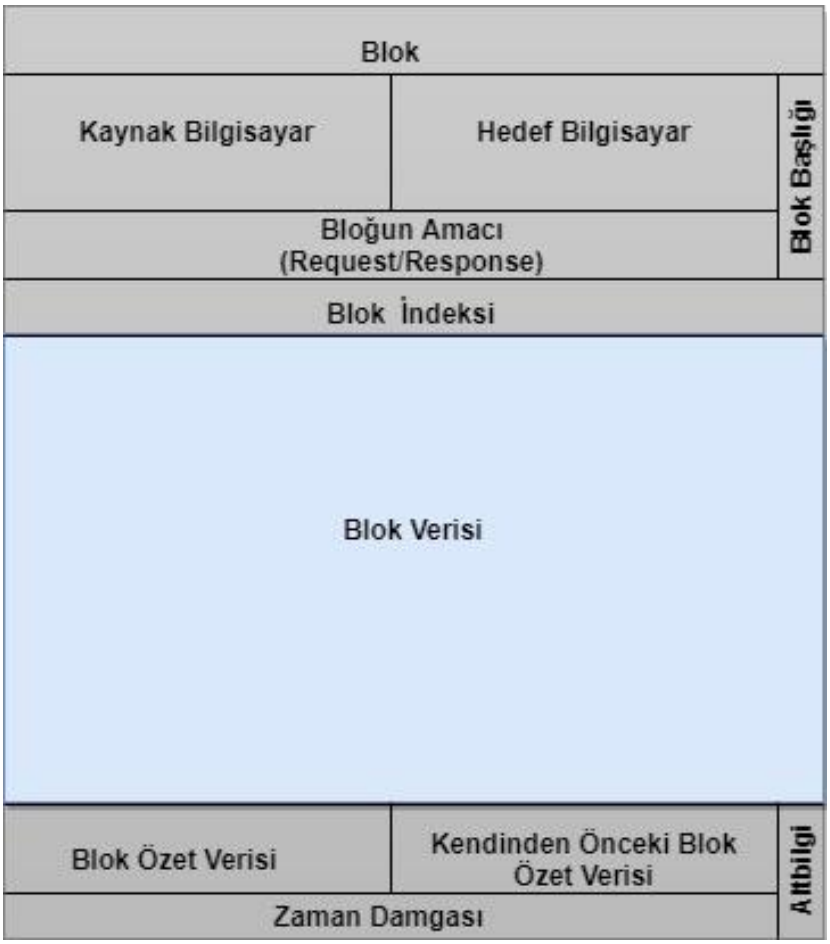
Şekil 1. *SSLChain çalışma yapısı*

SSLChain ağa bağlı düğümlerin HTTP üzerinden iletişim kurabilmesi için rest web servis olarak geliştirilmiştir. REST (Representational State Transfer), 2000 yılında Roy Fielding tarafından doktora tezinde tanıtılmış ve tanımlanmıştır. REST, dağıtık sistemler tasarlamak için kullanılan bir mimari tarzdır [12]. REST, istemci ile sunucu arasında HTTP/HTTPS protokolünün kullanılarak XML, JSON veya özelleştirilmiş formattaki verilerin aktarılmasını sağlayan bir mimaridir. REST mimarisini kullanan servislere ise RESTful servis denir.

Bir rest web servis geliştirmek için günümüzde birçok yazılım geliştirme dili mevcuttur. Aynı zamanda bu yazılım dilleri ile geliştirilmiş çok fazla framework bulunmaktadır. SSLChain için Java ile geliştirilmiş Springboot frameworku kullanılmıştır.

B. SSLCHAIN BLOK İÇERİĞİ

Bu çalışma kapsamında oluşturulan blokzinciri ile güvenli HTTP bağlantısı ekosisteminde her düğümde kullanıcının bilgisi dahilinde bilgisayarında çalışacak ve SSLChain sistemine dahil olacak blokların alınmasından ve gönderilmesinden sorumlu bir uygulama bulunur. Kullanıcılar her ne kadar kendi bilgisayarlarında çalışan uygulamayı bilseler dahi gelen giden veriler hakkında bilgi sahibi olamazlar. Gelen/giden blokların düğümlerde sabit disklere kaydedilmesi öngörülmemiştir. Bunun yerine düğümlerin belleklerinde tutulur. İletişim kuran iki düğüm oturumu bitirdiği anda bellekten silinir. Blokları sabit diske kaydedip sürekli olarak saklanması iletişim esnasında hassas verilerin olmasından dolayı daha sonra şifrelenmiş blokların içeriklerinin tespit edilmesi ihtimaline karşılık blokların silinmesi öngörülmüştür. Blokların içeriği önemli ölçüde transfer edilen verileri tutacak olsa da istemciye ve sunucuya ait veriler, oturuma ait veriler ve diğer verilere de sahip olacaktır. SSLChain blok yapısına Şekil 2'de yer verilmiştir.



Şekil 2. *SSLChain blok yapısı*

Şekil 2'de yer alan blok yapısı, bağlantı sırasında istemci ile sunucu arasında transfer edilecek bloğu ifade etmektedir.

Kaynak Bilgisayar: İstemci-sunucu arasında gerçekleşecek şifreli veri alışverişinde bloğu gönderen kişinin bilgilerinin tutulduğu alandır.

Hedef Bilgisayar: İstemci-sunucu arasında şifrelenmiş blok transferinde istek yapılacak ya da cevap verilecek düğümü ifade etmektedir.

Bloğun Amacı (Request/Response): Veri transferi sırasında isteğin ya da cevabın niteliğini belirleyen istemcinin ya da sunucunun bu alana göre cevap verme şekillerini değiştiren bir alandır. Bu alan anahtar paylaşımı sırasında anahtar paylaşımı, veri iletişimi sırasında ise istek veya cevap olarak değişecektir.

Blok İndeksi: Blokzinciri teknolojilerinde tüm zincirler tek bir örüntü üzerinde ilerlemektedir. Ancak SSLChain'deki bloklar oturum boyunca tutulup daha sonra yok edileceğinden ağdakilerin tamamını tek bir zincire bağlanmaması gerektiği öngörülmüştür. Bu alan her bir oturum için oturum kimlik alanı olarak öngörülmüştür.

Blok Verisi: İstemci-sunucu arasında gerçekleşecek veri transferi sırasında asıl verinin yani ekranlarda kullanılacak verilerin bloklarda tutulduğu alandır.

Blok Özet Verisi: Standart blokzinciri teknolojilerinde kullanılan blokların benzersizliğini sağlayan alanların doğruluğunu teyit etmek için kullanılır. Bu alan oluştururken yine blok içerisinde yer alan blok verisi, kendinden önceki blok özet verisi ve data alanlarının SHA-256 özetleme algoritması ile özetlenmesi ile oluşur.

Kendinden Önceki Blok Özet Verisi: Blokların birbirini bağlanması ve doğrulama için kullanılan alandır. Her bir blok yaratılırken önceki bloğun özet değeri alınarak bloğa eklenir.

Zaman Damgası: Blok oluşturulduğu anda oluşturulan veridir. Blok özeti oluşturulurken karmaşıklığı yükseltmek ve zaman bilgisini tutmak için kullanılmıştır.

IV. BULGULAR

Önerilen yöntemi kullanan restful servisinin işlevselliğini belirlemek için bir web sunucu ve 4 istemciden oluşan bir ağ oluşturulmuştur. Ağda saldırgan bir istemci tarafından trafik elde edilmeye çalışılmıştır.

Anahtar paylaşımı sırasında asimetrik şifreleme için AES algoritması 2048 bit anahtar uzunluğuyla kullanılmıştır. Anahtar istemci tarafından ortak anahtarı şifreleyip tekrar sunucuya göndermek için kullanılmıştır. Ortak anahtar için RSA algoritması 256 bit anahtar uzunluğu ile kullanılmıştır.

Saldırgan istemcinin haberleşme trafiğini elde edebilmesine rağmen verileri elde edemediği görülmüştür.

Oluşturulan ağda SSLChain ile sertifika ücreti ödemeksizin güvenli bağlantı kurulabilmiştir. Bağlantı sırasında uygulamanın hem avantajı hem de dezavantajı olan husus düğümlerin sayısının artmasıdır. Düğümlerin sayısının artması hem daha fazla düğümden doğrulama yapılması hem de her düğüm ile bilgi alışverişine olanak sağladığından çok fazla bekleme süresine sebep olmuştur.

Sistemin performans testi için sunucu rolüne bir bilgisayar, istemci olarak çalışacak dört terminalin olduğu bir ağ oluşturulmuştur. Aynı ağ bağlantısında gerçekleştirilen testte ağlar arası gecikme göz

ardı edilmiştir. Çalışan uygulama sadece anahtar paylaşımı ve basit bir istek ve cevaptan oluşturulmuştur. Uygulama içinde veritabanı sorgu gecikmesi olmamıştır. Tomcat uygulama sunucusunun ve kullanılan SpringBoot frameworkunun ortalama gecikme süreleri Tablo 1, Tablo 2 ve Tablo 3'te verilmiştir.

Tablo 1. *1 sunucu ve 2 istemciden oluşan ağ gecikmeleri*

1 sunucu ve 2 istemciden oluşan ağ

	Açık anahtarın üretilmesi ve düğümlere dağıtılması	**Gizli anahtarın sunucuya iletilmesi**	**İsteklerin cevaplanması**
sunucu	12ms	-	8ms
istemci-1	-	5ms	9ms
istemci-2	-	5ms	9ms

Tablo 2. *1 sunucu ve 3 istemciden oluşan ağ gecikmeleri*

1 sunucu ve 3 istemciden oluşan ağ

	Açık anahtarın üretilmesi ve düğümlere dağıtılması	**Gizli anahtarın sunucuya iletilmesi**	**İsteklerin cevaplanması**
sunucu	14ms	-	11ms
istemci-1	-	5ms	12ms
istemci-2	-	5ms	12ms
istemci-3	-	5ms	12ms

Tablo 3. *1 sunucu ve 4 istemciden oluşan ağ gecikmeleri*

1 sunucu ve 4 istemciden oluşan ağ

	Açık anahtarın üretilmesi ve düğümlere dağıtılması	**Gizli anahtarın sunucuya iletilmesi**	**İsteklerin cevaplanması**
sunucu	16ms	-	12ms
istemci-1	-	5ms	15ms
istemci-2	-	5ms	15ms
istemci-3	-	5ms	15ms
istemci-4	-	5ms	15ms

Yeni düğümlerin eklenmesi ile gecikmelerin doğrusal artış göstermediği sadece düğüm başına sunucuda her istemci için oluşturulan thread maliyetinin eklenmiş olduğu görülmüştür. Geniş ölçekli ağlarda her istemciye erişmek, açık anahtarı ve blokları dağıtmak maliyetli olacağından istemci grupları oluşturularak blokları dağıtmak daha sürdürülebilir olacaktır.

V. SONUÇ

Güvenli web istemci-sunucu haberleşmesi için önerilen blokzinciri teknolojisine dayalı SSLChain yöntemi HTTP bağlantılarının SSL/TLS sertifikaları kullanmadan verilerin güvenli bir şekilde

gönderilip alınmasını sağladığı görülmüştür. Ağdaki istemci sayısının artışının isteklerin cevaplanma zamanına olan etkisi incelenmiştir. Önerilen yöntemin kullanımı ile uygulama geliştiriciler ve uygulama sunucular için alternatif bir güvenli el sıkışma yöntemi elde edilmiştir.

Blokzincir yönteminin uygulanmasında ağdaki her istemciye erişim zaman gecikmelere yola açacağından önerilen yöntemin yüksek düğüm yoğunluğuna sahip ağlarda kullanımı için istemci ve sunucuyu içerisine alacak alt gruplar oluşturmaya yönelik yeni çalışmalara yol açabilecektir.

VI. KAYNAKLAR

[1] S. G. Dedeoğlu "Özgürlük, mahremiyet, demokrasinin değeri ve bilişim toplumunda maruz kaldığı tehditler," *Journal of Yasar University,* c. 9, s. 34, ss. 5889–5891, 2014.

[2] M. Işık, *İletişim Bilimine Giriş*, 4. baskı, Konya, Türkiye: Eğitim Yayınevi, 2018, ss. 150-220.

[3] B. Segal "A short history of Internet protocols at CERN," *CERN Computer Newsletter No.2001-001 In Section 'Internet Service'*, ss. 1-3, 1995.

[4] A. Shiranzaei ve R. Z. Khan "Internet protocol versions," *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, New Delhi, Hindistan, 2015, ss. 397-400.

[5] A. Goldberg, R. Buff ve A. Schmitt "A comparison of HTTP And HTTPS performance," *Computer Measurement Group (CMG98) International Conference*, New York, ABD, 1998.

[6] A. Çakmak "Web güvenliğinde SSL/TLS kriptografik protokolü: açıklıklar, saldırılar ve güvenlik önlemleri," Yüksek lisans tezi, Bilgi Güvenliği Mühendisliği ve Kriptografi, Fen Bilimleri Enstitüsü, İstanbul Şehir Üniversitesi, İstanbul, Türkiye, 2018.

[7] C. Brubaker, S. Jana, B. Ray, S. Khurshid ve V. Shmatikov "Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations," *2014 IEEE Symposium on Security and Privacy*, IEEE, Berkeley, CA, USA, 2014, ss. 114-129.

[8] D. Wagner ve B. Schneier "Analysis of the SSL 3.0 protocol," *The Second USENIX Workshop on Electronic Commerce*, Oakland, Kaliforniya, 1996.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," ss. 1-8, 2008. HTTPS://bitcoin.org/bitcoin.pdf (Erişim Tarihi: 08.10.2020)

[10] J. A. Jaoude ve R. G. Saade "Blockchain applications – usage in different domains," *IEEE Access*, c. 7, ss. 45360-45381, 2019.

[11] E. Karaarslan ve M. F. Akbaş "Blokzinciri tabanlı siber güvenlik sistemleri," *Uluslararası Bilgi Güvenliği Mühendisliği Dergisi*, c. 3, ss. 16-21, 2017.

[12] R. T. Fielding "Architectural Styles and the Design of Network-based Software Architectures," Doktora tezi, Bilgi ve Bilgisayar Bilimleri, Kaliforniya Üniversitesi, Irvine, Kaliforniya, 2000.

[13] D. S. V. Madala, M. P. Jhanwar ve A. Chattopadhyay "Certificate transparency using blockchain," *IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, Singapore, ss. 71-80, 2018.