Ethereum Pay ~Send encrypted messages with money.

Yasin Aktimuır
yasak@gmx.com

Bu çalışmada temel olarak, blockchain'e kayıt edilen ve gizli kalması gereken verilerin gizliliğini ve güvenliğini sağlamak amacı ile kullanılabilecek yöntemlerinden birisi olan Homomorfik şifreleme yöntemi tanıtılmıştır. Homomorfik şifrelemenin ne olduğu ve dijital pazarlarda kripto para ödemelerinin yanında sadece alıcının açabileceği şekilde şifreli mesajları nasıl göndereceğini de açıklayacağım. Homomorfik şifreleme algoritmalarından EDCSA veya RSA ile birlikte sadece ödeme alan kullanıcının gizli anahtarı ile açabileceği mesajların nasıl yaratılacağından ve tüm bu gizli verilerin blok zincirinin herkese açık bir doğası olmasına rağmen şifreli bir şekilde nasıl korunacağından bahsedeceğim.

1.Giriş

Blockchain ile birlikte hayatımıza giren merkeziyetsiz verilerin şeffaf bir şekilde tüm düğümler ile paylaşıldığı bir dünyada "düğüm" kavramı depolanan her tür veriye her an her yerden ulaşma imkânı sağlamaktadır. Blok zincir kavramı her ne kadar kullanıcılara istedikleri veriyi şeffaf bir şekilde sonsuza kadar depolama fırsatı sunuyor olsa da, bu verilerin şifrelenerek saklanması ve bunun sadece yetkili alıcılar tarafından gözlemlenebilmesi blok zinciri içerisinde çözümlenen bir problem değil.

Bu aşamada blok zincirine veri ekleyen bir kişi bu verilerin gizli kalmasından kendisi sorumludur. Bir kullanıcının gizlemek istediği bir veriyi şifreleyerek blok zincirine kayıt etmesi ve böylelikle verileri zincirde saklaması verileri gizlemeye yarar ama sizin anahtarınız sizdedir. Bu anahtarı ürününüzü satın alanlara paylaşırsanız sizin bütün ürünlerinizin şifrelerini çözmelerine neden olacaktır. Bu yüzden verileri şifrelerken sadece alıcıya özel ve sadece o alıcının şifreyi kendi gizli anahtarıyla çözebileceği şekilde şifrelemeniz gerekmektedir.

Homomorfik şifreleme

1.Homomorfik şifreleme

Homomorfik şifreleme, şifreli metinler üzerinde yapılan belirli matematiksel işlemlerin sonucunda oluşturulan şifreli sonucun şifresi çözüldüğünde elde edilecek sonuç ile bu işlemlerin açık metin üzerinde uygulanmasıyla elde edilecek sonucun aynı olmasını sağlayan bir şifreleme türüdür[2].

Homomorfik şifrelemeyi formül ile ifade etmek gerekirse.Enc şifreleme işlemini Dec şifre çözme işlemini ve K da şifreleme için kullanılan gizli anahtarı, bunların yanında + ve * işaretleri de Q kümesi üzerinde toplama ve çarpma işlemlerini temsil etsin. Bu durumda, [3]
Enc şifreleme fonksiyonunun; homomorfik toplama özelliği taşıması için
∀ a, b ϵ Q ifadesi ile
a + b = D ec κ (E ncκ(a) + Encκ(b)) eşitliğinin sağlanması, [3] ve homomorfik çarpma özelliği taşıması için de
∀ a, b ϵ Q ifadesi ile
a * b = D ecκ (E ncκ(a) * Encκ(b)) eşitliğinin Sağlanması gerekmektedir. [3]

Bu eşitliklerde kullanılan EncK(a)
, a sayısının K gizli anahtarı ile şifrelenmesini, Dec κ ise
Elde edilen şifreli toplam veya çarpım işlem sonucunun, yine K gizli anahtarı kullanılarak şifrenin çözülmesini ifade eder.

Bu şekilde veri içeriğinin gizliliği ve işlemlerin güvenli bir şekilde tamamlanabilmesi sağlanabilmektedir.

Farklı şirketlerdeki farklı servislerin, homomotfik şifreleme ile işlemlerini gerçekleştirmesi sayesinde güvenilir bir hesaplama zinciri oluşturulabilmektedir. Örneğin birinci servis vergilerin hesaplanmasıyla ilgili sorguları, ikincisi döviz değişim oranlarıyla ve üçünüsü de nakliye işlemleri ile ilgili sorguları işlettiğinde her servis sadece kendine ait sorguları işletmiş ve diğer servislerin verilerini elde etmemiş olacaktır. [2]

Çeşitli şifreleme sistemlerinde homomorfik şifreleme ile veri gizliliğinin sağlanması, güvenli elektronik oylama sistemlerinin geliştirilmesine.(her oy şifrelenir, sadece toplam sonucu deşifre edilir), gizli bilgi erişim yöntemlerine, tıbbi kayıtların tutulmasına, çakışma dirençli (collision-residant) hash fonksiyonları üretilmesine veri madenciliğine ve block zinciri alanında güvenli işlemlerin gerçekleşmesine yardımcı olmaktadır.[2] [5][6]

Şifreli mesajlar üzerinde basit işlemler gerçekleştirilmesi fikri ilk olarak Rivest Adleman ve Derouzous tarafından öne sürülmüştür. Bu şekilde bir yaklaşımın temelinde ise şifreli olarak saklanmış veri tabanına erişimin sadece yetkilendirilmiş kişiler tarafından şifre çözülmeden yapılmasına olanak tanıması bulunmaktadır. [5]

Homomorfik şifreleme, uygulanacak işlem operatörü sayısına göre, kısmi ve tam homomorfik şifreleme olarak ikiye ayrılmaktadır. Kısmi homomorfik şifrelemede (Partially-homomorphic encryption), şifreli metinler üzerinde gerçekleştirilebilecek tek bir işlem (ya toplama ya da çarpma) söz konusu iken ilk olarak 2009'da Craig Gentry'nin tez çalışmasında.[8] gündeme getirilen tam homomorfik şifrelemede(Fully-Homomorphic encryption) ise hem toplama hem de çarpma işlemlerinin bir arada uygulanması söz konusudur. Örneğin RSA ve El-Gamal algoritmaları çarpmaya göre kısmi şifreleme özelliği göstermektedir.

Gahi ve arkadaşları çalışmalarında, tam bir homomorfik şifreleme kullanarak güvenli bir veritabanı geliştirmişlerdir. Önerilen bu model, şifreli verileri giriş olarak almakta ve sonrasında kullanıcı sorgusunun içeriğini bilmeden, "görmeden işleme blind-processing" şeklinde işlemektedir. Şifreli olarak üretilen sonuç da sadece sorguyu başlatan kullanıcı tarafından deşifre edilebilmektedir. Böylece kullanıcıların gizlilikleri korunmuş ve uzak uygulamara güvenleri sağlanmış olmaktadır.[2][7]

Günümüzde kullanılan homomorfik şifreleme sistemleri toplama, çıkartma, çarpma, XOR ve üs alma gibi işlemleri gerçekleştirebilmektedir. Bu yöntemler, şirketlerin tüm veritabanlarını şifreleyerek buluta yükleyebilmelerinin ve bu veriler üzerinden şifre çözme işlemleri uygulamadan hesaplamalar yapabilmelerinin önünü açmıştır.

En yaygın kullanılan şifreleme sistemlerinden açık-anahtar kullanan RSA, bilinen ilk homomorfik şifreleme yöntemlerinden birisidir. Çarpmaya göre kısmi-homomorfik özellik gösteren rSA algoritmasında, m şifrelenecek mesaj, n modül, e açık anahtar, d gizli anahtar, E şifreli metin olarak alındığında yapılacak şifreleme (Enc) ve şifre çözme (Dec) işlemleri aşağıdaki formüllerden elde edilmektedir.[5]

Şifreleme : $Enc(m) = m^e \pmod n = E$

Şifre çözme: $Dec(E) = E^d \pmod n$

Burada kısmi-homomorfik şifreleme işlemi, RSA şifreli metinlerinin ayrı ayrı çarpılmasıyla bulunacak sonucun, şifresiz metinlerin çarpımının şifrelenmiş haline eşitliğini gerektirmektedir:

$Enc(m_1) * Enc(m_2) = m_1^e * m_2^e \pmod n = (m_1 * m_2)^e \pmod n = Enc(m_1 * m_2)$

Örneğin, $m_1 = 4$, $m_2 = 3$, $e = 7$, $d = 3$, $n = 33$ alındığında,

c_1 , m_1 verisinin şifreli metni olmak üzere;
$c_1 = Enc(m_1) = m_1^e \pmod n = 4^7 \pmod{33} = 16$

c_2 , m_2 verisinin şifreli metni olmak üzere;
$c_2 = Enc(m_2) = m_2^e \pmod n = 3^7 \pmod{33} = 9$

c_1 ve c_2 şifreli metinlerinin çarpımı;
$c_1 * c_2 = 16 * 9 = 144$

$(c_1 * c_2)$ şifreli metinlerinin çarpımının şifresi çözülürse;

$Dec(c_1 * c_2) = (c_1 * c_2)^d \pmod n$
$= (144)^3 \pmod{33} = 12$

m_1 ve m_2 şifresiz metinlerinin çarpımı;
$m_1 * m_2 = 4 * 3 = 12$

$Enc(m_1) * Enc(m_2) = Enc(m_1 * m_2)$

sonuçlarının eşitliği RSA kısmi -homomorfik şifreleme için sağlanmış olacaktır. [5]

Bir sonraki bölümde, homomorfik şifreleme ve homomorfik şifrelemenin bulut bilişim güvenliği açısından değerlendirilmesi ele alınmıştır.

EPAY PROTOCOL

Ethereum Pay nasıl çalışır?
Alice'in bir alıcı, bob' un da bir kahve satıcısı olduğunu düşünün alice satın aldığı ürünü hiç bir aracı olmadan blockchain üzerinden almak istiyor fakat bob hangi tür kahve alacağını ve alice'in adresi gibi bilgileri bilmiyor. Alice bu bilgileri açık bir şekilde paylaşamayacağı için epey aracılığı ile bob'un PUBLİC keyi ile şifreleyerek şifreli mesajı epey protokolü üzerinden şifreli bir şekilde gönderiyor. Bu mesajı dünyada sadece bob kendi private keyi ile decode edebiliyor. Böylece veriler blockchainde güvenle korunabilmiş oluyor.

EPAY cüzdanları EDCSA destekli olarak Public key ve private key ve adress ten oluşan üçlü şekilde üretiliyor.

Kaynaklar

[1] Brenner M., Wiebelitz J., Voigt G.V. ve Smith M., “Secret Program Execution in the Cloud Applying Homomorphic Encryption”, 5th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), Daejeon, Korea, 2011, pp 114 - 119.

[2] http://en.wikipedia.org/wiki/Homomor phi c_encryption (Son Erişim Tarihi: 14.10.2014)

[3] Özdemir S., “Kablosuz Algılayıcı

Ağlarında Homomorfik Şifreleme İle Gü venli Veri Kümeleme”, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi , Cilt 23, No 2, 2008, pp 365 - 373.

[4] Mell P. ve Grance T., The NIST Definition of Cloud Computing - Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800 -145, 2011.

[5] Ravindran S. ve Kalpana P., “Data Storage Security Using Par tially Homomorphic Encryption in a Cloud”, International Journal of Advanced Research in Computer Science and Software Engineering, 3(4), 2013, pp 603 - 606,.

[6] Tebaa M., Hajji S.E. ve Ghazi A.E., “Homomorphic Encryption Applied to the Cloud Computing Security”, Proceedings of the World Congress on Engineering 2012 Vol I WCE 2012, London, U.K., 2012, pp 536 - 539 .

[7] Gahi Y., Guennoun M. ve El Khatib K. , "A Secure Database System using Homomorphic Encryption Schemes" . The Third International Conference on Advances in Databases, Knowledg e, and Data Applications DBKDA, 2011, pp 54 – 58.

[8] Craig Gentry, “A Fully Homomorphic Encryption Scheme”, Phd. Thesis, Standford University, 2009.

[9] Ethereum Pay’s Pair