T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAYIN İLETİMİ İÇİN TEK YÖNLÜ BİR MELEZ ANAHTAR DAĞITIM ŞEMASI GELİŞTİRİLMESİ VE UYGULANMASI

HÜSEYİN BODUR

DOKTORA TEZİ
ELEKTRİK – ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

DANIŞMAN
PROF. DR. RESUL KARA

DÜZCE, 2020

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAYIN İLETİMİ İÇİN TEK YÖNLÜ BİR MELEZ ANAHTAR DAĞITIM ŞEMASI GELİŞTİRİLMESİ VE UYGULANMASI

Hüseyin BODUR tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik – Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Resul KARA
Düzce Üniversitesi

Jüri Üyeleri

Prof. Dr. Resul KARA
Düzce Üniversitesi _____

Prof. Dr. Pakize ERDOĞMUŞ
Düzce Üniversitesi _____

Doç. Dr. Sedat AKLEYLEK
Ondokuz Mayıs Üniversitesi _____

Dr. Öğr. Üyesi Esra ŞATIR
Düzce Üniversitesi _____

Dr. Öğr. Üyesi Murat İSKEFİYELİ
Sakarya Üniversitesi _____

Tez Savunma Tarihi: 31/12/2020

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

31/12/2020

Hüseyin BODUR

TEŞEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Prof. Dr. Resul KARA'ya en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Bu tez çalışması, TUBİTAK BİDEB 2211-C Öncelikli Alanlara Yönelik Yurt İçi Doktora Burs Programı ve Düzce Üniversitesi BAP-2018.06.01.793 numaralı Bilimsel Araştırma Projesiyle desteklenmiştir.

31/12/2020 **Hüseyin BODUR**

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ

Sayfa No

ÇİZELGE LİSTESİ

Sayfa No

KISALTMALAR

AAK	Açık Anahtar Kütüphanesi
AGDH	Ağaç-tabanlı Grup Diffie-Hellman
AŞA	Anahtar Şifreleme Anahtarı
DH	Diffie-Hellman
DÖGİ	Dağıtık Ölçeklendirilebilir Güvenli İletişim
EEAY	Eliptik Eğri Anahtar Yönetimi
EEDH	Eliptik Eğri Diffie-Hellman
GAY	Grup Anahtar Yönetimi
GGİ	Güvenli Grup İletişim
GÜY	Grup Üyelik Yönetimi
GY	Grup Yöneticisi
IoT	Nesnelerin İnterneti
KSA	Kablosuz Sensör Ağlar
MAH	Mantıksal Anahtar Hiyerarşisi
MGAD	Merkezi Grup Anahtar Dağıtım
MHTGG	Mantıksal Halka Tabanlı Güvenli Grup İletişim Şeması
SISA	Sıska Ağaç
TMAD	Tek Yönlü Melez Anahtar Dağıtım
TFA	Tek-yönlü Fonksiyon Ağacı
TFZ	Tek-yönlü Fonksiyon Zinciri
TŞA	Trafik Şifreleme Anahtarı
w-GGA	w-Oturumu Güvenilir Grup Anahtar Yönetimi
YM	Yayın Merkezi

ÖZET

YAYIN İLETİMİ İÇİN TEK YÖNLÜ BİR MELEZ ANAHTAR DAĞITIM ŞEMASI GELİŞTİRİLMESİ VE UYGULANMASI

Hüseyin BODUR
Düzce Üniversitesi
Fen Bilimleri Enstitüsü, Elektrik – Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi
Danışman: Prof. Dr. Resul KARA
Aralık 2020, 109 sayfa

Bir yayın haberleşme yönteminde bir kaynaktan çoklu kullanıcılara mesaj iletimi için genellikle içerisinde şifreleme yöntemlerinin kullanıldığı şemalardan yararlanılır. Bu şemalar anahtar sunucu işlemleri açısından merkezi, dağıtık ve birleşik olmak üzere üçe ayrılır. Bu çalışmada günümüzde yaygın olarak kullanılan merkezi şemalar olan Mantıksal Anahtar Hiyerarşisi (MAH), Tek-yönlü Fonksiyon Ağacı (TFA) ve Tek-yönlü Fonksiyon Zinciri (TFZ) şemalarına, dağıtık şemalar olan Ağaç-tabanlı Grup Diffie-Hellman (AGDH), Dağıtık Ölçeklendirilebilir Güvenli İletişim (DÖGİ) ve Sıska Ağaç (SISA) şemalarına ve birleşik şema olan Mantıksal Halka Tabanlı Güvenli Grup İletişim Şeması (MHTGG) şemasına değinilmiştir. Tek Yönlü Bir Melez Anahtar Dağıtım (TMAD) adında hem merkezi hem de dağıtık olmak üzere ikili ağaç temelli yeni bir yayın şifreleme şeması önerilmiştir. Şemada hem bir gizli anahtarlı şifreleme yöntemi hem de bir açık anahtarlı anahtar dağıtım protokolü birlikte kullanılmaktadır. Anahtar güncellemelerinde kullanıcı düğümlerden kök düğüme doğru simetrik anahtarların sol ya da sağ yarısı iletilmektedir. Bu durum iletilen toplam anahtar boyutunun azalmasını sağlamaktadır. TMAD şemasının merkezi modelinde bir Grup Yöneticisi (GY) bulunmaktadır. Dağıtık modelinde ise GY'nin görevini, kullanıcılar arasında oluşturulan fikir birliği mekanizması üstlenmektedir. TMAD şeması, kullanıcı ekleme-çıkarma ve toplu kullanıcı ekleme-çıkarma işlemlerinde anahtar iletim sayısı, işlem maliyeti, kullanıcılarda bulunan anahtar sayısı, kullanıcılarda bulunan anahtar boyutu ve anahtar iletim boyutları gibi çeşitli açılardan mevcut şemalar ile karşılaştırılmıştır. Sonuçlar grafiksel olarak gösterilmiştir.

Anahtar sözcükler: Bulut bilişimde veri güvenliği, Güvenli grup iletişim şemaları, Mobil programlamada veri güvenliği, Tek yönlü anahtar dağıtım, Yayın iletişim şemaları.

ABSTRACT

DEVELOPMENT AND IMPLEMENTATION OF AN ONE-WAY HYBRID KEY DISTRIBUTION SCHEME FOR BROADCAST TRANSMISSION

Hüseyin BODUR
Düzce University
Graduate School of Natural and Applied Sciences, Department of Electrical - Electronics and Computer Engineering
Doctoral Thesis
Supervisor: Prof. Dr. Resul KARA
December 2020, 109 pages

In broadcast communications, the schemes are used to transmit messages from a source to multiple users where the encryption methods are employed within the schemes. The schemes can be either tree based or have different topologies. There are three types of tree-based scheme categories: centralized, distributed and hybrid schemes. In this work, centralized, distributed and hybrid techniques which are widely used nowadays are investigated. The centralized techniques considered in this work are as follows: Logical Key Hierarchy (LKH), One-way Function Tree (OFT) and One-way Function Chain (OFC). The distributed techniques can be given as: Tree-based Group Diffie Hellman (TGDH), Distributed Scalable sEcure Communication (DISEG) and Skinny Tree (STR). The hybrid technique can be given as: The Logical Ring Based Secure Group Communication Scheme (RISeG). A new binary tree-based broadcast encryption schemes, both centralized and distributed, named One-way Hybrid Key Distribution (OHKD) have been proposed. Both a symmetric encryption and an asymmetric key distribution protocol are used in the OHKD scheme. In key updates, the left or right half of the symmetric keys are transmitted from the user nodes to the root node. This ensures that the total key size transmitted is reduced. The centralized model of the OHKD scheme includes a Group Manager. The role of Group Manager is undertaken by a consensus structure between users in the distributed model of the OHKD scheme. The OHKD scheme is compared with the existing schemes in terms of number of key transmissions, number of operations, number of user keys, size of user keys and size of key transmission in user adding-removing and batch user adding-removing operations. The results are shown graphically.

Keywords: Broadcast communication schemes, Data security in cloud computing, Data security in mobile programming, One-way key distribution, Secure group communication schemes.

EXTENDED ABSTRACT

DEVELOPMENT AND IMPLEMENTATION OF AN ONE-WAY HYBRID KEY DISTRIBUTION SCHEME FOR BROADCAST TRANSMISSION

Hüseyin BODUR
Düzce University
Graduate School of Natural and Applied Sciences, Department of Electrical - Electronics and Computer Engineering
Doctoral Thesis
Supervisor: Prof. Dr. Resul KARA
December 2020, 109 pages

1. INTRODUCTION

In broadcast communications, the schemes are used to transmit messages from a source to multiple users where the encryption methods are employed within the schemes. The schemes can be either tree based or have different topologies. There are three types of tree-based scheme categories: centralized, distributed and hybrid schemes. This study explores the currently most widely used centralized, distributed and hybrid techniques. The centralized techniques considered in this work are as follows: Logical Key Hierarchy (LKH), One-way Function Tree (OFT) and One-way Function Chain (OFC). The distributed techniques can be given as: Tree-based Group Diffie Hellman (TGDH), Distributed Scalable sEcure Communication (DISEG) and Skinny Tree (STR). The hybrid technique can be given as: The Logical Ring Based Secure Group Communication Scheme (RISeG). In addition, a new binary tree-based broadcast encryption scheme, both centralized and distributed, named One-way Hybrid Key Distribution (OHKD) has been proposed.

2. MATERIAL AND METHODS

The OHKD scheme has a binary tree structure, such as centralized and distributed schemes. The broadcast center is located in the root node, the users are located in the leaves. The keys are calculated from user nodes to the root node. The OHKD scheme uses an asymmetric key distribution protocol in the root node and the user nodes by employing a symmetric encryption method, which is not considered in other centralized schemes. It contains Hash and XOR functions in the intermediate node calculations as opposed to other distributed schemes. For the calculation of each intermediate symmetric key, the left half of the left child symmetric key and the right half of the right child symmetric key are combined.

3. RESULTS AND DISCUSSIONS

The OHKD scheme is compared with the existing schemes in terms of number of key transmissions, number of operations, number of user keys, size of user keys and size of key transmission in user adding/removing and batch user adding-removing operations. The schemes are written in Java by using a nosql database. Key updates after each membership change from 1 to 2^{21} users are performed.

4. CONCLUSION AND OUTLOOK

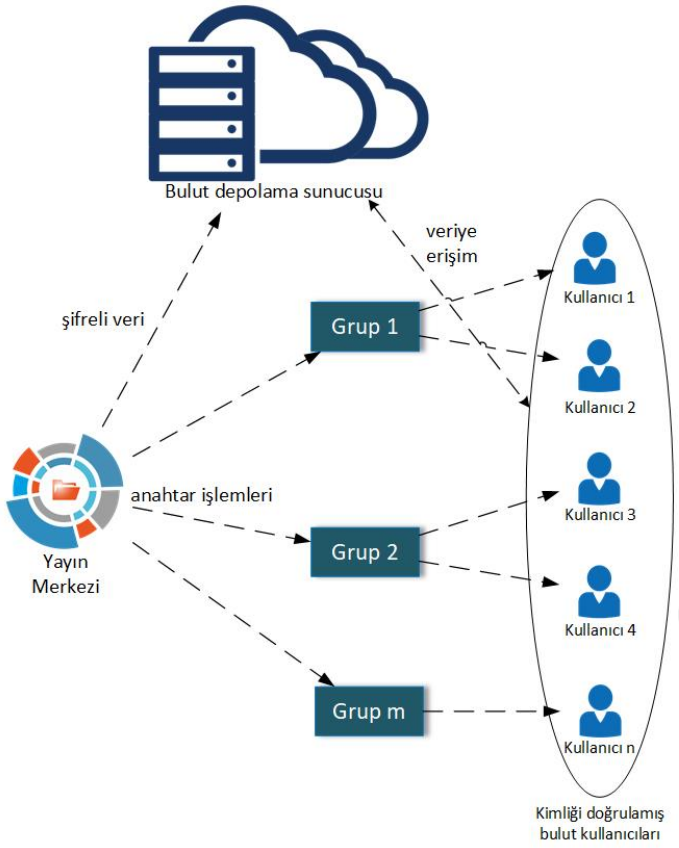
The OHKD scheme gives the best results in terms of the transmission size in user removing and batch user removing operations. It has an advantage in terms of transmission costs for user removing operation. It also performs better in terms of the number of operations in user adding/removing operations and batch user removing operation compared to distributed and hybrid schemes. Furthermore, unlike other centralized GKM schemes, the OHKD scheme utilizes a key distribution protocol both in the user nodes and in the root node. In contrast to other distributed GKM schemes, it utilizes Hash and XOR functions in the intermediate nodes.

1. GİRİŞ

Günümüzde bulut bilişimin kullanım oranı ve önemi sürekli artmaktadır. Bulut bilişim, temel olarak herhangi bir altyapı, yazılım veya cihaza gerek duymaksızın kullanıcıların bulundukları yerden hizmet almasını sağlayan bir sistemdir. Bulut bilişim, ölçeklenebilirlik, esneklik, kullandığın kadar öde özelliği gibi birçok açıdan hem akademik hem de endüstri açısından büyük bir öneme sahiptir. Güçlü sunuculara sahip olması nedeniyle depolama ve hizmet paylaşımı açısından insanların kolay bir şekilde bir grubun parçası olarak çalışmasına imkân sunmaktadır [1].

Kullanıcılar bulut bilişim hizmeti satın alarak, bulut üzerinde bulunan sistem veya yazılımları kullanabilir. Bunun yanı sıra kendi verilerini depolama ve işleme imkanına da sahip olabilir. Bulut alanında hizmet sağlayıcı kuruluşlar bulut bilişim sağlayıcı olarak isimlendirilir. Bulut bilişim sağlayıcıları, kullanıcılara bulut üzerinden altyapı sunmak ve bu altyapının güvenliğini sağlamaktan sorumludur. Kullanıcılar ihtiyaç duydukları sistem ya da yazılımları satın almak yerine, bulut üzerinden hizmet satın alma yoluna giderek ihtiyaçlarını daha düşük maliyet ile giderebilirler [2]. Bulut bilişim, kullanıcılara cihaz, zaman, mekân bağımsızlığı ile güçlü donanımsal bir altyapı sağlar. Ayrıca, bulut bilişim sınırsız hesaplama gücünün yanı sıra sınırsız saklama gücüne de sahiptir [3], [4], [5]. Fakat bu durumun yanı sıra bazı güvenlik sorunlarını da beraberinde getirir. Güvenlik sorunlarının temelinde kullanıcıların yararlandığı altyapıda bulunan sistem, yazılım veya verinin gizliliği yer almaktadır. Veri gizliliğinin korunması için, ortak yaklaşım bir verinin bulut sunucuya yüklenmeden önce şifrelenmesidir [6], [7], [8]. Bir bulut sunucu üzerindeki bir verinin gizliliği ve güvenliği için [9], [10], [11] gibi çeşitli şemalar önerilmiştir. Fakat bu şemalar tek veri sahipli güvenlik problemlerini ele almışlardır [12]. Çoğu bulut uygulamalarında veriler genellikle dinamik bir kullanıcı grubu içerisinde paylaşılmaktadır [13], [14], [15].

Şekil 1.1'de görüldüğü üzere bulut üzerinde yalnızca yetkili kullanıcıları içeren grubun veriler üzerindeki erişimlerine izin verilmelidir. Yetkili kullanıcılar haricinde hiçbir



Şekil 1.1. Güvenli Grup İletişim (GGİ) ile Bir Bulut Depolama Modeli.

kullanıcı bulut verilerine erişememelidir. Gruba katılmak isteyen kullanıcıların öncelikle bir kimlik doğrulama mekanizmasından geçmesi gerekir [16], [17]. Aynı şekilde gruptan çıkartılan kullanıcıların bulut üzerinde paylaşılan verilere erişimi engellenmelidir. Bunların yanı sıra verilerin gizliliği ve bütünlüğü sağlanmalıdır. Bulut altyapısı kötü niyetli kullanıcılar ya da bulut sağlayıcılarının ataklarına karşı güvenilir olmalıdır.

1.1. LİTERATÜRDE YER ALAN GELİŞTİRME ÇALIŞMALARI

Bir verinin çoklu kullanıcıya iletilmesi yayın haberleşme konusu içinde yer alır. Yayın haberleşmede çoklu kullanıcılara mesaj iletimi için genellikle şifreleme yöntemlerinden yararlanılır. Güvenli yayın haberleşmesi için geçmişten günümüze birçok yayın iletim şeması önerilmiştir.

Yayın haberleşmesi için yapılan çalışmalardan birinde Shanu ve Chandrasekaran, MAH'daki en önemli işlemin kullanıcı ekleme-çıkarma işlemlerinin ardından ileri ve geri gizliliğin sağlanması için yapılan yeniden anahtarlama olduğunu belirtmiş, yeniden

anahtarlama işlem maliyetini azaltmak için bir dağıtım fonksiyonundan yararlanmışlardır [18].

Prathap ve Vasudevan, çeşitli şemaları incelemişlerdir. Kullanıcı ekleme-çıkarma gibi işlemler üzerine bu şemaların avantajlı yönlerini bir araya getirerek yeni hibrit bir şema önermişlerdir [19]. [20]'de yapılan çalışmada Sakamoto ve arkadaşları, MAH şemasının kullanıcılardan yayın merkezine olan yol uzunluğunu azaltmak için Huffman algoritmasını kullandıkları bir şema önermişlerdir.

Gu ve arkadaşları Anahtar Ağacı Yeniden Kullanım isimli bir anahtar yönetim şeması önermiştir. Önerilen şema, kullanıcıların yayın sistemi içerisinde bulunan çoklu programlara aynı anahtar değeriyle kaydolmasına izin veren bir anahtar yönetim yaklaşımıdır. MAH tabanlı olmasına rağmen, MAH'a göre daha düşük yeniden anahtarlama maliyetine sahiptir [21].

Song ve arkadaşları dinamik bir kullanıcı grubunun bulut üzerinde şifreli veri paylaşımı yapılabilmesi için açık anahtar altyapısına sahip bir grup anahtar yönetim şeması önermiştir. Önerilen şema, bulut sunucu üzerinde kötü niyetli kullanıcıların saldırılarına maruz kalınsa dahi açık anahtarlı şifrelemenin avantajlarından yararlanarak veri güvenliği sağlanmaktadır [22].

Alyani ve arkadaşları MAH üzerinde Diffie-Hellman anahtar değişiminin nasıl yapılacağını açıklamış ve şemanın altkümelerinde bulunan kullanıcı sayısını arttırarak performansını geliştirmeye çalıştırmıştır [23]. [24]'de yapılan çalışmada Liu ve arkadaşları kullanıcı ekleme-çıkarma işlemlerinin anahtar güncelleme maliyetini azaltmak amacıyla MAH şemasını geliştirmişler, bunun için bir sezgisel arama algoritmasından yararlanmışlardır.

Sakamoto, bir anahtar ağacına eklenen veya çıkartılan ortalama kullanıcı sayısının bilindiği takdirde, anahtar güncelleme maliyetinin azaltılabileceğini savunan bir çalışma önermiştir [25]. [26]'da yapılan çalışmada, MAH'ın bir bulut sunucu üzerinde bulunan Nosql bir veritabanına uygulanması sırasında karşılaşılan sorunların çözümü için Diffie-Hellman anahtar değişiminden yararlanılmıştır.

[27]'de TFA şemasının güvenlik sorunlarına değinilmiştir. Tekrarlanan TFA ve Düğüm TFA adında iki geliştirilmiş TFA şeması önerilmiştir. TFA ile karşılaştırıldığında Tekrarlanan TFA ve Düğüm TFA şemalarının grup yönetiminde ekstra iletişim maliyetine ihtiyaç duymadığı tespit edilmiş, gizli anlaşma atağına karşı güçlü oldukları belirtilmiştir. [28]'de TFA şemasının gizli anlaşma atakları karşısındaki zayıflığına değinilmiştir. TFA şeması üzerine bir metot eklenerek yeni bir şema önerilmiştir. Bu metot kullanılarak önerilen şemada yapılan bir yayının ortalama boyutu minimize edilerek gizli anlaşma ataklarının engellenmesi amaçlanmıştır.

Hwang ve arkadaşları çalışmalarında makro ve mikro ödeme sistemlerini incelemiş, eliptik eğri şifreleme temelli bir yeni mikro ödeme yöntemi önermişlerdir. Önerilen yöntem TFA'dan türetilmiş olan TFZ tabanlıdır [29].

Lee ve arkadaşları bir ağ içinde konuk cihazların geçici erişimlerinde ortaya çıkan anahtar yönetim maliyetini azaltmak için Geçici Erişim Hakları Delegasyonu adında bir şema önermişlerdir. Bu şema bir kriptografik jetonun güvenli bir şekilde iletimi için TFZ şemasından yararlanmaktadır [30].

Benmalek ve Challal çalışmalarında, akıllı şebekelerin temel bileşeni olan gelişmiş ölçüm altyapısının siber saldırılara karşı güçlü olmasını sağlayan bir yeni anahtar yönetim şeması önermişlerdir. Bunun için TFZ nin bir çeşidini uyarlamışlardır [31]. Çalışmalarında Chen ve Tzeng, yeniden anahtarlama işleminin sürekli gerektiği, üye sayısının fazla ve dinamik olduğu şemalarda hesaplama ve saklama maliyetini azaltmak için KeyDer-GAY ve ReEnc-GAY adında iki Grup Anahtar Yönetimi (GAY) şeması önermişlerdir [32].

Benmalek ve arkadaşları, bir akıllı şebekede gelişmiş ölçüm altyapısı için kullanılacak dört anahtar yönetim şeması önermişlerdir. Bu şemaların temelinde açık anahtarlı bir anahtar yönetim protokolü olan Eliptik Eğri Diffie-Hellman (EEDH) bulunmaktadır. Bu protokol hem tekli hem de çoklu iletişimi desteklemektedir [33].

Kumar ve arkadaşları, anahtar güncelleme sırasında anahtar sunucunun hesaplama ve saklama maliyetini azaltan bir Merkezi Grup Anahtar Dağıtım (MGAD) protokolü önermişlerdir [34]. Çalışma içerisinde ayrıca, MGAD protokolünün büyük ölçüde kullanıcı içeren gruplar için kümelenmiş ağaç temelli bir genişletilmiş hali de yer almaktadır. [35]'de

yapılan çalışmada, Hanatani ve arkadaşları IEEE 802.21'de standardize edilmiş bir teknik önermişlerdir. Bu teknik MAH'a dayalıdır ancak şifreleme ve şifre çözme işlemleri için Tam Alt Ağaç kullanılmıştır.

Elhoseny ve arkadaşları çalışmalarında, Kablosuz Sensör Ağlar (KSA) üzerinde veri iletim güvenliği için bir şema önermişlerdir. Önerilen şema hem eliptik eğri hem de homomorfik şifreleme yöntemlerinden yararlanmakta, iletişim maliyeti, enerji tüketimi, bellek gereksinimi ve yaşam süresi gibi ağ performansının çeşitli yönlerini geliştirmektedir [36].

Lin ve arkadaşları çalışmalarında KSA üzerinde grup iletişim güvenliğini sağlamak adına bir küme tabanlı eliptik eğri anahtar yönetim şeması önermişlerdir. Önerilen şema, Diffie-Hellman (DH) ve RSA şifreleme sistemlerine kıyasla grup anahtarının etkili ve hızlı bir şekilde yeniden senkronizasyonunu sağlamıştır [37].

Islam ve Biswas, eliptik eğri temelli, eşleştirmesiz bir Kimlik tabanlı İki Taraflı Doğrulanmış Anahtar Anlaşması protokolü geliştirmiştir. Önerilen protokol, karşılıklı iki taraf arasında ortak bir gizli anahtarın oluşturmasına olanak sağlamıştır ve taraflar arası verimli ve güvenli iletişim için uygundur [38].

Chaudhari ve arkadaşları bir dinamik sensör ağı için bir grup merkezi GAY şeması seçip, bu şemaları güçlü saldırılar altında analiz etmişlerdir. Her bir sensör düğümünün ayrı bir gizli anahtara sahip olması gerektiğine dikkat çekerek mevcut şemaların güvende olmasında bir grup n sensör için güvenli kanalların gerekli olduğunu belirtmişlerdir [39].

Hur ve Lee, anahtarını kaybetmiş olan kullanıcıların güncel mesajları alabilmeleri için ihtiyaç duydukları mevcut grup anahtarının kurtarılmasını hedefleyen bir güvenilir w-oturumu grup anahtar yönetim şeması önermişlerdir. Bu şema kullanıcıların oturum bilgilerinden yararlanmaktadır. Oturum bilgisi, bir kullanıcının en son ki w-oturumlarında bulunan geçerli oturumun yol anahtarlarını kontrol etmesine olanak sağlamaktadır [40].

Zhang ve arkadaşları bulut bilişim ağları için Yönlendirilebilir Öznitelik kullanan bir hiyerarşik grup anahtar anlaşması protokolü önermişlerdir. Önerilen şema, grup anahtar

anlaşmaları için hesaplama maliyetini ortadan kaldıran grup anahtar faktörlerinden yararlanmıştır [41].

Vijayakumar ve arkadaşları hem mobil hem de bulut tabanlı ağ uygulamalarının anahtar yönetim ve dağıtım işlemlerinde yaşanan problemlere karşı önerilerde bulunmuşlardır [42]. Bu öneriler, güvenli ve etkili iletişim kurarak mevcut şemaları geliştirmeyi hedeflemektedir. [43]'de yapılan çalışmada Kung ve Hsia, dinamik Nesnelerin İnterneti (IoT) cihazları için GROUPIT adlı iki aşamalı bir GAY tasarlamışlardır. Her cihaz, önerilen şema içerisinde yer alan gruplardan birine atanmıştır. Çok sayıdaki IoT cihazının desteklenmesi ve anahtar hesaplama ve güncelleme maliyetlerinin azaltılması için her grup içerisinde anahtar yönetim işlemi gerçekleştirilmiştir.

Lee ve Park, basit varsayımlar altında, çıkartılan bir dizi kullanıcı ile ilişkili bir tekil kullanıcı iptali şifreleme şeması ile kimlik tabanlı bir kullanıcı iptali şifreleme şeması önermişlerdir [44]. Ardından önerdikleri şemaları güvenlik ve performans açısından birbirleriyle karşılaştırmışlardır.

Yan ve arkadaşları güvenli grup tabanlı iletişimi sağlamak için yaklaşımları araştırmışlardır [45]. Var olan iki şemayı, çift modlu yayın şifrelemesi adını verdikleri, yeni bir şifreleme sistemi içerisinde birleştirmişlerdir. Çift modlu şifreleme sisteminin hesaplama maliyetlerinin, tek modlu sistemden daha verimli olduğuna ulaşmışlardır.

Hongyong ve arkadaşları bulut ortamında kullanıma uygun, bir iptal edilebilir yayın şifreleme şeması önermişlerdir [46]. Şema sabit boyutlu şifreli bir metne ve özel bir anahtara sahiptir. Çift sistem şifreleme tekniğini kullanarak standart varsayımlar altında güvenliğini kanıtlamaktadır.

Maiti ve Misra, yaptıkları çalışmada, kimliğe dayalı yayın yeniden şifreleme için gizliliği koruyan bir şema önermişlerdir [47]. Önerilen şema, yayın yeniden şifreleme şemalarına ve gizliliği koruma şemalarına kıyasla şifre çözme süresini azaltmaktadır.

1.2. ÇALIŞMANIN AMACI, ÖNERİLEN ÇÖZÜM YÖNTEMİ VE KATKILARI

Bu çalışmada literatürde yer alan ve sıklıkla kullanılan GGİ şemaları incelenmiştir. Bu şemalar merkezi, dağıtık ve birleşik olmak üzere üç ayrı kategoriye ayrılmıştır.

Ayrıca TMAD adında, hem merkezi hem de dağıtık modellere sahip yeni bir GGİ şeması önerilmiştir. Şemalar anahtar iletim sayısı, anahtar iletim boyutu, işlem maliyeti, kullanıcıda bulunan anahtar sayısı ve boyutu açılarından karşılaştırılmıştır. Şemaların bulut bilişim içerisinde kullanılması durumunda yaşanacak zorluklar, kısıtlamalar, saldırı türleri ve grup anahtar yönetim gereksinimlerine değinilmiştir. Önerilen TMAD şemasının amacı, güvenlik sorunlarına neden olmadan anahtar güncelleme işlemi sonucu ortaya çıkan maliyetleri diğer şemalara kıyasla en aza indirmektir.

Literatürdeki merkezi şemalar genellikle simetrik şifreleme yöntemlerinden yararlanılır. Anahtar güncellemelerinde GY, şema yapısına göre ikili ağaç üzerinde yukarıdan aşağıya ya da aşağıdan yukarıya doğru benzer hesaplamaları gerçekleştirir. Tüm düğümlerde aynı şifreleme yöntemi oluşturulmuş anahtarlar bulunur. TMAD şemasının merkezi modelinde ise bir asimetrik anahtar dağıtımından ve bir simetrik şifreleme yönteminden yararlanılır. Asimetrik şifreleme yöntemi ile oluşturulmuş anahtarlar sadece kullanıcı düğümlerinde ve kök düğümde yer alır. Simetrik şifreleme yöntemi için kullanılan anahtarlar ise şemadaki tüm düğümlerde bulunur. Ayrıca kullanıcılardan kök düğüme doğru yapılan anahtar hesaplamalarında, her bir ebeveyn düğüm anahtarı, sol çocuğunun simetrik anahtarının sol yarısının, sağ çocuğunun simetrik anahtarının sağ yarısı ile birleştirilip özetinin alınmasıyla elde edilir.

TMAD şemasının dağıtık modelinde anahtar hesaplamaları merkezi model ile aynı olmakla birlikte GY bulunmaz. Anahtar üretim ve dağıtım işlemlerinden şemada bulunan kullanıcılar sorumludur. Şemaya üzerindeki değişiklikler kullanıcılar tarafından gerçekleştirilir ve onaylanır. Literatürdeki dağıtık şemalarda GY'nin görevleri bir sponsor düğüm ya da bir iş birliği grubu aracılığıyla gerçekleştirilmektedir. TMAD şemasında ise diğer şemaların aksine blok zinciri teknolojisi içerisinde kullanılan fikir birliği mekanizmasından yararlanılır. Fikir birliği mekanizması bir şema üzerindeki işlemlerin, şemada bulunan kullanıcıların belirli bir oranının onayı ile gerçekleşmesidir. TMAD şemasında bu oran %51 olarak belirlenmiştir.

1.3. TEZ ORGANİZASYONU

Bu tez çalışması aşağıda özetleri verilen bölümler şeklinde organize edilmiştir:

1.Bölümde, bulut bilişim ve GGİ hakkında bilgi verilmekte, literatürde yer alan GGİ çalışmalarına değinilmektedir.

2.Bölümde, bir şemanın karşılaşabileceği ataklar belirlenip, atakları içeren düşman modelleri oluşturulmaktadır. Ayrıca bir şemanın uyması gereken gereksinimlere değinilmektedir. Literatürde bulunan GGİ şemaları anahtar sunucu işlemleri açısından merkezi, dağıtık ve birleşik olmak üzere üçe ayrılmakta, her bir şemanın çalışma prensibi hakkında bilgi verilmektedir.

3.Bölümde, TMAD adında bir yeni ikili ağaç temelli yayın şifreleme şeması önerilmektedir. TMAD şeması hem merkezi hem de dağıtık modele sahiptir. Merkezi model için önerilen şemanın başlangıç adımları, kullanıcı ekleme-çıkarma ve toplu kullanıcı ekleme-çıkarma işlemleri ile güvenlik analizi hakkında bilgi verilmektedir. Dağıtık model için merkezi modelle olan farklılıklar belirtilmektedir. Bu bölümde ayrıca TMAD şemasının uygulandığı iki adet mobil uygulama bulunmaktadır.

4.Bölümde, TMAD şeması ile diğer şemalar anahtar iletim sayısı, anahtar iletim boyutu, işlem maliyeti, kullanıcıda bulunan anahtar sayısı ve boyutu gibi çeşitli kriterler açısından birbirleriyle karşılaştırılmaktadır.

5.Bölümde, önceki bölümde elde edilen performans sonuçları değerlendirilmekte ve çalışmanın literatüre getirdiği katkılara değinilmektedir.

2. GÜVENLİ GRUP İLETİŞİM ŞEMALARI

Bölüm 1'de literatür özeti verilen şemalardan performans değerlendirme işlemlerinde sık kullanılan şemaların sınıflandırılması ve çalışma prensipleri bu bölümde ele alınmıştır. Ayrıca grup iletişiminde karşılaşılacak ataklar ve düşman modelleri ile sağlanması gereken gereksinimler belirtilmiştir.

Bir GGİ şeması iki ana bileşenden oluşur. Bu bileşenler sırasıyla Grup Anahtar Yönetimi (GAY) ve Grup Üyelik Yönetimi (GÜY)'dir. GAY, GGİ içindeki grup üyeleri arasında ortak gizli bir anahtar oluşturulmasını sağlar. Ortak gizli anahtar GGİ şemasındaki mesaj güvenliği açısından büyük bir öneme sahiptir. Grup iletişiminde mesajları şifrelemek için kullanılır. GY, ortak gizli anahtar haricinde yayın merkezinde ve kullanıcılarda bulunan bazı gizli anahtarların yönetilmesinden de sorumludur.

Bir yayın şemasının dayanıklılığı, içerisinde kullanılan anahtar yönetim protokolüne ve ortak gizli anahtarın boyutuna bağlıdır. Anahtar yönetim protokolü bir grup anahtarının nasıl üretileceği, dağıtılacağı ve güncelleneceğini belirler. İleri ve geri gizliliğin sağlanması için ortak gizli anahtarın her üyelik işleminin ardından güncellenmesi gerekir.

GÜY bir kullanıcının gruba katılma ve ayrılma işlemlerini tanımlar. Gruba katılma işleminde ilk olarak kimlik doğrulama işleminin yapılması gerekir. Grup içerisindeki mesajlara yalnızca kimliği doğrulanmış kullanıcılar erişebilmelidir. Bir GGİ şemasının ataklara karşı dayanıklı olması gerekir. Bu ataklar Bölüm 2.1'de tanımlanmıştır.

Bu bölümde, bulut içerisinde GGİ'i için bilgi verilmiştir. Öncelikle grup iletişimini etkileyecek muhtemel ataklar ve düşman modelleri belirtilmiş, bir GGİ şemasının bu ataklardan kaçınması için ihtiyaç duyulan temel gereksinimler açıklanmıştır. Ardından literatürde bulunan GGİ şemaları merkezi, dağıtık ve birleşik olmak üzere üç kategoride sınıflandırılmış, şemalar detaylı olarak açıklanmıştır.

2.1. GRUP İLETİŞİM ATAKLARI

Ortadaki adam atağı: Bu atak çeşidinde saldırgan iletişim kanalına yerleşmeyi amaçlar. Eğer başarılı olursa bağlantıyı keser, kendisiyle kullanıcı ve kendisiyle bulut sunucu arasında iki ayrı bağlantı kurar. Kullanıcı bir istek yaptığında, bulut sunucu yerine saldırganla bağlantı kurar. Bulut sunucu ise kullanıcı yerine saldırgana cevap verir.

Kulak kabartma atağı: Saldırgan grup iletişimine pasif olarak katılarak iletilen mesajları dinlemeye çalışabilir. Kulak kabartma atağını engellemenin klasik çözümü iletimden önce bir grup anahtarı kullanarak mesajları şifrelemektir [48], [49].

Tekrarlama atağı: Saldırgan, grup iletişimini dinleyerek göndericiden alıcıya şifreli ya da şifresiz iletilen bir mesajı elde edip, alıcıya tekrar gönderebilir. Bu atak tipinde saldırgan mesajın içeriğinden daha çok mesajı gönderdiği kullanıcıları grup iletişiminin başarıyla gerçekleştiğine inandırmayı amaçlar [48], [49].

Taklit etme atağı: Saldırgan, grupta bulunan üyelerin kimliklerini taklit ederek grupla bağlantı kurmaya çalışabilir. Bu sayede grup içerisine izinsiz girerek, diğer atakları başlatmayı amaçlar [48], [49].

Sahte mesaj bırakma atağı: Saldırgan, grup içerisine sahte bir mesaj bırakarak, grubun yanlış bir karar almasına yol açabilir. Saldırıyı azaltmanın bir yolu orijinal mesaja bir mesaj bütünlüğü kodunun eklenmesidir. Bu kod mesaj bütünlüğünün yanı sıra kimlik doğrulaması da sağlamaktadır. [48], [49].

Uzlaşma atağı: Saldırgan grup içerisinden bir üye ile uzlaşarak gizli anahtar bilgisine ve tüm iletim mesajlarını elde edebilir. Grup yöneticisi mümkün olan en kısa süre içerisinde uzlaşma yapan düğümleri tespit edebilmeli ve gruptan çıkartmalıdır [48], [49].

Gizli anlaşma atağı: Kullanıcı gizli anahtarlarının kök düğüme doğru bir fonksiyonel bağımlılık ile hesaplanarak ortak grup anahtarının elde edildiği şemalarda görülür. Atağın ortaya çıkması için şema üzerinde sırasıyla t_1 zamanında gruptan bir üyenin ayrılması, t_2 zamanında ise gruba bir üyenin eklemesi gerekir. Eklenen ve ayrılan kullanıcılar kendi aralarında gizlice anlaşırlar ise $t_2 - t_1$ zaman aralığında güncel grup anahtarını

hesaplayabilirler. Bu atağı azaltmanın bir yolu şema üzerinde ayrılan kullanıcının pozisyon bilgisi tutularak, yeni kullanıcıyı bu pozisyona eklemektir.

2.2. DÜŞMAN MODELLERİ

Bu bölümde λ bir saldırgan olmak üzere beş tip düşman tanımlanmıştır.

Tip-I düşman modeli: Varsayalım bir kötü niyetli kullanıcı λ şemaya katılsın ve YM gibi davransın. Bu durumda saldırgan ortak gizli anahtarı elde etmeye çalışır. Eğer başarılı olursa, bu anahtarı kullanarak kullanıcılara mesaj gönderebilir. Ancak λ ortak gizli anahtarı değiştiremez. Çünkü anahtar hesaplaması merkezi şemalarda GY, dağıtık şemalarda ise GY'nin görevini üstlenmiş kullanıcı ya da belirli bir kullanıcı grubu tarafından gerçekleştirilir. Tip-I düşman modeli taklit etme ve sahte mesaj bırakma ataklarının bir örneğidir.

Tip-II düşman modeli: Varsayalım bir kötü niyetli kullanıcı λ, bir kullanıcı ile YM arasındaki iletişimi dinlesin. λ şema iletişimine gizlice katılır ve iletilen mesajları dinlemeye çalışır. Tip-II düşman modeli ortadaki adam ve kulak kabartma ataklarının bir örneğidir.

Tip-III düşman modeli: Varsayalım bir kullanıcı t_1 zamanında şemadan ayrılsın, bir diğer kullanıcı t_2 zamanında şemaya katılsın ve bir kötü niyetli kullanıcı λ bu kullanıcılarla gizlice anlaşarak $t_2 - t_1$ zaman aralığındaki güncel ortak gizli anahtarı öğrenmeye çalışsın. Tip-III düşman modeli gizli anlaşma atağının bir örneğidir.

Tip-IV düşman modeli: Bir kötü niyetli kullanıcı λ'nın, YM'den gelen bir mesajı yakalayıp, bu mesajı YM gibi davranarak şema kullanıcılarına ilettiğini varsayalım. Tip-IV düşman modeli tekrarlama atağının bir örneğidir.

Tip-V düşman modeli: Bir kötü niyetli kullanıcı λ'nın, şemaya katılmadan bir kullanıcıyla uzlaştığını ve böylece ortak gizli anahtara ve grup mesajlarına eriştiğini varsayalım. Tip-V düşman modeli uzlaşma atağının bir örneğidir.

2.3. GRUP İLETİŞİM GEREKSİNİMLERİ

Bu bölümde grup iletişim gereksinimleri güvenlik ve servis kalitesi olmak üzere iki alt bölüme ayrılarak açıklanmıştır.

2.3.1. Güvenlik Gereksinimleri

İleri gizlilik: Yayın ortamından ayrılan bir kullanıcının gelecek yayın mesajları çözmesini engellemektir.

Geri gizlilik: Yayın ortamına eklenen bir kullanıcının geçmiş mesajları çözmesini engellemektir.

Kimlik doğrulama: Bir GGİ şemasında, grup kullanıcılarına erişim izni verilmeden önce kimlik doğrulaması yapılmalıdır. Aksi halde kimlik tabanlı ataklara maruz kalınabilir [48], [50].

Grup mesaj bütünlüğü: Mesaj bütünlüğü korunmalı, kimliği doğrulanmamış varlıklar tarafından mesajın bir kısmı veya tamamı üzerinde ekleme, silme ve düzeltme yapılmasına izin verilmemelidir [48], [50].

Grup mesaj gizliliği: Sadece kimliği doğrulanmış yetkili kullanıcılar şifreli veriden anlamlı mesajı elde edebilmelidir. Gizlilik bir grup anahtarı ile şifreleme yapılarak sağlanabilir. Mesaj içeriğinin önemine göre, grup iletişimindeki mesajların bir kısmı güçlü şifreleme anahtarları ile şifrelenirken, bir kısmı daha zayıf anahtarlar ile şifrelenebilir [48].

Grup üyesiyle uzlaşmaya dayanıklılık: Saldırgan grupta bulunan bir üye ile anlaşma sağlayarak grup anahtarını elde edebilir ve gizli verilere erişim sağlayabilir. Bir GGİ şemasında düğümler sürekli kontrol edilmeli, gizli anlaşma yapan bir düğüm tespit edildiğinde hemen gruptan çıkartılmalıdır [48].

Yeniden anahtarlama: Her üyelik değişimi sonunda ortak gizli anahtar başta olmak üzere bazı anahtarların güncellenmesi gerekir. Güncelleme işlemi mümkün olduğunca hızlı yapılmalıdır. Aksi halde, ayrılan kullanıcılar grup anahtarı güncellenene kadar grup

mesajlarını elde edebilir. Eklenen kullanıcılar ise grup iletişimine hemen dahil olamayabilir [48].

Grup bağımsızlığı: Bir düğüm birden fazla gruba ait olabilir. Her grubun güvenlik parametreleri birbirinden farklı ve bağımsız olmalıdır [48].

2.3.2. Servis Kalitesi Gereksinimleri

Hizmet devamlılığı: Bir GGİ şemasında grup işlemleri sırasında ortaya çıkan tek bir sorun grup iletişim işlemini etkilememelidir [51], [52].

Ölçeklenebilirlik: İster küçük ister büyük bir kullanıcı grubu olsun bir GGİ şemasında anahtar yönetim, güvenlik ve etkililik özellikleri sağlanmalıdır. Grup yönetim veya üyelik işlemlerinin ardından güncellenen ortak gizli anahtarın ölçülebilir bir gecikme, kabul edilebilir bir hesaplama ve iletişim maliyetiyle iletimi sağlanmalıdır [51], [52], [53].

Güvenilirlik: Bir GGİ şemasında güncel anahtarların dağıtımı güvenli olmalıdır. Gruptaki tüm üyelere anahtarlar güvenli bir yoldan zamanında dağıtılmalıdır [51], [52].

Esneklik: Bir GGİ şeması farklı uygulamalar içerisinde kullanılabilir olmalıdır. Kullanıcı ekleme-çıkarma işlemleri herhangi bir zamanda yapılabilmelidir [48], [54].

2.4. MERKEZİ GÜVENLİ GRUP İLETİŞİM ŞEMALARI

Merkezi grup anahtar yönetim şemalarında, GY olarak isimlendirilen bir merkezi güvenli varlık bulunur. Bir GY, grup içerisinde bulunan anahtarların üretiminden, dağıtımından ve güncellenmesinden sorumludur. Merkezi grup anahtar yönetim şemalarında genellikle simetrik şifreleme yöntemlerinden yararlanılır.

2.4.1. İkili Anahtarlar

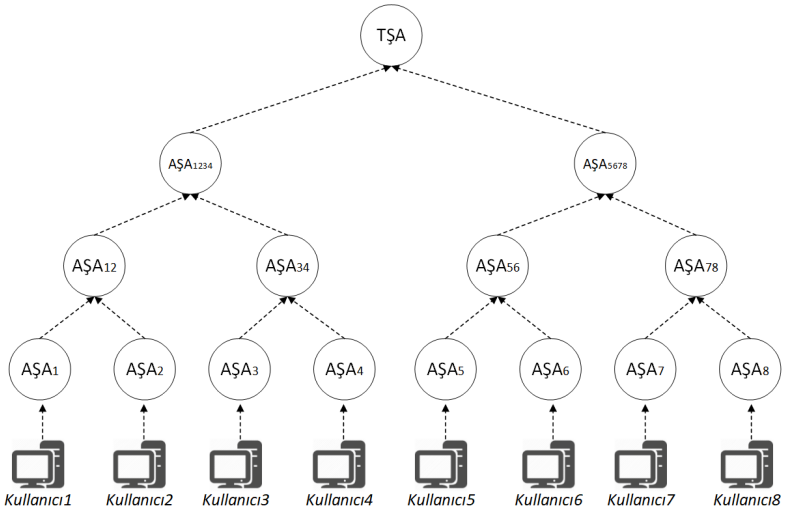
GY, gruptaki her üyeyle birebir iletişim kurar. Her üyenin kendisine ait bir gizli anahtarı bulunur. GY bu gizli anahtarı kullanarak üyelere güncel anahtarı iletir. Herhangi bir grup üyeliğinin sonunda anahtarın güncellenmesi gerekir. Bunun için GY, her üye ile birebir iletişime geçmeli ve üyenin gizli anahtarını kullanarak kendisine güncel anahtarı

iletmelidir. Bu yöntemde geri gizlilik sağlanırken, ileri gizlilik için yeniden anahtarlama maliyeti $O(n)$ dir. Bu nedenle yöntem dinamik ve geniş gruplar için elverişli değildir.

2.4.2. Mantıksal Anahtar Hiyerarşisi

Mantıksal Anahtar Hiyerarşisi (MAH) birbirlerinden bağımsız iki araştırma grubu olan Wong ve arkadaşları [55] ve Waller ve arkadaşları [56] tarafından yaklaşık olarak aynı zamanda önerilmiştir. Günümüzde popüler olarak kullanılan grup anahtar yönetim şemalarından bir tanesidir. MAH'ın ana fikri yetkili kullanıcıların eklendiği bir anahtar ağacı oluşturmaktır. Anahtar ağaç, bağlı düğüm içermeyen yani içerisinde başladığı yerde biten bir yol olmayan, bir yönlü döngüsüz ağaçtır.

Bir mantıksal anahtar ağacı GY tarafından yönetilir. Bir ağaç içerisinde, kullanıcı düğümleri ve anahtar düğümleri olmak üzere iki farklı düğüm vardır. Kullanıcı düğümleri ağaçtaki en alt düğümler olan yapraklarda yer alır. Bir yaprak düğümde, yaprakta yer alan kullanıcı düğümü ile ilişkili bireysel anahtarlar bulunur. Yayın merkezi ise kök düğümde yer alır. Yayın merkezi ile Trafik Şifreleme Anahtarı (TŞA) olarak isimlendirilen bir anahtar ilişkilendirilir. Yayın merkezinden kullanıcılara kadar olan yolda bulunan anahtar düğümler aradüğümler olarak isimlendirilir. Aradüğüm anahtarları Anahtar Şifreleme Anahtarı (AŞA) olarak isimlendirilir ve anahtar sunucu tarafından grup üyelerine TŞA anahtarını güvenli bir şekilde teslim etmek için kullanılır. Şema üzerinde bulunan her kullanıcıya, kök düğümde bulunan TŞA'yı hesaplayabilmesi için, kendisinden kök düğüme kadar olan yolda bulunan aradüğüm anahtarları güvenli bir yoldan iletilmelidir.



Şekil 2.1. 8 Kullanıcılı 3 Dereceli Bir Anahtar Ağacı.

İleri ve geri gizliliğin sağlanması için, şemaya bir kullanıcı eklendiğinde veya şemadan bir kullanıcı çıkartıldığında, kullanıcının bulunduğu noktadan kök düğüme kadar olan yol üzerinde bulunan anahtarlar güncellenmelidir. Şekil 2.1'de 8 kullanıcılı dolu bir ağaç bulunmaktadır.

Örneğin $Kullanıcı_8$ ağaca katıldığı zaman GY, $AŞA_8, AŞA_{78}, AŞA_{5678}$ ve TŞA anahtarlarını güncellemeli ve $Kullanıcı_8$'e iletmelidir.

Bir kullanıcı sunucuya katılma isteği gönderdiğinde, öncelikle GY ile kullanıcı arasında doğrulama işlemi gerçekleştirilir. Eğer kullanıcının kimliği doğrulanırsa, bir kullanıcı düğümü ve onunla ilişkili bir anahtar düğümü oluşturulur. Oluşturulan anahtar değeri kullanıcıya güvenli bir yolla iletilir. Bir sonraki adım kullanıcının ekleneceği düğümü bulmaktır. Eğer bir altkümedeki düğüm derecesi, ağacın derecesinden küçükse, bir diğer ifadeyle altkümedeki kullanıcı sayısı ağacın maksimum kullanıcı sayısından küçükse, yeni kullanıcı bu altküme içerisine eklenir. Eğer değilse, yeni bir altküme yaratılır ve yeni kullanıcı bu alt kümeye eklenir.

Ağaca eklenen veya çıkartılan her bir kullanıcı için o kullanıcının bireysel anahtarı oluşturulmalı veya silinmelidir. Kullanıcının bulunduğu yol üzerindeki diğer h adet düğüm anahtarı değişmek zorundadır. Yeni kullanıcıya eklendiği katılım noktasından köke kadar ki tüm anahtar değerleri verilmelidir. Aynı şekilde grup anahtarı değişen düğümlere de yeni grup anahtarları dağıtılmalıdır. n kullanıcı için anahtar dağıtım karmaşıklığı $O(logn)$'dir.

Eğer ağaca yeni kullanıcılar eklenir ve ağaçta yeniden anahtarlama işlemi yapılmazsa, yeni kullanıcılar önceki mesajları çözebilme imkânına sahip olur. Bunu engellemek için ağaç üzerinde kullanıcının eklendiği ekleme noktasından köke kadar ki tüm düğümlere yeni anahtar değerleri atanmak ve o düğümlerin kullanıcılarına güncel anahtar değerleri dağıtılmak zorundadır. Bu sayede geriye erişim kontrolü garanti altına alınmış olur.

$Kullanıcı_8$'in ağaca eklendiğini varsayalım. $Kullanıcı_8$ ağaca dahil olduğunda aşağıdaki adımlar sırasıyla gerçekleştirilir.

$(AŞA'_{78})_{AŞA_8} \rightarrow AŞA'_{78}$ anahtarı $AŞA_8$ anahtarıyla şifrelenerek $Kullanıcı_8$'e gönderilir.

$(AŞA'_{78})_{AŞA_7} \rightarrow AŞA'_{78}$ anahtarı $AŞA_7$ anahtarıyla şifrelenerek $Kullanıcı_7$'e gönderilir.

$(AŞA'_{5678})_{AŞA_8} \rightarrow AŞA'_{5678}$ anahtarı $AŞA_8$ anahtarıyla şifrelenerek $Kullanıcı_8$'e gönderilir.

$(AŞA'_{5678})_{AŞA_{5678}} \rightarrow AŞA'_{5678}$ anahtarı $AŞA_{5678}$ anahtarıyla şifrelenerek $Kullanıcı_5$, $Kullanıcı_6$ ve $Kullanıcı_7$'e gönderilir.

$(TŞA')_{AŞA_8} \rightarrow TŞA'$ anahtarı $AŞA_8$ anahtarıyla şifrelenerek $Kullanıcı_8$'e gönderilir.

$(TŞA')_{TŞA} \rightarrow TŞA'$ anahtarı $TŞA$ anahtarıyla şifrelenerek $Kullanıcı_1$, $Kullanıcı_2$, $Kullanıcı_3$, $Kullanıcı_4$, $Kullanıcı_5$, $Kullanıcı_6$ ve $Kullanıcı_7$'ye gönderilir.

Bir MAH şemasında bulunan kullanıcı sayısı n ise, ağacın yüksekliği $h = log_2 n$'dir. Ağacın derinliği onun derecesi olarak ifade edilir. Her kullanıcı, bir tanesi kendi bireysel anahtarı olmak üzere, bulunduğu düğümden kök düğüme kadar olan yolda bulunan toplam $h+1$ adet anahtarı saklamaya ihtiyaç duyar. Diğer h adet anahtar değeri o düğümden kök düğüme olan aradüğümlerin anahtarlarıdır. Bu nedenle gereken saklama alanı boyutu $O(h)$'dır.

Bir kullanıcıyı silmek ekleme işlemine benzer yolla yapılır. Öncelikle kullanıcıyla ilişkili düğümler ağaçtan silinir. Silinen kullanıcının gelecek mesajları çözmesini engellemek için, bulunduğu noktadan kök düğüme kadar ki yol üzerindeki tüm aradüğümler için yeni anahtarlar hesaplanır. Ardından bu anahtarlar yol üzerinde bulunan kullanıcılara dağıtılır. Bu sayede ileri gizlilik sağlanmış olur.

Şekil 2.1 üzerinde $Kullanıcı_8$'in ağaçtan silindiğini varsayalım. Bu durumda $Kullanıcı_8$'in ağaç üzerinde yayınlanacak mesajları okumasını engellemek için, kullanıcının bulunduğu konumdan kök düğüme kadar olan düğümler üzerinde yeniden anahtarlama işlemlerinin yapılması gerekir.

Öncelikle $Kullanıcı_8$'e ait $AŞA_8$ anahtarı silinir. Ardından $AŞA_{78}$, $AŞA_{5678}$ ve $TŞA$ anahtarları güncellenir. Güncelleme işleminin ardından yeni $AŞA'_{78}$, $AŞA'_{5678}$ ve $TŞA'$ anahtarlarının diğer kullanıcılara güvenli bir şekilde iletilmesi gerekir. Bunun için aşağıdaki işlem adımları uygulanır.

$(AŞA'_{78})_{AŞA_7} \rightarrow AŞA'_{78}$ anahtarı $AŞA_7$ anahtarıyla şifrelenerek $Kullanıcı_7$'e gönderilir.
$(AŞA'_{5678})_{AŞA'_{78}} \rightarrow AŞA'_{5678}$ anahtarı $AŞA'_{78}$ anahtarıyla şifrelenerek $Kullanıcı_7$'e

gönderilir.
$(AŞA'_{5678})_{AŞA_{56}} \rightarrow AŞA'_{5678}$ anahtarı $AŞA_{56}$ anahtarıyla şifrelenip $Kullanıcı_5$ ve $Kullanıcı_6$'ya gönderilir.
$(TŞA')_{AŞA'_{5678}} \rightarrow TŞA'$ anahtarı $AŞA'_{5678}$ anahtarıyla şifrelenerek $Kullanıcı_5$, $Kullanıcı_6$ ve $Kullanıcı_7$'ye gönderilir.
$(TŞA')_{AŞA_{1234}} \rightarrow TŞA'$ anahtarı $AŞA_{1234}$ anahtarıyla şifrelenerek $Kullanıcı_1$, $Kullanıcı_2$, $Kullanıcı_3$ ve $Kullanıcı_4$'e gönderilir.

Bir mesaj tüm yetkili kullanıcılara yayınlandığında, basit bir şekilde bir kök anahtar ile şifrelenir. Çünkü yalnız yetkili kullanıcılar kök anahtarına sahiptir ve herhangi bir ek hesaplama yapmadan yayın mesajını çözebilirler. Bu nedenle bir yayın şifreleme ve çözme karmaşıklığı $O(1)$'dir.

2.4.3. Tek Yönlü Fonksiyon Ağacı

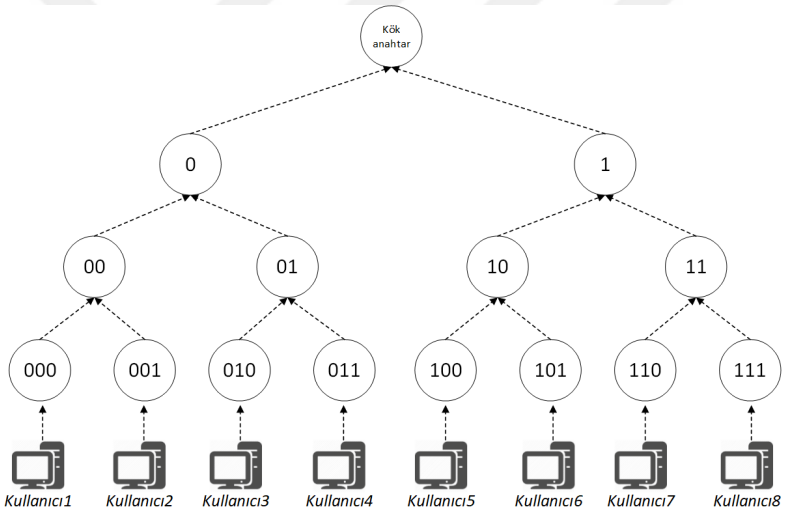
Tek-yönlü Fonksiyon Ağacı (TFA) geniş ve dinamik gruplar için etkili bir anahtar yönetim şemasıdır. Sherman ve McGrew tarafından önerilmiştir [57]. Ağaç üzerindeki kullanıcı anahtarları aşağıdan yukarıya doğru bir tek yönlü fonksiyon $g()$, bir birleştirme fonksiyonu $f()$ ve bir anahtar oluşturma fonksiyonu $anahtar()$ kullanılarak hesaplanır. Kullanıcılar yapraklarda bulunur. Her kullanıcı düğümü (x) ile ilişkilendirilmiş üç kriptografik anahtar bulunur:

- n_x düğüm sırrıdır.
- n'_x ise kör (blinded) düğüm sırrıdır. Bir tek yönlü fonksiyon $g()$ kullanılarak hesaplanır: $n'_x = g(n_x)$
- Her düğüm anahtarı k, düğüm sırrının $anahtar()$ fonksiyonuna girmesiyle elde edilir: $k_x = anahtar(n_x)$
- $f()$ fonksiyonu ise karıştırma (ör. XOR) fonksiyonudur.

MAH şemasında anahtar değerleri kökten yapraklarda bulunan kullanıcılara yukarıdan aşağıya doğru dağıtılırken, TFA şemasında yapraklarda bulunan kullanıcılardan köke doğru aşağıdan yukarıya bir yol izler.

Her düğüm sırrı n_x, sol ve sağ çocuklarının kör düğüm sırlarının birleştirilmesinden elde edilir. Ara düğüm hesaplaması Denklem 2.1'deki gibidir.

$$
\begin{aligned}
n_x &= f(g(n_{x_{sol_{cocuk}}}), g(n_{x_{sag_{cocuk}}})) = f(n'_{x_{sol_{cocuk}}}, n'_{x_{sag_{cocuk}}}) \\
n_{00} &= f(g(n_{000}), g(n_{001})) \\
n_{01} &= f(g(n_{010}), g(n_{011})) \\
n_{10} &= f(g(n_{100}), g(n_{101})) \\
n_{11} &= f(g(n_{110}), g(n_{111})) \\
n_0 &= f(g(n_{00}), g(n_{01})) \\
n_1 &= f(g(n_{10}), g(n_{11})) \\
n_{kök_{anahtar}} &= f(g(n_0), g(n_1))
\end{aligned}
\tag{2.1}
$$



Şekil 2.2. Kullanıcılardan Kök Düğüme Gizli Anahtar Hesaplama.

Yapraklarda bulunan her kullanıcının, düğüm sırlarını hesaplayabilmesi için kendisinden kök düğüme kadarki yolda kardeş kör düğüm sırlarını bilmesi gerekir. Örneğin; Şekil 2.2'de $Kullanıc_7$'ye n'_{111}, n'_{10} ve n'_0 değerleri iletilmelidir. Aksi halde kullanıcı grup anahtarını hesaplayamaz. Sistem güvenliği her üyenin anahtar ağacının mevcut durumu hakkındaki bilgisine dayalıdır.

Ağaca bir kullanıcının eklenmesi için, öncelikle ağaç üzerinde konumunun belirlenmesi gerekir. Ardından o konumdaki yaprak düğüm $x, sol(x)$ ve $sağ(x)$ olmak üzere ikiye bölünür. Mevcut düğüm sol kısımla yeni kullanıcı ise sağ kısımla ilişkilendirilir. Yeni anahtarlar oluşturulup her iki kullanıcıya da verilir. Güncellenen kör düğüm sırları ilgili kullanıcılara güvenli bir şekilde dağıtılır. Yeni eklenen kullanıcıya grup anahtarını

hesaplayabilmesi için güvenli bir yoldan kendisinden kök düğüme kadar ki yolda bulunan kardeş düğüm kör anahtar sırları iletilir. Ağacın yüksekliğini mümkün olduğunca düşük tutmak için, bir kullanıcı eklenmek istendiğinde kök düğüme en yakın düğüm bölünerek ekleme işlemi yapılır.

Bir kullanıcı gruptan ayrıldığında kardeşi grubun ebeveyni konumuna getirilir ve kendisine yeni bir yaprak anahtar değeri verilir. Değişen kör düğüm anahtarlarının yeni değerleri ihtiyaç duyan kullanıcılara güvenli bir şekilde aktarılır.

Eğer ağacın yüksekliği h ise, bir ekleme veya çıkarma işleminin ardından yaklaşık olarak h adet yeni anahtar iletimi yapılır. MAH ağacının aksine anahtar değerleri fonksiyonel olarak birbirine bağlı olduğundan, TFA ağacında bir ekleme veya çıkarma işleminin ardından x bitlik anahtar iletimleri söz konusu iken bu değer MAH ağacında $2x$ boyutundadır.

2.4.4. Tek Yönlü Fonksiyon Zinciri

Tek-yönlü Fonksiyon Zinciri (TFZ) Canetti ve arkadaşları tarafından 1999 yılında önerilmiş [58], Sherman ve arkadaşları tarafından isimlendirilmiştir [59]. TFA'nın aksine TFZ de kör düğüm sırları bulunmaz. Biri düğüm sırrı (r_x) diğeri de düğüm anahtarı (k_x) olmak üzere her düğüm ile iki anahtar ilişkilendirilir. Kullanıcılar yapraklarda bulunur. Ağaçtan ayrılan kullanıcının ebeveyni ile kök düğüm arasında bulunan düğüm sırları arasında her zaman fonksiyonel bir ilişki bulunur. Bu ilişki zincir olarak isimlendirilir.

TFZ içerisinde kullanılan f tek yönlü bir rasgele üreteçtir. İçerisine giren bir verinin iki katı boyutunda çıktı üretir. Bu fonksiyon düğüm anahtarları ve düğüm sırlarını üretmek için kullanılır.

Şekil 2.2 görüldüğü üzere, ağaçta 8 kullanıcı olduğunu ve ağaçtan $Kullanıc1_1$'in ayrıldığını varsayalım. Öncelikle $Kullanıc1_1$ ile ilişkili düğüm ağaç üzerinden silinmelidir. Ardından silinen düğümün ebeveyninden kök düğüme kadar ki yol üzerinde bulunan düğüm anahtar ve düğüm sır değerlerinin güncellenmesi ve güncel anahtarların ihtiyaç duyan kullanıcılara dağıtılması gerekir.

GY yeni bir rasgele r_{00} değeri oluşturur. Oluşturulan r_{00}, k_{001} ile şifrelenip $Kullanıc1_2$'ye gönderilir. Ardından r_{00} bir $f()$ fonksiyonuna sokulur. Bu fonksiyonun sol yarısı ilgili

düğümün düğüm anahtarı olan $k'_{00} = f(r_{00})|_L$ ile, sağ yarısı ise üst düğümün düğüm sırrı olan $r_0 = f(r_{00})|_R$ ile ilişkilidir. r_0 değeri, k_{01} ile şifrelenip $Kullanıcı_3$ ve $Kullanıcı_4$'e gönderilir. $k'_0 = f(r_0)|_L$ işlemi ile bu düğümün düğüm anahtarı, $r_{kök_{anahtar}} = f(r_0)|_R$ işlemi ile kök düğümün düğüm sırrı elde edilir. $r_{kök_{anahtar}}$ değeri k_1 ile şifrelenip $Kullanıcı_5 - Kullanıcı_8$'e gönderilir. Son olarak $k'_{kök_{anahtar}} = f(r_{kök_{anahtar}})|_L$ işlemiyle kök düğümün düğüm anahtarı elde edilir. Ağaçta bulunan kullanıcılar ihtiyaç duydukları anahtarları kolaylıkla hesaplayabilirler.

Örneğin $Kullanıcı_2$, r_{00} değerini elde etmek için k_{001} anahtarını kullanır. Ardından sırasıyla $k'_{00} = f(r_{00})|_L$, $k'_0 = f(f(r_{00})|_R)|_L$'yi ve $k'_{kök_{anahtar}} = f(f(f(r_{00})|_R)|_R)|_L$ işlemlerini yaparak kök düğümün güncel anahtarını elde eder. İletişim maliyeti $logn$'dir. TFZ üzerinde düğüm sırları arasındaki zincir ilişki genellikle bir kullanıcı eklendiğinde değil, bir kullanıcı ayrıldığında ortaya çıkmaktadır.

2.5. DAĞITIK GÜVENLİ GRUP İLETİŞİM ŞEMALARI

Dağıtık GGİ şemalarında merkezi bir varlık bulunmaz. Tüm grup üyeleri aralarında iş birliği yaparak, bir GGİ iş yükünü paylaşırlar. Dağıtık şemaların çoğu GÜY ile ilgilenmezler yalnızca GAY çözümleri sunarlar. Merkezi grup anahtar yönetimi ile karşılaştırıldığında, dağıtık grup anahtar yönetim şeması hata toleransı açısından avantaja sahip iken, hesaplama maliyeti açısından dezavantaja sahiptir [60]. Literatürde sıklıkla önerilen dağıtık GGİ şemaları aşağıda sunulmuştur.

2.5.1. Ağaç Tabanlı Grup Diffie-Hellman

Ağaç-tabanlı Grup Diffie-Hellman (AGDH), ikili ağaç temellidir. 2004 yılında Kim ve arkadaşları [61] tarafından önerilmiştir. Yöntemde açık anahtarlı şifreleme algoritmaları kullanılır. AGDH içerisinde kullanılan notasyonlar şunlardır:

- $< l, v >$: ağacın l.seviyesindeki v.düğüm anlamına gelir.
- $k_{<l,v>} :< l, v >$ düğümünün anahtarı (gizli anahtar) anlamına gelir.
- $bk_{<l,v>} :< l, v >$ düğümünün kör anahtarı (açık anahtar) anlamına gelir.
- $T_{<i,j>} :< i, j >$ düğümünün kök olduğu alt ağaç anlamına gelir.

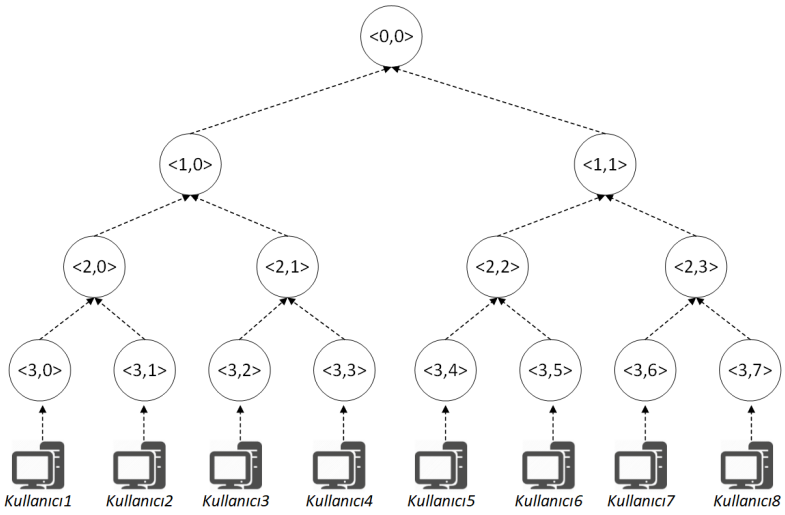
AGDH yönteminde $< 0,0 >$ düğümü kök düğümdür. Kullanıcılar yapraklarda bulunur. Diğer düğümler aradüğümlerdir. Her aradüğümün iki çocuğu bulunur. $< l,v >$ düğümünün sol çocuğu $< l+1,2v >$ konumunda, sağ çocuğu ise $< l+1,2v+1 >$ konumunda bulunmaktadır. Z_p içerisinde g primitif bir kök olmalı ve p değeri olarak büyük bir asal sayı seçilmelidir. $< l,v >$ konumundaki düğümün kör anahtarı $bk_{<l,v>} = g^{k_{<l,v>}}$ denklemiyle hesaplanır. Kök düğüm haricindeki her düğümün bir gizli birde açık anahtarı bulunur. Aradüğüm anahtarları aşağıdan yukarıya anahtarların kullanılmasıyla hesaplanır. Bu nedenle AGDH, TFA ile benzerlik göstermesine rağmen aralarında beş önemli fark bulunur.

- TFA'daki kör anahtar ortaya çıkarılmaması gereken bir anahtar değeri iken, AGDH'daki tüm kör anahtarların, açık anahtar altyapısına sahip olduklarından, gizli tutulmasına gerek yoktur.
- TFA merkezi iken AGDH dağıtıktır. AGDH'da gizli anahtarı taşıyan hiçbir merkezi varlığa gerek yoktur.
- TFA'dan farklı olarak AGDH'da ayrıca birleştirme ve bölme protokolleri tanımlanmıştır.
- Yapıların güvenliği birbirinden farklıdır.
- TFA bir k-ary temelli olduğundan farklı k değerlerinde ağaç oluşturulabilirken, AGDH'de bir aradüğümün en fazla 2 çocuğu bulunabilir.

Aradüğüm anahtarları, Diffie-Hellman anahtar değişimi kullanılarak hesaplanır. Denklem 2.2'de görüldüğü üzere aradüğüm gizli anahtarları ise, aradüğümlerin çocuklarından birinin açık anahtar değerinin, diğer çocuğun gizli anahtar değeriyle üs işlemine tabi tutulup, ağacın kurulum aşamasında belirlenen p değeriyle modu alınarak elde edilir.

$$\begin{aligned} k_{<l,v>} &= (bk_{<l+1,2v>})^{k_{<l+1,2v+1>}} mod\, p \\ &= (bk_{<l+1,2v+1>})^{k_{<l+1,2v>}} mod\, p \\ &= g^{(k_{<l+1,2v>} * k_{<l+1,2v+1>})} mod\, p \end{aligned} \tag{2.2}$$

8 elemanlı bir AGDH ağacı Şekil 2.3'deki gibidir. $k_{<2,0>}$ düğümünün gizli anahtarının hesaplanması için Denklem 2.3'deki işlem gerçekleştirilir. Her kullanıcı, kendisinden kök düğüme kadarki yol üzerinde bulunan düğümlerin gizli anahtarlarını hesaplayabilmesi için, kendisinden kök düğüme kadarki yol üzerinde bulunan kardeş düğüm açık anahtarlarını



Şekil 2.3. 8 Elemanlı AGDH Şeması.

bilmelidir. Bir yeni üye AGDH'a katılmak isterse, gruba içinde kendi açık anahtarının da yer aldığı bir istek mesajı göndermelidir.

$$\begin{aligned} k_{<2,0>} &= (bk_{<3,0>})^{k_{<3,1>}} mod p \\ &= (bk_{<3,1>})^{k_{<3,0>}} mod p \\ &= g^{(k_{<3,0>} * k_{<3,1>})} mod p \end{aligned} \tag{2.3}$$

Gruptaki mevcut kullanıcılar, yeni üyenin ekleneceği konumu ve sponsor düğümü belirler. Sponsor düğüm eklenen ya da ayrılan kullanıcının konumuna göre mevcut üyelerden seçilir. Sponsor düğüm kullanıcı ekleme-çıkarma işlemlerinin ardından, AGDH üzerinde ileri ve geri gizliliğin sağlanması için, anahtar güncelleme ve güncel anahtarları yayınlama görevini geçici olarak üstlenir.

Bir sponsor aşağıdaki gibi tanımlanır:

- Bir alt ağacın sponsoru, alt ağaçtaki en sağdaki yaprakta bulunan üyedir.
- Bir yaprak düğümün sponsoru, kendisinin üyesi olduğu en alt ağacın sağındaki (kendisi hariç) yaprak düğümdür.

Ekleme işleminde, ağaçta bulunan her üye, yeni eklenen kullanıcı için bir yeni yaprak düğüm ve bir yeni aradüğüm oluşturur. Aradüğümü yeni eklenen düğümün ebeveyni olacak şekilde ağacı günceller. Ayrıca her üye sponsor düğümden kök düğüme kadar ki yol üzerinde bulunan tüm kör anahtarları ve gizli anahtarları silerler.

Sponsor düğüm, bulunduğu konumdan kök düğüme kadarki yol üzerinde bulunan gizli ve kör anahtarları hesaplar. Ardından, kör anahtarları içeren yeni ağacı yayınlar. Ağaçta bulunan diğer kullanıcılar güncel kör anahtarları ile yeni grup anahtarını hesaplarlar.

Gruptan bir üye ayrıldığında, ilk olarak sponsor düğüm belirlenir. Ağaçta bulunan diğer kullanıcılar ayrılan kullanıcı düğümünü siler. Silinen kullanıcının kardeş düğümü ebeveyn düğümü konumuna getirilir. Sponsor düğüm kendisinden kök düğüme kadar tüm gizli anahtar ve kör anahtar değerlerini hesaplar ve yeni kör anahtar setini yayınlar. Ağaçta bulunan diğer kullanıcılar güncel kör anahtarları ile yeni grup anahtarını hesaplarlar.

2.5.2. Dağıtık Ölçeklendirilebilir Güvenli İletişim

Dağıtık Ölçeklendirilebilir Güvenli İletişim (DÖGİ), Dondeti ve arkadaşları tarafından önerilmiş dağıtık bir şemadır [62]. TFA şeması ile aralarında terminoloji ve notasyon açısından bazı farklılıklara sahiptir. TFA şemasındaki bir tek yönlü fonksiyon f(), DÖGİ şemasında E olarak ifade edilir.

TFA içindeki bir düğüm sırrı, DÖGİ içerisinde gizli anahtar olarak isimlendirilir. TFA içindeki bir kör düğüm sırrı, DÖGİ içinde kör anahtar (açık anahtar) olarak isimlendirilir. TFA içindeki düğüm anahtarları, DÖGİ içindeki düğüm sırlarına denk gelmektedir.

E fonksiyonu, basit karıştırma işlemi yapan bir tek yönlü fonksiyondur; $mix(x,y) = x \oplus y$. Her X düğümünde iki anahtar bulunur: bir gizli anahtar k_x ve bir kör anahtar $E(k_x)$. Bir kullanıcı, düğümünün gizli anahtarını rasgele seçer ve gizli tutar. Bir aradüğümün gizli anahtarı çocuklarının kör anahtarları karıştırılarak hesaplanır. Her düğümün kör anahtarı, gizli anahtarının E fonksiyonuna sokulmasıyla elde edilir.

Her kullanıcı ikilik düzende bir kimliğe sahiptir. Kullanıcı düğümden kök düğüme kadar anahtar hesaplaması yapabilmesi için, kullanıcıların komşularının belirlenmesi gerekir. Bunun için ikilik düzendeki kimlik bilgisi kullanılarak AnahtarIsbirligiBul() algoritması ile her düğümün anahtar iş birliği grubu belirlenir. Kullanıcı bu gruptaki üyelerden sırasıyla kör anahtar bilgilerini alarak kök düğüme kadar anahtar hesaplama işlemini gerçekleştirir.

Ağaca yeni bir kullanıcının (x düğüm) katılması durumunda öncelikle kullanıcının ekleneceği pozisyon belirlenir. Kendisine, kök düğümünden bulunduğu konuma doğru bir

ikilik düzende bir kimlik verilir. Kullanıcı düğümünden kök düğüme kadarki yol üzerinde bulunan anahtar hesaplamaları aşağıdaki gibidir:

1. AnahtarIsbirligiBul() algoritması kullanılarak x düğümün anahtar işbirliği grubu belirlenir.
2. x düğümü, gizli anahtar k_x için bir rasgele değer seçer.
3. k_x'den kör anahtar değeri $E(k_x)$ hesaplanır.
4. x düğümü, sırasıyla anahtar işbirliği grubundaki komşusundan ilgili düğümün kör anahtar bilgisini alır.
5. Bir üst düğümün gizli anahtarı hesaplanır, $k_{ebeveyn} = mix(E(k_x), E(k_y)) = E(k_x) \oplus E(k_y)$
6. Bir üst düğümün kör anahtarı hesaplanır, $E(k_{ebeveyn})$.

4-6 arası adımlar anahtar işbirliği grubundaki komşuların sayısı kadar devam eder. Bu işlemin sonucunda yeni kullanıcı güncel kök düğüm kör anahtarı ve gizli anahtar değerlerini hesaplar. Ağaçtaki mevcut kullanıcılar yukarıdaki adımlara benzer şekilde komşularının güncel kör anahtar değerlerini elde ederek güncel kök düğüm kör anahtar ve gizli anahtar değerlerini hesaplarlar.

Kullanıcı çıkarma işleminde ise, bir üyenin ayrılmasının ardından kardeşi yeni bir kimliğe sahip olarak ebeveyn pozisyonuna gelir. Yeni bir gizli anahtar üretir. Ardından AnahtarIsbirligiBul() algoritması ile önceden tespit edilmiş anahtar işbirliği grubu üyelerine sırasıyla kör anahtar bilgilerini hesaplayıp göndermesi gerekir. Güncel kör anahtar bilgilerini alan üyeler kendi alt gruplarındaki üyelere güncel anahtar bilgisini göndermekten sorumludur.

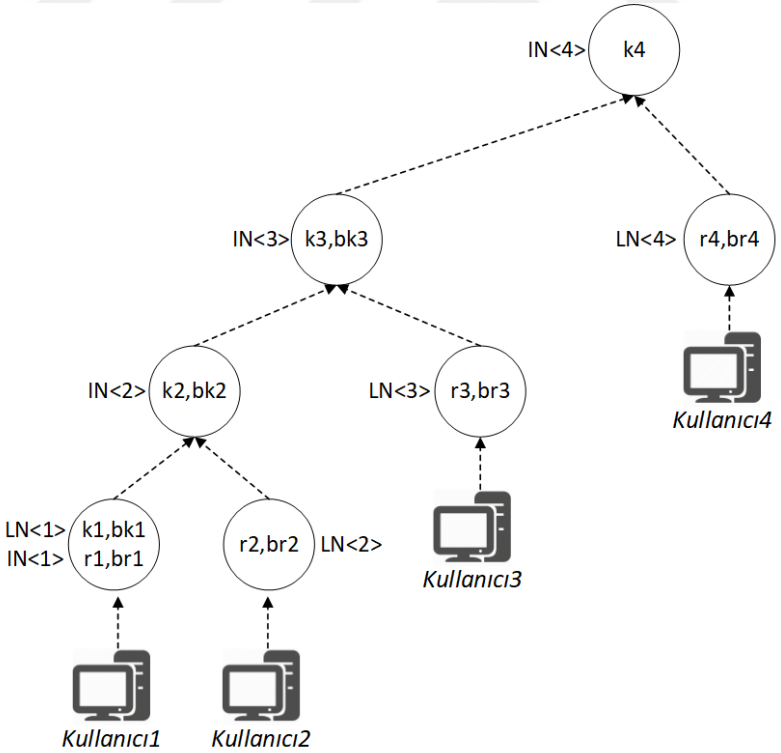
2.5.3. Sıska Ağaç

SISA şeması ilk olarak [63] çalışmasıyla ortaya atılmış, [64] çalışması bu yöntemi dinamik grup işlemlerini kapsayacak şekilde genişletmiştir. Bir SISA şemasında aşağıdaki notasyonlar kullanılır.

- n, N: grup üyelerinin sayısı
- $Kullanıcı_i$: i'inci grup üyesi

- p: büyük bir asal sayı
- α: üs alma tabanı
- r_i: $Kullanıcı_i$'nin gizli anahtarı
- br_i: $Kullanıcı_i$'nin kör (açık) anahtarı $(\alpha^{r_i} mod p)$
- k_j: $Kullanıcı_1 \ldots Kullanıcı_j$ arasında paylaşılan gizli anahtar
- bk_j: k_j'nin kör anahtarı $(\alpha^{k_i} mod p)$
- $N_{<j>}$: j ağaç düğümü
- $IN_{<1>}$: l seviyesindeki iç ağaç düğümü
- $LN_{<i>}$: $Kullanıcı_i$ üyesi ile ilişkili yaprak düğüm
- $T_{<i>}$: $Kullanıcı_i$ üyesinin ağacı
- $BT_{<i>}$: $Kullanıcı_i$ üyesinin tüm kör anahtarlarını içeren ağaç

Bir SISA ağacında, yaprak düğüm ve aradüğüm olmak üzere iki çeşit düğüm bulunur. En yüksek indisli aradüğüm kök düğüm olarak isimlendirilir. Kök düğümde bulunan gizli anahtar grup anahtarı olarak paylaşılır.



Şekil 2.4. Bir SISA Şeması.

Şekil 2.4'de görüldüğü üzere bir aradüğüm $IN_{<i>}$'ın her zaman iki çocuğu bulunur: Bu çocuklardan biri aradüğüm $IN_{<i-1>}$ diğeri ise yaprak düğüm $LN_{<i+1>}$'dir. Bir istisna $IN_{<1>}$ aradüğüm olmakla birlikte yaprak düğüm $Kullanıcı_1$ ile de ilişkilidir.

Her yaprak düğüm $LN_{<i>}$ bir gizli anahtar r_i'e sahiptir. $LN_{<i>}$'nin kör anahtarı r_i kullanılarak elde edilir: $br_i = \alpha^{r_i} mod p$ Her aradüğüm $IN_{<j>}$, bir gizli anahtar k_j ile ilişkilidir. $IN_{<j>}$'ın kör anahtarı, k_j kullanılarak elde edilir: $bk_j = \alpha^{k_j} mod p$ Kullanıcılardan kök düğüme k_i anahtarının hesaplanması için Denklem 2.4'deki işlem gerçekleştirilir (eğer i>1).

$$k_i = (bk_{i-1})^{r_i} mod p = (br_i)^{k_{i-1}} mod p = \alpha^{r_i k_{i-1}} mod p \tag{2.4}$$

Her üyenin grup anahtarını hesaplayabilmesi için kendi gizli anahtarını ve kardeş alt ağacının kör anahtarlarını bilmesi gerekir. Bunun için öncelikle iki üye $Kullanıc1_1$ ve $Kullanıc1_2$ grup anahtarını hesaplar.

$Kullanıc1_1$ üyesinin grup anahtarını hesaplayabilmesi için sırasıyla Denklem 2.5'deki işlemleri gerçekleştirmesi gerekir.

$$\begin{aligned} k_2 &= (br_2)^{k_1} mod p = \alpha^{r_1 r_2} mod p, bk_2 = \alpha^{k_2} mod p \\ k_3 &= (br_3)^{k_2} mod p = \alpha^{r_1 r_2 r_3} mod p, bk_3 = \alpha^{k_3} mod p \\ k_4 &= (br_4)^{k_3} mod p = \alpha^{r_1 r_2 r_3 r_4} mod p \\ k_N &= (br_N)^{k_{N-1}} mod p \end{aligned} \tag{2.5}$$

Ardından $Kullanıc1_1$ tüm kör anahtarları $(1 \leq i \leq N-1; bk_i)$ yayınlar. Bu sayede her üye k_N'i hesaplayabilir.

Her üye $Kullanıc1_i (i > 2)$ yayın mesajından kendi gizli anahtarı r_i'yi ve bk_{i-1}'i bilir. $k_i = {bk_{i-1}}^{r_i} mod p$ işlemlerini gerçekleştirerek k_N'i hesaplayabilir.

Ağaca üye ekleme ya da ağaçtan üye çıkarma işlemlerinde anahtar güncelleme adımları belirlenen bir sponsor kullanıcı tarafından gerçekleştirilir. Ekleme işleminde sponsor, mevcut gruba eklenmiş son üyedir ($Kullanıc1_N$). Bir yeni üye ($Kullanıc1_{N+1}$) kendi kör anahtarını (bk_{N+1}) içeren bir katılma istek mesajı yayınlar. Bunun üzerine sponsor mevcut grup anahtarının kör versiyonunu (bk_N) hesaplar ve yeni üyeye tüm kör anahtarları ve kör oturum anahtarlarını gönderir. Ağaçta bulunan diğer kullanıcılara yeni kullanıcının kör anahtarını gönderir. Yeni üyeye mevcut grup anahtarının kör versiyonu verildiğinden ekleme işleminde geri gizlilik sağlamaktadır.

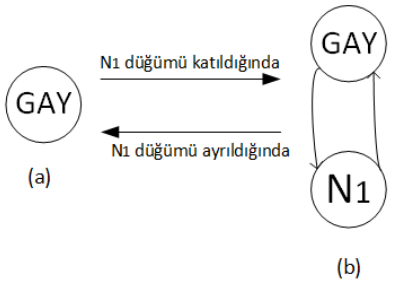
n kullanıcılı bir gruptan $Kullanıcı_d (d \leq n)$ üyesi ayrılsın. Eğer $d > 1$ ise sponsor düğüm, ayrılan kullanıcı düğümünün altında yer alan yaprak düğümdür, yani $Kullanıcı_{d-1}$. Ayrılma işleminin ardından ağaçtaki üyeler, $Kullanıcı_d$ ve ebeveyn düğümü $IN_{<d>}$ ile ilişkili olan $LN_{<d>}$ düğümünü silerek ağacı günceller. $Kullanıcı_d$'nin kardeşi $IN_{<d-1>}$, çıkan düğüm $Kullanıcı_d$'nin ebeveyni konumuna yerleşir. Sponsor düğüm bir yeni gizli anahtar belirler, kök düğüme kadarki yol üzerindeki anahtarları ve kör anahtarları hesaplayıp, kör anahtarları gruba yayınlar. Bu sayede grupta bulunan diğer üyeler güncel grup anahtarını hesaplayabilirler. Gruptan çıkartılan üye sponsorun güncellediği gizli anahtara sahip olmadığından güncel grup anahtarını hesaplayamaz. Bu sayede ileri gizlilik sağlanmış olur.

2.6. BİRLEŞİK GÜVENLİ GRUP İLETİŞİM ŞEMALARI

Birleşik grup anahtar yönetiminde grup anahtar üretimi, bir merkezi grup anahtar yönetimi gibi bir GY tarafından gerçekleştirilir. Ancak grup anahtar dağıtımı bir dağıtık grup anahtar yönetiminde olduğu gibi grupta bulunan kullanıcılar tarafından gerçekleştirilir.

2.6.1. Mantıksal Halka Tabanlı Güvenli Grup İletişim Şeması

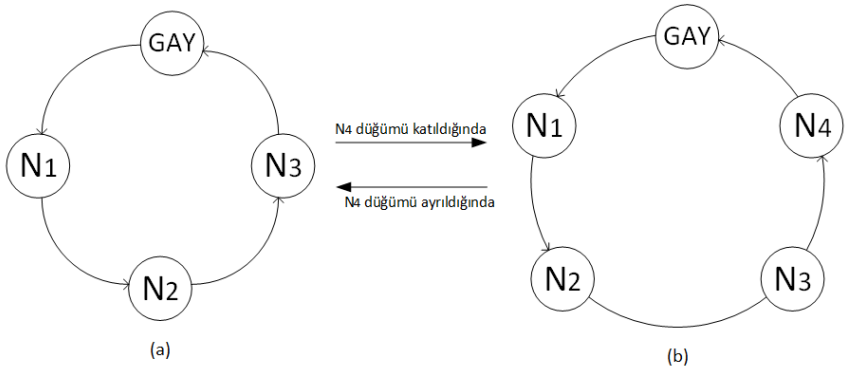
Mantıksal Halka Tabanlı Güvenli Grup İletişim Şeması (MHTGG) diğer şemaların aksine ikili ağaçtan farklıdır. Halka biçiminde sıralanmış kullanıcı düğümlerinden oluşur. Bu halka bir GY tarafından yönetilir. Mantıksal halka, bir mesajın GY'den tüm kullanıcı düğümlerine iletimi için görev dağıtımına izin verir. Mantıksal halka topolojisinde, bir mesaj gönderilen kaynağa tekrar ulaşana kadar düğümden düğüme dolaşır. Bu sayede GY, n kullanıcı için $O(n)$ mesaj göndermek yerine yalnızca $O(1)$ mesaj gönderir [65], [66].



Şekil 2.5. Mantıksal Halka İçerisinde Bir Düğümü Ekleme-Silme İşlemleri 1.

Bir mantıksal halkada başlangıçta GY'ı içeren tek bir halka bulunur (Şekil 2.5a). Gruba yeni bir kullanıcı katılmak istediğinde, halkanın kuyruğuna yeni bir düğüm eklenir (Şekil 2.5b).

GY kuyruğunda yer alan düğüme bir mesaj iletir. Her düğüm kendisine gelen mesajı kuyruğuna bağlı olan bir diğer düğüme iletir. Mesaj başlangıç düğüm olan GY'e ulaştığında mesaj iletimi tamamlanmış olur. Gruba yeni bir kullanıcı eklendiğinde ya da bir kullanıcı ayrıldığında, GY yeni bir anahtar üretir ve grupta bulunan tüm kullanıcılara dağıtır. Bu sayede yeni kullanıcı önceki mesajları çözemez. Gruptan çıkan kullanıcı ise gelecek mesajlara erişemez. Bu sayede ileri ve geri gizlilik sağlanmış olur.



Şekil 2.6. Mantıksal Halka İçerisinde Bir Düğümü Ekleme-Silme İşlemleri 2.

GY, tüm grup kullanıcı düğümlerinin adreslerini yönetir. Her düğüm yalnızca bir önceki ve bir sonraki düğümün adresini bilir. Örneğin, Şekil 2.6a'da N_2'de bir önceki düğüm olan N_1'in adresi ve bir sonraki düğüm olan N_3'ün adresi bulunur. Halkaya yeni bir düğüm eklendiğinde, GY yeni eklenen düğüme ağaca en son eklenmiş olan düğümün adresini verir. GY, ağaçtaki son kullanıcıya yeni düğümün bilgisini gönderir. Şekil 2.6b de görüldüğü üzere, N_4'ün halkaya eklenmesinin ardından, GY, N_4'e bir önceki düğüm olan N_3'ün adresini gönderir. N_4'den sonraki düğüm GY'dir. GY ayrıca N_3'e kendi bir sonraki düğüm adresini güncellemesi için bir sonraki düğüm olan N_4'ün adresini gönderir.

Şemadan bir kullanıcı ayrıldığında, ayrılan kullanıcı düğümü mantıksal halkadan silinmelidir. Bunun için GY, ayrılan düğümün üst düğümüne, ayrılan düğümün alt düğümünün adresini gönderir. Örneğin; Şekil 2.6a'da halkadan N_2 düğümü ayrılırsa, GY N_3'ü önceki düğüm adresinin N_1 olduğu ve N_1'i sonraki düğüm adresinin N_3 olduğu konusunda bilgilendirir.

2.7. FİKİR BİRLİĞİ MEKANİZMASI

Fikir birliği mekanizması ilk olarak blok zinciri teknolojisi içerisinde kullanılmıştır. Blok zinciri teknolojisi, iki kullanıcı ağ işlemlerinin üçüncü bir tarafın onaylama gereksinimini ortadan kaldıran, dağıtık uzlaşma mekanizmasına sahip, merkezi olmayan bir veri yönetim teknolojisidir [67], [68]. Yapılan tüm işlemler bir açık deftere kaydedilir. İşlemler aracı bir üçüncü kuruluş yerine ağda bulunan kullanıcıların çoğunluğunun onayı ile doğrulanır ve gerçekleştirilir. Buna fikir birliği denir. Fikir birliği mekanizması blok zinciri teknolojisinin dağıtık olmasını sağlayan ve teknolojiye olan ilgili arttıran en önemli özelliktir [69]. Fikir birliği mekanizmasına göre ağda bulunan kullanıcıların çoğunluğunun yapılan işlemin doğruluğuna onay vermesi gerekir. Eğer çoğunluk sağlanmaz ise işlem gerçekleştirilmez. Bu duruma blok zinciri üzerinde örnek vermek gerekirse, açık deftere bir diğer ifade ile zincire bir yeni blok eklenmesi gereksin. Madenciler sürekli blok üreterek, bu bloğun uygun olarak oluşturulup oluşturulmadığını kontrol ederler. Eğer bir madenci doğru bir blok oluşturduğunu ağda bulunan kullanıcılara bildirirse, kullanıcıların bu bloğun zincire eklenecek doğrulukta olmadığını kontrol etmesi gerekir. Eğer ağda bulunan kullanıcıların çoğunluğu oluşturulan bloğun doğru blok olduğunu onaylarsa blok zincirin sonuna eklenir ve madenci ödüllendirilir. Fikir birliği dağıtık bir kullanıcı grubunun çoğunluğunun onayını gerektiren bir mutabakat yaklaşımıdır.

3. TEK YÖNLÜ MELEZ ANAHTAR DAĞITIM ŞEMASI (TMAD)

Bölüm 2'de GGİ'i için geliştirilmiş şemalar merkezi, dağıtık ve birleşik GGİ şemaları olmak üzere üç kategoride sınıflandırılmıştı. TMAD şeması merkezi ve dağıtık olmak üzere iki model olarak önerildiğinden, merkezi modeli merkezi GGİ şemaları, dağıtık modeli dağıtık GGİ şemaları kategorisinde yer alır. TMAD, her üyelik değişimi sonrası gerçekleştirilen anahtar üretim ve dağıtım işlemlerinin ardından ortaya çıkan anahtar iletim sayısı ve boyutu, işlem maliyeti, kullanıcılarda bulunan anahtar sayısı ve boyutunu azaltarak, güvenlik sorunlarına yol açmadan, yayın iletişiminin etkin bir şekilde gerçekleştirilmesini hedeflemektedir. Bu hedefleri gerçekleştirmek için hem merkezi hem de dağıtık bir GGİ şemalarının çalışma esaslarından faydalanılmıştır. Ancak önerilen yöntem bu iki şema türünün dışında yeni bir yaklaşım içermektedir.

TMAD şeması ikili ağaç temellidir. Yayın Merkezi (YM) kök düğümde, kullanıcılar yapraklarda bulunmaktadır. Anahtar hesaplaması TFA ve TFZ şemalarında olduğu gibi yapraklardan kök düğüme doğrudur. Yöntemde bir açık anahtar dağıtımından ve bir gizli anahtarlı şifreleme yönteminden yararlanılır. Kullanıcıların simetrik anahtar değerlerini elde etmelerini sağlamak için Eliptik Eğri Diffie-Hellman (EEDH) protokolünden yararlanılır. Ardından kök düğüme kadarki yol üzerinde bulunan aradüğümlerin ve kök düğüm ortak gizli anahtarının hesaplanmasında özetleme fonksiyonundan yararlanılır.

Ağaca eklenen her kullanıcı, EEDH protokolünü kullanarak açık anahtarını ($pu_{kullanici_x}$) ve gizli anahtarını ($pr_{kullanici_x}$) oluşturur. Ardından simetrik anahtar değerini oluşturmak için $pr_{kullanici_x}$ anahtarını, Denklem 3.1'de görüldüğü üzere bir rasgele üreteç kullanılarak oluşturulmuş, bir rasgele veri ile tuzlayarak bir $ozet()$ fonksiyonuna sokar. $ozet()$ fonksiyonu ilk olarak içerisine aldığı verinin özetini çıkartır. Ardından özet verisini belirli bir boyuta indirger.

$$n_{kullanici_x} = ozet(pr_{kullanici_x} + rasgele_{veri}) \tag{3.1}$$

Elde edilen $n_{kullanici_x}$ kullanıcının simetrik değeridir. [70]'daki yöntemin aksine, tuzlama işleminde kullanıcının pozisyon değerinin yerine bir rasgele oluşturulmuş veri kullanılır.

Kullanıcı $pu_{kullanici_x}$, $pr_{kullanici_x}$ anahtarlarına ve $n_{kullanici_x}$ değerine sahiptir. Bu anahtarlardan $pu_{kullanici_x}$ 'i ve şema üzerindeki pozisyonuna göre $n_{kullanici_x}$ değerinin sol ya da sağ yarısını önceden tanımlı Açık Anahtar Kütüphanesi (AAK)'ne yükler.

TMAD şemasında YM'de bulunan ortak gizli anahtarı hesaplamak için, TFA şemasına benzer bir şekilde, kullanıcılardan kök düğüme doğru bir anahtar hesaplama işlemi gerçekleştirilir. Denklem 3.2'de görüldüğü üzere, her aradüğüm değeri n_x sol çocuğunun simetrik değerinin sol yarısı ile sağ çocuğunun simetrik değerinin sağ yarısının birleştirilip özetinin alınmasıyla elde edilir.

$$n_x = ozet(n_{x_{solCocukAnahtarininSolYarisi}} + n_{x_{sagCocukAnahtarininSagYarisi}}) \tag{3.2}$$

Şekil 2.2'ye benzer bir şekilde, ikili ağaç üzerinde yaprak düğümlerden kök düğüme doğru aradüğüm değerlerinin hesaplanması Denklem 3.3'deki gibidir.

$$\begin{aligned}
n_x &= ozet(n_{x_{solCocukAnahtarininSolYarisi}} + n_{x_{sagCocukAnahtarininSagYarisi}}) \\
n_{00} &= ozet(n_{000_{solYari}} + n_{001_{sagYari}}) \\
n_{01} &= ozet(n_{010_{solYari}} + n_{011_{sagYari}}) \\
n_{10} &= ozet(n_{100_{solYari}} + n_{101_{sagYari}}) \\
n_{11} &= ozet(n_{110_{solYari}} + n_{111_{sagYari}}) \\
n_0 &= ozet(n_{00_{solYari}} + n_{01_{sagYari}}) \\
n_1 &= ozet(n_{10_{solYari}} + n_{11_{sagYari}}) \\
n_{kok_{deger}} &= ozet(n_{0_{solYari}} + n_{1_{sagYari}}) \\
n_{kok_{anahtar}} &= anahtar(n_{kok_{deger}})
\end{aligned} \tag{3.3}$$

Kök düğümde bulunan $n_{kok_{deger}}$ değeri bir $anahtar()$ fonksiyonuna sokularak, ortak gizli anahtar bir diğer ifade ile simekrik anahtar elde edilir. $anahtar()$ fonksiyonu yalnızca YM'de ve kullanıcılarda bulunur. TMAD şemasında kullanıcılardan kök düğüme doğru simetrik değerlerin yalnızca yarısı iletilir. Bu durum anahtar güncelleme işlemlerinde iletilen toplam anahtar boyutunun azalmasını sağlar.

3.1. MERKEZİ SİSTEM MODELİ

Şema, bir YM, bir GY, bir dizi dinamik grup kullanıcıları, bir dizi gruba katılmak ya da gruptan ayrılmak isteyen kullanıcılar ve bir AAK'dan oluşur.

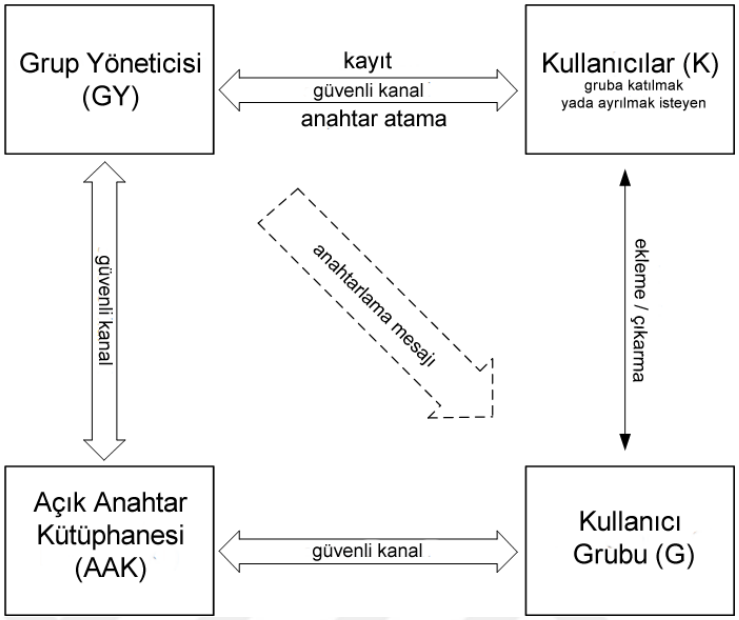
YM, yayın mesajı iletiminden sorumludur. Yayın mesajı iletimi kök düğümde bulunan ortak gizli anahtar kullanılarak gerçekleştirilir. Denklem 3.4'de görüldüğü üzere, YM mesajı şifreleyip göndermeden önce mesajın özetini çıkarır ve özetini imzalar. İmzayı mesajın sonuna ekler. Ardından $n_{kok_{anahtar}}$ ile şifreleme işlemini gerçekleştirir.

$$\begin{aligned} ozet_{veri} &= mesajOzet(mesaj_{veri}) \\ imza_{veri} &= imza(ozet_{veri})pr_{kok_{anahtar}} \\ yayin_{veri} &= mesaj_{veri} + imza_{veri} + rasgele_{veri} \\ sifreli_{veri} &= sifrele(yayin_{veri})n_{kok_{anahtar}} \end{aligned} \tag{3.4}$$

Mesajın sonuna ayrıca bir rasgele üreteç kullanılarak oluşturulmuş bir $rasgele_{veri}$ ekler. Mesaj sonuna bir $rasgele_{veri}$ eklemek, mesaj güvenliğini sağlamak için etkili bir yöntemdir. Şekil 3.1'de üyelik işlemleri gösterilmiştir. GY, kullanıcıları gruba ekleme-çıkarma işleminden ve anahtar dağıtımından sorumludur. Bir kullanıcı gruba dahil olduğunda AAK'a açık anahtarı $pu_{kullanici_x}$ 'ı ve pozisyonuna göre simetrik değerinin sol ya da sağ yarısını yükler. GY ise yeni ortak gizli anahtarının hesaplanması için diğer kullanıcılara bir güvenli kanal üzerinden yeniden anahtarlama mesajı iletir.

Bir kullanıcı gruptan ayrıldığında, GY ağaç üzerinde bulunan kullanıcı düğümünü ve AAK üzerinde bulunan kullanıcı anahtarlarını siler. Silinen kullanıcı düğümünün kardeşi konumunda bulunan düğümü ebeveyn pozisyonuna getirir. GY, gruba bir yeni kullanıcı ekleme ya da gruptan bir kullanıcı çıkarma işleminin ardından anahtar güncelleme işlemlerini gerçekleştirir. Grup kullanıcılarına ortak gizli anahtarı hesaplayabilmeleri için bir yeniden anahtarlama mesajı iletir. Anahtarlama işlemleri sırasında yayın mesajı iletimi gerçekleştirilmez. Bu sayede herhangi bir kullanıcı, anahtarlama işlemleri esnasında oluşabilecek veri kaybından etkilenmez.

Eğer bir kullanıcı anahtarlama işlemlerini kaçırırsa, ortak grup anahtarını hesaplaması için ihtiyaç duyduğu anahtarları AAK'a erişerek elde edebilir. AAK, grup kullanıcılarının gruptan ayrılmadıkları müddetçe her zaman erişebilecekleri bir kütüphanedir.



Şekil 3.1. Üyelik İşlemleri.

3.1.1. Başlangıç Adımları

GY, EEDH protokolünü kullanarak kök düğüme ait birbirleriyle ilişkili biri açık ($pu_{kok_{anahtar}}$) biri gizli ($pr_{kok_{anahtar}}$) iki anahtar oluşturur. Bu anahtarları YM'ye gönderir. Ayrıca $pu_{kok_{anahtar}}$'ı AAK'ya yükler. Başlangıç $n_{kok_{deger}}$ 'i, Denklem 3.5'de görüldüğü üzere, kök düğüm gizli anahtarı $pr_{kok_{anahtar}}$'a rasgele bir veri eklenip $ozet()$ fonksiyonuna sokulmasıyla elde edilir. Simetrik anahtar olan $n_{kok_{anahtar}}$ ise, $n_{kok_{deger}}$'in $anahtar()$ fonksiyonuna sokulmasıyla elde edilir.

$$\begin{aligned} n_{kok_{deger}} &= ozet(pr_{kok_{anahtar}} + rasgele_{veri}) \\ n_{kok_{anahtar}} &= anahtar(n_{kok_{deger}}) \end{aligned} \tag{3.5}$$

3.1.2. Kullanıcı Ekleme-Çıkarma

Bir yeniden anahtarlama işlemi gruba bir kullanıcı eklendiğinde ya da gruptan bir kullanıcı ayrıldığında gerçekleştirilir. Her işlem için, ilgili kullanıcı düğümünden kök düğüme kadar yol üzerindeki tüm simetrik değerler güncellenir. GY, yol üzerinde bulunan kullanıcılar ile

güvenli bir kanal üzerinden iletişime geçerek ihtiyaç duydukları anahtarları güvenli bir şekilde iletir. GY ve tüm kullanıcılar ortak gizli anahtarı bireysel olarak hesaplarlar.

Şekil 2.2'de 8 kullanıcılı bir ağaç mevcuttur. Örneğin $Kullanici_8$ ağaca katılmak istediğinde sırasıyla aşağıdaki işlemler gerçekleşir: Ağaç üzerinde $Kullanici_8$'in ekleneceği pozisyon belirlenir; 111. $Kullanici_8$, EEDH protokolünü kullanarak açık anahtarı $pu_{kullanici_8}$'i ve gizli anahtarı $pr_{kullanici_8}$'i üretir. Ardından $pr_{kullanici_8}$'i ile üretilen bir rasgele veriyi kullanarak Denklem 3.6'daki gibi simetrik değerini hesaplar.

$$n_{kullanici_8} = n_{111} = ozet(pr_{kullanici_8} + rasgele_{veri}) \tag{3.6}$$

$Kullanici_8$, $pu_{kullanici_8}$ anahtarını ve şema üzerindeki pozisyonu gereği simetrik değerin sağ yarısı $n_{111_{sagYari}}$ değerini AAK'a yükler. GY, $Kullanici_8$'in kardeş düğüm değerleri olan $n_{110_{solYari}}, n_{10_{solYari}}$ ve $n_{0_{solYari}}$'ı $pu_{kullanici_8}$ ile şifreler ve güvenli bir kanal üzerinden $Kullanici_8$'e gönderir. $Kullanici_8$, gizli anahtarı $pr_{kullanici_8}$'ı kullanarak kardeş düğüm anahtarının şifresini çözer ve ortak gizli anahtarı Denklem 3.7'deki işlemleri gerçekleştirerek hesaplar:

$$\begin{aligned}
n'_{11} &= ozet(n_{110_{solYari}} + n_{111_{sagYari}}) \\
n'_{1} &= ozet(n_{10_{solYari}} + n_{11'_{sagYari}}) \\
n'_{kok_{deger}} &= ozet(n_{0_{solYari}} + n_{1'_{sagYari}}) \\
n'_{kok_{anahtar}} &= anahtar(n_{kok'_{deger}})
\end{aligned} \tag{3.7}$$

GY, $Kullanici_8$'in simetrik anahtarının sağ yarısı olan $n_{111_{sagYari}}$'ı n_{11} ile şifreleyip $Kullanici_7$'e iletir. GY, $n'_{11_{sagYari}}$'ı n_1 ile şifreleyip $Kullanici_5$ ve $Kullanici_6$'a gönderir. GY, $n'_{1_{sagYari}}$'ı $n_{kok_{anahtar}}$ ile şifreleyip $Kullanici_1, Kullanici_2$, $Kullanici_3$ ve $Kullanici_4$'e gönderir. Bu sayede ağaç üzerinde bulunan tüm kullanıcılar kendilerinden kök düğüme kadar ki yol üzerinde bulunan tüm aradüğüm anahtarlarını ve ortak gizli anahtar $n_{kok_{anahtar}}$'ı hesaplayabilir.

$Kullanici_8$ ağaçtan ayrılmak istediğinde GY, AAK üzerinde bulunan ve $Kullanici_8$'e ait olan $pu_{kullanici_8}$ anahtarı ile $n_{111_{sagYari}}$ değerini siler. n_{110} düğümü n_{11} pozisyonuna yerleşir ve kullanıcı düğümlerinden kök düğüme anahtar güncelleme işlemi gerçekleştirilir. ($Kullanici_7$, bir üst düğüm olan ebeveyninin konumuna yerleşir.)

YM, $pu_{kok_{anahtar}}$, $pr_{kok_{anahtar}}$ ve $n_{kok_{anahtar}}$ anahtarlarını saklar. Bir kullanıcı ise $pu_{kullanici_x}$, $pr_{kullanici_x}$ anahtarlarını, $n_{kullanici_x}$ değerini, kendisinden kök düğüme kadar ki yol üzerinde bulunan kardeş düğüm simetrik değerleri ve $n_{kok_{anahtar}}$'ı saklar.

Tüm kullanıcıların ve kök düğümün açık anahtarları AAK'da saklanır. Ayrıca kullanıcıların pozisyonlarına göre simetrik değerlerinin sağ ya da sol kısımları da AAK'da saklanır.

Merkezi modelin kullanıcı ekleme işlemi açısından sözde kodu Şekil 3.2'de, kullanıcı çıkarma işlemi açısından sözde kodu Şekil 3.3'de verilmiştir.

yeniKullanici
GY
aradugumler
if *GY* onaylarsa *yeniKullanici* **then**
  $pu_{yeniKullanici}$
  $pr_{yeniKullanici}$
  $rasgele_{veri}$
  $n_{yeniKullanici} \leftarrow ozet(pr_{yeniKullanici} + rasgele_{veri})$
  for i = 1 'den *aradugumler*' e **do**
    $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-i)_{solCocukAnahtarininSolYarisi}} +$
    $n_{aradugumler(aradugumler.sayi-i)_{sagCocukAnahtarininSagYarisi}})$
  end for
  $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
  $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
else
  $sil(yeniKullanici)$
end if

Şekil 3.2. Sözde Kod - Merkezi Model Kullanıcı Ekleme.

3.1.3. Toplu Kullanıcı Ekleme-Çıkarma

Kullanıcıların sırasıyla eklenmesi ya da çıkartılması yerine toplu bir şekilde eklenmesi ya da çıkartılması da mümkündür. Bu durumda anahtar güncelleme işlemleri her kullanıcı ekleme ya da çıkarma işleminin ardından gerçekleştirilmez. Anahtar güncelleme işlemleri toplu kullanıcı ekleme ya da çıkarma işlemlerinin ardından gerçekleştirilir.

Bu ise anahtar iletim sayısı, anahtar iletim boyutu ve işlem maliyeti açısından daha az maliyet sağlar. Toplu kullanıcı ekleme ya da çıkarma işlemleri maliyet açısından avantajlıdır.

ayrilacakKullanici
GY
aradugumler
if *GY* onaylarsa *ayrilacakKullanici* **then**
  sil(*ayrilacakKullanici*'nın anahtarları)
  tasi(*ayrilacakKullanici*'nın kardeşini ebebeyn düğüm konumuna)
  $rasgele_{veri}$
  $n_{yeniEbeveynDugum} \leftarrow ozet(pr_{yeniEbeveynDugum} + rasgele_{veri})$
  for i = 1'den *aradugumler*'e **do**
    $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-i)_{solCocukAnahtarininSolYarisi}} +$
    $n_{aradugumler(aradugumler.sayi-i)_{sagCocukAnahtarininSagYarisi}})$
  end for
  $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
  $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
end if

Şekil 3.3. Sözde Kod - Merkezi Model Kullanıcı Çıkarma.

Kullanıcıların toplu bir şekilde eklenmesi için gruba eklenecek kullanıcıların bir süre beklemesi gerekir. Aynı şekilde gruptan ayrılacak kullanıcıların da bir süre daha grupta kalması gerekir. Bu durum gruba eklenecek kullanıcıların eklenecekleri süre zarfı boyunca yayın iletiminden faydalanamamasına, gruptan çıkacak kullanıcıların ise iletim mesajlarını almaya devam etmesine yol açar. Güvenlik zafiyetine neden olmaması için toplu işlem bekleme süresinin iyi belirlenmesi gerekir.

Merkezi modelin toplu kullanıcı ekleme işlemi açısından sözde kodu Şekil 3.4'de, toplu kullanıcı çıkarma işlemi açısından sözde kodu Şekil 3.5'de verilmiştir.

3.2. DAĞITIK SİSTEM MODELİ

Dağıtık GGİ şemalarında merkezi bir varlık olan GY bulunmaz. GY'nin görevi grup üyeleri arasında iş birliği yapılarak paylaşılır. Bu iş birliği literatürdeki şemalarda genellikle bir sponsor düğümün ya da iş birliği grubunun belirlenmesi şeklindedir. TMAD şemasını dağıtık olarak tasarladığımızda GY'nin görevi blok zinciri teknolojisinin fikir birliği mekanizması kullanılarak gerçekleştirilir. Şema üzerindeki işlemler, şemada bulunan mevcut üyelerin çoğunluğunun fikir birliği ile doğrulanır ve onaylanır. Bu durum merkezi bir otoritesinin rolünü ortadan kaldırır. TMAD şemasının dağıtık versiyonunda, GY'nin görevini şemada bulunan kullanıcıların %51'inin fikir birliği ile gerçekleştirilmiştir.

yeniKullanicilar
GY
aradugumler
if *GY* onaylarsa *yeniKullanicilar* **then**
  for i = 1'den *yeniKullanicilar*'a **do**
    $pu_{yeniKullanicilar(i)}$
    $pr_{yeniKullanicilar(i)}$
    $rasgele_{veri}$
    $n_{yeniKullanicilar(i)} \leftarrow ozet(pr_{yeniKullanicilar(i)} + rasgele_{veri})$
  end for
  for j = 1'den *aradugumler*'e **do**
    $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-i)_{solCocukAnahtarininSolYarisi}} +$
    $n_{aradugumler(aradugumler.sayi-i)_{sagCocukAnahtarininSagYarisi}})$
  end for
  $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
  $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
else
  $sil(yeniKullanicilar)$
end if

Şekil 3.4. Sözde Kod - Merkezi Model Toplu Kullanıcı Ekleme.

Dağıtık modelde, şema içerisinde bulunan kullanıcı sayısı arttıkça, yayın iletiminden yararlanan kullanıcılar şemayı kullanma sıklığına göre aktif ve pasif olmak üzere ikiye ayrılır. Aktif kullanıcıların yayın iletiminden yararlanmaları pasif kullanıcılara göre çok daha fazladır. Bu durum yayın iletimi için bir soruna neden olmasa da özellikle kullanıcı ekleme-çıkarma işlemlerinin fikir birliği ile sağlanması açısından bir sorun oluşturabilir. Yayın iletimine erişim sağlamayan kullanıcıların şemaya eklenen yeni kullanıcılar ya da şemadan ayrılan mevcut kullanıcılar için onay vermesi uygun değildir. Ayrıca fazla sayıda pasif kullanıcının bulunduğu gruplarda kullanıcı ekleme-çıkarma işlemlerinin tamamlanması uzun süreler alabilir.

TMAD şemasının dağıtık versiyonunda fikir birliği işleminin başarıyla uygulanabilmesi için şemada bulunan mevcut kullanıcılar arasında aktif kullanıcılar için bir ayrım oluşturulmuş, GY işlemini yalnızca şemada aktif olarak bulunan kullanıcılar üstlenmiştir. Şemada bulunan kullanıcıları yayın iletiminden yararlanma noktasında aktif ve pasif olmak üzere ikiye ayırabilmek için, her kullanıcının yayın iletiminden ne ölçüde yararlandığını belirleyen bir puan değeri tutulmuştur. Bu değer kullanıcıların yanı sıra yayın merkezi için de kullanılmıştır. Yayın merkezi için tutulan puan değeri iletimini yaptığı her veri

için bir arttırılmıştır. Kullanıcılar için tutulan puan değeri ise eriştikleri her mesaj için bir arttırılmıştır. Belirli bir ölçüde puan değeri, yayın merkezinin puan değerine yakın olan kullanıcılar aktif kullanıcılar olarak belirlenmiştir. Bu sayede GY işlemleri aktif kullanıcıların fikir birliği ile gerçekleştirilmiştir.

$ayrilacakKullanicilar$
GY
$aradugumler$
if GY onaylarsa $ayrilacakKullanicilar$ **then**
  for i = 1'den $ayrilacakKullanicilar$'a **do**
    $sil(ayrilacakKullanicilar(i)$'nin anahtarları$)$
    $tasi(ayrilacakKullanicilar(i)$'nin kardeşini ebebeyn düğüm konumuna$)$
    $rasgele_{veri}$
    $n_{yeniEbeveynDugum(i)} \leftarrow ozet(pr_{yeniEbeveynDugum(i)} + rasgele_{veri})$
  end for
  for i = 1'den $aradugumler$'e **do**
    $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-i)_{solCocukAnahtarininSolYarisi}} +$
    $n_{aradugumler(aradugumler.sayi-i)_{sagCocukAnahtarininSagYarisi}})$
  end for
  $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
  $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
end if

Şekil 3.5. Sözde Kod - Merkezi Model Toplu Kullanıcı Çıkarma.

Dağıtık modelin kullanıcı ekleme-çıkarma işlemlerinin pseudo kodu aşağıdaki gibidir. Kullanıcı ekleme-çıkarma işlemleri dağıtık bir kullanıcı grubunun onayı ile sağlandığından dağıtık modelde toplu kullanıcı ekleme-çıkarma işlemleri bulunmaz.

Dağıtık modelin kullanıcı ekleme işlemi açısından sözde kodu Şekil 3.6'de, kullanıcı çıkarma işlemi açısından sözde kodu Şekil 3.7'da verilmiştir.

3.3. GÜVENLİK DEĞERLENDİRMESİ

Diğer merkezi GGİ şemalarında olduğu gibi, TMAD şemasının merkezi modelinde de GY, anahtar dağıtımı için güvenilir bir varlık olarak kabul edilir. Dağıtık GGİ şemalarında ise GY'nin görevi geçici bir süre ile şema içerisinde yer alan bir kullanıcıya ya da kullanıcı grubuna verilir. Bu görevi üstlenen kullanıcı ya da kullanıcı grubu da merkezi GGİ şemalarında olduğu gibi güvenilir bir varlık olarak kabul edilir.

Kullanıcı ekleme işlemi sırasında, yeni eklenen kullanıcıya veri gizliliği ve güvenliğinin sağlandığı güvenli bir kanal üzerinden ihtiyaç duyduğu anahtarlar iletilir. Kullanıcı, anahtarlarının bazılarını lokal olarak kendisinde saklar. Şemada bulunan iki varlık

$yeniKullanici$
$yayin_{sayisi}$
$aradugumler$
$kullanicilar$
$esik_{deger} \leftarrow 0.1$
$aktif_{kullanici_{kumesi}} \leftarrow aktif_{kullanicilari_{bul}}(kullanicilar, yayin_{sayisi}, esik_{deger})$
$aktif_{kullanici_{sayisi}} \leftarrow sayi(aktif_{kullanici_{kumesi}})$
$aktif_{kullanici_{kumesi}}$'ne $yeniKullanici$'nın bilgilerini gönder.
$i \leftarrow 0$
$onaylayan_{kullanici_{sayisi}} \leftarrow 0$
while $aktif_{kullanici_{kumesi}}(i)$ onaylarsa $yeniKullanici$ **do**
  $onaylayan_{kullanici_{sayisi}} \leftarrow +1$
  if $onaylayan_{kullanici_{sayisi}} > aktif_{kullanici_{sayisi}}/2$ **then**
    $pu_{yeniKullanici}$
    $pr_{yeniKullanici}$
    $rasgele_{veri}$
    $n_{yeniKullanici} \leftarrow ozet(pr_{yeniKullanici} + rasgele_{veri})$
    for j = 1 'den $aradugumler$'e **do**
      $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-j)_{solCocukAnahtarininSolYarisi}} +$
      $n_{aradugumler(aradugumler.sayi-j)_{sagCocukAnahtarininSagYarisi}})$
    end for
    $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
    $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
  end if
  if $i == aktif_{kullanici_{sayisi}}$ **then**
    $sil(yeniKullanici)$
    $break$
  end if
end while

Şekil 3.6. Sözde Kod - Dağıtık Model Kullanıcı Ekleme.

arasındaki tüm iletişim kanallarının, gönderici kimliğinin ve mesaj bütünlüğünün doğrulandığı varsayılır.

Bu bölümde TMAD, Bölüm 2.2'de tanımlanan düşman modellerine göre incelenmiştir.

Teorem TMAD şeması tip-I düşman modeline karşı güçlüdür.

Kanıt. λ'nın TMAD şeması üzerinde tip-I düşman modeli ile başarı sağladığını ve $n_{kok_{anahtar}}$'ı elde ettiğini varsayalım.

λ'nın bir mesajı $n_{kok_{anahtar}}$ ile şifrelemeden önce $pr_{kok_{anahtar}}$ ile imzalaması gerekir. Ancak $pr_{kok_{anahtar}}$ AAK içerisinde bulunmaz. $pr_{kok_{anahtar}}$ YM'nin gizli anahtarıdır ve yalnızca YM içerisinde gizli olarak tutulur.

$ayrilacakKullanici$
$yayin_{sayisi}$
$aradugumler$
$kullanicilar$
$esik_{deger} \leftarrow 0.1$
$aktif_{kullanici_{kumesi}} \leftarrow aktif_{kullanicilari_{bul}}(kullanicilar, yayin_{sayisi}, esik_{deger})$
$aktif_{kullanici_{sayisi}} \leftarrow sayi(aktif_{kullanici_{kumesi}})$
$aktif_{kullanici_{kumesi}}$'ne $ayrilacakKullanici$'nın bilgilerini gönder.
$i \leftarrow 0$
$onaylayan_{kullanici_{sayisi}} \leftarrow 0$
while $aktif_{kullanici_{kumesi}}(i)$ onaylarsa $ayrilacakKullanici$ **do**
  $onaylayan_{kullanici_{sayisi}} \leftarrow +1$
  if $onaylayan_{kullanici_{sayisi}} > aktif_{kullanici_{sayisi}}/2$ **then**
    $sil(ayrilacakKullanici$'nın anahtarları)
    $tasi(ayrilacakKullanici$'nın kardeşini ebebeyn düğüm konumuna)
    $rasgele_{veri}$
    $n_{yeniEbeveynDugum} \leftarrow ozet(pr_{yeniEbeveynDugum} + rasgele_{veri})$
    for j = 1 'den $aradugumler$ 'e **do**
      $n_x \leftarrow ozet(n_{aradugumler(aradugumler.sayi-j)_{solCocukAnahtarininSolYarisi}} +$
      $n_{aradugumler(aradugumler.sayi-j)_{sagCocukAnahtarininSagYarisi}})$
    end for
    $n_{kokdeger} \leftarrow ozet(n_{0_{solYari}} + n_{1_{sagYari}})$
    $n_{kok\_anahtar} \leftarrow anahtar(n_{kokdeger})$
  end if
  $i \leftarrow +1$
  if $i == aktif_{kullanici_{sayisi}}$ **then**
    $break$
  end if
end while

Şekil 3.7. Sözde Kod - Dağıtık Model Kullanıcı Çıkarma.

Eğer λ mesajı YM'ye ait olmayan farklı bir anahtar kullanarak imzalayıp kullanıcılara gönderirse, kullanıcılar $pu_{kok_{anahtar}}$'ı kullanarak mesajın YM'den gelmediğini ispat edebilirler.

$pr_{kok_{anahtar}}$ bir mesajı imzalamak, $pu_{kok_{anahtar}}$ ise bir imzayı doğrulamak için kullanılır. Denklem 3.8'de görüldüğü üzere, YM kullanıcılara bir mesaj göndermek istediğinde, öncelikle bir $mesajOzet()$ fonksiyonu kullanarak mesajın bir özetini çıkartır.

$$ozet_{veri} = mesajOzet(mesaj_{veri}) \tag{3.8}$$

Ardından bu mesajı $pr_{kok_{anahtar}}$ ile imzalar. Denklem 3.9'da görüldüğü üzere, imza verisini mesajın sonuna ekler ve şifreleme işleminin ardından mesajı tüm kullanıcılara gönderir.

$$\begin{aligned} imza_{veri} &= imza(ozet_{veri})pr_{kok_{anahtar}} \\ yayin_{veri} &= mesaj_{veri} + imza_{veri} + rasgele_{veri} \\ sifreli_{veri} &= sifrele(yayin_{veri})n_{kok_{anahtar}} \end{aligned} \tag{3.9}$$

Denklem 3.10'da görüldüğü üzere, kullanıcılar şifreli veriyi $n_{kok_{anahtar}}$ ile çözdükten sonra imza verisi ile rasgele veriyi mesajdan ayırırlar ve imzayı AAK üzerinden eriştikleri $pu_{kok_{anahtar}}$ ile çözerek özet değerini elde ederler. Ardından mesajın da özetini çıkartırlar ve iki bu iki özet değerini karşılaştırırlar.

$$\begin{aligned} yayin_{veri} &= sifreCoz(sifreli_{veri})n_{kok_{anahtar}} \\ yayin_{veri}(mesaj_{veri} + imza_{veri} + rasgele_{veri}) \\ ilk_{ozet_{veri}} &= mesajOzet(mesaj_{veri}) \\ ikinci_{ozet_{veri}} &= imzaCoz(imza_{veri})pu_{kok_{anahtar}} \end{aligned} \tag{3.10}$$

Eğer iki özet değeri birbirine eşitse ($ilk_{ozet_{veri}} == ikinci_{ozet_{veri}}$), mesaj yayın merkezi tarafından gönderilmiştir. Eğer özetler birbirine eşit değilse, şema üzerinde bir saldırı olduğu anlaşılır.

Teorem TMAD şeması tip-II düşman modeline karşı güçlüdür.

Kanıt. λ'nın TMAD şeması üzerinde tip-II düşman modeli ile başarı sağladığını ve şemayı gizli bir şekilde dinlediğini varsayalım.

n kullanıcı sayısı olmak üzere λ'ın YM'den $G(Kullanici_1..Kullanici_{n)}$ kümesine iletilen bir mesajı dinlediğini ve $mesaj_{veri}$'ı elde ettiğini varsayalım. λ'ın $mesaj_{veri}$'dan anlamlı bir mesaj elde edebilmesi için güncel $n_{kok_{anahtar}}$'a ihtiyacı vardır. λ'ın $n_{kok_{anahtar}}$'ı elde edebilmesi için anahtar güncellemeleri sırasında $GY-K$ veya $GY-G$ iletişim kanallarından birini dinlemelidir. Fakat $GY-K$ ve $GY-G$ üzerindeki anahtar iletimi açık değildir. GY'den ya da GY görevini üstlenmiş üye/ler'den şemadaki diğer kullanıcılara iletilen anahtar mesajları kullanıcıların şemadaki pozisyon değerine göre belirlenen ebeveyn aradüğüm anahtarıyla şifrelenerek iletilir.

Teorem TMAD şeması tip-III düşman modeline karşı güçlüdür.

Kanıt. Tip-III düşman modeli anahtar hesaplamalarının kullanıcılardan kök düğüme doğru olduğu, anahtarların birbirlerinin arasında matematiksel bir bağlantının olduğu şemalarda etkilidir. Her üyelik değişiminin ardından ileri ve geri gizlilik sağlanmalıdır. Bu nedenle gruba bir kullanıcı eklendiğinde ya da gruptan bir kullanıcı ayrıldığında kullanıcıdan kök düğüme kadarki yol üzerinde bulunan tüm düğüm anahtarları güncellenmelidir. Tip-III düşman modelinin başarıya ulaşması için ağaçtan ayrılan kullanıcı ile ağaca eklenen kullanıcı kök düğüme göre farklı altkümelerde yer almalıdır.

Şemadan t_1 zamanında bir üyenin ayrıldığını, t_2 zamanında ise bir üyenin eklendiğini, t_1 ve t_2 zaman aralığında ise başka bir kullanıcı işleminin gerçekleşmediğini varsayalım.

Şekil 2.2'de 000 pozisyonundaki $Kullanici_1$, t_1 zamanında şemadan ayrılsın. Bu durumda $Kullanici_1$'in 001 pozisyonundaki kardeş düğümü ebeveyn düğüm pozisyonuna geçer ve düğümden kök düğüme tüm anahtarlar (0 aradüğümü ve kök anahtar) güncellenir. 1 numaralı aradüğüm güncellenen düğümler arasında yer almaz. Ayrılan kullanıcı $Kullanici_1$'de ise kendisinden kök düğüme kadar ki yol üzerinde içerisinde 1 numaralı aradüğümünde yer aldığı tüm kardeş düğüm anahtarları bulunur. Kısaca $Kullanici_1$, t_1 zamanında şemadan ayrılır fakat kendisinde 1 numaralı güncel düğüm anahtarı hala bulunmaktadır. 111 pozisyonundaki $Kullanici_8$'in ise t_2 zamanında şemaya yeni eklendiğini varsayalım. Bu durumda t_2 zamanında eklenen kullanıcıdan kök düğüme kadar ki yol üzerindeki aradüğüm anahtarları (11 ve 1 aradüğüm anahtarları) ile kök anahtar güncellenir. Ardından $Kullanici_8$'e kardeş düğüm anahtarları olan 110 düğümü ile 10 ve 0 aradüğümlerinin anahtarları verilir. $Kullanici_1$ şemadan ayrılmış olsa da kendisinde şemadan ayrıldığı t_1 zamanından, $Kullanici_8$'in şemaya eklendiği t_2 zamanında kadar 1 numaralı aradüğümün güncel anahtarı bulunurken, $Kullanici_8$'de ise 0 numaralı düğümün güncel anahtarı bulunur. Eğer $Kullanici_1$ ve $Kullanici_8$ aralarında gizlice anlaşırsa $t_1 - t_2$ zaman aralığında güncel kök anahtarı olan ortak gizli anahtarı hesaplayabilirler [27]. [27]'de gizli anlaşma atağının engellenmesi için, $Kullanici_8$ t_2 zamanında şemaya eklendiğinde, tüm kardeş düğüm anahtarları güncellenir. Fakat bu durum ek işlem maliyetine neden olur. Önerilen TMAD şemasında bir kullanıcı şemadan ayrıldığında pozisyon değeri geçici olarak tutulur. Eğer çıkan bir kullanıcının ardından şemaya yeni bir

kullanıcı eklenirse kendisine geçici tutulan pozisyon değeri verilir. Bu işlem ile şemadan ayrılan ve şemaya eklenen kullanıcılar aynı alt küme içerisinde yer alır. Bu durum gizli anlaşma atağının oluşmasını engeller.

Teorem TMAD şeması tip-IV düşman modeline karşı güçlüdür.

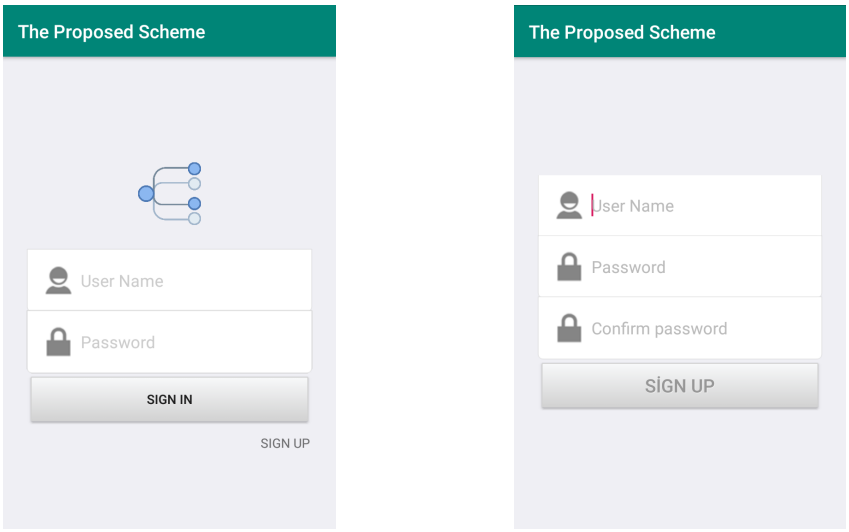
Kanıt. Tip-IV düşman modeli genellikle iki taraf arasında, bir taraftan diğerine iletilen veri üçüncü kötü niyetli bir kişi tarafından dinlenerek elde edilmesi ve alıcıya tekrar gönderilmesi şeklindedir.Tip-IV düşman modelini engellemenin iki yöntemi vardır.

Birincisi her yayın iletimi tek seferlik şifreler kullanmaktır. Bu sayede YM kullanıcılara yeni bir mesaj gönderse de gönderilen mesaj yeni oluşturulan tek seferlik şifre ile şifreleneceğinden, salgırganın eski mesajları kullanıcılara yeniden iletmesi etkisiz olacaktır. Ancak bu yöntem yayın şifreleme şemalarında üyelik değişimlerinin ardından yapılan anahtar güncelleme işleminin ayrıca her mesaj iletiminin ardından da yapılması gerekliliğini doğrurur. Bu durum şema üzerinde büyük ölçüde işlem maliyetine yol açar.

Tip-IV düşman modelini engelleyecek ikinci yöntem ise YM'den kullanıcılara iletilecek mesajların sonuna bir zaman damgası eklenmesidir. Bu sayede kullanıcılar kendilerine gelen mesaj belirli bir zaman aralığındaysa mesajı kabul edeceklerdir. Saldırgan mesajı belirli bir zaman sonra kullanıcılara tekrar gönderdiğinde kullanıcılar zaman damgasını kontrol edecek ve mesajı kabul etmeyeceklerdir.

TMAD şemasında Tip-IV düşman modelini engellemek için zaman damgasına benzer olarak bir rasgele veri kullanılır. YM mesajı kullanıcılara göndermek için şifrelemeden önce mesajın sonuna bir rasgele veri ekler. Rasgele veri mesaj ile şifrelenerek kullanıcılara gönderilir. YM ayrıca rasgele veriyi bulut sunucu üzerinde yer alan veritabanında bir alana kaydeder. Kullanıcılar kendilerine gelen mesajın şifresini çözdükten sonra mesajın sonunda yer alan rasgele veriyi, sunucu üzerinde yer alan rasgele veri alanı ile karşılaştırır. Eğer iki değer birbirine eşit değilse kullanıcılar kendilerine gelen mesajın YM tarafından gönderilen doğru bir mesaj olmadığını tespit ederler.

Teorem TMAD şeması tip-V düşman modeline karşı zayıftır.



Şekil 3.8. Kullanıcı Giriş ve Kayıt Sayfaları.

Kanıt. Literatürdeki diğer şemalarda olduğu gibi TMAD şeması da tip-V düşman modeline karşı zayıftır. Tip-V düşman modeli kolay bir şekilde tespit edilmez. Bu modeli engellemek merkezi modelde GY'nin, dağıtık modelde ise GY'nin görevini üstlenen kullanıcıların görevidir. Şema içerisinde bir saldırgan ile uzlaşan kullanıcı varsa hemen tespit edilmeli ve ilgili kullanıcı şemadan çıkartıp anahtar güncelleme işlemi gerçekleştirilmelidir.

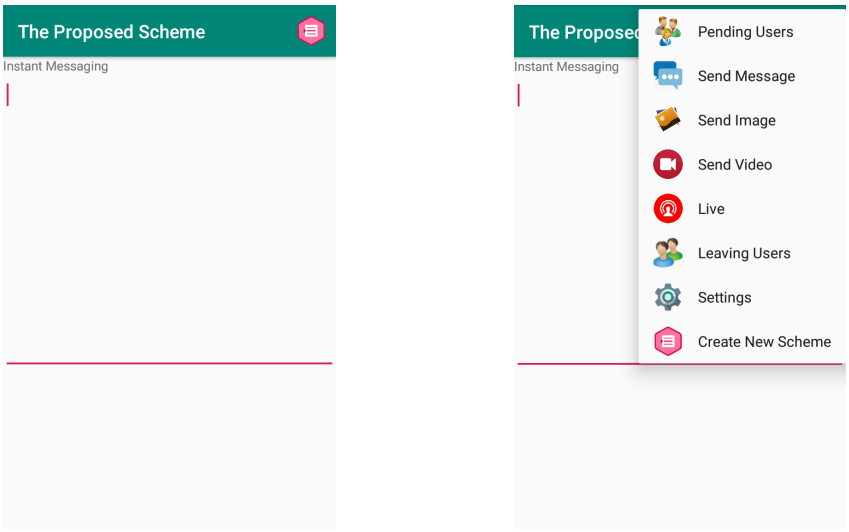
3.4. MOBİL UYGULAMA

Bu kısımda Android Studio kullanılarak biri yayın merkezi, diğeri kullanıcı uygulaması olmak üzere iki farklı uygulama geliştirilmiştir. Uygulamalar TMAD şemasının merkezi modeline göre oluşturulmuştur. Dağıtık model için uygulama üzerinde yapılan değişiklikler ayrıca belirtilmiştir.

3.4.1. Yayın Merkezi Uygulaması

Yayın merkezi uygulamasında on sayfa bulunur. Bu sayfalar sırasıyla "Sign In", "Sign Up", "Pending Users", "Send Message", "Send Image", "Send Video", "Live", "Leaving Users", "Settings" ve "Create New Scheme" 'dir.

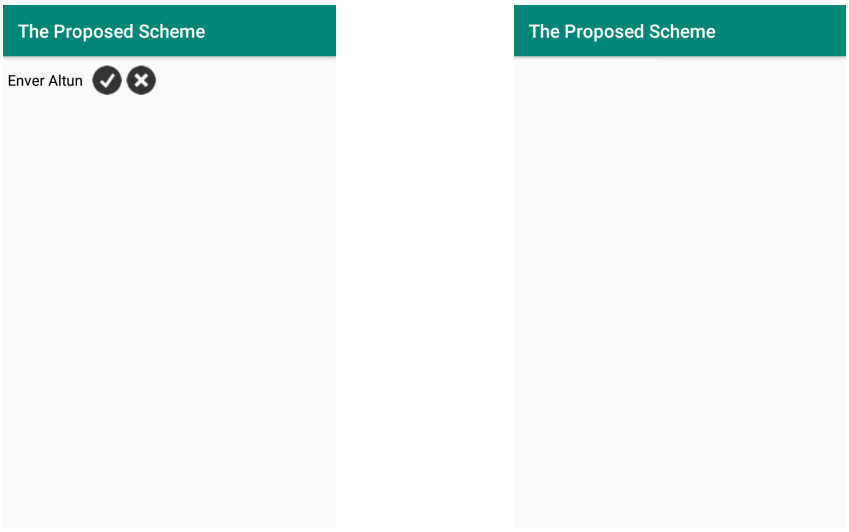
Şekil 3.8'de görüldüğü üzere, yayın merkezi "Sign In" sayfasını kullanarak kullanıcı adı ve şifresi ile giriş yapar. Eğer daha önce bir hesap oluşturmadıysa "Sign Up" sayfasını kullanarak kayıt işlemini gerçekleştirir.



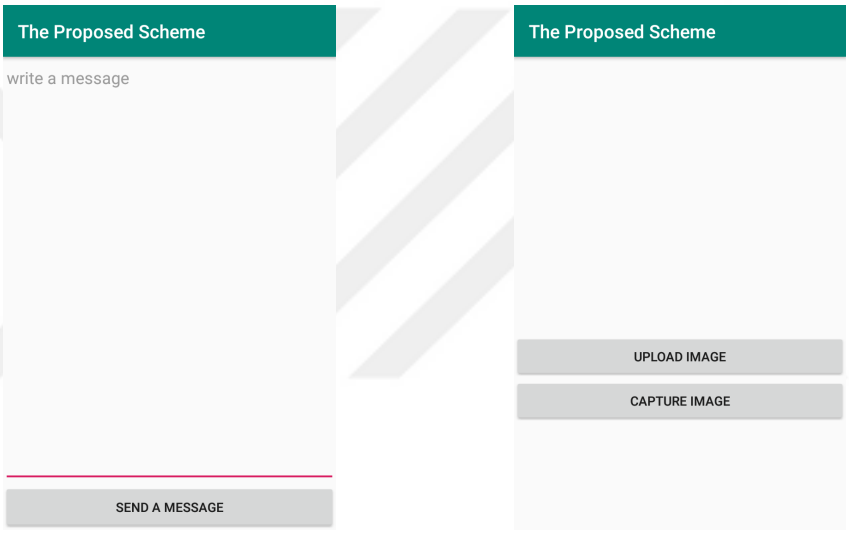
Şekil 3.9. Anasayfa ve Menüler.

Başarılı bir girişin ardından Şekil 3.9'da görüldüğü üzere, anasayfaya ulaşılır. Sağ üst köşede menü butonuna tıklayarak menü seçenekleri listelenir. Yayın merkezi "Create New Scheme" sayfasını kullanarak yeni bir ağaç şeması oluşturabilir. Bu durumda eski ağaç şeması ve kullanıcılar silinecektir.

Şekil 3.10'da görüldüğü üzere, yayın merkezi "Pending Users" sayfası girerek şemaya katılmak isteyen yeni bir kullanıcı olup olmadığını görebilir. Yayın merkezi onay vermeden yeni bir kullanıcı şemaya katılamaz. Onay işleminin ardından yeni kullanıcının ağaçta ekleneceği konumu belirlenir. Kullanıcıdan yayın merkezine kadar ki yol üzerinde bulunan anahtarlar güncellenir. Mevcut kullanıcılara güncel anahtar değerleri iletilir. Yeni kullanıcı eklendikten sonraki mesajları görebilir, eklenmeden önceki mesajlara erişmesi mümkün değildir. "Leaving Users" sayfasında yayın merkezi şemadan ayrılmak isteyen kullanıcı olup olmadığını görebilir. Yayın merkezi onay vermeden herhangi bir kullanıcı şemadan ayrılamaz. Gruptan ayrılma talebinde bulunan kullanıcının yayın merkezi ayrılmasını onayladıktan sonra yeni mesajlara ulaşması mümkün değildir. Gruptan ayrılmadan önceki mesajlara erişebilir. Şemadan ayrılan kullanıcının pozisyon değeri boşa çıkartılır. Şekil 3.11'de görüldüğü üzere, "Send Message" sayfasında yayın merkezi şemada bulunan kullanıcılara metin mesajları gönderebilir. Mesaj iletimi "Settings" sayfasında belirlenen şifreleme algoritmaları kullanılarak şifreli bir şekilde gerçekleştirilir. Yayın merkezi şemada bulunan kullanıcılara "Send Image" sayfasını kullanarak resim gönderebilir. Resim şifreli olarak gönderilir. Veritabanında şifreli olarak tutulur.



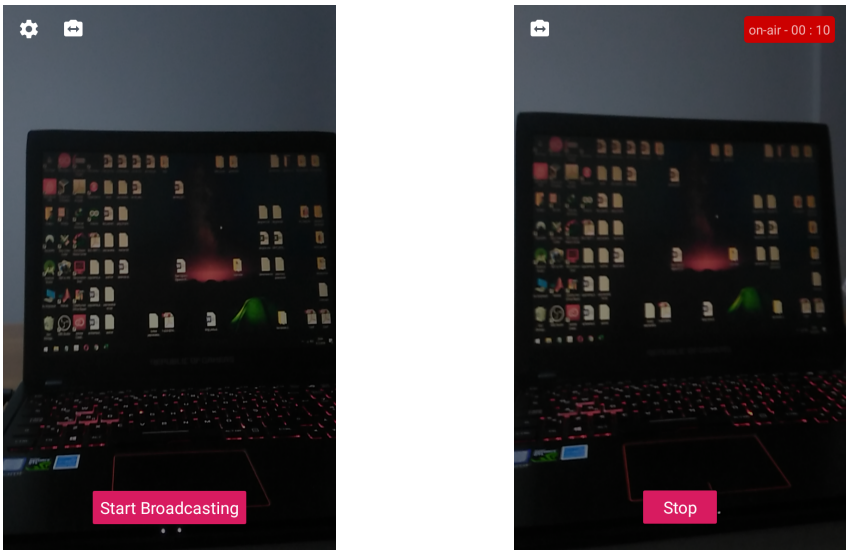
Şekil 3.10. Şemaya Katılmak ve Ayrılmak İçin Bekleyen Kullanıcılar.



Şekil 3.11. Mesaj ve Resim Gönderme Sayfaları.

Şekil 3.12'de görüldüğü üzere, yayın merkezi "Live" sayfasını kullanarak canlı yayın başlatabilir. Canlı yayın linki her yayın için rasgele oluşturulur. Yayın linki kullanıcılara şifreli olarak gönderilir. Şekil 3.13'de görüldüğü üzere, yayın merkezi şemada bulunan kullanıcılara "Send Video" sayfasını kullanarak video gönderebilir. Video şifreli olarak gönderilir. Veritabanında şifreli olarak tutulur.

Şekil 3.13 ve Şekil 3.14'de görüldüğü üzere, "Settings" sayfasını kullanarak uygulama içerisinde hangi gizli ve açık anahtarlı şifreleme yönteminin kullanılacağı, anahtar boyutunun ne olacağı belirlenir. Ayrıca mesaj iletiminin imzalı olarak yapılıp yapılmayacağı belirlenir.



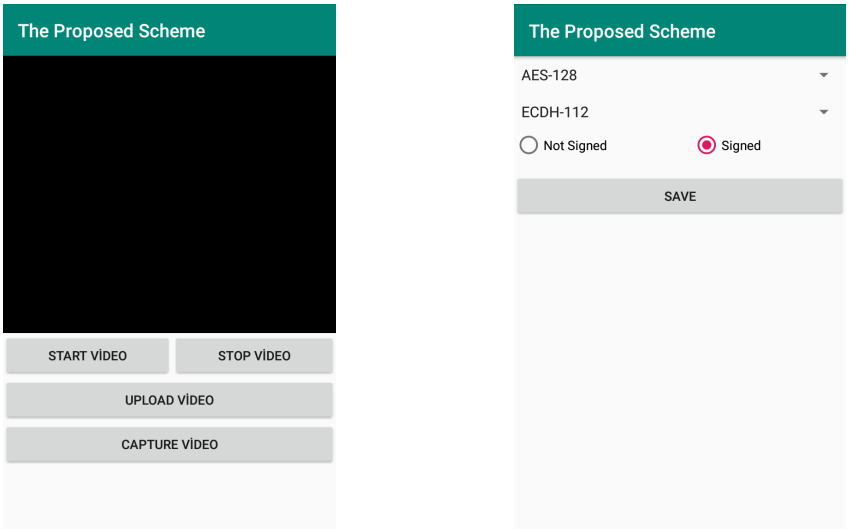
Şekil 3.12. Canlı Yayın Sayfaları.

Şekil 3.15'de görüldüğü üzere, yayın merkezi kullanıcılara metin, resim video ya da canlı yayın türünde bir veri gönderdiğinde kullanıcıların mobil cihazlarına bildirim düşer. Bu sayede kullanıcılar yayın merkezinin kendileriyle haberleşmeye geçtiğini anlayabilir. Eğer bir kullanıcı mobil cihazının internet bağlantısı mevcut değil ise canlı yayın haricindeki diğer veri türlerine dilediği zaman erişebilmesi mümkündür.

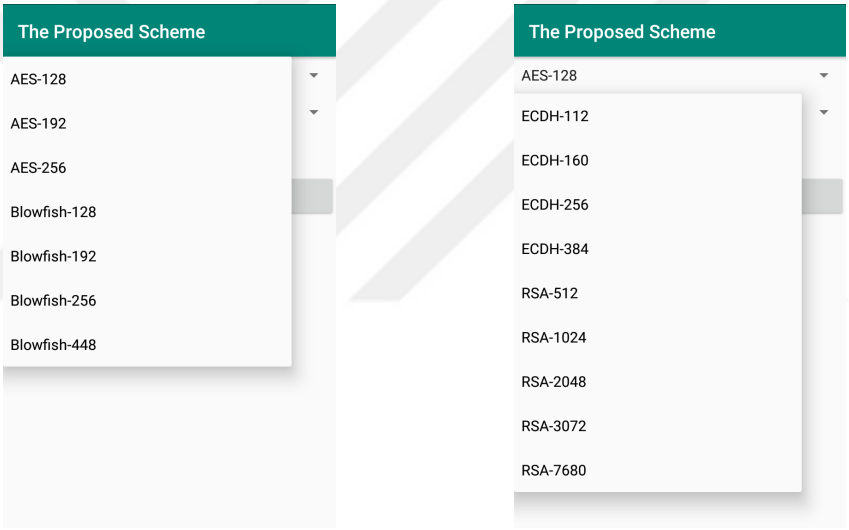
3.4.2. Kullanıcı Uygulaması

Kullanıcı uygulamasında yedi sayfa bulunur. Bu sayfalar sırasıyla "Sign In", "Sign Up", "Read Message", "See Image", "Watch Video", "Live" ve "Leave the Group" 'dur. Şekil 3.16'da görüldüğü üzere, kullanıcı "Sign In" sayfasını kullanarak kullanıcı adı ve şifresi ile giriş yapar. Eğer daha önce bir hesap oluşturmadıysa, "Sign Up" sayfasını kullanarak kayıt işlemini gerçekleştirir. Eğer kullanıcının şemaya katılımı onaylanırsa kullanıcının açık anahtarı, gizli anahtarları ve simetrik anahtar değeri oluşturulur.

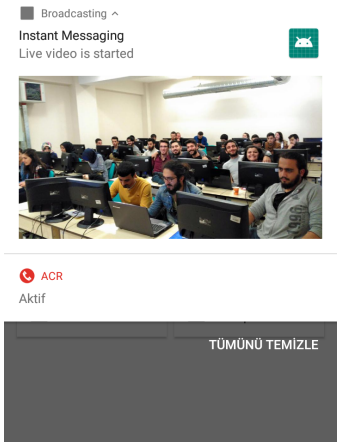
Başarılı bir girişin ardından Şekil 3.17 'de görüldüğü üzere anasayfaya ulaşılır. Kullanıcı, sağ üst köşede menü butonuna tıklayarak menü seçenekleri listelenir. Şekil 3.18'de görüldüğü üzere, "Read Message" sayfasını kullanarak yayın merkezinden iletilen metin mesajlarını okuyabilir. Şekil 3.18 ile Şekil 3.19'da görüldüğü üzere,"See Image" sayfasını kullanarak yayın merkezinden iletilen resimleri görebilir. Şekil 3.19 ile Şekil 3.20'de görüldüğü üzere, "Watch Video" sayfasını kullanarak yayın merkezinden iletilen videoları izleyebilir.



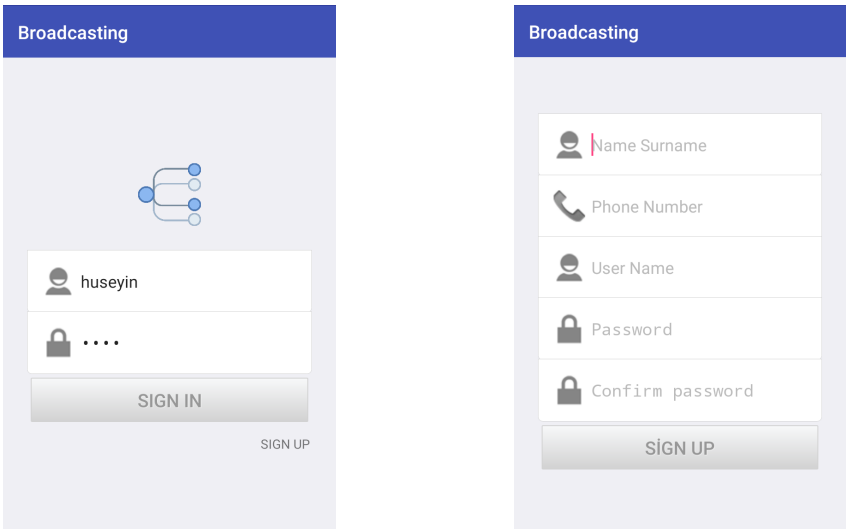
Şekil 3.13. Video Gönderme ve Ayarlar Sayfaları.



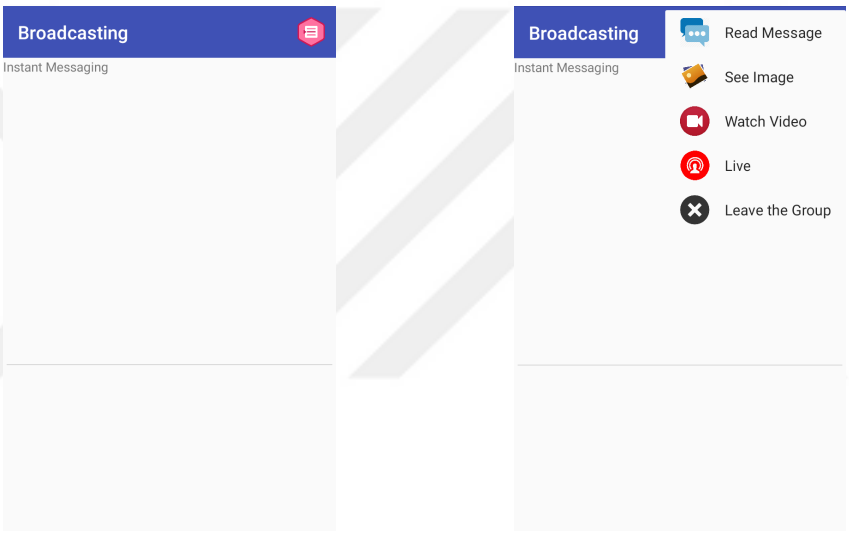
Şekil 3.14. Açık ve Gizli Anahtar Seçimleri.



Şekil 3.15. Canlı Yayın Bildirim.

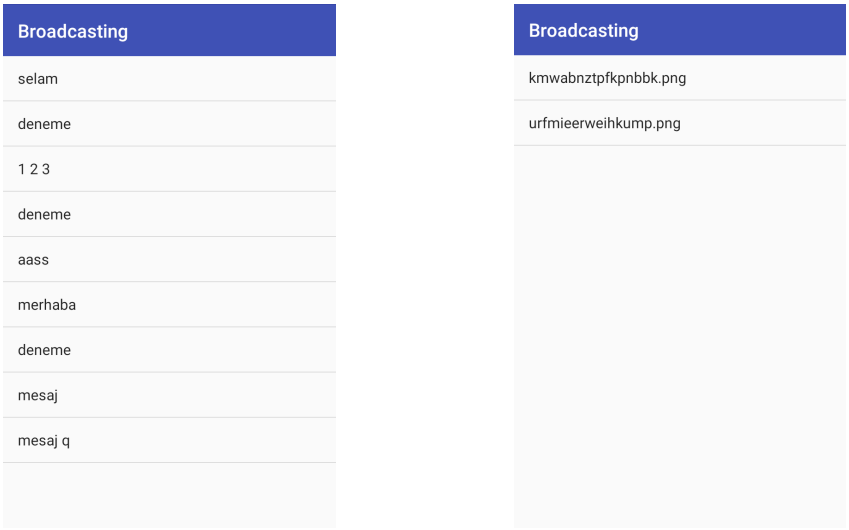


Şekil 3.16. Kullanıcı Uygulaması Giriş ve Kayıt Sayfaları.

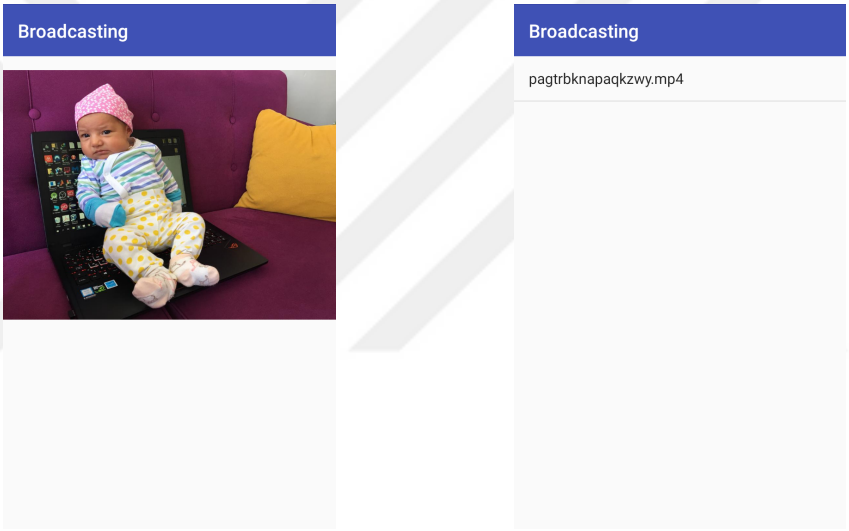


Şekil 3.17. Kullanıcı Uygulaması Anasayfa ve Menüler.

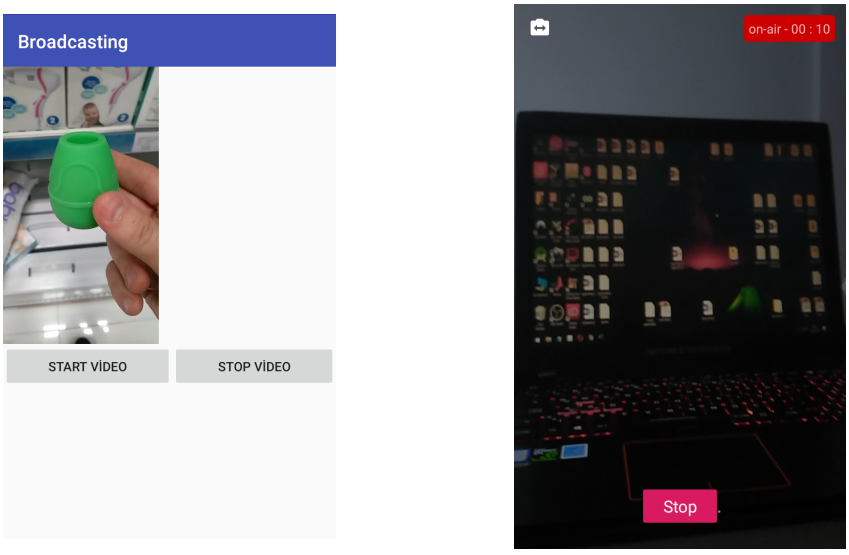
Kullanıcı gönderilen resim ve videolara erişmek istediğinde öncelikle veriler şifreli olarak uygulama tarafından indirilir. Şifre çözme işleminin ardından verinin açık hali de ayrıca kaydedilir. Bu sayede kullanıcı aynı resim veya videoya tekrar erişmek istediğinde uygulama yeniden şifre çözme işlemi gerçekleştirmez. Resim ve video verilerinin şifreli ve açık halleri uygulamanın kullanıldığı mobil cihazda bir dosya içerisinde tutulur. Şekil 3.20'de görüldüğü üzere, kullanıcı "Live" sayfasını kullanarak yayın merkezinin başlattığı canlı yayına erişip izleyebilir. Kullanıcı "Leave the Group" sayfasını kullanarak şemadan ayrılma talebinde bulunabilir.



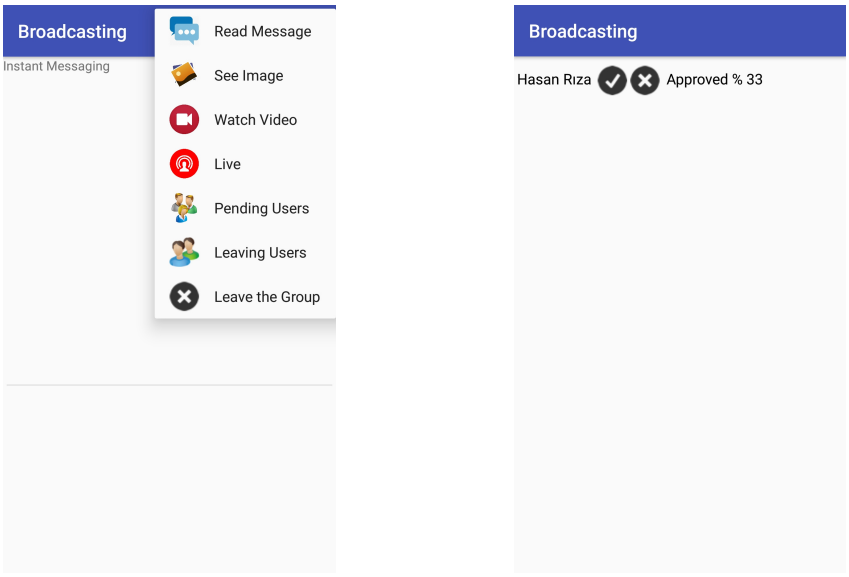
Şekil 3.18. Mesajlara ve Resimlere Erişme Sayfaları.



Şekil 3.19. Resimlere ve Videolara Erişme Sayfaları.



Şekil 3.20. Videolara ve Canlı Yayına Erişim Sayfaları.



Şekil 3.21. Şemaya Katılmak ve Ayrılmak İçin Bekleyen Kullanıcılar.

TMAD şemasının dağıtık versiyonu için yayın merkezi uygulaması içerisinde bulunan şemaya katılmak ve ayrılmak için bekleyen kullanıcılar sayfalarının kullanıcı uygulamasına taşınması gerekir. Çünkü kullanıcı ekleme-çıkarma işlemi merkezi bir yöntem yerine dağıtık olarak aktif kullanıcıların onayı alınarak sağlanmaktadır. Şekil 3.21'de görüldüğü üzere kullanıcı ekleme-çıkarma işlemleri kullanıcı uygulaması içerisine taşınmıştır. İşlemin gerçekleştirilmesi için veritabanında bazı değişikliklerin yapılması gerekir. Bu değişiklikler Bölüm 7.1.2'de anlatılmıştır.

4. BULGULAR VE TARTIŞMA

Literatürde bulunan MAH, TFA, TFZ, AGDH, DÖGİ, SISA ve MHTGG şemaları ile TMAD şeması benzer açık anahtar dağıtımı, gizli anahtarlı şifreleme yöntemi ve bir bulut sunucu üzerinde bulunan benzer bir Nosql veritabanı kullanılarak Java dilinde gerçekleştirilmiştir. Şemalar, literatürde yer alan performans ölçüm kriterleri [1], [21], [27], [48], [71] kullanılarak anahtar iletim sayısı, anahtar iletim boyutu, işlem maliyeti ve kullanıcıda bulunan anahtar sayısı ve boyutu açılarından karşılaştırılmıştır. Şemalar ayrıca literatürde yer alan güvenlik ölçüm kriterleri [48] kullanılarak grup iletişim gereksinimleri ve düşman modelleri açılarından karşılaştırılmıştır.

4.1. PERFORMANS DEĞERLENDİRME İŞLEM ADIMLARI

Şemalara öncelikle her bir adımda bir kullanıcı ekleyerek, sırasıyla $2^n (0 \leq n \leq 21)$ kullanıcı ekleme işlemi gerçekleştirilmiştir. Ardından her bir adımda bir kullanıcı çıkartarak, sırasıyla 2^n kullanıcı çıkarma işlemi gerçekleştirilmiştir. Ayrıca şemalara tek seferde 2^n kullanıcının eklendiği ve çıkartıldığı toplu kullanıcı ekleme ve çıkarma işlemleri gerçekleştirilmiştir.

Anahtar İletim Sayısı: Ağaç üzerinde 2^n kullanıcının oluşturulması sonucu kullanıcılara iletilen toplam anahtar sayısıdır. Bu hesaplama, kullanıcıların sırasıyla eklendiğinde ortaya çıkan anahtar güncelleme sayısını belirtir. Anahtar güncelleme kullanıcı anahtarı ile başlar, kullanıcıdan kök düğüme kadar devam eder.

İşlem Maliyeti: Ağaç üzerinde 2^n kullanıcının oluşturulması sonucu ortaya çıkan toplam işlem sayısıdır. Bu hesaplama, kullanıcıların sırasıyla eklendiğinde gerçekleştirilen anahtar üretim ve dağıtım işlemi ile kullanıcıdan kök düğüme anahtarların güncellenmesi ve ihtiyaç duyan kullanıcılara dağıtım işlemini ifade eder.

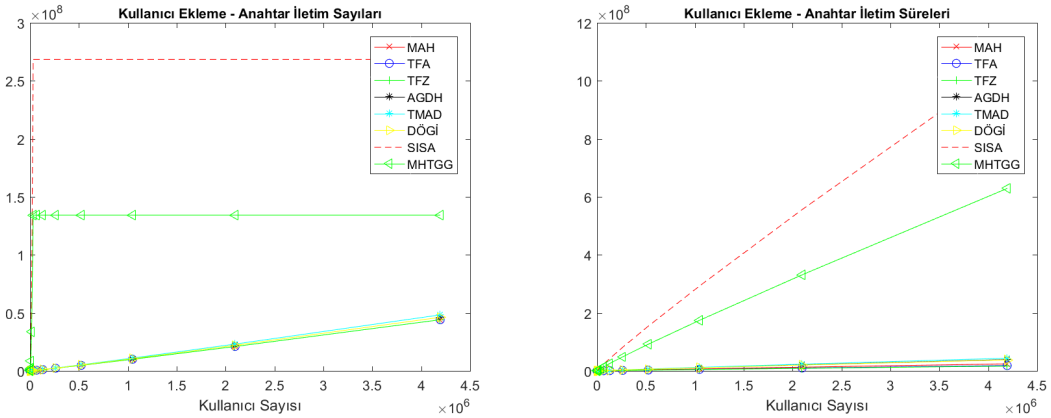
Kullanıcılarda Bulunan Anahtar Sayısı ve Boyutu: Ağaç üzerindeki bulunan her 2^n kullanıcı için, kullanıcılarda bulunması gereken anahtar sayısı ve boyutunu içermektedir.

Anahtar İletim Boyutu: Ağaç üzerinde 2^n kullanıcının oluşturulmasının ardından kullanıcılara iletilen anahtarların toplam boyutudur. Hesaplama için anahtar iletim sayılarından yararlanılır. Şifreleme işleminde kullanılan anahtarların boyutları kullanılan yönteme göre farklılık göstermekle beraber, sabittir ve birbirlerine eşittir. Bu nedenle 2^n kullanıcı için toplam anahtar iletim boyutu, anahtar iletim sayısının, bir anahtar boyutu ile çarpımı sonucu elde edilir.

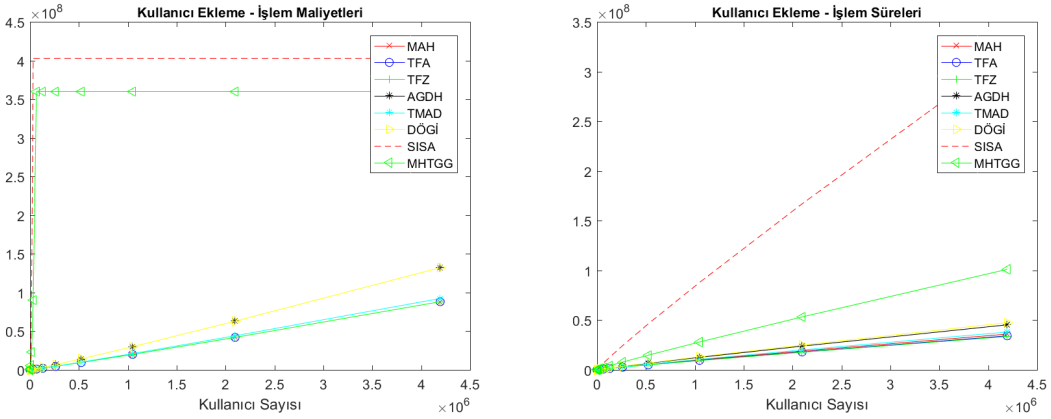
4.2. PERFORMANS DEĞERLENDİRME

Sonuçların hesaplanmasında, ilgili şemalarda gizli anahtarlı şifreleme yöntemi olarak AES-128'den, anahtar dağıtık protokolü olarak EEDH-112'den yararlanılmıştır. Gizli ve açık anahtarlı yöntemlerle oluşturulmuş anahtarlar genellikle oluşturuldukları formatta kullanıma uygun değildirler. Bu nedenle anahtar güncellemelerinde anahtarlar üzerinde işlem yapabilmek için, bu anahtarların öncelikle uygun bir formata dönüştürülmesi gerekir. Bunun için Base64 sınıfından yararlanılır. Base64 sınıfı ikili veriler ile UTF-8 kodlu metinler arasında dönüşüm işlemi gerçekleştirir. Bu sayede anahtarlar, üzerlerinde işlem yapılabilecek metinsel ifadeler haline dönüşür. Base64 sınıfında UTF-8 formatında bir metinsel ifadeye dönüştürülen anahtarın boyutu ilk boyutuna göre daha büyüktür. Bu nedenle özellikte anahtar iletim boyutu hesaplamalarında, sonuçların bayt cinsinden daha büyük anahtar boyutları ile elde edilmesine neden olmaktadır.

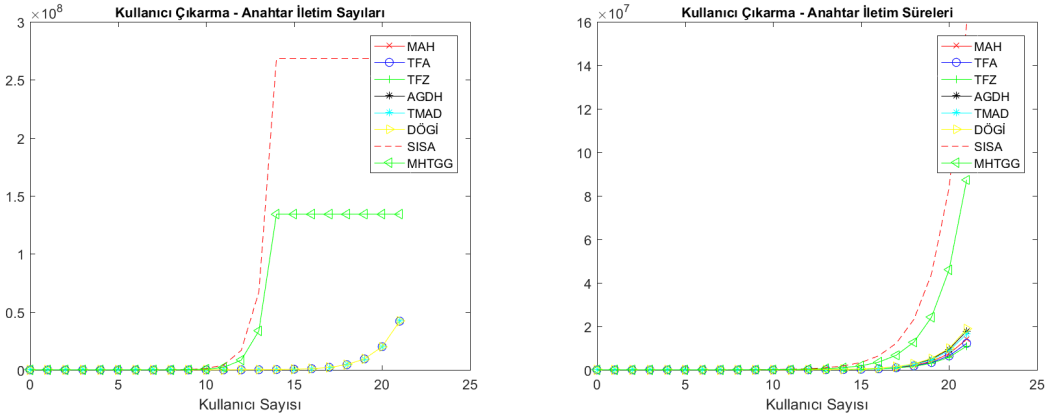
Anahtar boyutları, anahtarların uygulama içerisinde kullanım boyutu esas alınarak belirlenmiştir. Örneğin 16 baytlık (128 bit) bir AES anahtarı Base64 sınıfı kullanılarak, UTF-8 formatında 24 bayt boyutunda bir anahtara dönüştüğünden, hesaplamalarda AES-128 için 24 bayt esas alınmıştır. EEDH-112 ile oluşturulmuş ve Base64 sınıfından geçirilmiş bir açık anahtar 104 bayt, gizli anahtar ise 92 bayt olmaktadır. Sonuçların tablosal gösterimi ile formüller arasındaki ilişki incelendiğinde 2^0 satırlarında istisna oluşabilmektedir. Sonuç grafikleri için Matlab programından yararlanılmıştır. K.S. kısaca "Kullanıcı Sayısı" anlamına gelmektedir. Formül isimleri içerisinde yer alan kısaltmalar sırasıyla "KE" kullanıcı ekleme, "KC" kullanıcı çıkarma, "TKE" toplu kullanıcı ekleme ve "TKC" toplu kullanıcı çıkarma anlamına gelmektedir.



Şekil 4.1. Kullanıcı Ekleme – Anahtar İletim Sayıları ve Süreleri.



Şekil 4.2. Kullanıcı Ekleme – İşlem Maliyetleri ve Süreleri.



Şekil 4.3. Kullanıcı Çıkarma – Anahtar İletim Sayıları ve Süreleri.

Çizelge 4.1. Kullanıcı Ekleme – Anahtar İletim Sayıları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^0	2	32	2	32	2	58	3	286
2^1	4	42	4	32	4	26	6	74
2^2	10	70	10	72	10	60	14	165
2^3	26	128	26	132	26	114	34	334
2^4	66	283	66	230	66	238	82	680
2^5	162	627	162	451	162	504	194	1020
2^6	386	1265	386	995	386	1033	450	2550
2^7	898	2997	898	2250	898	2046	1026	4845
2^8	2050	5694	2050	4275	2050	3887	2306	9206
2^9	4610	10819	4610	8123	4610	7386	5122	17490
2^{10}	10242	20556	10242	15433	10242	14034	11266	33232
2^{11}	22530	39057	22530	29322	22530	26664	24578	63141
2^{12}	49154	74209	49154	55712	49154	50661	53250	119967
2^{13}	106498	140997	106498	105853	106498	96256	114690	227937
2^{14}	229378	267893	229378	201121	229378	182886	245762	433081
2^{15}	491522	508997	491522	382130	491522	347484	524290	822854
2^{16}	1048578	967095	1048578	726047	1048578	660219	1114114	1563422
2^{17}	2228226	1837481	2228226	1379490	2228226	1254416	2359298	2970502
2^{18}	4718594	3491213	4718594	2621031	4718594	2383391	4980738	5643953
2^{19}	9961474	6633305	9961474	4979959	9961474	4528442	10485762	10723511
2^{20}	20971522	12603279	20971522	9461921	20971522	8604040	22020098	20374670
2^{21}	44040194	23946230	44040194	17977650	44040194	16347677	46137346	38711874

Çizelge 4.2. Kullanıcı Ekleme – Anahtar İletim Sayıları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^0	4	79	3	25	5	142	2	25
2^1	8	72	6	1	9	314	5	48
2^2	18	136	14	3	23	720	14	66
2^3	42	249	34	7	75	919	44	138
2^4	98	709	82	323	275	19275	152	11479
2^5	226	1230	194	980	1059	36623	560	21810
2^6	514	2867	450	2490	4163	69583	2144	41439
2^7	1154	5447	1026	4731	16515	132207	8384	78734
2^8	2562	10350	2306	8989	65795	251194	33152	149595
2^9	5634	19665	5122	17079	262659	477268	131840	284231
2^{10}	12290	37363	11266	32450	1049603	906809	525824	540040
2^{11}	26626	70990	24578	61655	4196355	1722938	2100224	1026075
2^{12}	57346	134881	53250	117144	16781315	3273582	8394752	1949543
2^{13}	122882	256273	114690	222574	67117059	6219805	33566720	3704132
2^{14}	262146	486919	245762	422891	268451843	11817630	134242304	7037851
2^{15}	557058	925146	524290	803492	1073774595	22453497	536920064	13371917
2^{16}	1179650	1757777	1114114	1526635	4295032835	42661645	2147581952	25406642
2^{17}	2490370	3339776	2359298	2900607	17180000259	81057126	8590131200	48272620
2^{18}	5242882	6345574	4980738	5511154	68719738883	154008539	34360131584	91717977
2^{19}	11010050	12056590	10485762	10471193	274878431235	292616224	137439739904	174264157
2^{20}	23068674	22907522	22020098	19895266	1099512676355	555970825	549757386752	331101899
2^{21}	48234498	43524291	46137346	37801006	4398048608259	1056344567	2199026401280	629093607

Çizelge 4.3. Kullanıcı Ekleme – İşlem Maliyetleri (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^0	4	34	4	39	4	59	6	293
2^1	8	43	8	44	8	28	12	167
2^2	20	173	20	184	20	169	30	187
2^3	52	345	52	328	52	322	78	392
2^4	132	656	132	623	132	612	198	826
2^5	324	1245	324	1184	324	1162	486	1569
2^6	772	2366	772	2250	772	2209	1158	2982
2^7	1796	4496	1796	4275	1796	4196	2694	5666
2^8	4100	8543	4100	8122	4100	7973	6150	10765
2^9	9220	16231	9220	15431	9220	15149	13830	20453
2^{10}	20484	30839	20484	29319	20484	28783	30726	38860
2^{11}	45060	58593	45060	55706	45060	54687	67590	73834
2^{12}	98308	111327	98308	105842	98308	103905	147462	140284
2^{13}	212996	211522	212996	201099	212996	197420	319494	266540
2^{14}	458756	401891	458756	382088	458756	375099	688134	506426
2^{15}	983044	763594	983044	725967	983044	712687	1474566	962210
2^{16}	2097156	1450828	2097156	1379338	2097156	1354106	3145734	1828198
2^{17}	4456452	2756573	4456452	2620742	4456452	2572802	6684678	3473576
2^{18}	9437188	5237489	9437188	4979410	9437188	4888323	14155782	6599795
2^{19}	19922948	9951229	19922948	9460878	19922948	9287814	29884422	12539611
2^{20}	41943044	18907335	41943044	17975669	41943044	17646846	62914566	23825261
2^{21}	88080388	35923936	88080388	34153771	88080388	33529007	132120582	45267995

Çizelge 4.4. Kullanıcı Ekleme – İşlem Maliyetleri (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^0	6	82	6	34	7	7	6	40
2^1	12	78	12	8	13	56	14	71
2^2	28	139	30	113	34	1492	36	100
2^3	68	365	78	245	112	4029	104	244
2^4	164	694	198	880	412	5789	304	1844
2^5	388	1318	486	1645	1588	10999	896	3504
2^6	900	2504	1158	3126	6244	20898	2720	6657
2^7	2052	4757	2694	5938	24772	39707	8672	12648
2^8	4612	9038	6150	11283	98692	75443	29280	24031
2^9	10244	17172	13830	21438	393988	143341	104288	45659
2^{10}	22532	32626	30726	40732	1574404	272349	387424	86753
2^{11}	49156	61990	67590	77390	6294532	517462	1482080	164830
2^{12}	106500	117781	147462	147042	25171972	983178	5776736	313177
2^{13}	229380	223784	319494	279380	100675588	1868039	22771040	595036
2^{14}	491524	425189	688134	530821	402677764	3549274	90346848	1130569
2^{15}	1048580	807860	1474566	1008560	1610661892	6743621	359781728	2148080
2^{16}	2228228	1534934	3145734	1916265	6442549252	12812880	1435653472	4081353
2^{17}	4718596	2916374	6684678	3640903	25770000388	24344472	5735142752	7754570
2^{18}	9961476	5541111	14155782	6917716	103079608324	46254497	22924580192	14733683
2^{19}	20971524	10528112	29884422	13143660	412317646852	87883544	91664242016	27993998
2^{20}	44040196	20003412	62914566	24972954	1649269014532	166978734	366585000000	53188597
2^{21}	92274692	38006483	132120582	47448613	6597072912388	317259595	1466190000000	101058334

Çizelge 4.5. Kullanıcı Çıkarma – Anahtar İletim Sayıları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^{21}	41943042	14042543	41943042	12176862	41943042	10930411	41943042	17865789
2^{20}	19922946	7390812	19922946	6408875	19922946	5752848	19922946	9403047
2^{19}	9437186	3889901	9437186	3373092	9437186	3027815	9437186	4948972
2^{18}	4456450	2047316	4456450	1775312	4456450	1593587	4456450	2604722
2^{17}	2097154	1077535	2097154	934374	2097154	838730	2097154	1370906
2^{16}	983042	567124	983042	491776	983042	441437	983042	721530
2^{15}	458754	298486	458754	258830	458754	232335	458754	379752
2^{14}	212994	157098	212994	136226	212994	122282	212994	199870
2^{13}	98306	82683	98306	71698	98306	64359	98306	105195
2^{12}	45058	43517	45058	37736	45058	33873	45058	55366
2^{11}	20482	22904	20482	19861	20482	17828	20482	29140
2^{10}	9218	12055	9218	10453	9218	9383	9218	15337
2^9	4098	6345	4098	5502	4098	4938	4098	8072
2^8	1794	3339	1794	2896	1794	2599	1794	4248
2^7	770	1758	770	1524	770	1368	770	2236
2^6	322	925	322	1258	322	1288	322	824
2^5	130	226	130	227	130	146	130	353
2^4	50	112	50	119	50	122	50	193
2^3	18	23	18	9	18	11	18	99
2^2	6	11	6	2	6	20	6	58
2^1	2	10	2	5	2	19	2	17
2^0	1	9	1	5	1	6	1	25

Çizelge 4.6. Kullanıcı Çıkarma – Anahtar İletim Sayıları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^{21}	41943042	16499488	41943042	18784647	4398040219651	159007125	2199024304128	87337024
2^{20}	19922946	8683941	19922946	9886656	1099508482051	83687960	549756338176	45966855
2^{19}	9437186	4570495	9437186	5203503	274876334083	44046295	137439215616	24193081
2^{18}	4456450	2405524	4456450	2738686	68718690307	23182260	34359869440	12733201
2^{17}	2097154	1266065	2097154	1441414	17179475971	12201190	8590000128	6701685
2^{16}	983042	666350	983042	758639	4294770691	6421679	2147516416	3527202
2^{15}	458754	350711	458754	399284	1073643523	3379831	536887296	1856422
2^{14}	212994	184585	212994	210149	268386307	1778858	134225920	977064
2^{13}	98306	97150	98306	110605	67084291	936241	33558528	514244
2^{12}	45058	51131	45058	58213	16764931	492759	8390656	270655
2^{11}	20482	26911	20482	30638	4188163	259347	2098176	142450
2^{10}	9218	14164	9218	16126	1045507	136498	524800	74974
2^9	4098	7455	4098	8487	260611	71841	131328	39460
2^8	1794	3924	1794	4467	64771	37811	32896	20768
2^7	770	2065	770	2351	16003	19901	8256	10931
2^6	322	2170	322	959	3907	10474	2080	5753
2^5	130	680	130	872	931	983	528	761
2^4	50	151	50	217	211	274	136	541
2^3	18	174	18	118	43	222	36	224
2^2	6	32	6	44	7	11	10	23
2^1	2	16	2	4	1	5	3	5
2^0	1	13	1	5	1	0	1	10

Çizelge 4.1 ve Çizelge 4.2, kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim sayısını ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.1'deki gibidir. Çizelge 4.5 ve Çizelge 4.6, kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim sayısını ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.3'deki gibidir.

Şekil 4.1'de görüldüğü üzere, kullanıcı ekleme işlemleri anahtar iletim sayıları açısından değerlendirildiğinde, TMAD şeması sırasıyla, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi üçüncü performans sonucuna sahiptir. MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının iletim sayıları eşittir.

MAH şeması kullanıcı ekleme işlemi açısından incelendiğinde, GY yeni eklenen kullanıcıya gizli anahtarını ilettikten sonra, eklenen kullanıcıdan kök düğüme kadar şemada bulunan tüm anahtarları günceller. Güncel anahtarları hem yeni eklenen kullanıcıya hem de ağaçta bulunan diğer kullanıcılara iletir. TFA ve TFZ şemalarında ise, GY sırasıyla yeni eklenen kullanıcının düğüm sırrı, kör düğüm sırrı ve düğüm anahtarını hesaplar ve kullanıcıya iletir. Eklenen kullanıcıdan kök düğüme kadar şemada bulunan tüm anahtarları günceller. Yeni kullanıcıya güncel anahtarları hesaplaması için kardeş kör düğüm sırlarını iletir. Ayrıca şemada bulunan diğer kullanıcılara da ihtiyaç duydukları kardeş düğüm sırlarını gönderir. AGDH ve DÖGİ şemalarında, kullanıcılarda açık anahtar altyapısı ile oluşturulmuş anahtar çiftleri bulunmaktadır. Bu nedenle kullanıcı ekleme işleminin ardından kullanıcılara iletilen anahtar sayıları gizli anahtarlı şifreleme altyapısını kullanan merkezi şemalara göre daha fazladır.

Dağıtık şemalarda olduğu gibi, önerilen şemada kullanıcılarda açık anahtar altyapısı ile oluşturulmuş iki anahtarın yanı sıra, merkezi şemalarda olduğu gibi bir de simetrik anahtar değeri bulunur. Bu nedenle üretilen anahtarların iletimi sonucu oluşan iletim sayısı MAH, TFA, TFZ, AGDH ve DOGİ şemalarına göre daha fazladır. MAH, TFA ve TFZ şemalarında anahtar iletim sayıları birbirlerine benzerlik gösterse de anahtar oluşturma süreçleri birbirlerinden farklı olduğundan iletilen anahtar boyutları farklıdır.

$$Anahtar_{sayisi_{KE}} = \begin{cases} 2^{n-1}(1+n) & \text{MAH, TFA, TFZ şemaları için} \\ 2^{n-1}(2+n) & \text{AGDH, DÖGİ şemaları için} \\ 2^{n-1}(3+n) & \text{TMAD şeması için} \\ 2^n * (2^n+1)+3 & \text{SISA şeması için} \\ ((2^n+1)*(2^n+2)/2)-1 & \text{MHTGG şeması için} \end{cases} \quad (4.1)$$

Denklem 4.1'de TMAD şemasında her 2^n kullanıcı ekleme işleminde iletimi yapılan anahtar sayısının formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 3 değeri her bir kullanıcı ekleme işleminde iletilmesi gereken açık anahtar, gizli anahtar ve simetrik değeri ifade etmektedir. n değeri ise bir kullanıcıdan kök düğüme olan uzaklığı, bir diğer ifade ile ağacın derinliğini ifade etmektedir. Aşağıdan yukarıya her adımda anahtar iletim sayısı bir önceki adımın iletim sayısı üzerine ekleme yapılarak hesaplandığından, şemaya 2^n kullanıcı ekleme işlemi şemaya eklenmiş 2^{n-1} kullanıcıya sırasıyla 2^{n-1} kullanıcı daha ekleyerek gerçekleştirilir. Bu nedenle 3 değeri 2^{n-1} ile çarpma işlemine tabi tutulur. $2^{n-1} * n$ ise kullanıcı ekleme işleminin ardından iletilen aradüğüm anahtar sayısıdır. Çizelge 4.1 ve Çizelge 4.2'deki her bir adımdaki sonuç bir önceki adımın üzerine eklenerek elde edilmektedir.

Şekil 4.3'de görüldüğü üzere, kullanıcı çıkarma işlemleri anahtar iletim sayıları açısından değerlendirildiğinde, TMAD şeması ile MAH, TFA, TFZ, AGDH ve DÖGİ şemaları aynı sonucu vermektedir. Bunun nedeni kullanıcı çıkarma işleminde şemaya eklenecek yeni kullanıcılar olmadığından yeni anahtarların oluşturulmaması ve sadece aradüğüm anahtarlarının güncellenmesidir. Ayrıca dağıtık şemalarda aradüğümlerde açık anahtar altyapısı kullanılarak oluşturulmuş anahtar çifti bulunsa da, anahtar iletiminde bu anahtarlardan yalnızca kör anahtar olarak adlandırılan açık anahtar kullanılmaktadır. Güncellenen ve iletilen aradüğüm anahtarı sayısı ikili ağaç şemaları için eşit sayıda olmaktadır.

Denklem 4.2'de TMAD şemasında her 2^n kullanıcı çıkarma işleminde iletimi yapılan anahtar sayısının formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. n değeri ise bir kullanıcıdan kök düğüme olan uzaklığı, bir diğer ifade ile ağacın derinliğini ifade etmektedir.

$$Anahtar_{sayisi_{KC}} = \begin{cases} 2^{n-1} * n & \text{MAH, TFA, TFZ, AGDH, DÖGİ,} \\ & \text{TMAD şemaları için} \\ (2^n - 1) * (2^n - 2) + 1 & \text{SISA şeması için} \\ 2^n * (2^n + 1)/2 & \text{MHTGG şeması için} \end{cases} \quad (4.2)$$

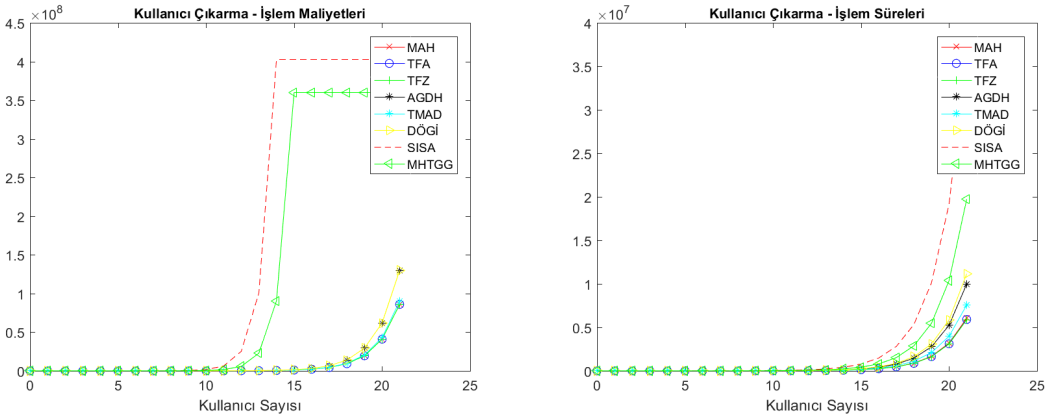
Kullanıcı çıkarma işleminde, şemadan ayrılan kullanıcılar için anahtar güncellemeye ihtiyaç olmadığından yalnızca aradüğüm anahtarları güncellenir. Bu nedenle anahtar iletim sayısı güncellenen aradüğüm anahtar sayısıyla ilişkilidir.

Çizelge 4.7. Kullanıcı Çıkarma – İşlem Maliyetleri (a).

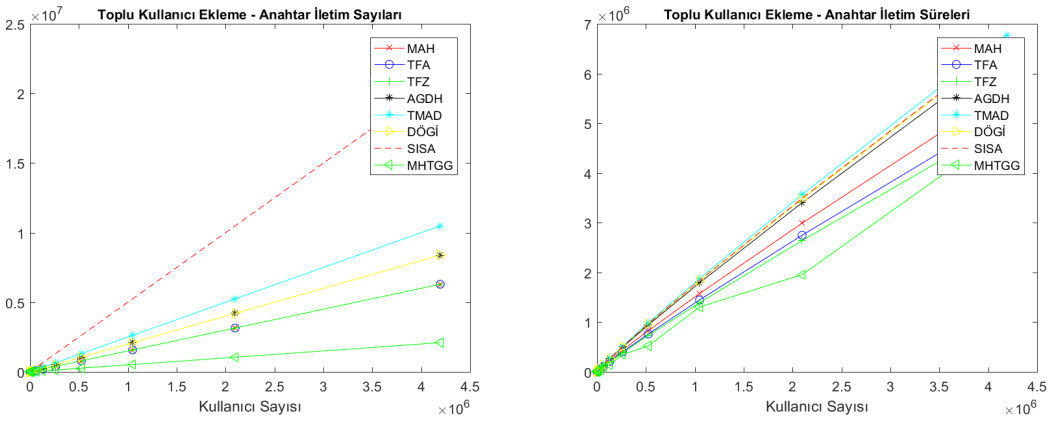
	MAH		TFA		TFZ		AGDH	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^{21}	85983236	6042089	85983236	5905458	85983236	5875096	130023430	9974000
2^{20}	40894468	3180047	40894468	3108136	40894468	3092156	61865990	5249474
2^{19}	19398660	1673709	19398660	1635861	19398660	1627450	29360134	2762881
2^{18}	9175044	880899	9175044	860980	9175044	856553	13893638	1454148
2^{17}	4325380	463631	4325380	453147	4325380	450817	6553606	765341
2^{16}	2031620	244016	2031620	238498	2031620	237272	3080198	402811
2^{15}	950276	128430	950276	125526	950276	124880	1441798	212006
2^{14}	442372	67595	442372	66066	442372	65726	671750	111582
2^{13}	204804	35576	204804	34772	204804	34593	311302	58727
2^{12}	94212	18724	94212	18301	94212	18207	143366	30909
2^{11}	43012	9855	43012	9632	43012	9583	65542	16268
2^{10}	19460	5187	19460	5069	19460	5043	29702	8562
2^{9}	8708	2730	8708	2668	8708	2654	13318	4506
2^{8}	3844	1437	3844	1404	3844	1397	5894	2372
2^{7}	1668	756	1668	739	1668	735	2566	1248
2^{6}	708	398	708	389	708	387	1094	657
2^{5}	292	66	292	90	292	149	454	358
2^{4}	116	16	116	49	116	27	182	253
2^{3}	44	38	44	19	44	21	70	125
2^{2}	16	18	16	9	16	21	26	70
2^{1}	6	18	6	7	6	19	10	22
2^{0}	3	20	3	11	3	8	5	28

Çizelge 4.8. Kullanıcı Çıkarma – İşlem Maliyetleri (b).

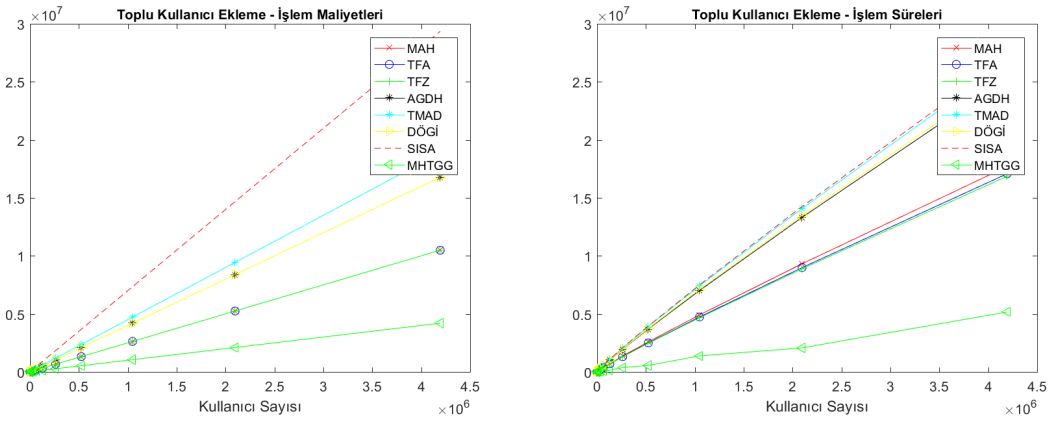
	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^{21}	90177540	7560201	130023430	11146143	6597060329478	36498625	1466190000000	19719485
2^{20}	42991620	3979053	61865990	5866391	1649262723078	19209803	366585000000	10378676
2^{19}	20447236	2094239	29360134	3087574	412314501126	10110423	91664242016	5462461
2^{18}	9699332	1102231	13893638	1625039	103078035462	5321275	22924580192	2874980
2^{17}	4587524	580121	6553606	855284	25769213958	2800671	5735142752	1513147
2^{16}	2162692	305327	3080198	450149	6442156038	1474037	1435653472	796393
2^{15}	1015812	160698	1441798	236921	1610465286	775809	359781728	419154
2^{14}	475140	84578	671750	124695	402579462	408321	90346848	220608
2^{13}	221188	44515	311302	65629	100626438	214906	22771040	116109
2^{12}	102404	23429	143366	34542	25147398	113108	5776736	61110
2^{11}	47108	12331	65542	18180	6282246	59531	1482080	32163
2^{10}	21508	6490	29702	9568	1568262	31332	387424	16928
2^{9}	9732	3416	13318	5036	390918	16490	104288	8909
2^{8}	4356	1798	5894	2651	97158	8679	29280	4689
2^{7}	1924	946	2566	1395	24006	4568	8672	2468
2^{6}	836	498	1094	231	5862	6943	2720	4453
2^{5}	356	175	454	101	1398	4362	896	1181
2^{4}	148	83	182	263	318	1064	304	121
2^{3}	60	89	70	25	66	236	104	48
2^{2}	24	38	26	9	12	501	36	35
2^{1}	10	18	10	6	3	53	14	9
2^{0}	5	14	5	13	2	57	6	13



Şekil 4.4. Kullanıcı Çıkarma – İşlem Maliyetleri ve Süreleri.



Şekil 4.5. Toplu Kullanıcı Ekleme – Anahtar İletim Sayıları ve Süreleri.



Şekil 4.6. Toplu Kullanıcı Ekleme – İşlem Maliyetleri ve Süreleri.

Çizelge 4.9. Toplu Kullanıcı Ekleme – Anahtar İletim Sayıları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^0	2	48	2	8	2	42	2	34
2^1	4	82	4	39	4	79	6	90
2^2	10	112	10	90	10	66	14	135
2^3	22	259	22	161	22	214	30	340
2^4	46	609	46	338	46	259	62	212
2^5	94	197	94	181	94	174	126	224
2^6	190	374	190	344	190	331	254	426
2^7	382	711	382	653	382	628	510	809
2^8	766	1351	766	1241	766	1193	1022	1536
2^9	1534	2567	1534	2359	1534	2268	2046	2919
2^{10}	3070	4878	3070	4482	3070	4308	4094	5546
2^{11}	6142	9268	6142	8515	6142	8186	8190	10538
2^{12}	12286	17609	12286	16179	12286	15553	16382	20023
2^{13}	24574	33458	24574	30740	24574	29551	32766	38043
2^{14}	49150	63569	49150	58406	49150	56148	65534	72282
2^{15}	98302	120782	98302	110972	98302	106681	131070	137336
2^{16}	196606	229486	196606	210847	196606	202693	262142	260938
2^{17}	393214	436023	393214	400610	393214	385117	524286	495783
2^{18}	786430	828444	786430	761159	786430	731722	1048574	941987
2^{19}	1572862	1574043	1572862	1446202	1572862	1390272	2097150	1789775
2^{20}	3145726	2990682	3145726	2747784	3145726	2641516	4194302	3400572
2^{21}	6291454	5682296	6291454	5220790	6291454	5018881	8388606	6461088

Çizelge 4.10. Toplu Kullanıcı Ekleme – Anahtar İletim Sayıları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^0	3	67	2	4	2	1	2	2
2^1	8	118	6	31	12	12	3	14
2^2	18	273	14	118	32	35	5	36
2^3	38	218	30	122	72	168	9	56
2^4	78	250	62	226	152	218	17	90
2^5	158	235	126	229	312	230	33	125
2^6	318	447	254	435	632	437	65	187
2^7	638	848	510	827	1272	830	129	468
2^8	1278	1612	1022	1571	2552	1578	257	703
2^9	2558	3063	2046	2984	5112	2997	513	1757
2^{10}	5118	5819	4094	5670	10232	5695	1025	2636
2^{11}	10238	11056	8190	10774	20472	10821	2049	6591
2^{12}	20478	21006	16382	20470	40952	20559	4097	9887
2^{13}	40958	39911	32766	38892	81912	39062	8193	24719
2^{14}	81918	75832	65534	73895	163832	74218	16385	37078
2^{15}	163838	144080	131070	140401	327672	141015	32769	92697
2^{16}	327678	273752	262142	266763	655352	267928	65537	139045
2^{17}	655358	520129	524286	506849	1310712	509062	131073	347614
2^{18}	1310718	988245	1048574	963013	2621432	967219	262145	521421
2^{19}	2621438	1877666	2097150	1829725	5248872	1837715	524289	1303553
2^{20}	5242878	3567565	4194302	3476478	10485752	3491659	1048577	1955330
2^{21}	10485758	6778373	8388606	6605308	20971512	6634153	2097153	4888325

Çizelge 4.11. Toplu Kullanıcı Ekleme – İşlem Maliyetleri (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^0	4	48	4	56	4	42	4	37
2^1	7	84	7	60	7	81	12	99
2^2	17	114	17	137	17	80	28	142
2^3	37	170	37	164	37	162	60	243
2^4	77	323	77	312	77	308	124	462
2^5	157	614	157	592	157	585	252	877
2^6	317	1166	317	1125	317	1111	508	1667
2^7	637	2215	637	2137	637	2111	1020	3167
2^8	1277	4209	1277	4061	1277	4011	2044	6017
2^9	2557	7998	2557	7716	2557	7621	4092	11432
2^{10}	5117	15196	5117	14659	5117	14481	8188	21721
2^{11}	10237	28872	10237	27853	10237	27513	16380	41270
2^{12}	20477	54857	20477	52921	20477	52275	32764	78413
2^{13}	40957	104228	40957	100549	40957	99323	65532	148985
2^{14}	81917	198033	81917	191044	81917	188714	131068	283071
2^{15}	163837	376264	163837	362984	163837	358557	262140	537836
2^{16}	327677	714901	327677	689669	327677	681258	524284	1021887
2^{17}	655357	1358311	655357	1310371	655357	1294391	1048572	1941586
2^{18}	1310717	2580792	1310717	2489705	1310717	2459343	2097148	3689014
2^{19}	2621437	4903504	2621437	4730439	2621437	4672751	4194300	7009126
2^{20}	5242877	9316658	5242877	8987834	5242877	8878227	8388604	13317340
2^{21}	10485757	17701650	10485757	17076885	10485757	16868631	16777212	25302946

Çizelge 4.12. Toplu Kullanıcı Ekleme – İşlem Maliyetleri (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^0	8	71	4	5	4	8	6	3
2^1	15	126	12	49	18	86	8	15
2^2	33	292	28	34	46	768	12	40
2^3	69	257	60	246	102	260	20	58
2^4	141	488	124	467	214	494	36	96
2^5	285	928	252	888	438	939	68	132
2^6	573	1763	508	1687	886	1783	132	198
2^7	1149	3349	1020	3206	1782	3388	260	495
2^8	2301	6364	2044	6091	3574	6438	516	742
2^9	4605	12091	4092	11573	7158	12232	1028	1856
2^{10}	9213	22973	8188	21989	14326	23241	2052	2784
2^{11}	18429	43648	16380	41780	28662	44157	4100	6960
2^{12}	36861	82931	32764	79381	57334	83899	8196	10441
2^{13}	73725	157568	65532	150824	114678	159408	16388	26103
2^{14}	147453	299380	131068	286566	229366	302875	32772	39155
2^{15}	294909	568822	262140	544475	458742	575462	65540	97888
2^{16}	589821	1080762	524284	1034503	917494	1093378	131076	146832
2^{17}	1179645	2053447	1048572	1965556	1834998	2077417	262148	367080
2^{18}	2359293	3901550	2097148	3734557	3670006	3947093	524292	550621
2^{19}	4718589	7412944	4194300	7095659	7340022	7499477	1048580	1376552
2^{20}	9437181	14084594	8388604	13481752	14680054	14249006	2097156	2064828
2^{21}	18874365	26760729	16777212	25615328	29360118	27073111	4194308	5162072

Çizelge 4.9 ve Çizelge 4.10, toplu kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim sayısını ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.5'deki gibidir. Toplu kullanıcı ekleme işlemleri anahtar iletim sayısı açısından değerlendirildiğinde, ilk sırada MHTGG şeması yer almaktadır. Toplu kullanıcı ekleme işlemi şemaya tek seferde 2^n kullanıcı ekleme işlemidir. Şekil 2.6'da görüldüğü üzere MHTGG şemasının halka topolojisine sahip olması nedeniyle, şemaya 2^n kullanıcı ekleme işleminin ardından GY'den kullanıcılara anahtarın 2^n+1 kez dolaştırılması yeterli olmaktadır. İkinci sırada MAH, TFA ve TFZ şemaları yer almaktadır. Üçüncü sırada TMAD şeması, dördüncü sırada AGDH ve DÖGİ şemaları bulunmaktadır. SISA şeması ise beşinci sırada yer almaktadır. TMAD şemasının MHTGG şeması ile MAH, TFA ve TFZ şemalarının ardında yer almalarının nedeni bu şemalara kıyasla toplu kullanıcı ekleme işlemlerinde kullanıcılara iletilmesi gereken anahtar sayısının daha fazla olmasıdır.

$$Anahtar_{sayisi_{TKE}} = \begin{cases} 1*2^n+2*(2^n-1) & \text{MAH, TFA, TFZ şemaları için} \\ 2*2^n+2*(2^n-1) & \text{AGDH, DÖGİ şemaları için} \\ 3*2^n+2*(2^n-1) & \text{TMAD şeması için} \\ 2*2^n+8*(2^n-1) & \text{SISA şeması için} \\ 2^n+1 & \text{MHTGG şeması için} \end{cases} \tag{4.3}$$

Denklem 4.3'de TMAD şemasında her toplu 2^n kullanıcı ekleme işleminde iletimi yapılan anahtar sayısının formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 3 değeri her bir kullanıcı ekleme işleminde iletilmesi gereken açık anahtar, gizli anahtar ve simetrik değeri ifade etmektedir. 2^n kullanıcı tek bir seferde oluşturulduğundan 3 ile çarpma işlemine tabi tutulur. $2(2^n - 1)$ toplu kullanıcı ekleme işleminin ardından aradüğüm güncellemeleri için iletilen anahtar sayısını ifade etmektedir.

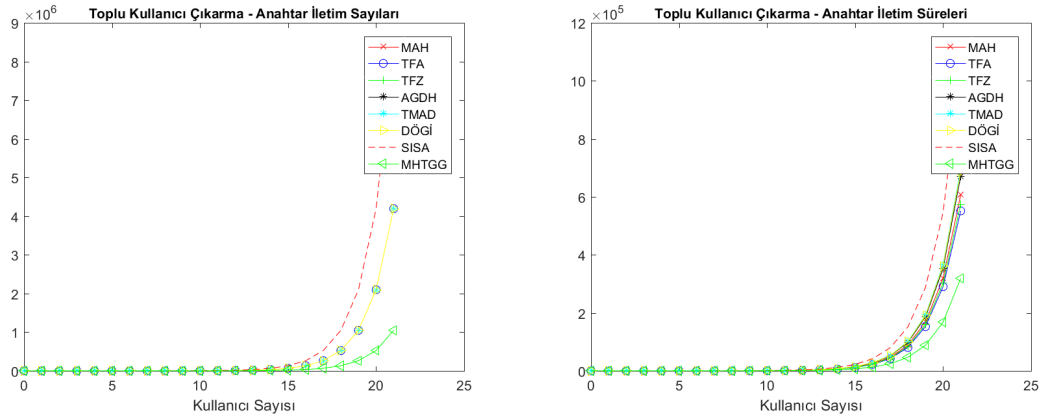
Çizelge 4.13. Toplu Kullanıcı Çıkarma – Anahtar İletim Sayıları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^{21}	4194302	607245	4194302	551315	4194302	575285	4194302	671166
2^{20}	2097150	319603	2097150	290166	2097150	302781	2097150	353245
2^{19}	1048574	168212	1048574	152719	1048574	159359	1048574	185918
2^{18}	524286	88533	524286	80378	524286	83873	524286	97852
2^{17}	262142	46596	262142	42304	262142	44144	262142	51501
2^{16}	131070	24524	131070	22265	131070	23234	131070	27106
2^{15}	65534	12908	65534	11719	65534	12228	65534	14266
2^{14}	32766	6793	32766	6168	32766	6436	32766	7509
2^{13}	16382	3575	16382	3246	16382	3387	16382	3952
2^{12}	8190	1882	8190	1709	8190	1783	8190	2080
2^{11}	4094	990	4094	899	4094	938	4094	1095
2^{10}	2046	521	2046	473	2046	494	2046	576
2^9	1022	274	1022	249	1022	260	1022	303
2^8	510	144	510	131	510	137	510	160
2^7	254	76	254	69	254	72	254	84
2^6	126	88	126	87	126	95	126	77
2^5	62	34	62	115	62	86	62	95
2^4	30	49	30	10	30	70	30	35
2^3	14	41	14	66	14	33	14	27
2^2	6	9	6	28	6	25	6	17
2^1	2	12	2	16	2	12	2	27
2^0	0	6	0	13	0	4	0	16

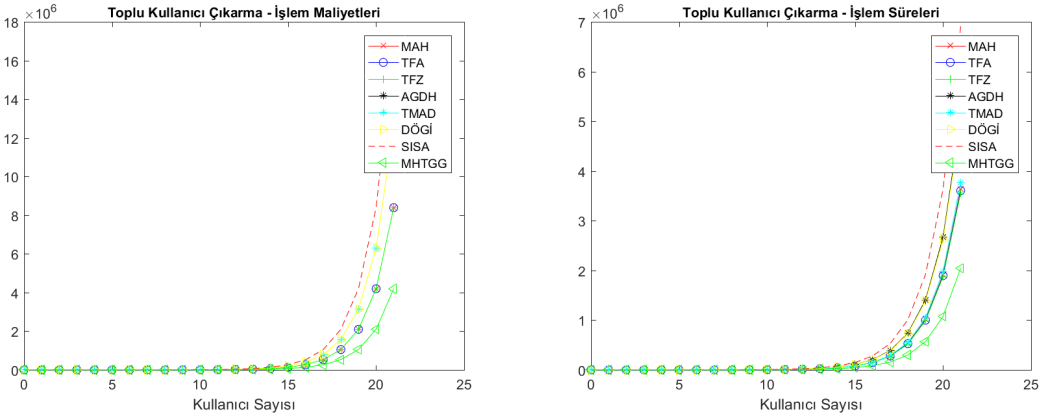
Çizelge 4.14. Toplu Kullanıcı Çıkarma – Anahtar İletim Sayıları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)	İletim Sayısı	Süre(ms)
2^{21}	4194302	695136	4194302	687146	8388604	1038709	1048577	318804
2^{20}	2097150	365861	2097150	361656	4194300	546689	524289	167791
2^{19}	1048574	192558	1048574	190345	2097148	287731	262145	88311
2^{18}	524286	101347	524286	100182	1048572	151437	131073	46480
2^{17}	262142	53340	262142	52727	524284	79704	65537	24463
2^{16}	131070	28074	131070	27751	262140	41949	32769	12875
2^{15}	65534	14776	65534	14606	131068	22079	16385	6776
2^{14}	32766	7777	32766	7687	65532	11620	8193	3567
2^{13}	16382	4093	16382	4046	32764	6116	4097	1877
2^{12}	8190	2154	8190	2129	16380	3219	2049	988
2^{11}	4094	1134	4094	1121	8188	1694	1025	520
2^{10}	2046	597	2046	590	4092	892	513	274
2^9	1022	314	1022	310	2044	469	257	144
2^8	510	165	510	163	1020	247	129	76
2^7	254	87	254	86	508	130	65	40
2^6	126	96	126	91	252	94	33	21
2^5	62	94	62	100	124	63	17	9
2^4	30	73	30	62	60	36	9	9
2^3	14	56	14	17	28	9	5	8
2^2	6	19	6	11	12	5	3	4
2^1	2	26	2	3	4	0	1	4
2^0	0	19	0	4	0	0	0	1

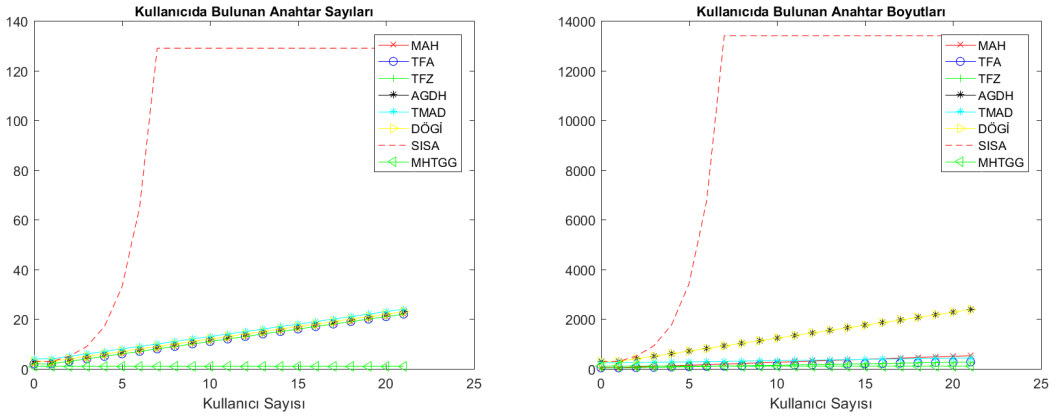
Çizelge 4.13 ve Çizelge 4.14, toplu kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim sayısını ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.7'deki gibidir. Toplu kullanıcı çıkarma işlemleri iletim sayısı açısından değerlendirildiğinde, TMAD şeması ile MAH, TFA, TFZ, AGDH ve DÖGİ şemaları MHTGG şemasının ardından en iyi ikinci performans



Şekil 4.7. Toplu Kullanıcı Çıkarma – Anahtar İletim Sayıları ve Süreleri.



Şekil 4.8. Toplu Kullanıcı Çıkarma – İşlem Maliyetleri ve Süreleri.



Şekil 4.9. Kullanıcılarda Bulunan Anahtar Sayıları ve Boyutları.

Çizelge 4.15. Toplu Kullanıcı Çıkarma – İşlem Maliyetleri (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^{21}	8388605	3643470	8388605	3603520	8388605	3563570	12582908	5073692
2^{20}	4194301	1917616	4194301	1896590	4194301	1875563	6291452	2670364
2^{19}	2097149	1009272	2097149	998205	2097149	987138	3145724	1405455
2^{18}	1048573	531196	1048573	525371	1048573	519547	1572860	739713
2^{17}	524285	279577	524285	276511	524285	273446	786428	389323
2^{16}	262141	147146	262141	145532	262141	143919	393212	204907
2^{15}	131069	77445	131069	76596	131069	75747	196604	107846
2^{14}	65533	40761	65533	40314	65533	39867	98300	56761
2^{13}	32765	21453	32765	21218	32765	20982	49148	29874
2^{12}	16381	11291	16381	11167	16381	11043	24572	15723
2^{11}	8189	5943	8189	5877	8189	5812	12284	8275
2^{10}	4093	3128	4093	3093	4093	3059	6140	4355
2^{9}	2045	1646	2045	1628	2045	1610	3068	2292
2^{8}	1021	866	1021	857	1021	847	1532	1207
2^{7}	509	456	509	451	509	446	764	635
2^{6}	253	154	253	993	253	304	380	303
2^{5}	125	141	125	337	125	136	188	282
2^{4}	61	72	61	229	61	77	92	131
2^{3}	29	44	29	108	29	38	44	104
2^{2}	13	10	13	43	13	29	20	85
2^{1}	5	13	5	26	5	12	8	30
2^{0}	3	6	3	22	3	4	5	18

Çizelge 4.16. Toplu Kullanıcı Çıkarma – İşlem Maliyetleri (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)	İşlem Sayısı	Süre(ms)
2^{21}	12582909	3767947	12582908	4985802	16777210	6911408	4194308	2045457
2^{20}	6291453	1983130	6291452	2624106	8388602	3637583	2097156	1076556
2^{19}	3145725	1043753	3145724	1381109	4194298	1914517	1048580	566609
2^{18}	1572861	549344	1572860	726899	2097146	1007641	524292	298215
2^{17}	786429	289128	786428	382579	1048570	530337	262148	156955
2^{16}	393213	152173	393212	201357	524282	279125	131076	82608
2^{15}	196605	80091	196604	105977	262138	146908	65540	43478
2^{14}	98301	42153	98300	55778	131066	77320	32772	22883
2^{13}	49149	22186	49148	29357	65530	40695	16388	12044
2^{12}	24573	11677	24572	15451	32762	21418	8196	6339
2^{11}	12285	6146	12284	8132	16378	11273	4100	3336
2^{10}	6141	3235	6140	4280	8186	5933	2052	1756
2^{9}	3069	1702	3068	2253	4090	3123	1028	924
2^{8}	1533	896	1532	1186	2042	1644	516	486
2^{7}	765	396	764	624	1018	865	260	256
2^{6}	381	358	380	372	506	301	132	73
2^{5}	189	239	188	248	250	234	68	49
2^{4}	93	20	92	93	122	173	36	17
2^{3}	45	63	44	35	58	566	20	16
2^{2}	21	84	20	29	26	197	12	10
2^{1}	9	32	8	17	10	74	8	7
2^{0}	3	22	5	7	2	52	6	3

sonucuna sahiptir. İlk sırada yer alan MHTGG şemasının halka topolojisine sahip olması toplu kullanıcı çıkarma işleminde daha düşük anahtar iletim sayısına sahip olmasına neden olmaktadır. İkinci sırada bulunan TMAD şeması ile MAH, TFA, TFZ, AGDH ve DÖGİ şemaları aynı sonucu vermektedir. Bunun nedeni toplu kullanıcı çıkarma işleminde yeni kullanıcı olmadığından yeni anahtarların oluşturulmaması ve sadece aradüğüm anahtarlarının güncellenmesidir. Ayrıca dağıtık şemalarda aradüğümlerde açık anahtar altyapısı kullanılarak oluşturulmuş anahtar çifti bulunsa da, anahtar iletiminde bu anahtarlardan yalnızca kör anahtar olarak adlandırılan açık anahtar kullanılmaktadır. Güncellenen ve iletilen aradüğüm anahtarı sayısı ikili ağaç şemaları için eşit sayıda olmaktadır.

$$Anahtar_{sayisi_{TKC}} = \begin{cases} 2*(2^n-1) & \text{MAH, TFA, TFZ, AGDH,} \\ & \text{DÖGİ, TMAD şemaları için} \\ 4*(2^n-1) & \text{SISA şeması için} \\ n+1 & \text{MHTGG şeması için} \end{cases} \quad (4.4)$$

Denklem 4.4'de TMAD şemasında her toplu 2^n kullanıcı çıkarma işleminde iletimi yapılan anahtar sayısının formülü gösterilmektedir. Aşağıdan yukarıya her adımda anahtar sayısı bir önceki adımın anahtar sayısının üzerine ekleme yapılarak hesaplandığından, şemadan 2^n kullanıcı çıkarma işlemi şemadan ayrılmış 2^{n-1} kullanıcının ardından sırasıyla 2^{n-1} kullanıcı daha çıkartarak gerçekleştirilmektedir. Toplu kullanıcı çıkarma işleminde şemadan ayrılan kullanıcılara anahtar iletimine gerek yoktur. Yalnızca şemadan ayrılmamış kullanıcıların aradüğüm anahtarlarının güncellenmesi yeterlidir. Çizelge 4.3 ve Çizelge 4.4, kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan işlem maliyetini ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.2'deki gibidir. Kullanıcı ekleme işlemleri işlem maliyeti açısından değerlendirildiğinde, TMAD şeması sırasıyla, MAH, TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir.

$$Islem_{maliyeti_{KE}} = \begin{cases} 2^n(1+n) & \text{MAH,TFA,TFZ için} \\ 3*2^{n-1}(1+n) & \text{AGDH,DÖGİ için} \\ 2^{n+1}+2^n*n & \text{TMAD şeması için} \\ 3*2^{n-1}+3*(2^n-1+2^{n-1})*2^{n-2} & \text{SISA şeması için} \\ 2^{n-1}*((4n+2)+(2^{n-1}-1)) & \text{MHTGG şeması için} \end{cases} \quad (4.5)$$

Denklem 4.5'de TMAD şemasında her bir 2^n kullanıcı ekleme için gerçekleştirilen işlem maliyetinin formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. n değeri ise bir kullanıcıdan kök düğüme olan uzaklığı, bir diğer ifade ile ağacın derinliğini ifade etmektedir. İşlem maliyeti anahtar iletim sayısının yanısıra üretilen ve güncellenen anahtar sayısını da içerir. Şemaya 2^n kullanıcı sırasıyla eklendiğinde üretilen ve kullanıcılara iletilen anahtar sayısı 2^{n+1}'dir. Ayrıca aradüğüm anahtarı güncelleme ve iletim maliyeti 2^n*n olmaktadır.

Çizelge 4.7 ve Çizelge 4.8, kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan işlem maliyetini ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.4'deki gibidir. Kullanıcı çıkarma işlemleri işlem maliyeti açısından değerlendirildiğinde, TMAD şeması MAH, TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir.

$$Islem_{maliyeti_{KC}} = \begin{cases} 2^{n-1} * (1+2n) & \text{MAH, TFA, TFZ için} \\ 2^{n-1} * (2+3n) & \text{AGDH, DÖGİ için} \\ 2^{n-1} * (3+2n) & \text{TMAD şeması için} \\ 3 * (2^n - 2 + 2^{n-1} - 1) * 2^{n-2} & \text{SISA şeması için} \\ 2^{n-1} * ((2+4n) + (2^{n-1} - 1)) & \text{MHTGG şeması için} \end{cases} \tag{4.6}$$

Denklem 4.6'da TMAD şemasında her bir 2^n kullanıcı çıkarma için gerçekleştirilen işlem maliyetinin formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 3 değeri her bir kullanıcı çıkarma işleminde kullanıcıda bulunan açık anahtar, gizli anahtar ve simetrik değerin şemadan silinmesi durumunda ortaya çıkar. Aşağıdan yukarıya her adımda işlem maliyeti bir önceki adımın işlem maliyeti üzerine ekleme yapılarak hesaplandığından, şemadan 2^n kullanıcı çıkarma işlemi şemadan ayrılmış 2^{n-1} kullanıcının ardından sırasıyla 2^{n-1} kullanıcı daha çıkartarak gerçekleştirilir. Bu nedenle 3 değeri 2^{n-1} ile çarpma işlemine tutulur. $2^{n-1}(2n)$ ise kullanıcı çıkarma işleminin ardından güncellenen ve iletilen aradüğüm anahtar sayısıdır.

İşlem maliyeti bir şemada anahtar üretim ve iletim süreçlerinin tamamıdır. Şemalarda üretilen anahtar sayılarının birbirlerinden farklı olması iletilen anahtar sayılarının da farklı olmasına neden olur. TMAD şemasının kullanıcı ekleme ve çıkarma işlemlerinde işlem maliyeti açısından MAH, TFA ve TFZ şemalarının ardında yer almasının nedeni MAH, TFA ve TFZ şemalarında anahtar güncellemesi gerektiren kullanıcı işlemlerinde üretilen ve güncellenen anahtar sayısının TMAD şemasına göre daha az olmasıdır. Örneğin kullanıcı ekleme işlemi için MAH, TFA ve TFZ şemalarında kullanıcının kendisine ait bir anahtarı bulunurken, TMAD şemasında bir kullanıcının kendisine ait üç anahtarı bulunur.

TMAD şemasının kullanıcı ekleme ve çıkarma işlemlerinde işlem maliyeti açısından AGDH ve DÖGİ şemalarından daha iyi sonuç vermesinin nedeni, anahtar

güncellemelerinde AGDH ve DÖGİ şemalarında her aradüğümde açık anahtar altyapısı ile oluşturulmuş iki anahtarın güncellenmesi gerekirken, TMAD şemasında aradüğümlerde güncellenecek yalnızca bir adet simetrik anahtar değerinin bulunmasıdır. TMAD şemasında aradüğüm simetrik anahtar değerlerinin üretimi ya da güncellenmesi dağıtık şemalara göre daha az işlem gerektirmektedir.

Çizelge 4.11 ve Çizelge 4.12, toplu kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan işlem maliyetini ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.6'deki gibidir. Toplu kullanıcı ekleme işlemleri işlem maliyeti açısından değerlendirildiğinde, TMAD şeması sırasıyla MHTGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. İlk sırada MHTGG şeması yer almaktadır. Bunun nedeni MHTGG şemasına toplu olarak eklenecek 2^n kullanıcı için bir halka topolojisinin oluşturulması, kullanıcıların birbirlerine göre önceki-sonraki adresleri oluşturularak bir halka sıra düzenin oluşturulmasıdır. Bu düzen ikili ağaç yapısına sahip şemaların oluşumu için daha az işlem maliyeti gerektirmektedir. İkinci ve üçüncü sırada bulunan MAH, TFA ve TFZ şemaları ile AGDH ve DÖGi şemalarının sonuçları birbirlerine eşittir. TMAD şemasının bu şemaların ardından dördüncü sırada yer almasının nedeni kullanıcılarda üretilmesi ve iletilmesi gereken anahtar sayısının fazla olmasıdır. Ayrıca toplu kullanıcı ekleme işleminde güncellenip iletilmesi gereken aradüğüm anahtarı sayısı kullanıcı ekleme işlemine göre çok az sayıdadır. TMAD şemasının ardından SISA şeması gelmektedir.

$$Islem_{maliyeti_{TKE}} = \begin{cases} 2*2^n + 3*(2^n - 1) & \text{MAH, TFA, TFZ şemaları için} \\ 4*2^n + 4*(2^n - 1) & \text{AGDH, DÖGİ şemaları için} \\ 6*2^n + 3*(2^n - 1) & \text{TMAD şeması için} \\ 4*2^n + 10*(2^n - 1) & \text{SISA şeması için} \\ 2*(n+1) + 2 & \text{MHTGG şeması için} \end{cases} \tag{4.7}$$

Denklem 4.7'de TMAD şemasında her toplu 2^n kullanıcı ekleme işleminin ardından ortaya çıkan işlem maliyetinin formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 6 değeri her bir kullanıcı ekleme işleminde üretilmesi ve iletilmesi gereken açık anahtar, gizli anahtar

ve simetrik değeri ifade etmektedir. $3*(2^n-1)$ değeri aradüğüm güncellemeleri sonucu ortaya çıkan işlem maliyetidir.

Çizelge 4.15 ve Çizelge 4.16, toplu kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan işlem maliyetini ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.8'deki gibidir. Toplu kullanıcı çıkarma işlemleri işlem maliyeti açısından değerlendirildiğinde, TMAD şeması en iyi dördüncü performans sonucuna sahiptir. İlk sırada MHTGG şeması, ikinci sırada MAH,TFA ve TFZ şemaları yer almaktadır. Bunun nedeni MHTGG şemasına toplu olarak çıkartılacak 2^n kullanıcının bir halka topolojisi içerisinde yer almasıdır. Halka topolojisi toplu kullanıcı çıkarma işleminde ikili ağaç yapısına sahip şemalara kıyasla daha az işlem maliyeti gerektirmektedir. Toplu kullanıcı çıkarma işleminde kullanıcılarda silinmesi gereken anahtar sayısının daha az olması MAH, TFA ve TFZ şemalarının TMAD şemasına göre daha iyi sonuç vermesine neden olmaktadır. AGDH ve DÖGİ şemaları üçüncü sırada yer alsa dördüncü sırada bulunan TMAD şemasıyla yaklaşık sonuca sahip olmasının nedeni kullanıcılarda silinmesi gereken anahtar sayısı daha az olsa bile, güncellenmesi gereken aradüğüm sayısının daha fazla olmasıdır.

$$Islem_{maliyeti_{TKC}} = \begin{cases} 2^n + 3*(2^n-1) & \text{MAH, TFA, TFZ şemaları için} \\ 2*2^n + 4*(2^n-1) & \text{AGDH, DÖGİ şemaları için} \\ 3*2^n + 3*(2^n-1) & \text{TMAD şeması için} \\ 2*2^n + 6*(2^n-1) & \text{SISA şeması için} \\ 2*(n+1)+2 & \text{MHTGG şeması için} \end{cases} \tag{4.8}$$

Denklem 4.8'de TMAD şemasında her toplu 2^n kullanıcı çıkarma işleminin ardından ortaya çıkan işlem maliyetinin formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 3 değeri her bir kullanıcı çıkarma işleminde kullanıcıda bulunan açık anahtar, gizli anahtar ve simetrik değerin şemadan silinmesi durumunda ortaya çıkmaktadır. Bu nedenle $3*2^n$ değeri 2^n kullanıcıyı toplu olarak çıkarma işleminin ardından silinen anahtar sayısını ifade eder. $3*(2^n-1)$ ise güncellenen ve iletilen aradüğüm anahtar sayısıdır.

Çizelge 4.17, 2^n kullanıcılı şemalarda kullanıcılarda bulunan anahtar sayısı ve boyutunu ifade eder. İlgili çizelgenin grafiksel gösterimi Şekil 4.9'deki gibidir. Kullanıcılarda

Çizelge 4.17. Kullanıcılarda Bulunan Anahtar Sayıları ve Boyutları.

	MAH		TFA		TFZ		AGDH		TMAD		DÖGİ		SISA		MHTGG	
K.S.	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut	Sayı	Boyut
2^0	2	48	2	36	2	36	3	300	4	240	3	300	3	300	1	104
2^1	2	48	2	36	2	36	3	300	4	240	3	300	3	300	1	104
2^2	3	72	3	48	3	48	4	404	5	250	4	404	5	508	1	104
2^3	4	96	4	60	4	60	5	508	6	260	5	508	9	924	1	104
2^4	5	120	5	72	5	72	6	612	7	270	6	612	17	1756	1	104
2^5	6	144	6	84	6	84	7	716	8	280	7	716	33	3420	1	104
2^6	7	168	7	96	7	96	8	820	9	290	8	820	65	6748	1	104
2^7	8	192	8	108	8	108	9	924	10	300	9	924	129	13404	1	104
2^8	9	216	9	120	9	120	10	1028	11	310	10	1028	257	26716	1	104
2^9	10	240	10	132	10	132	11	1132	12	320	11	1132	513	53340	1	104
2^{10}	11	264	11	144	11	144	12	1236	13	330	12	1236	1025	106588	1	104
2^{11}	12	288	12	156	12	156	13	1340	14	340	13	1340	2049	213084	1	104
2^{12}	13	312	13	168	13	168	14	1444	15	350	14	1444	4097	426076	1	104
2^{13}	14	336	14	180	14	180	15	1548	16	360	15	1548	8193	852060	1	104
2^{14}	15	360	15	192	15	192	16	1652	17	370	16	1652	16385	1704028	1	104
2^{15}	16	384	16	204	16	204	17	1756	18	380	17	1756	32769	3407964	1	104
2^{16}	17	408	17	216	17	216	18	1860	19	390	18	1860	65537	6815836	1	104
2^{17}	18	432	18	228	18	228	19	1964	20	400	19	1964	131073	13631580	1	104
2^{18}	19	456	19	240	19	240	20	2068	21	410	20	2068	262145	27263068	1	104
2^{19}	20	480	20	252	20	252	21	2172	22	420	21	2172	524289	54526044	1	104
2^{20}	21	504	21	264	21	264	22	2276	23	430	22	2276	1048577	109051996	1	104
2^{21}	22	528	22	276	22	276	23	2380	24	440	23	2380	2097153	218103900	1	104

bulunan anahtar sayısı bakımından TMAD şeması sırasıyla, MHTGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından dördüncü sırada yer almaktadır. En kötü sonucu ise TMAD şemasının ardından gelen SISA şeması vermektedir. MHTGG şeması halka topolojisine sahip olduğundan GY'den çıkan bir anahtar sırasıyla kullanıcıları dolaşarak GY'e geri döner. Kullanıcılar güncel anahtarları bu şekilde öğrenir ve her kullanıcıda yalnızca bir anahtar bulunur. İkinci sırada yer alan merkezi şemalarda gizli anahtarlı şifreleme yöntemiyle oluşturulmuş anahtarlar, üçüncü sırada yer alan dağıtık şemalarda ise açık anahtarlı anahtar dağıtım yöntemiyle oluşturulmuş anahtarlar saklanır. TMAD şemasında hem gizli anahtarlı şifreleme yöntemiyle oluşturulmuş bir anahtar hem de açık anahtarlı anahtar dağıtım yöntemiyle oluşturulmuş anahtar çifti saklanmaktadır. Bu durum şemanın kullanıcılarda bulunan anahtar sayısı bakımından dördüncü sırada yer almasına neden olmaktadır.

$$Kullanici_{anahtar_{sayisi}} = \begin{cases} n+1 & \text{MAH, TFA, TFZ şemaları için} \\ n+2 & \text{AGDH, DÖGİ şemaları için} \\ n+3 & \text{TMAD şeması için} \\ 2^n+1 & \text{SISA şeması için} \\ 1 & \text{MHTGG şeması için} \end{cases} \tag{4.9}$$

Denklem 4.9'da TMAD şemasında her bir 2^n kullanıcı için kullanıcıda bulunan anahtar sayısının formülü gösterilmektedir. Kullanıcının kendisinde 3 adet anahtar bulunur.

Kendisinden kök düğüme kadarki yol üzerinde bulunan düğüm sayısı ise n'dir. TMAD şeması kullanıcılarda bulunan anahtarların boyutu açısından değerlendirildiğinde, MHTGG şeması ile TFA ve TFZ şemalarının ardından üçüncü sırada yer almaktadır. MHTGG şemasında kullanıcılarda yalnızca bir anahtar bulunması anahtar boyutu açısından da en iyi sonucu vermesini sağlamaktadır. İkinci sırada anahtar güncelleme işlemi TMAD şemasına benzer olan TFA ve TFZ şemaları bulunur. TFZ ve TFZ şemalarında her bir işlemde üst düğüme iletilen aradüğüm anahtarının boyutu TMAD şemasından daha büyük olsa da kullanıcıların kendilerine ait olan anahtar boyutunun daha küçük olması TMAD şemasına göre daha iyi sonuç vermesine neden olmaktadır.

$$Kullanici_{anahtar_{boyutu}} = \begin{cases} 24 * (n+1) & \text{MAH şeması için} \\ 24 + n * 12 & \text{TFA, TFZ şemaları için} \\ 196 + n * 104 & \text{AGDH, DÖGİ şemaları için} \\ 240 + (n-1) * 10 & \text{TMAD şeması için} \\ 196 + (2^n - 1) * 104 & \text{SISA şeması için} \\ 104 & \text{MHTGG şeması için} \end{cases} \tag{4.10}$$

Denklem 4.10'da TMAD şemasında her bir 2^n kullanıcı için kullanıcıda bulunan anahtar boyutunun formülü bayt cinsinden gösterilmektedir. 240 bayt değeri bir kullanıcının açık anahtarı, gizli anahtarı, simetrik anahtar değeri ile kök düğüm simetrik anahtarının toplam boyutudur. $n-1$ işlemi kök düğüm dışında (bu nedenle 1 çıkartılır) kullanıcıdan kök düğüme kadarki yol üzerinde bulunan aradüğüm sayısını ifade etmektedir.

Çizelge 4.18 ve Çizelge 4.19, kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim boyutunu ifade eder.

Çizelge 4.20 ve Çizelge 4.21, kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim boyutunu ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.10'deki gibidir.

Kullanıcı ekleme işlemi anahtar iletim boyutu açısından incelendiğinde, TMAD şeması TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir. TMAD

Çizelge 4.18. Kullanıcı Ekleme – Anahtar İletim Boyutları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^0	2	48	2	36	2	36	3	300
2^1	4	96	4	72	4	72	6	600
2^2	10	240	10	168	10	168	14	1108
2^3	26	624	26	408	26	408	34	2840
2^4	66	1584	66	984	66	984	82	6928
2^5	162	3888	162	2328	162	2328	194	16352
2^6	386	9264	386	5400	386	5400	450	37696
2^7	898	21552	898	12312	898	12312	1026	85376
2^8	2050	49200	2050	27672	2050	27672	2306	190720
2^9	4610	110640	4610	61464	4610	61464	5122	421376
2^{10}	10242	245808	10242	135192	10242	135192	11266	922624
2^{11}	22530	540720	22530	294936	22530	294936	24578	2004992
2^{12}	49154	1179696	49154	639000	49154	639000	53250	4329472
2^{13}	106498	2555952	106498	1376280	106498	1376280	114690	9297920
2^{14}	229378	5505072	229378	2949144	229378	2949144	245762	19873792
2^{15}	491522	11796528	491522	6291480	491522	6291480	524290	42303488
2^{16}	1048578	25165872	1048578	13369368	1048578	13369368	1114114	89718784
2^{17}	2228226	53477424	2228226	28311576	2228226	28311576	2359298	189661184
2^{18}	4718594	113246256	4718594	59768856	4718594	59768856	4980738	399769600
2^{19}	9961474	239075376	9961474	125829144	9961474	125829144	10485762	840433664
2^{20}	20971522	503316528	20971522	264241176	20971522	264241176	22020098	1762656256
2^{21}	44040194	1056964656	44040194	553648152	44040194	553648152	46137346	3688890368

Çizelge 4.19. Kullanıcı Ekleme – Anahtar İletim Boyutları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^0	4	240	3	300	5	612	2	300
2^1	8	466	6	600	9	1120	5	704
2^2	18	938	14	1108	23	2760	14	1824
2^3	42	1922	34	2840	75	8536	44	5312
2^4	98	3970	82	6928	275	30072	152	17280
2^5	226	8226	194	16352	1059	113080	560	61184
2^6	514	17058	450	37696	4163	438840	2144	228864
2^7	1154	35362	1026	85376	16515	1729336	8384	883712
2^8	2562	73250	2306	190720	65795	6866232	33152	3471360
2^9	5634	151586	5122	421376	262659	27363640	131840	13758464
2^{10}	12290	313378	11266	922624	1049603	109252920	525824	54779904
2^{11}	26626	647202	24578	2004992	4196355	436609336	2100224	218611712
2^{12}	57346	1335330	53250	4329472	16781315	1745633592	8394752	873431040
2^{13}	122882	2752546	114690	9297920	67117059	6980927800	33566720	3491692544
2^{14}	262146	5668898	245762	19873792	268451843	27920499000	134242304	13962706944
2^{15}	557058	11665442	524290	42303488	1073774595	111675572536	536920064	55842701312
2^{16}	1179650	23986210	1114114	89718784	4295032835	446689444152	2147581952	223354552320
2^{17}	2490370	49283106	2359298	189661184	17180000259	1786732085560	8590131200	893385703424
2^{18}	5242882	101187618	4980738	399769600	68719738883	7146876961080	34360131584	3573477801984
2^{19}	11010050	207618082	10485762	840433664	274878431235	28587405082936	137439739904	14293781184512
2^{20}	23068674	425721890	22020098	1762656256	1099512676355	114349414809912	549757386752	57174864691200
2^{21}	48234498	872415266	46137346	3688890368	4398048608259	457397248196920	2199026401280	228698938671104

Çizelge 4.20. Kullanıcı Çıkarma – Anahtar İletim Boyutları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^{21}	41943042	1006633008	41943042	503316504	41943042	503316504	41943042	4362076368
2^{20}	19922946	478150704	19922946	239075352	19922946	239075352	19922946	2071986384
2^{19}	9437186	226492464	9437186	113246232	9437186	113246232	9437186	981467344
2^{18}	4456450	106954800	4456450	53477400	4456450	53477400	4456450	463470800
2^{17}	2097154	50331696	2097154	25165848	2097154	25165848	2097154	218104016
2^{16}	983042	23593008	983042	11796504	983042	11796504	983042	102236368
2^{15}	458754	11010096	458754	5505048	458754	5505048	458754	47710416
2^{14}	212994	5111856	212994	2555928	212994	2555928	212994	22151376
2^{13}	98306	2359344	98306	1179672	98306	1179672	98306	10223824
2^{12}	45058	1081392	45058	540696	45058	540696	45058	4686032
2^{11}	20482	491568	20482	245784	20482	245784	20482	2130128
2^{10}	9218	221232	9218	110616	9218	110616	9218	958672
2^9	4098	98352	4098	49176	4098	49176	4098	426192
2^8	1794	43056	1794	21528	1794	21528	1794	186576
2^7	770	18480	770	9240	770	9240	770	80080
2^6	322	7728	322	3864	322	3864	322	33488
2^5	130	3120	130	1560	130	1560	130	13520
2^4	50	1200	50	600	50	600	50	5200
2^3	18	432	18	216	18	216	18	1872
2^2	6	144	6	72	6	72	6	624
2^1	2	48	2	24	2	24	2	208
2^0	1	24	1	12	1	12	1	104

Çizelge 4.21. Kullanıcı Çıkarma – Anahtar İletim Boyutları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^{21}	41943042	419430420	41943042	4362076368	4398040219651	457396182843704	2199024304128	228698527629312
2^{20}	19922946	199229460	19922946	2071986384	1099508482051	114348882133304	549756338176	57174659170304
2^{19}	9437186	94371860	9437186	981467344	274876334083	28587138744632	137439215616	14293678424064
2^{18}	4456450	44564500	4456450	463470800	68718690307	7146743791928	34359869440	3573426421760
2^{17}	2097154	20971540	2097154	218104016	17179475971	1786665500984	8590000128	893360013312
2^{16}	983042	9830420	983042	102236368	4294770691	446656151864	2147516416	223341707264
2^{15}	458754	4587540	458754	47710416	1073643523	111658926392	536887296	55836278784
2^{14}	212994	2129940	212994	22151376	268386307	27912175928	134225920	13959495680
2^{13}	98306	983060	98306	10223824	67084291	6976766264	33558528	3490086912
2^{12}	45058	450580	45058	4686032	16764931	1743552824	8390656	872628224
2^{11}	20482	204820	20482	2130128	4188163	435568952	2098176	218210304
2^{10}	9218	92180	9218	958672	1045507	108732728	524800	54579200
2^{9}	4098	40980	4098	426192	260611	27103544	131328	13658112
2^{8}	1794	17940	1794	186576	64771	6736184	32896	3421184
2^{7}	770	7700	770	80080	16003	1664312	8256	858624
2^{6}	322	3220	322	33488	3907	406328	2080	216320
2^{5}	130	1300	130	13520	931	96824	528	54912
2^{4}	50	500	50	5200	211	21944	136	14144
2^{3}	18	180	18	1872	43	4472	36	3744
2^{2}	6	60	6	624	7	728	10	1040
2^{1}	2	20	2	208	1	104	3	312
2^{0}	1	10	1	104	1	104	1	104

şemasında anahtar hesaplama işlemi TFA ve TFZ şemalarında olduğu gibi yapraklarda bulunan kullanıcılardan kök düğüme doğru gerçekleştirilir. Fakat bu şemalardan farklı olarak hesaplama işleminde, her bir aradüğüm anahtar değeri sol çocuğunun gizli anahtar değerinin sol yarısı ile sağ çocuğunun gizli anahtar değerinin sağ yarısının birleştirilmesinin ardından bir özetleme fonksiyonuna sokulmasıyla elde edilir. Yapraklardan üst düğümlere her bir adımda anahtar değerinin yarısının iletiliyor olması anahtar iletim boyutunun azalmasını sağlamaktadır. TMAD şemasının kullanıcı ekleme işleminde anahtar iletim boyutu açısından TFA ve TFZ şemalarının ardında yer almasının nedeni kullanıcılarda bulunan anahtar boyutlarının TFA ve TFZ şemalarına kıyasla TMAD şemasında daha büyük olmasıdır. TMAD şemasının ardından sırasıyla AGDH ve DÖGİ şemaları, MHTGG şeması ile SISA şeması gelmektedir.

$$Anahtar_{boyutu_{KE}} = \begin{cases} 2^{n-1} * 24 + 2^{n-1} * n * 24 & \text{MAH için} \\ 2^{n-1} * 24 + 2^{n-1} * n * 12 & \text{TFA, TFZ için} \\ 2^{n-1} * 196 + 2^{n-1} * n * 104 & \text{AGDH, DÖGİ} \\ 2^{n-1} * 216 + 2^{n-1} * n * 10 & \text{TMAD için} \\ 2^{n} * 196 + (2^{n} * (2^{n}+1) - 2^{n} + 3) * 104 & \text{SISA için} \\ 2^{n} * 196 + \frac{2^{n}(2^{n}+1)}{2} * 104 & \text{MHTGG için} \end{cases} \tag{4.11}$$

Denklem 4.11'de TMAD şemasında her 2^n kullanıcı ekleme işleminde iletimi yapılan anahtar boyutunun formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. n değeri ise

bir kullanıcıdan kök düğüme olan uzaklığı, bir diğer ifade ile ağacın derinliğini ifade etmektedir. 216 değeri bir kullanıcıda bulunan açık ve gizli anahtar ile simetrik anahtar değerinin bayt cinsinden toplamıdır. Çizelge 4.18 ve Çizelge 4.19'daki her bir adımdaki sonuç bir önceki adımın üzerine eklenerek elde edilmektedir. Çünkü ağaç üzerinde 2^n kullanıcı ekleme işlemi, şemada bulunan 2^{n-1} kullanıcıya sırasıyla 2^{n-1} kullanıcı daha eklenerek gerçekleştirilir. Bu nedenle 216 değeri 2^{n-1} ile çarpma işlemine tutulur. Ardından şemada bulunan aradüğüm sayısı olan $2^{n-1} * n$, her aradüğüm işleminde bir kullanıcıdan kök düğüme 10 baytlık simetrik değer iletildiğinden, 10 değeri ile çarpma işlemine tabi tutularak anahtar boyutu elde edilir.

Kullanıcı çıkarma işlemi anahtar iletim boyutu açısından incelendiğinde, TMAD şeması en iyi performans sonucuna sahiptir. Kullanıcı çıkarma işleminde güncellenen anahtarlar arasında şemadan çıkan kullanıcı anahtarlarının olmaması, yalnızca aradüğüm anahtarlarının güncellenmesi ve diğer şemalara kıyasla önerilen şemada aradüğüm anahtarlarının daha düşük anahtar boyutuyla güncellenmesi TMAD şemasının kullanıcı çıkarma işleminde anahtar iletim boyutu açısından en iyi sonucu vermesini sağlamaktadır.

$$Anahtar_{boyutu_{KC}} = \begin{cases} 2^{n-1} * n * 24 & \text{MAH şeması için} \\ 2^{n-1} * n * 12 & \text{TFA, TFZ şemaları için} \\ 2^{n-1} * n * 104 & \text{AGDH, DÖGİ şemaları için} \\ 2^{n-1} * n * 10 & \text{TMAD şeması için} \\ ((2^n - 1)(2^n - 2) + 1) * 104 & \text{SISA şeması için} \\ \frac{2^n(2^n+1)}{2} * 104 & \text{MHTGG şeması için} \end{cases} \tag{4.12}$$

Denklem 4.12'de TMAD şemasında her 2^n kullanıcı çıkarma işleminde iletimi yapılan anahtar boyutunun formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. n değeri ise bir kullanıcıdan kök düğüme olan uzaklığı, bir diğer ifade ile ağacın derinliğini ifade etmektedir. Kullanıcı çıkarma işleminde yalnızca aradüğüm anahtarları güncellendiğinden, her aradüğüm işleminde bir kullanıcıdan kök düğüme yalnızca 10 baytlık simetrik değer iletilir. Bu nedenle iletimi yapılan aradüğüm anahtar sayısı 10 değeri ile çapılarak iletimi yapılan anahtar boyutu elde edilir. Kullanıcı ekleme işleminde olduğu gibi her bir adımdaki sonuç bir önceki adımın üzerine eklenerek elde edilmektedir.

Çizelge 4.22. Toplu Kullanıcı Ekleme - Anahtar İletim Boyutları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^0	2	48	2	36	2	36	2	300
2^1	4	96	4	72	4	72	6	600
2^2	10	240	10	168	10	168	14	1408
2^3	22	528	22	360	22	360	30	3024
2^4	46	1104	46	744	46	744	62	6256
2^5	94	2256	94	1512	94	1512	126	12720
2^6	190	4560	190	3048	190	3048	254	25648
2^7	382	9168	382	6120	382	6120	510	51504
2^8	766	18384	766	12264	766	12264	1022	103216
2^9	1534	36816	1534	24552	1534	24552	2046	206640
2^{10}	3070	73680	3070	49128	3070	49128	4094	413488
2^{11}	6142	147408	6142	98280	6142	98280	8190	827184
2^{12}	12286	294864	12286	196584	12286	196584	16382	1654576
2^{13}	24574	589776	24574	393192	24574	393192	32766	3309360
2^{14}	49150	1179600	49150	786408	49150	786408	65534	6618928
2^{15}	98302	2359248	98302	1572840	98302	1572840	131070	13238064
2^{16}	196606	4718544	196606	3145704	196606	3145704	262142	26476336
2^{17}	393214	9437136	393214	6291432	393214	6291432	524286	52952880
2^{18}	786430	18874320	786430	12582888	786430	12582888	1048574	105905968
2^{19}	1572862	37748688	1572862	25165800	1572862	25165800	2097150	211812144
2^{20}	3145726	75497424	3145726	50331624	3145726	50331624	4194302	423624496
2^{21}	6291454	150994896	6291454	100663272	6291454	100663272	8388606	847249200

Çizelge 4.23. Toplu Kullanıcı Ekleme - Anahtar İletim Boyutları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^0	3	240	2	300	2	196	2	208
2^1	8	466	6	600	12	1224	3	312
2^2	18	938	14	1408	32	3280	5	520
2^3	38	1882	30	3024	72	7392	9	936
2^4	78	3770	62	6256	152	15616	17	1768
2^5	158	7546	126	12720	312	32064	33	3432
2^6	318	15098	254	25648	632	64960	65	6760
2^7	638	30202	510	51504	1272	130752	129	13416
2^8	1278	60410	1022	103216	2552	262336	257	26728
2^9	2558	120826	2046	206640	5112	525504	513	53352
2^{10}	5118	241658	4094	413488	10232	1051840	1025	106600
2^{11}	10238	483322	8190	827184	20472	2104512	2049	213096
2^{12}	20478	966650	16382	1654576	40952	4209856	4097	426088
2^{13}	40958	1933306	32766	3309360	81912	8420544	8193	852072
2^{14}	81918	3866618	65534	6618928	163832	16841920	16385	1704040
2^{15}	163838	7733242	131070	13238064	327672	33684672	32769	3407976
2^{16}	327678	15466490	262142	26476336	655352	67370176	65537	6815848
2^{17}	655358	30932986	524286	52952880	1310712	134741184	131073	13631592
2^{18}	1310718	61865978	1048574	105905968	2621432	269483200	262145	27263080
2^{19}	2621438	123731962	2097150	211812144	5248872	539591232	524289	54526056
2^{20}	5242878	247463930	4194302	423624496	10485752	1077935296	1048577	109052008
2^{21}	10485758	494927866	8388606	847249200	20971512	2155871424	2097153	218103912

Çizelge 4.24. Toplu Kullanıcı Çıkarma - Anahtar İletim Boyutları (a).

	MAH		TFA		TFZ		AGDH	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^{21}	4194302	100663248	4194302	50331624	4194302	50331624	4194302	436207408
2^{20}	2097150	50331600	2097150	25165800	2097150	25165800	2097150	218103600
2^{19}	1048574	25165776	1048574	12582888	1048574	12582888	1048574	109051696
2^{18}	524286	12582864	524286	6291432	524286	6291432	524286	54525744
2^{17}	262142	6291408	262142	3145704	262142	3145704	262142	27262768
2^{16}	131070	3145680	131070	1572840	131070	1572840	131070	13631280
2^{15}	65534	1572816	65534	786408	65534	786408	65534	6815536
2^{14}	32766	786384	32766	393192	32766	393192	32766	3407664
2^{13}	16382	393168	16382	196584	16382	196584	16382	1703728
2^{12}	8190	196560	8190	98280	8190	98280	8190	851760
2^{11}	4094	98256	4094	49128	4094	49128	4094	425776
2^{10}	2046	49104	2046	24552	2046	24552	2046	212784
2^9	1022	24528	1022	12264	1022	12264	1022	106288
2^8	510	12240	510	6120	510	6120	510	53040
2^7	254	6096	254	3048	254	3048	254	26416
2^6	126	3024	126	1512	126	1512	126	13104
2^5	62	1488	62	744	62	744	62	6448
2^4	30	720	30	360	30	360	30	3120
2^3	14	336	14	168	14	168	14	1456
2^2	6	144	6	72	6	72	6	624
2^1	2	48	2	24	2	24	2	208
2^0	0	0	0	0	0	0	0	0

Çizelge 4.25. Toplu Kullanıcı Çıkarma - Anahtar İletim Boyutları (b).

	TMAD		DÖGİ		SISA		MHTGG	
K.S.	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut	İletim Sayısı	Boyut
2^{21}	4194302	41943020	4194302	436207408	8388604	872414816	1048577	109052008
2^{20}	2097150	20971500	2097150	218103600	4194300	436207200	524289	54526056
2^{19}	1048574	10485740	1048574	109051696	2097148	218103392	262145	27263080
2^{18}	524286	5242860	524286	54525744	1048572	109051488	131073	13631592
2^{17}	262142	2621420	262142	27262768	524284	54525536	65537	6815848
2^{16}	131070	1310700	131070	13631280	262140	27262560	32769	3407976
2^{15}	65534	655340	65534	6815536	131068	13631072	16385	1704040
2^{14}	32766	327660	32766	3407664	65532	6815328	8193	852072
2^{13}	16382	163820	16382	1703728	32764	3407456	4097	426088
2^{12}	8190	81900	8190	851760	16380	1703520	2049	213096
2^{11}	4094	40940	4094	425776	8188	851552	1025	106600
2^{10}	2046	20460	2046	212784	4092	425568	513	53352
2^{9}	1022	10220	1022	106288	2044	212576	257	26728
2^{8}	510	5100	510	53040	1020	106080	129	13416
2^{7}	254	2540	254	26416	508	52832	65	6760
2^{6}	126	1260	126	13104	252	26208	33	3432
2^{5}	62	620	62	6448	124	12896	17	1768
2^{4}	30	300	30	3120	60	6240	9	936
2^{3}	14	140	14	1456	28	2912	5	520
2^{2}	6	60	6	624	12	1248	3	312
2^{1}	2	20	2	208	4	416	1	104
2^{0}	0	0	0	0	0	0	0	0

Çizelge 4.22, Çizelge 4.23, toplu kullanıcı ekleme işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim boyutunu ifade eder. Çizelge 4.24 ve Çizelge 4.25, toplu kullanıcı çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan toplam anahtar iletim boyutunu ifade eder. İlgili çizelgelerin grafiksel gösterimi Şekil 4.11'deki gibidir.

Toplu kullanıcı ekleme işlemi anahtar iletim boyutu açısından incelendiğinde TMAD şeması sırasıyla, TFA ve TFZ şemaları, MAH şeması ve MHTGG şemasının ardından en iyi dördüncü performans sonucuna sahiptir. Toplu kullanıcı ekleme işleminde kullanıcı ekleme işlemine kıyasla az sayıda aradüğüm anahtarının güncellenmesi TMAD şemasının anahtar iletim boyutu açısından TFA ve TFZ ile MAH şemalarının ardında yer almasına neden olmaktadır. TMAD şemasının MHTGG şemasının ardında yer almasının nedeni ise, MHTGG şemasının topolojik yapısı nedeniyle toplu kullanıcı ekleme işleminde gerçekleşen anahtar iletim sayısının TMAD şemasına oranla daha küçük olmasıdır.

Denklem 4.13'de TMAD şemasında her toplu 2^n kullanıcı ekleme işleminde iletimi yapılan anahtar boyutunun formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. 216 değeri bir kullanıcıda bulunan açık ve gizli anahtar ile simetrik anahtar değerinin bayt cinsinden toplamıdır. Bu değer şemaya toplu olarak eklenen 2^n kullanıcı ile çarpılır. Ardından şemada $2 * (2^n - 1)$ aradüğüm güncelleme işlemi her bir adımda 10 baytlık anahtar kullanılarak elde edilir.

$$Anahtar_{boyutu_{TKE}} = \begin{cases} 2^n * 24 + (2^n - 1) * 48 & \text{MAH şeması için} \\ 2^n * 24 + (2^n - 1) * 24 & \text{TFA, TFZ şemaları için} \\ 2^n * 196 + (2 * (2^n - 1)) * 104 & \text{AGDH, DÖGİ için} \\ 2^n * 216 + 2 * (2^n - 1) * 10 & \text{TMAD şeması için} \\ 2^n * 196 + 8 * (2^n - 1) * 104 & \text{SISA şeması için} \\ (2^n + 1) * 104 & \text{MHTGG şeması için} \end{cases} \quad (4.13)$$

AGDH ve DÖGİ şemaları ile SISA şeması toplu kullanıcı ekleme işleminde anahtar iletim boyutu açısından TMAD şemasının ardında yer almaktadırlar.

Toplu kullanıcı çıkarma işlemi anahtar iletim boyutu açısından incelendiğinde TMAD şeması, en iyi performans sonucuna sahiptir. Toplu kullanıcı çıkarma işleminde güncellenen anahtarlar arasında şemadan çıkan kullanıcı anahtarlarının olmaması, yalnızca aradüğüm anahtarlarının güncellenmesi ve diğer şemalara kıyasla önerilen şemada aradüğüm anahtarlarının daha düşük anahtar boyutuyla güncellenmesi TMAD şemasının toplu kullanıcı çıkarma işleminde anahtar iletim boyutu açısından en iyi sonucu vermesini sağlamaktadır.

Denklem 4.14'de TMAD şemasında her toplu 2^n kullanıcı çıkarma işleminde iletimi yapılan anahtar boyutunun formülü gösterilmektedir. 2^n değeri kullanıcı sayısıdır. $2 * (2^n - 1)$ aradüğüm güncelleme işlemi, şemadaki aradüğüm güncellemelerinin her bir adımda 10 baytlık anahtar iletimiyle yapılması sonucu ortaya çıkmaktadır.

$$Anahtar_{boyutu_{TKC}} = \begin{cases} 2 * (2^n - 1) * 24 & \text{MAH şeması için} \\ 2 * (2^n - 1) * 13 & \text{TFA, TFZ şemaları için} \\ 2 * (2^n - 1) * 104 & \text{AGDH, DÖGİ şemaları için} \\ 2 * (2^n - 1) * 10 & \text{TMAD şeması için} \\ 4 * (2^n - 1) * 104 & \text{SISA şeması için} \\ (2^n + 1) * 104 & \text{MHTGG şeması için} \end{cases} \quad (4.14)$$

TMAD şemasının hesaplamalarında EEDH-112 protokolünden yararlanılmıştır. Bir diğer ifade ile hesaplamalarda kullanılan EEDH protokolünün anahtar boyutu 112 bittir. Sonuçlar 112 bitlik anahtar boyutuna göre elde edilmiştir. Fakat aynı protokolü kullanarak farklı anahtar boyutları ile hesaplamalar yapmak ve sonuçlar elde etmek mümkündür. Ayrıca farklı bir anahtar dağıtım ya da şifreleme yöntemi de kullanılabilir.

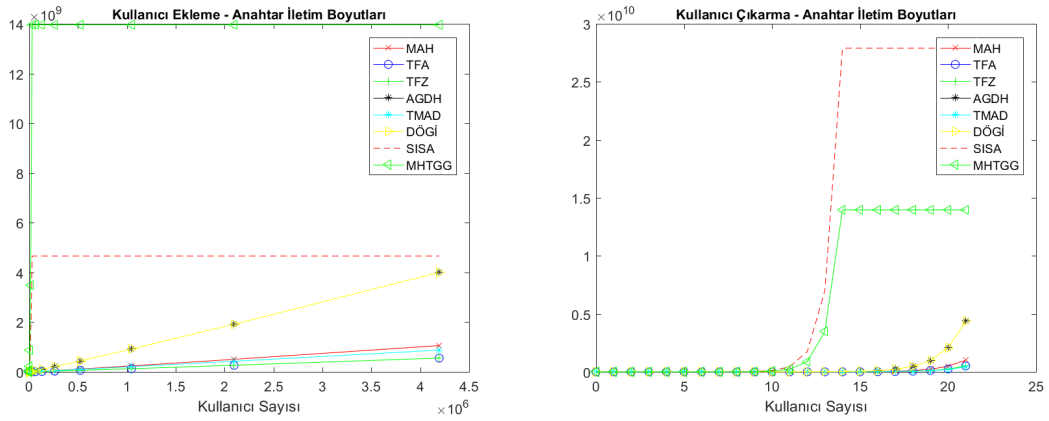
Çizelge 4.26, Çizelge 4.27, Çizelge 4.28 ve Çizelge 4.29 sırasıyla kullanıcı ekleme, kullanıcı çıkarma, toplu kullanıcı ekleme ve toplu kullanıcı çıkarma işlemleri için TMAD şemasının iki farklı yöntem ve farklı anahtar boyutları ile anahtarlama hesaplamalarının sonucu göstermektedir. Anahtarlama hesaplamaları, bir diğer ifade ile şemada 2^n kullanıcı için toplam anahtar iletim boyutlarının belirlenmesi, önceden hesaplanmış iletim sayılarının anahtar boyutuyla çarpma işlemine tabi tutularak elde edilmektedir.

Hesaplamalar EEDH protokolü ve RSA yönteminin farklı anahtar boyutları kullanılarak sırasıyla EEDH-112, EEDH-160, EEDH-256, EEDH-384, RSA-512, RSA-1024, RSA-2048, RSA-3072 ve RSA-7680 ile elde edilmiştir. Şekil 4.12 ve Şekil 4.13'de ilgili çizelgelerin grafiksel gösterimi yer almaktadır.

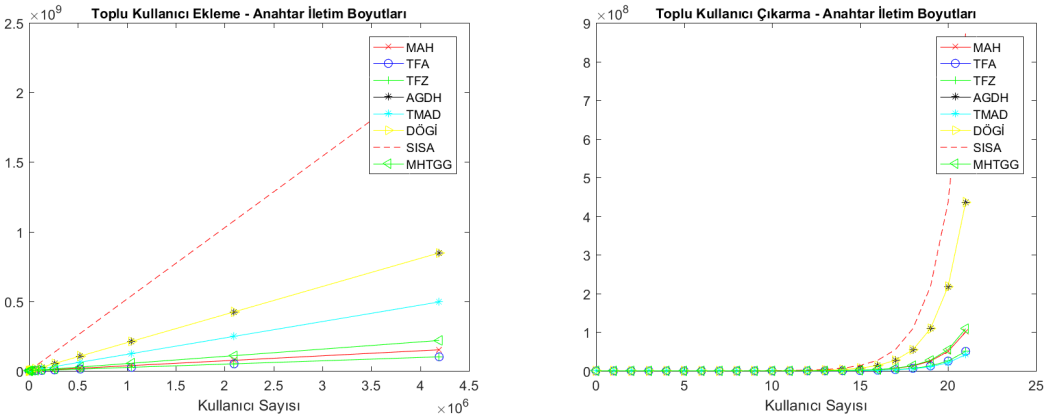
Çizelge 4.26. TMAD Şeması – Kullanıcı Ekleme – Anahtar İletim Boyutları.

		TMAD								
K.S.	İletim Sayısı	EEDH-112	EEDH-160	EEDH-256	EEDH-384	RSA-512	RSA-1024	RSA-2048	RSA-3072	RSA-7680
2^0	4	226	262	346	430	985	1342	2058	2762	5935
2^1	8	266	302	386	470	1025	1382	2098	2802	5975
2^2	18	366	402	486	570	1125	1482	2198	2902	6075
2^3	42	606	642	726	810	1365	1722	2438	3142	6315
2^4	98	1166	1202	1286	1370	1925	2282	2998	3702	6875
2^5	226	2446	2482	2566	2650	3205	3562	4278	4982	8155
2^6	514	5326	5362	5446	5530	6085	6442	7158	7862	11035
2^7	1154	11726	11762	11846	11930	12485	12842	13558	14262	17435
2^8	2562	25806	25842	25926	26010	26565	26922	27638	28342	31515
2^9	5634	56526	56562	56646	56730	57285	57642	58358	59062	62235
2^{10}	12290	123086	123122	123206	123290	123845	124202	124918	125622	128795
2^{11}	26626	266446	266482	266566	266650	267205	267562	268278	268982	272155
2^{12}	57346	573646	573682	573766	573850	574405	574762	575478	576182	579355
2^{13}	122882	1229006	1229042	1229126	1229210	1229765	1230122	1230838	1231542	1234715
2^{14}	262146	2621646	2621682	2621766	2621850	2622405	2622762	2623478	2624182	2627355
2^{15}	557058	5570766	5570802	5570886	5570970	5571525	5571882	5572598	5573302	5576475
2^{16}	1179650	11796686	11796722	11796806	11796890	11797445	11797802	11798518	11799222	11802395
2^{17}	2490370	24903886	24903922	24904006	24904090	24904645	24905002	24905718	24906422	24909595
2^{18}	5242882	52429006	52429042	52429126	52429210	52429765	52430122	52430838	52431542	52434715
2^{19}	11010050	110100686	110100722	110100806	110100890	110101445	110101802	110102518	110103222	110106395
2^{20}	23068674	230686926	230686962	230687046	230687130	230687685	230688042	230688758	230689462	230692635
2^{21}	48234498	482345166	482345202	482345286	482345370	482345925	482346282	482346998	482347702	482350875

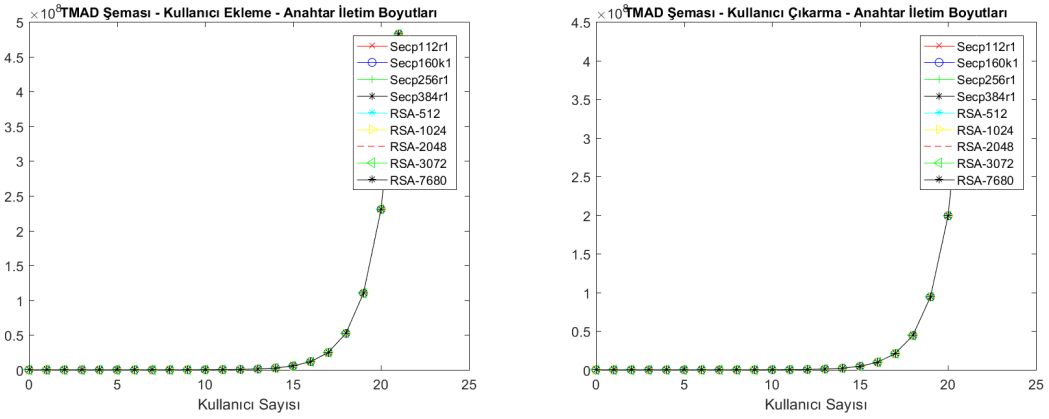
Çizelge 4.30, 2^n kullanıcı için EEDH protokolü ve RSA yönteminin farklı anahtar boyutları kullanıldığında bir kullanıcıda bulunması gereken toplam anahtar boyutunu ifade eder. İlgili çizelgenin grafiksel gösterimi Şekil 4.14'deki gibidir. Çizelge 4.31, TMAD şemasının kullanıcı ekleme ve çıkarma işlemlerinin ardından gerçekleştirilen anahtar güncellemelerinin sonucu olarak ortaya çıkan işlem maliyetini ifade eder.



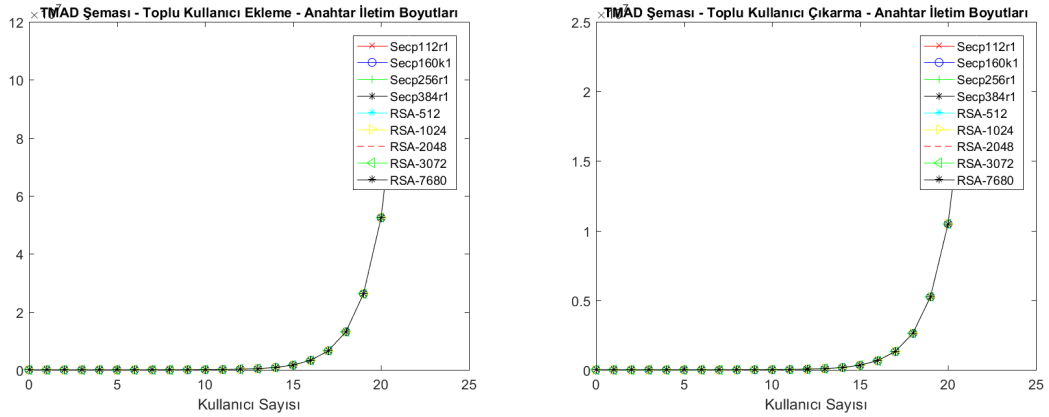
Şekil 4.10. Kullanıcı Ekleme ve Çıkarma – Anahtar İletim Boyutları.



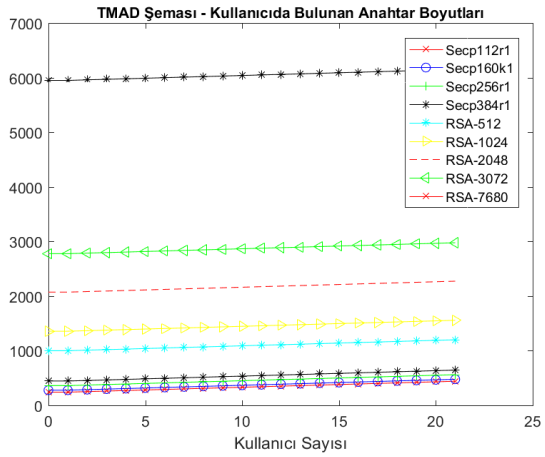
Şekil 4.11. Toplu Kullanıcı Ekleme ve Çıkarma – Anahtar İletim Boyutları.



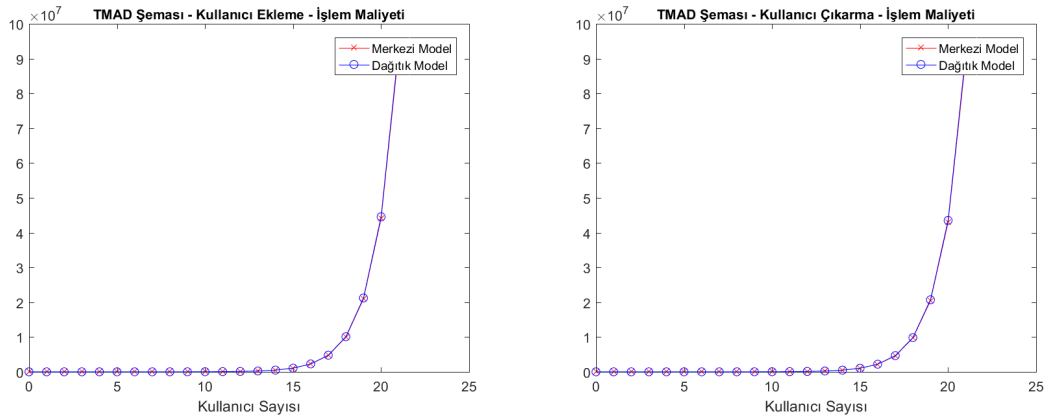
Şekil 4.12. TMAD Şeması - Kullanıcı Ekleme ve Çıkarma – Anahtar İletim Boyutları.



Şekil 4.13. TMAD Şeması - Toplu Kullanıcı Ekleme ve Çıkarma – Anahtar İletim Boyutları.



Şekil 4.14. TMAD Şeması – Kullanıcıda Bulunan Anahtar Boyutları.



Şekil 4.15. TMAD Şemasının Merkezi - Dağıtık Model Karşılaştırması.

Çizelge 4.27. TMAD Şeması – Kullanıcı Çıkarma – Anahtar İletim Boyutları.

		TMAD								
K.S.	İletim Sayısı	EEDH-112	EEDH-160	EEDH-256	EEDH-384	RSA-512	RSA-1024	RSA-2048	RSA-3072	RSA-7680
2^{21}	41943042	419430606	419430642	419430726	419430810	419431365	419431722	419432438	419433142	419436315
2^{20}	19922946	199229646	199229682	199229766	199229850	199230405	199230762	199231478	199232182	199235355
2^{19}	9437186	94372046	94372082	94372166	94372250	94372805	94373162	94373878	94374582	94377755
2^{18}	4456450	44564686	44564722	44564806	44564890	44565445	44565802	44566518	44567222	44570395
2^{17}	2097154	20971726	20971762	20971846	20971930	20972485	20972842	20973558	20974262	20977435
2^{16}	983042	9830606	9830642	9830726	9830810	9831365	9831722	9832438	9833142	9836315
2^{15}	458754	4587726	4587762	4587846	4587930	4588485	4588842	4589558	4590262	4593435
2^{14}	212994	2130126	2130162	2130246	2130330	2130885	2131242	2131958	2132662	2135835
2^{13}	98306	983246	983282	983366	983450	984005	984362	985078	985782	988955
2^{12}	45058	450766	450802	450886	450970	451525	451882	452598	453302	456475
2^{11}	20482	205006	205042	205126	205210	205765	206122	206838	207542	210715
2^{10}	9218	92366	92402	92486	92570	93125	93482	94198	94902	98075
2^{9}	4098	41166	41202	41286	41370	41925	42282	42998	43702	46875
2^{8}	1794	18126	18162	18246	18330	18885	19242	19958	20662	23835
2^{7}	770	7886	7922	8006	8090	8645	9002	9718	10422	13595
2^{6}	322	3406	3442	3526	3610	4165	4522	5238	5942	9115
2^{5}	130	1486	1522	1606	1690	2245	2602	3318	4022	7195
2^{4}	50	686	722	806	890	1445	1802	2518	3222	6395
2^{3}	18	366	402	486	570	1125	1482	2198	2902	6075
2^{2}	6	246	282	366	450	1005	1362	2078	2782	5955
2^{1}	2	206	242	326	410	965	1322	2038	2742	5915
2^{0}	1	196	232	316	400	955	1312	2028	2732	5905

Çizelge 4.28. TMAD Şeması – Toplu Kullanıcı Ekleme – Anahtar İletim Boyutları.

		TMAD								
K.S.	İletim Sayısı	EEDH-112	EEDH-160	EEDH-256	EEDH-384	RSA-512	RSA-1024	RSA-2048	RSA-3072	RSA-7680
2^{0}	3	216	252	336	420	975	1332	2048	2752	5925
2^{1}	8	266	302	386	470	1025	1382	2098	2802	5975
2^{2}	18	366	402	486	570	1125	1482	2198	2902	6075
2^{3}	38	566	602	686	770	1325	1682	2398	3102	6275
2^{4}	78	966	1002	1086	1170	1725	2082	2798	3502	6675
2^{5}	158	1766	1802	1886	1970	2525	2882	3598	4302	7475
2^{6}	318	3366	3402	3486	3570	4125	4482	5198	5902	9075
2^{7}	638	6566	6602	6686	6770	7325	7682	8398	9102	12275
2^{8}	1278	12966	13002	13086	13170	13725	14082	14798	15502	18675
2^{9}	2558	25766	25802	25886	25970	26525	26882	27598	28302	31475
2^{10}	5118	51366	51402	51486	51570	52125	52482	53198	53902	57075
2^{11}	10238	102566	102602	102686	102770	103325	103682	104398	105102	108275
2^{12}	20478	204966	205002	205086	205170	205725	206082	206798	207502	210675
2^{13}	40958	409766	409802	409886	409970	410525	410882	411598	412302	415475
2^{14}	81918	819366	819402	819486	819570	820125	820482	821198	821902	825075
2^{15}	163838	1638566	1638602	1638686	1638770	1639325	1639682	1640398	1641102	1644275
2^{16}	327678	3276966	3277002	3277086	3277170	3277725	3278082	3278798	3279502	3282675
2^{17}	655358	6553766	6553802	6553886	6553970	6554525	6554882	6555598	6556302	6559475
2^{18}	1310718	13107366	13107402	13107486	13107570	13108125	13108482	13109198	13109902	13113075
2^{19}	2621438	26214566	26214602	26214686	26214770	26215325	26215682	26216398	26217102	26220275
2^{20}	5242878	52428966	52429002	52429086	52429170	52429725	52430082	52430798	52431502	52434675
2^{21}	10485758	104857766	104857802	104857886	104857970	104858525	104858882	104859598	104860302	104863475

Çizelge 4.29. TMAD Şeması – Toplu Kullanıcı Çıkarma – Anahtar İletim Boyutları.

		TMAD								
K.S.	İletim Sayısı	EEDH-112	EEDH-160	EEDH-256	EEDH-384	RSA-512	RSA-1024	RSA-2048	RSA-3072	RSA-7680
21	4194302	41943206	41943242	41943326	41943410	41943965	41944322	41945038	41945742	41948915
20	2097150	20971686	20971722	20971806	20971890	20972445	20972802	20973518	20974222	20977395
19	1048574	10485926	10485962	10486046	10486130	10486685	10487042	10487758	10488462	10491635
18	524286	5243046	5243082	5243166	5243250	5243805	5244162	5244878	5245582	5248755
17	262142	2621606	2621642	2621726	2621810	2622365	2622722	2623438	2624142	2627315
16	131070	1310886	1310922	1311006	1311090	1311645	1312002	1312718	1313422	1316595
15	65534	655526	655562	655646	655730	656285	656642	657358	658062	661235
14	32766	327846	327882	327966	328050	328605	328962	329678	330382	333555
13	16382	164006	164042	164126	164210	164765	165122	165838	166542	169715
12	8190	82086	82122	82206	82290	82845	83202	83918	84622	87795
11	4094	41126	41162	41246	41330	41885	42242	42958	43662	46835
10	2046	20646	20682	20766	20850	21405	21762	22478	23182	26355
9	1022	10406	10442	10526	10610	11165	11522	12238	12942	16115
8	510	5286	5322	5406	5490	6045	6402	7118	7822	10995
7	254	2726	2762	2846	2930	3485	3842	4558	5262	8435
6	126	1446	1482	1566	1650	2205	2562	3278	3982	7155
5	62	806	842	926	1010	1565	1922	2638	3342	6515
4	30	486	522	606	690	1245	1602	2318	3022	6195
3	14	326	362	446	530	1085	1442	2158	2862	6035
2	6	246	282	366	450	1005	1362	2078	2782	5955
1	2	206	242	326	410	965	1322	2038	2742	5915
0	0	186	222	306	390	945	1302	2018	2722	5895

TMAD şemasının merkezi versiyonu dağıtık versiyonuna göre daha iyi sonuç verse de, sonuçlar birbirlerine oldukça yakındır. Dağıtık versiyonunda tabloda elde edilen sonuçlarda tüm kullanıcıların aktif kullanıcılar olduğu varsayılmıştır. İlgili çizelgenin grafiksel gösterimi Şekil 4.15'deki gibidir.

Çizelge 4.30. TMAD Şeması – Kullanıcıda Bulunan Anahtar Boyutları.

		TMAD								
K.S.	Anahtar Sayısı	EEDH-112	EEDH-160	EEDH-256	EEDH-384	RSA-512	RSA-1024	RSA-2048	RSA-3072	RSA-7680
2^0	3	240	276	360	444	999	1356	2072	2776	5949
2^1	4	240	276	360	444	999	1356	2072	2776	5949
2^2	5	250	286	370	454	1009	1366	2082	2786	5959
2^3	6	260	296	380	464	1019	1376	2092	2796	5969
2^4	7	270	306	390	474	1029	1386	2102	2806	5979
2^5	8	280	316	400	484	1039	1396	2112	2816	5989
2^6	9	290	326	410	494	1049	1406	2122	2826	5999
2^7	10	300	336	420	504	1059	1416	2132	2836	6009
2^8	11	310	346	430	514	1069	1426	2142	2846	6019
2^9	12	320	356	440	524	1079	1436	2152	2856	6029
2^{10}	13	330	366	450	534	1089	1446	2162	2866	6039
2^{11}	14	340	376	460	544	1099	1456	2172	2876	6049
2^{12}	15	350	386	470	554	1109	1466	2182	2886	6059
2^{13}	16	360	396	480	564	1119	1476	2192	2896	6069
2^{14}	17	370	406	490	574	1129	1486	2202	2906	6079
2^{15}	18	380	416	500	584	1139	1496	2212	2916	6089
2^{16}	19	390	426	510	594	1149	1506	2222	2926	6099
2^{17}	20	400	436	520	604	1159	1516	2232	2936	6109
2^{18}	21	410	446	530	614	1169	1526	2242	2946	6119
2^{19}	22	420	456	540	624	1179	1536	2252	2956	6129
2^{20}	23	430	466	550	634	1189	1546	2262	2966	6139
2^{21}	24	440	476	560	644	1199	1556	2272	2976	6149

Çizelge 4.31. TMAD Şemasının Merkezi - Dağıtık Model Karşılaştırması

	Kullanıcı Ekleme İşlem Maliyeti		Kullanıcı Çıkarma İşlem Maliyeti		
K.S.	Merkezi	Dağıtık	K.S.	Merkezi	Dağıtık
2^0	6	6	2^{21}	90177540	91226116
2^1	12	13	2^{20}	42991620	43515908
2^2	28	30	2^{19}	20447236	20709380
2^3	68	72	2^{18}	9699332	9830404
2^4	164	172	2^{17}	4587524	4653060
2^5	388	404	2^{16}	2162692	2195460
2^6	900	932	2^{15}	1015812	1032196
2^7	2052	2116	2^{14}	475140	483332
2^8	4612	4740	2^{13}	221188	225284
2^9	10244	10500	2^{12}	102404	104452
2^{10}	22532	23044	2^{11}	47108	48132
2^{11}	49156	50180	2^{10}	21508	22020
2^{12}	106500	108548	2^9	9732	9988
2^{13}	229380	233476	2^8	4356	4484
2^{14}	491524	499716	2^7	1924	1988
2^{15}	1048580	1064964	2^6	836	868
2^{16}	2228228	2260996	2^5	356	372
2^{17}	4718596	4784132	2^4	148	156
2^{18}	9961476	10092548	2^3	60	64
2^{19}	20971524	21233668	2^2	24	26
2^{20}	44040196	44564484	2^1	10	11
2^{21}	92274692	93323268	2^0	5	1

4.3. FARKLI ANAHTAR BOYUTLARI AÇISINDAN DEĞERLENDİRME

Önerilen TMAD şeması ile literatürde yer alan şemaların sonuçları alınırken AES-128 ve EEDH-112'den yararlanılmıştır. Hesaplama maliyetlerini azaltmak adına TMAD şeması PRESENT-80 ve BLOWFISH-32 gibi gizli anahtarlı şifreleme yöntemleri kullanılarak sonuçlar yeniden hesaplanmıştır. Şema içerisinde yalnızca gizli anahtarlı şifreleme yöntemleri kullanıldığından anahtar güncelleme işlemlerinde gerçekleştirilen anahtar iletim

sayıları ile işlem maliyetleri merkezi GGİ şemalarıyla aynı sonuca sahip olmaktadır. Bu durum TMAD şemasının kullanıcı ekleme, kullanıcı çıkarma, toplu kullanıcı ekleme ve toplu kullanıcı çıkarma işlemlerinde anahtar iletim sayısı ve işlem maliyeti açılarından merkezi şemalarla aynı sonucu vermesine neden olmaktadır.

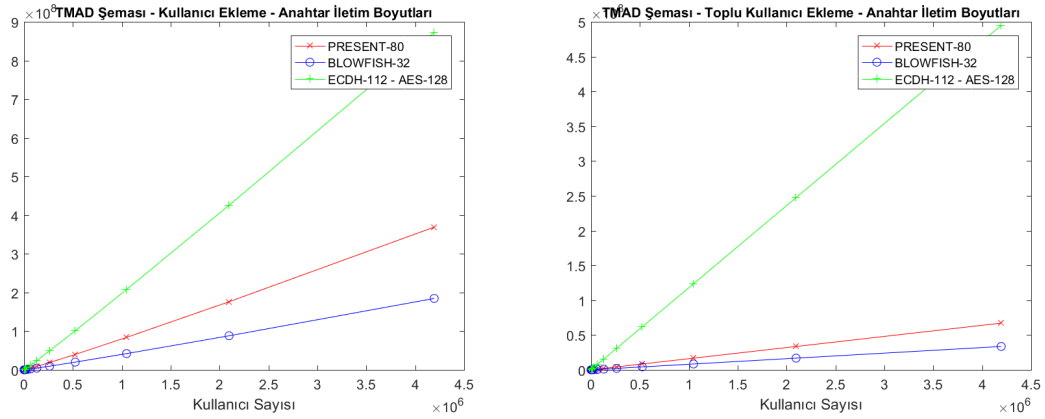
TMAD şemasında kullanıcı anahtarları PRESENT-80 ve BLOWFISH-32 gizli anahtarlı şifreleme yöntemleri kullanılarak oluşturulmuştur. Kullanıcılardan kök düğüme anahtar değerlerinin pozisyonlarına göre sadece sol ya da sağ yarısı iletilir. PRESENT-80 ile oluşturulmuş bir anahtar Base64 sınıfının ardından 16 baytlık metinsel ifadeye, BLOWFISH-32 ile oluşturulmuş bir anahtar Base64 sınıfının ardından 8 baytlık metinsel ifadeye dönüşmektedir.

Çizelge 4.32. TMAD Şeması - Toplu/Kullanıcı Ekleme - Anahtar İletim Boyutları.

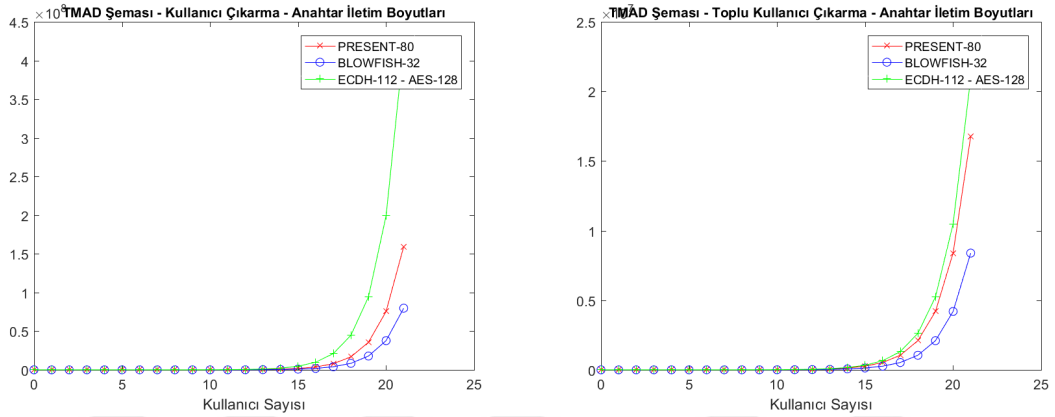
	Kullanıcı Ekleme				Toplu Kullanıcı Ekleme			
K.S.	Anahtar İletim Sayısı	PRESENT-80	BLOWFISH-32	İşlem Maliyeti	Anahtar İletim Sayısı	PRESENT-80	BLOWFISH-32	İşlem Maliyeti
2^0	2	24	12	4	2	24	12	4
2^1	4	48	24	8	4	48	24	7
2^2	10	112	56	20	10	112	56	17
2^3	26	272	136	52	22	240	120	37
2^4	66	656	328	132	46	496	248	77
2^5	162	1552	776	324	94	1008	504	157
2^6	386	3600	1800	772	190	2032	1016	317
2^7	898	8208	4104	1796	382	4080	2040	637
2^8	2050	18448	9224	4100	766	8176	4088	1277
2^9	4610	40976	20488	9220	1534	16368	8184	2557
2^{10}	10242	90128	45064	20484	3070	32752	16376	5117
2^{11}	22530	196624	98312	45060	6142	65520	32760	10237
2^{12}	49154	426000	213000	98308	12286	131056	65528	20477
2^{13}	106498	917520	458760	212996	24574	262128	131064	40957
2^{14}	229378	1966096	983048	458756	49150	524272	262136	81917
2^{15}	491522	4194320	2097160	983044	98302	1048560	524280	163837
2^{16}	1048578	8912912	4456456	2097156	196606	2097136	1048568	327677
2^{17}	2228226	18874384	9437192	4456452	393214	4194288	2097144	655357
2^{18}	4718594	39845904	19922952	9437188	786430	8388592	4194296	1310717
2^{19}	9961474	83886096	41943048	19922948	1572862	16777200	8388600	2621437
2^{20}	20971522	176160784	88080392	41943044	3145726	33554416	16777208	5242877
2^{21}	44040194	369098768	184549384	88080388	6291454	67108848	33554424	10485757

Çizelge 4.33. TMAD Şeması - Toplu/Kullanıcı Çıkarma - Anahtar İletim Boyutları.

	Kullanıcı Çıkarma				Toplu Kullanıcı Çıkarma			
K.S.	Anahtar İletim Sayısı	PRESENT-80	BLOWFISH-32	İşlem Maliyeti	Anahtar İletim Sayısı	PRESENT-80	BLOWFISH-32	İşlem Maliyeti
2^{21}	41943042	159383568	79691784	85983236	4194302	16777200	8388600	4194301
2^{20}	19922946	75497488	37748744	40894468	2097150	8388592	4194296	2097149
2^{19}	9437186	35651600	17825800	19398660	1048574	4194288	2097144	1048573
2^{18}	4456450	16777232	8388616	9175044	524286	2097136	1048568	524285
2^{17}	2097154	7864336	3932168	4325380	262142	1048560	524280	262141
2^{16}	983042	3670032	1835016	2031620	131070	524272	262136	131069
2^{15}	458754	1703952	851976	950276	65534	262128	131064	65533
2^{14}	212994	786448	393224	442372	32766	131056	65528	32765
2^{13}	98306	360464	180232	204804	16382	65520	32760	16381
2^{12}	45058	163856	81928	94212	8190	32752	16376	8189
2^{11}	20482	73744	36872	43012	4094	16368	8184	4093
2^{10}	9218	32784	16392	19460	2046	8176	4088	2045
2^9	4098	14352	7176	8708	1022	4080	2040	1021
2^8	1794	6160	3080	3844	510	2032	1016	509
2^7	770	2576	1288	1668	254	1008	504	253
2^6	322	1040	520	708	126	496	248	125
2^5	130	400	200	292	62	240	120	61
2^4	50	144	72	116	30	112	56	29
2^3	18	48	24	44	14	48	24	13
2^2	6	16	8	16	6	16	8	5
2^1	2	8	4	6	2	8	4	3
2^0	1	0	0	3	1	0	0	0



Şekil 4.16. TMAD Şeması - Toplu/Kullanıcı Ekleme - Anahtar İletim Boyutları.



Şekil 4.17. TMAD Şeması - Toplu/Kullanıcı Çıkarma - Anahtar İletim Boyutları.

Çizelge 4.32 ve Çizelge 4.33'de sırasıyla kullanıcı ekleme, kullanıcı çıkarma, toplu kullanıcı ekleme ve toplu kullanıcı çıkarma işlemleri için TMAD şemasında iki farklı gizli anahtarlı şifreleme yöntemi PRESENT-80 ve BLOWFISH-32 kullanıldığında ortaya çıkan anahtar iletim boyutlarını göstermektedir. İlgili çizelgelerin grafiksel gösterimi Şekil 4.16 ve Şekil 4.17'deki gibidir. Sonuçlar göz önüne alındığında hem PRESENT-80 hem de BLOWFISH-32 yöntemi için TMAD şeması tüm kullanıcı işlemlerinde anahtar boyutu açısından en iyi performans sonucuna sahip olmaktadır.

4.4. GÜVENLİK KRİTERLERİ AÇISINDAN DEĞERLENDİRME

Bölüm 2.2'de belirlenen düşman modellerinin şemalar üzerindeki etkisi incelenmiş, sonuçlar Çizelge 4.34'de belirtilmiştir. Bir şemanın tip-I düşman modelinden etkilenmemek için kimlik doğrulama işlemine sahip olması gerekir. Bir saldırgan özellikle YM gibi davranmaya çalıştığında kullanıcılar bir saldırı olup olmadığını fark edebilmelidir. Bu durum bir açık anahtarlı yöntem kullanan şemalar için kolaydır. Çünkü bir açık anahtarlı

yöntem ile imzalama ve imza ispatı yapılabilir. Gizli anahtarlı şifreleme yöntemini kullanan merkezi şemalarda ise kimlik doğrulama işlemini güvenilir kabul edilen GY gerçekleştirir. Bu nedenle tüm şemalar tip-I düşman modeline karşı güçlüdür.

Çizelge 4.34. Düşman Modelleri.

	GCİ Şemaları							
Düşman Modelleri	MAH	TFA	TFZ	AGDH	DÖGİ	SISA	MHTGG	TMAD
Tip-I Düşman Modeli	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü
Tip-II Düşman Modeli	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü
Tip-III Düşman Modeli	güçlü	zayıf	zayıf	zayıf	zayıf	zayıf	güçlü	güçlü
Tip-IV Düşman Modeli	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü	güçlü
Tip-V Düşman Modeli	zayıf	zayıf	zayıf	zayıf	zayıf	zayıf	zayıf	zayıf

Tüm şemalarda hem anahtar dağıtımı hem de yayın iletiminde şifreleme metotlarından yararlanılmaktadır. Şifreleme metotları güvenli bir yayın iletimi için büyük bir öneme sahiptir. Yayın iletimi şifreli yapılmadığı takdirde açık mesajlar kolaylıkla dinlenebilir. Bir şema yayın iletimini şifreli gerçekleştirmediği takdirde, kullanıcı güncellemelerinin ardından maliyet oluşturacak anahtar güncelleme hesaplamalarına gerek kalmaz. Tüm şemalar, yayın iletiminde birbirlerinden farklı da olsa şifreleme metotlarından yararlanırlar. Bu nedenle tüm şemalar tip-II düşman modeline karşı güçlüdürler.

TFA, TFZ, AGDH, DÖGİ ve SISA şemalarında ortak gizli anahtar kullanıcılardan kök düğüme doğru anahtarların bir fonksiyona sokulmasıyla elde edilir. Anahtarlar arasında fonksiyonel ilişki bulunan şemalar tip-III düşman modeline karşı zayıftır. TMAD şemasında da ortak gizli anahtar kullanıcılardan kök düğüme doğru hesaplanıyor olsa da tip-III düşman modelinde anlatılan $t_2 - t_1$ zaman aralığındaki kullanıcı işlemleri aynı altküme içerisinde gerçekleşir. Bu durum tip-III düşman modelinin TMAD şeması üzerindeki etkisini yok eder.

Tip-IV düşman modeli genellikle iki taraf arasında, bir taraftan diğerine iletilen verinin üçüncü kötü niyetli bir kişi tarafından dinlenerek elde edilmesi ve alıcıya tekrar gönderilmesi şeklindedir. Kötü niyetli kişi tarafından elde edilen bir mesaj şema üzerinde bulunan kullanıcılara tekrar tekrar gönderilebilir. Tüm şemalarda YM, her seferinde farklı bir şifreleme anahtarı ile yayın iletimi yaparak tip-IV düşman modelini azaltabilir. Fakat her yayın iletimi için farklı bir şifreleme anahtarı kullanmak şemaların işlem maliyetini arttıracaktır. Bir diğer seçenek mesajların sonuna bir zaman damgası eklemektir. Bu durumda kullanıcılar gelen mesajların belirli bir zaman aralığında olup olmadığının kontrolünü yapabilirler. Bir mesajın zaman damgası kısmındaki değer güncel zamanın

belirli bir aralığının dışında ise mesaj kabul edilmez. Zaman damgası eklemek literatürde yer alan şemaların tip-IV düşman modelini azaltmasının bir yöntemidir.

TMAD şemasında ise tip-IV düşman modelini azaltmak için bir rasgele veriden yararlanılır. YM, kullanıcılara ilettiği mesajın sonuna bir rasgele veri ekler. Rasgele veriyi ayrıca sunucu üzerinde yer alan veritabanındaki bir alana kaydeder. Kullanıcılar kendilerine gelen mesajda bulunan rasgele veri ile veritabanı üzerinde yer alan rasgele veriyi karşılaştırır. Bu sayede mesajın YM'den gönderilmiş güncel bir mesaj olup olmadığı anlaşılır. YM kullanıcılara ikinci bir mesaj gönderdiğinde veritabanında yer alan rasgele veri alanını güncelleyecektir. Bu sayede bir saldırgan bir önceki mesajı kullanıcılara tekrar iletmek istediğinde mesaj içerisinde yer alan rasgele veri ile veritabanı üzerinde tutulan rasgele veri değerleri aynı olmayacak, kullanıcılar gelen mesajın YM tarafından gönderilen doğru bir mesaj olmadığını tespit edebileceklerdir.

Hem TMAD şeması hem de diğer şemalar tip-V düşman modeline karşı zayıftır. Tip-V düşman modeli kolay bir şekilde tespit edilmez. Şema içerisinde bir saldırgan ile uzlaşan kullanıcı varsa hemen tespit edilebilmeli ve ilgili kullanıcı şemadan çıkartılarak anahtar güncelleme işlemi gerçekleştirmelidir. Bu görev merkezi ve birleşik şemalarda GY'nin, dağıtık şemalarda GY'nin görevini üstlenen bir kullanıcının ya da kullanıcı grubunun görevidir.

Çizelge 4.35. Grup İletişim Gereksinimleri.

	GCİ Şemaları							
Güvenlik Gereksinimleri	MAH	TFA	TFZ	AGDH	DÖGİ	SISA	MHTGG	TMAD
İleri Gizlilik	e	e	e	e	e	e	e	e
Geri Gizlilik	e	e	e	e	e	e	e	e
Kimlik Doğrulama	e	e	e	e	e	e	e	e
Grup Mesaj Bütünlüğü	e	e	e	e	e	e	e	e
Grup Mesaj Gizliliği	e	e	e	e	e	e	e	e
Grup Üyesiyle Uzlaşmaya Dayanıklılık	h	h	h	h	h	h	h	h
Yeniden Anahtarlama	e	e	e	e	e	e	e	e
Grup Bağımsızlığı	e	e	e	e	e	e	e	e
Servis Kalitesi Gereksinimleri								
Hizmet Devamlılığı	e	h	h	h	h	h	h	h
Ölçeklenebilirlik	e	e	e	e	e	e	h	e
Güvenilirlik	e	e	e	e	e	e	h	e
Esneklik	e	e	e	e	e	e	h	e

Tüm şemalar Bölüm 2.3'de belirlenen güvenlik gereksinimleri ve servis kalitesi gereksinimleri açılarından incelenmiştir. Çizelge 4.35'de görüldüğü üzere, bir şemanın ilgili gereksinime sahip olduğunu belirtmek için, evet anlamına gelen "e" harfi, sahip değilse hayır anlamına gelen "h" harfi kullanılmıştır. Güvenlik ve servis kalitesi gereksinimleri niteliksel, ölçülemeyen gereksinimlerdir. İlgili gereksinimler şemalar için aşağıdaki gibi sıralanırlar:

Güvenlik gereksinimleri

1. İleri gizlilik şemadan ayrılan bir kullanıcının gelecek yayın mesajlarını çözmesini engellemek iken, geri gizlilik şemaya eklenen bir kullanıcının geçmiş mesajları çözmesini engellemektir. Her iki gereksinimde anahtar güncelleme işlemiyle sağlanır. Bu gereksinimleri tüm şemaların sağlaması gerekir.
2. Kimlik doğrulama bir şemada olması gereken en önemli gereksimlerden bir tanesidir. Merkezi veya birleşik bir şemada bu görev güvenilir kabul edilen GY'e aittir. Dağıtık bir şemada ise bu görev belirlenen bir kullanıcıya ya da kullanıcı grubuna aittir. TMAD şeması ile diğer şemalar bu gereksinime sahiptir.
3. Grup mesaj bütünlüğü yalnızca kimlik doğrulaması yapılmış kişilerin grup mesajlarına erişiminin sağlanmasıdır. Kimlik doğrulamasına sahip tüm şemalar bu gereksinimi karşılarlar.
4. Grup mesaj gizliliği yalnızca kimlik doğrulaması yapılmış kişilerin şifreli veriden anlamlı mesajı elde etmesidir. Kimlik doğrulamasına sahip tüm şemalar bu gereksinimi karşılarlar.
5. Grup üyesiyle uzlaşmaya dayanıklılık, bir saldırganın bir kullanıcıyla gizlice anlaşmasına karşılık kullanıcının şemadan çıkartılmasıdır. Merkezi veya birleşik bir şemada bu görev merkezi bir varlık olan GY'e aittir. Ancak dağıtık şemalarda bu görev şemadaki diğer kullanıcıların görevidir. Tüm şemalar için tespit edilmesi zor bir gereksinimdir.
6. Yeniden anahtarlama, şemada gerçekleşen kullanıcı değişikliğinin ardından yayın merkezinde bulunan ortak gizli anahtar başta olmak üzere bazı anahtarların güncellenmesidir. Bu işlem ileri ve geri gizliliğin sağlanması için hızlı yapılmalıdır. Tüm şemaların mutlaka bu özelliğe sahip olması gerekir.
7. Grup bağımsızlığı bir kullanıcının birden fazla şemaya üye olabilmesidir. Hem literatürdeki şemalarda hem de önerilen şemada bununla ilgili bir kısıtlama bulunmamaktadır.

Servis kalitesi gereksinimleri

1. Hizmet devamlılığı şema üzerinde gerçekleşen işlemlerin birinde bir sorun ortaya çıktığında, bu sorunun grup iletişimini etkilememesidir. Şema üzerinde bir kullanıcı

anahtarının oluşturulmasında sırasında bir sorunla karşılaşıldığını varsayalım. Bu durum, merkezi bir şema olan MAH üzerindeki grup iletişimi için sorun oluşturmaz. Çünkü anahtar dağıtımı kök düğümden kullanıcılara doğrudur ve anahtarlar birbirinden bağımsızdır. Anahtarların kullanıcıdan kök düğüme doğru, matematiksel bir bağlantı ile hesaplandığı şemalarda bu durum yayın iletimini etkiler. Bu nedenle TMAD şeması dahil diğer tüm şemalar bu durumdan etkilenir.

2. Ölçeklenebilirlik şema büyüklüğüne bakılmaksızın anahtar güncellemelerinin ölçülebilir bir gecikme ile gerçekleşmesidir. Merkezi ve dağıtık şemalar ikili ağaç temelli olduğu için bu gereksinime sahiptir.
3. Güvenilirlik anahtar güncelleme işlemlerinde şemada bulunan üyelere anahtarların güvenli bir yoldan dağıtılmasıdır. Merkezi bir şemada bu görev güvenilir kabul edilen GY'e ait iken, dağıtık ya da birleşik bir şemada bu görev belirlenen bir kullanıcıya ya da bir kullanıcı grubuna aittir.
4. Esneklik şema üzerindeki işlemlerin herhangi bir zamanda yapılabilmesidir. Bu gereksinime tüm şemalar sahip olsalar da, şema üzerinde birçok işlemin zamanlaması doğru seçilmelidir. Çünkü zamanlama şema güvenliği için önemli bir konudur.

5. SONUÇLAR VE ÖNERİLER

Bölüm 4'de elde edilen performans değerlendirmeleri ile çalışmanın getirdiği katkılar bu bölümde incelenmiş ve gelecek çalışmalar için önerilerde bulunulmuştur.

5.1. SONUÇLAR

Bu çalışmada "Tek Yönlü Hibrit Anahtar Dağıtım şeması (One Way Hybrid Key Distribution scheme-OHKD)" adı verilen hem merkezi hem de dağıtık modele sahip yeni bir GGİ şeması önerilmiştir. Şema, mobil uygulama platformu olan Android Studio üzerinde gerçekleştirilerek bir bulut sunucu üzerinde bulunan Nosql veritabanına bağlı bir mobil uygulama geliştirilmiştir. Hem önerilen şemanın hem de literatürde yer alan şemaların mobil uygulama aracılığıyla karşılaştırmaları yapılmış, bu karşılaştırmaların performans değerlendirmeleri gerçekleştirilmiştir. Önerilen şema hem gizli anahtarlı bir şifreleme yönteminden hem de açık anahtarlı bir anahtar dağıtım protokolünden yararlanmaktadır ve diğer şemalara kıyasla düşük sayı ve boyutta aradüğüm değerlerine sahiptir. Ayrıca şemanın merkezi modelinde anahtar dağıtımından sorumlu olan GY'nin görevleri dağıtık modelde kullanıcıların fikir birliği ile gerçekleştirilmektedir. Aşağıda, yapılan çalışmalar, tezin sunumuna uygun olarak, kısaca hatırlatılmış ve elde edilen bulgular sunulmuştur.

1. Bölüm 1 ve Bölüm 2'de yeni bir şema tasarımı yapabilmek için, literatürde yer alan şemaların çalışma prensipleri incelenmiş, merkezi, dağıtık ve birleşik GGİ şemalarına değinilmiş, ikili ağaç temelli olduklarından merkezi ve dağıtık şemalar üzerinde durulmuştur.
2. Bölüm 2'de bir şemanın karşılaşabileceği ataklar belirlenmiş, atakları içeren düşman modelleri oluşturulmuştur.
3. Bölüm 2'de bir şemanın uyması gereken gereksinimlere değinilmiş, bu gereksinimler güvenlik ve servis kalitesi olmak üzere iki kategoride incelenmiştir.

4. Bölüm 3'de gizli ve açık anahtarlı yöntemler kullanılarak, TMAD adı verilen şemanın ayrıntılı çalışma adımları verilmiştir. TMAD şeması merkezi ve dağıtık olarak iki modelde oluşturulmuştur.
5. Bölüm 3'de TMAD şeması iki mobil uygulama üzerinde gerçekleştirilmiştir. Bu uygulamalardan biri veri iletimi için yayın merkezi tarafından, diğeri veri erişimi için kullanıcılar tarafından kullanılmaktadır.
6. Bölüm 4'de TMAD şemasının değerlendirme işlemini yapmak için öncelikli olarak, literatürde yayınlanmış performans ölçüm kriterleri verilmiştir [1], [21], [27], [48], [71]. Şemalar, kullanıcı ekleme ve çıkarma ile toplu kullanıcı ekleme ve çıkarma işlemlerinde, anahtar iletim sayısı, anahtar iletim boyutu, işlem maliyeti, kullanıcıda bulunan anahtar sayısı ve boyutu gibi yönlerden birbirleri ile karşılaştırılmıştır. Şemalarla yapılan performans karşılaştırma sonuçları aşağıda verilmiştir.
    a) TMAD şeması, kullanıcı çıkarma işleminde anahtar iletim boyutları açısından merkezi şemalardan MAH'a göre en az %150, TFA ve TFZ'ye göre en az %20 daha iyi performans sonucuna sahiptir. Önerilen şema ayrıca AGDH ve DÖGİ şemalarına göre en az dokuz kat, SISA ve MHTGG şemalarına göre yüz kattan çok daha fazla performans sonucuna sahiptir. TMAD şeması SISA ve MHTGG şemalarına kıyasla yüksek ölçüde performans başarısı göstermektedir.
    b) TMAD şeması, toplu kullanıcı çıkarma işleminde anahtar iletim boyutları açısından merkezi şemalardan MAH ve MHTGG şemalarına göre en az %150, TFA ve TFZ'ye göre en az %20 daha iyi performans sonucuna sahiptir. Önerilen şema ayrıca dağıtık şemalar olan AGDH ve DÖGİ'ye göre en az on kat, SISA'ya göre en az yirmi kat daha iyi performans sonucuna sahiptir.
    c) TMAD şeması, kullanıcı çıkarma işleminde anahtar iletim sayısı açısından MAH, TFA, TFZ, AGDH ve DÖGİ şemalarıyla ilk sırada, aynı performans sonucuna sahiptir.
    d) TMAD şeması, toplu kullanıcı çıkarma işleminde anahtar iletim sayısı açısından MAH, TFA, TFZ, AGDH ve DÖGİ şemalarıyla ikinci sırada, aynı performans sonucuna sahiptir. Bu şemalar ilk sırada halka topolojisine sahip MHTGG şemasına göre yaklaşık dört kat daha kötü performans göstermektedir.
    e) TMAD şeması, kullanıcı ekleme işleminde işlem maliyeti açısından MAH, TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir.

İlk sırada yer alan merkezi şemalar, TMAD şemasına göre %30 daha iyi performans sonucuna sahiptir. TMAD şemasının ardından sırasıyla AGDH ve DÖGİ şemaları, MHTGG şeması ve SISA şeması gelmektedir. Önerilen şema, AGDH ve DÖGİ şemalarına göre en az %13 daha iyi performans sonucuna sahiptir. Önerilen şema ayrıca SISA ve MHTGG şemalarına göre yüz kattan çok daha fazla performans sonucuna sahiptir. TMAD şeması SISA ve MHTGG şemalarına kıyasla yüksek ölçüde performans göstermektedir.

f) TMAD şeması, kullanıcı çıkarma işleminde işlem maliyeti açısından MAH, TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir. İlk sırada yer alan merkezi şemalar TMAD şemasına göre %5 daha iyi performans sonucuna sahiptir. TMAD şemasının ardından sırasıyla AGDH ve DÖGİ şemaları, MHTGG şeması ve SISA şeması gelmektedir. Önerilen şema, AGDH ve DÖGİ şemalarına göre en az %44 daha iyi performans sonucuna sahiptir. Önerilen şema ayrıca SISA ve MHTGG şemalarına göre yüz kattan çok daha fazla performans sonucuna sahiptir. TMAD şeması SISA ve MHTGG şemalarına kıyasla yüksek ölçüde performans göstermektedir.

g) Kullanıcı ekleme işlemi anahtar iletim boyutu açısından incelendiğinde, TMAD şeması TFA ve TFZ şemalarının ardından en iyi ikinci performans sonucuna sahiptir. İlk sırada yer alan merkezi şemalar TMAD şemasına göre %50 daha iyi performans sonucuna sahiptir. TMAD şemasının ardından sırasıyla MAH şeması, AGDH ve DÖGİ şemaları, SISA şeması ve MHTGG şeması gelmektedir. Önerilen şema MAH şemasına göre en az %20 daha iyi performans sonucuna sahiptir. TMAD şeması ayrıca AGDH ve DÖGİ şemalarına göre en az dört kat daha fazla performans sonucuna sahiptir.

h) TMAD şeması, kullanıcı ekleme işleminde anahtar iletim sayısı açılarından sırasıyla MAH, TFA ve TFZ ile AGDH ve DÖGİ şemalarının ardından en iyi üçüncü performans sonucuna sahiptir. TMAD şeması ilk sırada yer alan merkezi şemalara göre en az %9, ikinci sırada yer alan dağıtık şemalar olan AGDH ve DÖGİ şemalarına göre en az %4 daha kötü performans değerlerine sahiptir. TMAD şemasının ardından sırasıyla MHTGG ve SISA şemaları gelmektedir. Önerilen şema ayrıca SISA ve MHTGG şemalarına göre yüz

kattan çok daha fazla performans sonucuna sahiptir. TMAD şeması SISA ve MHTGG şemalarına kıyasla yüksek ölçüde performans göstermektedir.

i) TMAD şeması kullanıcılarda bulunan anahtar boyutu açısından değerlendirildiğinde, MTHGG şeması ile TFA ve TFZ şemalarının ardından en iyi üçüncü performans sonucuna sahiptir. İlk sırada MHTGG şeması yer alırken, ikinci sırada TFA ve TFZ şemaları bulunmaktadır. Önerilen şemanın ardından sırasıyla MAH şeması, AGDH ve DÖGİ şemaları ile SISA şeması gelmektedir.

j) Toplu kullanıcı çıkarma işlemleri işlem maliyeti açısından değerlendirildiğinde, TMAD şeması MHTGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. TMAD şeması ilk sırada halka topolojisine sahip MHTGG şemasına göre üç kat daha kötü performans göstermektedir. İkinci sırada yer alan MAH, TFA ve TFZ şemaları TMAD şemasına göre %50 daha iyi performans sonucuna sahiptir. AGDH ve DÖGİ şemaları TMAD şemanın önünde üçüncü sırada yer alsa da sonuçlar neredeyse birbirine eşittir. TMAD şemasının ardından SISA gelmektedir. Önerilen şema, SISA şemasına göre en az %30 daha iyi performans değerlerine sahiptir.

k) TMAD şeması, toplu kullanıcı ekleme işleminde anahtar iletim sayısı açısından sırasıyla MHTGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. TMAD şeması ilk sırada yer alan MHTGG şemasına göre en az dört kat daha kötü performans göstermektedir. Önerilen şema ayrıca ikinci sırada yer alan merkezi şemalar olan MAH, TFA ve TFZ şemalarına göre en az %50, üçüncü sırada olan AGDH ve DÖGi şemalarından %20 daha kötü performans değerine sahiptir. Önerilen şema ayrıca SISA şemasına göre %100 daha iyi performans sonucuna sahiptir.

l) TMAD şeması kullanıcılarda bulunan anahtar sayısı açısından değerlendirildiğinde, MTHGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. İlk sırada MHTGG şeması yer almaktadır.

m) TMAD şeması, toplu kullanıcı ekleme işleminde işlem maliyeti açısından sırasıyla MHTGG şeması, MAH, TFA ve TFZ şemaları ile AGDH ve DÖGİ şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. MHTGG şeması TMAD şemasına göre en az dört kat daha iyi performans sonucuna sahiptir. TMAD şeması ikinci sırada yer alan MAH, TFA ve TFZ şemalarına göre en az %80, üçüncü sırada yer alan AGDH ve DÖGi şemalarına göre ise en az %10 daha kötü performans sonucuna sahiptir. Önerilen şema ayrıca SISA şemasına göre en az %50 daha iyi performans sonucuna sahiptir.

n) Toplu kullanıcı ekleme işlemi anahtar iletim boyutu açısından incelendiğinde TMAD şeması, TFA ve TFZ şemaları, MAH şeması ile MHTGG şemalarının ardından en iyi dördüncü performans sonucuna sahiptir. İlk sırada yer alan TFA ve TFZ şemaları TMAD şemasına göre en az dört kat, ikinci sırada yer alan MAH şeması en az üç kat, üçüncü sırada yer alan MHTGG şeması en az iki kat daha iyi performans sonucuna sahiptir. TMAD şemasının ardından sırasıyla AGDH ve DÖGİ şemaları ile SISA şeması gelmektedir. Önerilen şema AGDH ve DÖGİ'ye göre %70, SISA'ya göre en az dört kat daha iyi performans sonucuna sahiptir.

o) TMAD şeması ayrıca PRESENT-80 ve BLOWFISH-32 gizli anahtarlı şifreleme yöntemleriyle düşük anahtar boyutları ile hesaplanmış ve sonuçları karşılaştırılmıştır. TMAD şemasının yalnızca bir gizli anahtarlı şifreleme yöntemiyle oluşturulması, kullanıcı işlemlerinde anahtar iletim sayısı ve işlem maliyeti açısından gizli anahtarlı şifreleme yöntemine sahip MAH, TFA ve TFZ gibi şemalarla benzer sonuçlar vermesine neden olmaktadır. Anahtar boyutları açısından ise, kullanıcılarda ve anahtar güncellemelerinde işlemlerin düşük anahtar boyutları ile yapılıyor olması şemanın diğer tüm şemalardan daha iyi sonuçlar vermesini sağlamaktadır.

Yukarıda belirtilen sonuçlar kapsamında önerilen TMAD şemasının özellikle kullanıcı çıkarma ve toplu kullanıcı çıkarma işlemlerinde en iyi sonuçlara sahip olduğu tespit edilmiştir. TMAD şeması yüksek sayıda aradüğüm anahtarı güncellemesi gerektiren işlemlerde ve düşük sayıda kullanıcı anahtarı güncellemesi gerektiren işlemlerde iyi sonuçlar vermektedir. Bu nedenle önerilen şema kullanıcı anahtarı güncellemesinin yüksek sayıda ve aradüğüm anahtarı güncellemesinin düşük sayıda olduğu toplu kullanıcı ekleme

işlemlerinde performans açısından diğer birçok şemanın gerisinde kalmaktadır. Önerilen şemanın performansı kullanıcı ekleme işlemleri açısından değerlendirildiğinde, şema genellikle merkezi ve dağıtık şemaların arasında yer almaktadır.

5.2. ÇALIŞMANIN GETİRDİĞİ KATKILAR

Bu çalışmanın bilime ve teknolojiye getirdiği ana katkılar aşağıda maddeler halinde sunulmaktadır:

1. Bilime ve teknolojiye yenilik getirme:
    a) Çalışmada hem merkezi hem de dağıtık modellere sahip yeni bir GGİ şeması geliştirilmiştir. Geliştirilen şemanın anahtar güncelleme işlemlerini daha düşük anahtar iletim boyutuyla gerçekleştirilmesi amaçlanmıştır.
    b) Şemada hem bir gizli anahtarlı şifreleme yöntemi hem de bir açık anahtarlı anahtar dağıtım protokolü birlikte kullanılmıştır. Merkezi model için sonuçlar incelendiğinde, önerilen şemada bir açık anahtarlı anahtar dağıtım protokolü ile oluşturulmuş anahtar çifti yer alsa da performans olarak merkezi özelliklere sahip şemalara benzer ya da bu şemalardan daha iyi sonuçlar verdiği belirlenmiştir.
2. Bilinen bir yöntemi yeni bir alana uygulama:
    a) Şemanın dağıtık modeli için, blok zinciri teknolojisinde yer alan fikir birliği mekanizması kullanılmış, fikir birliği mekanizması ile anahtar güncelleme işlemlerinde, anahtar iletim ve dağıtımından sorumlu olan grup yöneticisinin görevleri şemada bulunan kullanıcıların çoğunluğunun onayı ile gerçekleştirilmiştir.

5.3. TARTIŞMALAR VE ÖNERİLER

Tez çalışmasında geliştirilen şema sonuçlarından faydalanarak gelecekte yapılabilecek çalışmalara ışık tutmak üzere aşağıda üç öneri sunulmaktadır:

1. Bu tez çalışmasında TMAD hem bir gizli anahtarlı şifreleme yöntemi hem de bir açık anahtarlı anahtar dağıtım protokolünden yararlanır. TMAD şemasında her iki yöntemini kullanmak yerine yalnızca gizli anahtarlı şifreleme yöntemi ile

oluşturulmuş anahtarlar kullanılarak kullanıcılarda saklanan anahtarların boyutları azaltılabilir, toplu kullanıcı ekleme işlemleri için daha iyi sonuçlar alınması sağlanabilir. Bölüm 4.3'de TMAD şemanın düşük anahtar boyutları ile hesaplaması durumunda ortaya çıkacak sonuçlar gösterilmektedir. Bu çalışma daha kapsamlı hale getirilebilir. Özellikle işlem kapasitesinin kısıtlı olduğu donanımsal yapılarda önerilen yöntemin düşük anahtar boyutlarıyla kullanımı tercih edilebilir.

2. TMAD, mobil bir uygulama üzerinde uygulanmış ve sonuçları alınmıştır. Şema farklı platformlara uyarlanarak performans katkısı sağlayıp sağlamadığı araştırılabilir.
3. TMAD ikili ağaç temellidir. Bu nedenle merkezi ve dağıtık modelleri önerilmiştir. İkili ağaç üzerinde yayın merkezi kök düğümde bulunurken, kullanıcılar ağacın en alt dalları olan yapraklarda yer almaktadır. Şema ikili ağaç dışında farklı topolojilerde oluşturulabilir, performans katkısı sağlayıp sağlamadığı araştırılabilir.

6. KAYNAKLAR

[1] W. Song, H. Zou, H. Liu, ve J. Chen, "A practical group key management algorithm for cloud data sharing with dynamic group," *China Communications*, c. 13, sayı 6, ss. 205–216, 2016.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, ve M. Zaharia, "A view of cloud computing," *Communications of the ACM*, c. 53, sayı 4, ss. 50–58, 2010.

[3] J. Shen, T. Zhou, X. Chen, J. Li, ve W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Transactions on Information Forensics and Security*, c. 13, sayı 4, ss. 912–925, 2018.

[4] J. Yu, K. Ren, C. Wang, ve V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," *IEEE Transactions on Information Forensics and Security*, c. 10, sayı 6, ss. 1167–1179, 2015.

[5] X. Chen, J. Li, X. Huang, J. Ma, ve W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Transactions on Dependable and Secure Computing*, c. 12, sayı 5, ss. 546–556, 2015.

[6] S. Kamara, ve K. Lauter, "Cryptographic cloud storage," *International Conference on Financial Cryptography and Data Security*, Tenerife, Spain, 2010, ss. 136–149.

[7] B. Cui, Z. Liu, ve L. Wang, "Key-aggregate searchable encryption (kase) for group data sharing via cloud storage," *IEEE Transactions on Computers*, c. 65, sayı 8, ss. 2374–2385, 2016.

[8] L. Jiang, ve D. Guo, "Dynamic encrypted data sharing scheme based on conditional proxy broadcast re-encryption for cloud storage," *IEEE Access*, c. 5, ss. 13 336–13 345, 2017.

[9] L. Zhou, V. Varadharajan, ve M. Hitchens, "Trust enhanced cryptographic role-based access control for secure cloud data storage," *IEEE Transactions on Information Forensics and Security*, c. 10, sayı 11, ss. 2381–2395, 2015.

[10] F. Chen, T. Xiang, Y. Yang, ve S. S. Chow, "Secure cloud storage meets with secure network coding," *IEEE Transactions on Computers*, c. 65, sayı 6, ss. 1936–1948, 2016.

[11] D. He, S. Zeadally, ve L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, c. 12, sayı 1, ss. 64–73, 2018.

[12] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, ve Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, c. 16, sayi 6, ss. 996-1010, 2017.

[13] S. Xu, G. Yang, Y. Mu, ve R. H. Deng, "Secure fine-grained access control and data sharing for dynamic groups in the cloud," *IEEE Transactions on Information Forensics and Security*, c. 13, sayı 8, ss. 2101–2113, 2018.

[14] Z. Zhu, ve R. Jiang, "A secure anti-collusion data sharing scheme for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, c. 27, sayı 1, ss. 40–50, 2016.

[15] X. Liu, Y. Zhang, B. Wang, ve J. Yan, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, c. 24, sayı 6, ss. 1182–1191, 2013.

[16] M. Singh, ve S. Singh, "Design and implementation of multi-tier authentication scheme in cloud," *International Journal of Computer Science Issues (IJCSI)*, c. 9, sayı 5, ss. 181, 2012.

[17] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, ve K. Sakurai, "Authentication in mobile cloud computing: A survey," *Journal of Network and Computer Applications*, c. 61, ss. 59–80, 2016.

[18] P. K. Shanu, ve K. Chandrasekaran, "Distribution function based efficient secure group communication using key tree," *2016 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, India, 2016, ss. 1–6.

[19] M. Joe Prathap, ve V.Vasudevan, "Analysis of the various key management algorithms and new proposal in the secure multicast communications," *International Journal of Computer Science and Information Security*, c. 2, sayı 1, 2009.

[20] T. Sakamoto, T. Tsuji, ve Y. Kaji, "Group key rekeying using the lkh technique and the huffman algorithm," *2008 International Symposium on Information Theory and Its Applications*, Auckland, New Zealand, 2008, ss. 1–6.

[21] Q. Gu, L. Peng, L. Wang-Chien, ve C. Chao-Hsien, "Ktr: An efficient key management scheme for secure data access control in wireless broadcast services," *IEEE Transactions on Dependable and Secure Computing*, c. 6, sayı 3, ss. 188–201, 2008.

[22] W. Song, H. Zou, H. Liu, ve J. Chen, "A practical group key management algorithm for cloud data sharing with dynamic group," *China Communications*, c. 13, sayı 6, ss. 205–216, 2016.

[23] N. Alyani, K. Seman, N. M. Nawawi, ve M. N. S. M. Sayuti, "The improvement of key management based on logical key hierarchy by implementing diffie hellman algorithm," *Journal of Emerging Trends in Computing and Information Sciences*, c. 3, sayı 3, 2012.

[24] H. Liu, J. Li, X. Hao, ve G. Zou, "A novel lkh key tree structure based on heuristic search algorithm," *2014 IEEE International Conference on Communiction Problem-Solving*, Beijing, China, 2014, ss. 35–38.

[25] N. Sakamoto, "An efficient structure for lkh key tree on secure multicast communications," *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Las Vegas, USA, 2014, ss. 1–7.

[26] H. Bodur, ve R. Kara, "Implementing diffie-hellman key exchange method on logical key hierarchy for secure broadcast transmission," *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, Girne, Kıbrıs, 2017, ss. 144–147.

[27] Y. Sun, M. Chen, A. Bacchus, ve X. Lin, "Towards collusion-attack-resilient group

key management using one-way function tree," *Computer Networks*, c. 104, ss. 16–26, 2016.

[28] X. Xu, L. Wang, A. Youssef, ve B. Zhu, "Preventing collusion attacks on the one-way function tree (oft) scheme," *International Conference on Applied Cryptography and Network Security*, Zhuhai, China, 2007, ss. 177–193.

[29] M.-S. Hwang ve P.-C. Sung, "A study of micro-payment based on one-way hash chain," *IJ Network Security*, c. 2, sayı 2, ss. 81–90, 2006.

[30] J. Lee, J. W. Seo, H. Ko, ve H. Kim, "Tard: Temporary access rights delegation for guest network devices," *Journal of Computer and System Sciences*, c. 86, ss. 59–69, 2017.

[31] M. Benmalek, ve Y. Challal, "Mk-ami: efficient multi-group key management scheme for secure communications in ami systems," *2016 IEEE Wireless Communications and Networking Conference*, Doha, Qatar, 2016, ss. 1–6.

[32] Y.-R. Chen, ve W.-G. Tzeng, "Group key management with efficient rekey mechanism: a semi-stateful approach for out-of-synchronized members," *Computer Communications*, c. 98, ss. 31–42, 2017.

[33] M. Benmalek, Y. Challal, A. Derhab, ve A. Bouabdallah, "Versami: Versatile and scalable key management for smart grid ami systems," *Computer Networks*, c. 132, ss. 161–179, 2018.

[34] V. Kumar, R. Kumar, ve S. Pandey, "A computationally efficient centralized group key distribution protocol for secure multicast communications based upon rsa public key cryptosystem," *Journal of King Saud University-Computer and Information Sciences*, c. 32, sayı 10, ss. 1081-1094, 2018.

[35] Y. Hanatani, N. Ogura, Y. Ohba, L. Chen, ve S. Das, "Secure multicast group management and key distribution in ieee 802.21," *International Conference on Research in Security Standardisation*, Gaithersburg, USA, 2016, ss. 227–243.

[36] M. Elhoseny, H. Elminir, A. Riad, ve X. Yuan, "A secure data routing schema for wsn using elliptic curve cryptography and homomorphic encryption," *Journal of King Saud University-Computer and Information Sciences*, c. 28, sayı 3, ss. 262–275, 2016.

[37] H.-Y. Lin, M.-Y. Hsieh, ve K.-C. Li, "The cluster-based key management mechanism with secure data transmissions scheme in wireless sensor networks," *International Conference on Applied Mechanics and Mechanical Automation (AMMA)*, Hong Kong, China, 2017, ss. 302-309.

[38] S. H. Islam, ve G. Biswas, "A pairing-free identity-based two-party authenticated key agreement protocol for secure and efficient communication," *Journal of King Saud University-Computer and Information Sciences*, c. 29, sayı 1, ss. 63-73, 2017.

[39] A. Chaudhari, G. Pareek, ve B. Purushothama, "Security analysis of centralized group key management schemes for wireless sensor networks under strong active outsider adversary model," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Karnataka, India, 2017, ss. 1576–1581.

[40] J. Hur, ve Y. Lee, "A reliable group key management scheme for broadcast encryption," *Journal of Communications and Networks*, c. 18, sayı 2, ss. 246–260, 2016.

[41] Q. Zhang, X. Wang, J. Yuan, L. Liu, R. Wang, H. Huang, ve Y. Li, "A hierarchical group key agreement protocol using orientable attributes for cloud computing," *Information Sciences*, c. 480, ss. 55–69, 2019.

[42] P. Vijayakumar, V. Chang, L. J. Deborah, ve B. S. R. Kshatriya, "Key management and key distribution for secure group communication in mobile and cloud network," *Future Generation Computer Systems*, c. 84, ss. 123-125, 2018.

[43] Y.-H. Kung, ve H.-C. Hsiao, "Groupit: Lightweight group key management for dynamic iot environments," *IEEE Internet of Things Journal*, c. 5, sayı 6, ss. 5155–5165, 2018.

[44] K. Lee, ve J. H. Park, "Identity-based revocation from subset difference methods under simple assumptions," *IEEE Access*, c. 7, ss. 60333-60347, 2019.

[45] Y. Zhu, R. Yu, E. Chen, ve D. Huang, "An efficient broadcast encryption supporting designation and revocation mechanisms," *Chinese Journal of Electronics*, c. 28, sayı 3, ss. 445–456, 2019.

[46] H. Jia, Y. Chen, K. Yang, Y. Guo, ve Z. Wang, "Revocable broadcast encryption with constant ciphertext and private key size," *Chinese Journal of Electronics*, c. 28, sayı 4, ss. 690–697, 2019.

[47] S. Maiti, ve S. Misra, "P2b: Privacy preserving identity-based broadcast proxy re-encryption," *IEEE Transactions on Vehicular Technology*, c. 69, sayı 5, ss. 5610–5617, 2020.

[48] O. Cheikhrouhou, "Secure group communication in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, c. 61, ss. 115–132, 2016.

[49] K. Venkatraman, J. V. Daniel, ve G. Murugaboopathi, "Various attacks in wireless sensor network: Survey," *International Journal of Soft Computing and Engineering (IJSCE)*, c. 3, sayı 1, ss. 208–212, 2013.

[50] P. Sakarindr, ve N. Ansari, "Security services in group communications over wireless infrastructure, mobile ad hoc, and wireless sensor networks," *IEEE Wireless Communications*, c. 14, sayı. 5, ss. 8–20, 2007.

[51] B. Daghighi, M. L. M. Kiah, S. Shamshirband, ve M. H. U. Rehman, "Toward secure group communication in wireless mobile environments: Issues, solutions, and challenges," *Journal of Network and Computer Applications*, c. 50, ss. 1–14, 2015.

[52] B. Daghighi, M. L. M. Kiah, S. Shamshirband, S. Iqbal, ve P. Asghari, "Key management paradigm for mobile secure group communications: Issues, solutions, and challenges," *Computer Communications*, c. 72, ss. 1–16, 2015.

[53] X. He, M. Niedermeier, ve H. De Meer, "Dynamic key management in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, c. 36, sayı 2, ss. 611–622, 2013.

[54] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, ve M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, c. 30, sayı 11-12, ss. 2314–2341, 2007.

[55] C. K. Wong, M. Gouda, ve S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, c. 8, sayı 1, ss. 16–30, 2000.

[56] D. Wallner, E. Harder, ve R. Agee, "Key management for multicast: Issues and

architectures," *National Security Agency*, USA, Teknik Rapor, 1999.

[57] A. T. Sherman, ve D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, c. 29, sayı 5, ss. 444–458, 2003.

[58] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, ve B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," *IEEE INFOCOM'99. Conference on Computer Communications*, New York, USA, 1999, ss. 708–716.

[59] A. T. Sherman, ve D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, c. 29, sayı 5, ss. 444–458, 2003.

[60] X. Zou, B. Ramamurthy, ve S. S. Magliveras, "Secure group communications over data networks," *Springer Science & Business Media*, Florida, USA, 2007, ss. 1-165.

[61] Y. Kim, A. Perrig, ve G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security (TISSEC)*, c. 7, sayı 1, ss. 60–96, 2004.

[62] L. R. Dondeti, S. Mukherjee, ve A. Samal, "Disec: a distributed framework for scalable secure many-to-many communication," *Proceedings ISCC 2000. Fifth IEEE Symposium on Computers and Communications*, Juan Les Pins, France, 2000, ss. 693–698.

[63] D. G. Steer, L. Strawczynski, W. Diffie, ve M. Wiener, "A secure audio teleconference system," *Conference on the Theory and Application of Cryptography*, New York, USA, 1988 , ss. 520–528.

[64] Y. Kim, A. Perrig, ve G. Tsudik, "Communication-efficient group key agreement," *IFIP International Information Security Conference*, Paris, France, 2001, ss. 229–244.

[65] O. Cheikhrouhou, A. Koubâa, O. Gaddour, G. Dini, ve M. Abid, "Riseg: A logical ring based secure group communication protocol for wireless sensor networks," *International Conference on Communication in Wireless Environments and Ubiquitous Systems: New Challenges (ICWUS)*, Sousse, Tunisia, 2010, ss. 1-5.

[66] O. Cheikhrouhou, A. Koubâa, G. Dini, ve M. Abid, "Riseg: a ring based secure group communication protocol for resource-constrained wireless sensor networks," *Personal and Ubiquitous Computing*, c. 15, sayı 8, ss. 783–797, 2011.

[67] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," USA, Teknik Rapor, 2019, Erişim: https://bitcoin.org/bitcoin.pdf

[68] E. Ünsal, ve Ö. Kocaoğlu, "Blok zinciri teknolojisi: Kullanım alanları, açık noktaları ve gelecek beklentileri," *Avrupa Bilim ve Teknoloji Dergisi*, sayı 13, ss. 54–64, 2018.

[69] L. Guo, H. Xie, ve Y. Li, "Data encryption based blockchain and privacy preserving mechanisms towards big data," *Journal of Visual Communication and Image Representation*, c. 70, ss. 102741, 2020.

[70] H. Bodur, ve K. Resul, "Yayın şifreleme şemaları üzerinde bir karşılaştırma: Bir yeni yayın şifreleme şeması," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 7, sayı 1, ss. 861–871.

[71] S. Tang, L. Xu, N. Liu, X. Huang, J. Ding, ve Z. Yang, “Provably secure group key management approach based upon hyper-sphere,” *IEEE Transactions on Parallel and Distributed Systems*, c. 25, sayı 12, ss. 3253–3263, 2014.

7. EKLER

Bölüm 3.4'de mobil uygulaması verilen TMAD şemasının bulut üzerinde bulunan bir NoSql veritabanına uygulanması sonucu veritabanında tutulan bilgiler bu bölümde ele alınmıştır.

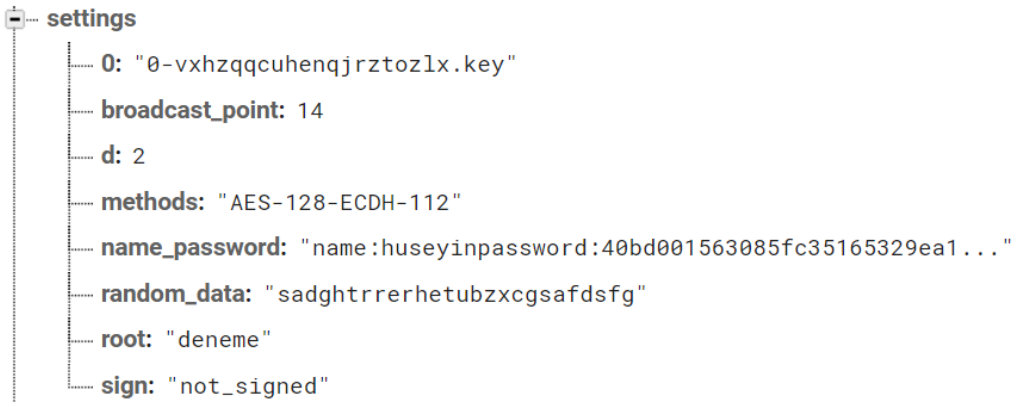
7.1. EK 1: VERİTABANI YAPISI

Veritabanı yapısı TMAD şemasının merkezi modeli için tasarlanmıştır. Ardından dağıtık modelin veritabanına uyarlanması için yapılan değişiklikler açıklanmıştır.

7.1.1. Merkezi Model

TMAD şemasının merkezi modelinin veritabanına uygulanması sonucunda veritabanında yedi adet tablo tutulur. Bu tablolar sırasıyla şöyledir:

1. Şemada yer alan kullanıcıların bilgileri "users" tablosunda bulunur.
2. Ayarlar bilgilerini oluşturan, şemada kullanılan şifreleme metotları, şema ağacının derinlik bilgisi, yayın merkezinin adı ve şifresi, kök düğümde kullanılan simetrik anahtar bilgisi, bir rasgele veri ve yapılan yayın sayısı "settings" tablosunda yer alır.
3. Şemaya eklenecek kullanıcılara verilmek üzere uygun pozisyonlar "positions" tablosunda yer alır.
4. Şemada kullanıcılar ile yayın merkezi arasında bulunan aradüğümlerin pozisyon ve şifreleme anahtar bilgileri "intermediates" tablosunda yer alır. Bu tablo bulut üzerinde anahtarların tutulduğu "keys" klasörü ile ilişkilidir. Anahtarlar fiziksel olarak "keys" klasöründe tutulurken, "intermediates" tablosunda bir anahtarın hangi aradüğüm pozisyonunda bulunduğunun bilgisi yer alır.
5. Yayın merkezinden gönderilen metin mesajları, şifrelendikleri anahtar bilgileri ile birlikte "messages" tablosunda tutulur.
6. Yayın merkezinden gönderilen resimler, şifrelendikleri anahtar bilgileri ile birlikte "images" tablosunda tutulur. Bu tablo bulut üzerinde resimlerin tutulduğu "images"



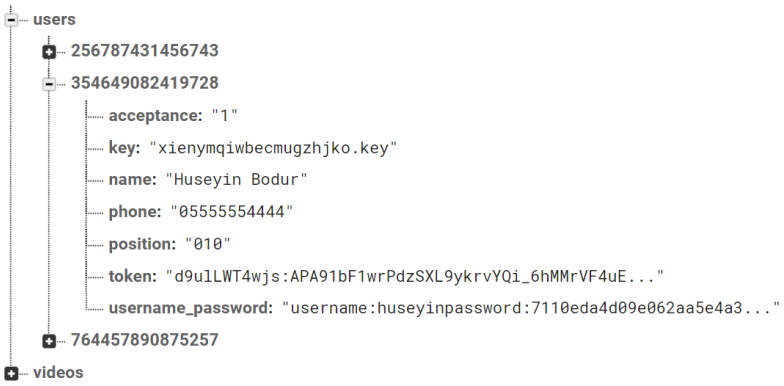
Şekil 7.1. Veritabanı Ayarlar Tablosu.

klasörü ile ilişkilidir. Resimler fiziksel olarak "images" klasöründe tutulurken, "images" tablosunda bir resmin hangi anahtar ile şifrelendiğini belirten bilgi yer alır.

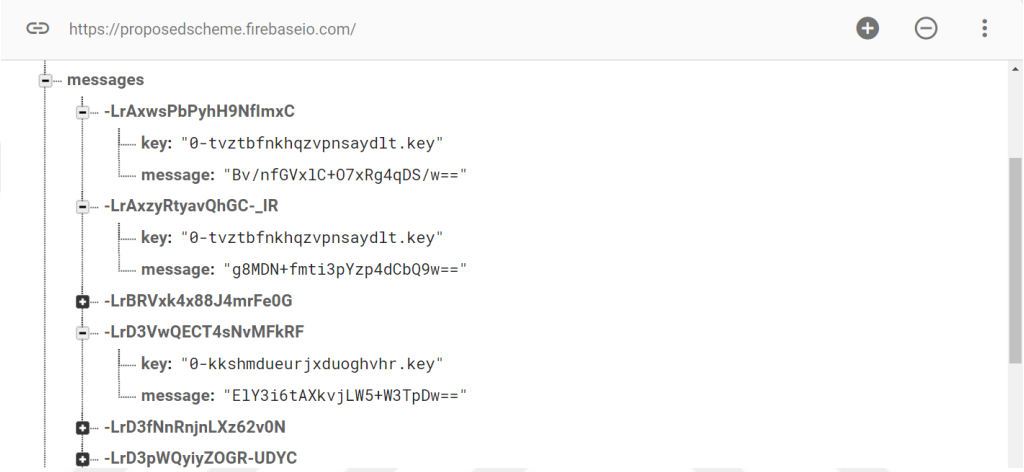
7. Yayın merkezinden gönderilen videolar, şifrelendikleri anahtar bilgileri ile birlikte "videos" tablosunda tutulur. Bu tablo bulut üzerinde resimlerin tutulduğu "videos" klasörü ile ilişkilidir. Videolar fiziksel olarak "videos" klasöründe tutulurken, "videos" tablosunda bir videonun hangi anahtar ile şifrelendiğini belirten bilgi yer alır.

Şekil 7.1'de görüldüğü üzere, ayarlar tablosu olan "settings"'de "0" alanı güncel ortak gizli anahtar bilgisini tutar. "d" değeri ağaç derinliğini ifade eder. "methods" alanı şemada kullanılan gizli ve açık anahtarlı yöntemleri ve anahtar boyutlarını ifade eder. "name_password" alanı yayın merkezinin kullanıcı adı ve şifre bilgilerini tutar. "random_data" alanı mesaj iletimlerinde son iletilen mesajın içerisine eklenen bir rasgele veriyi tutar. Bu sayede tekrarlama atakları önemli ölçüde engellenir. "sign" alanı şemada yayın merkezinden kullanıcılara aktarılan mesajların, aktarımdan önce imzalanıp imzalanmadığının bilgisini tutar.

Şekil 7.2'de görüldüğü üzere, şema üzerinde bulunan kullanıcıların bilgileri "users" tablosunda tutulur. Bu tabloda her kullanıcı için isim (name), telefon numarası (phone), kullanıcı adı ve şifre (username_password) ile token bilgisi yer alır. Token bilgisi kullanıcıya bildirim gönderilmesini mümkün kılar. Bu bilgilerin yanı sıra kullanıcıdan ağaca katılma isteği geldiğinde ilk değeri "0" olan kabul (acceptance) bilgisi bulunur. Eğer GY, kullanıcıyı ağaca katılmayı kabul ederse bu bilgi "1" olarak güncellenir. Kullanıcıya uygun pozisyon değeri pozisyon (position) alanına atanır. Kullanıcının anahtarı keys tablosuna yüklenir ve bu anahtarın adı anahtar (key) alanına eklenir. Şekil 7.3'da görüldüğü



Şekil 7.2. Veritabanı Kullanıcılar Tablosu.

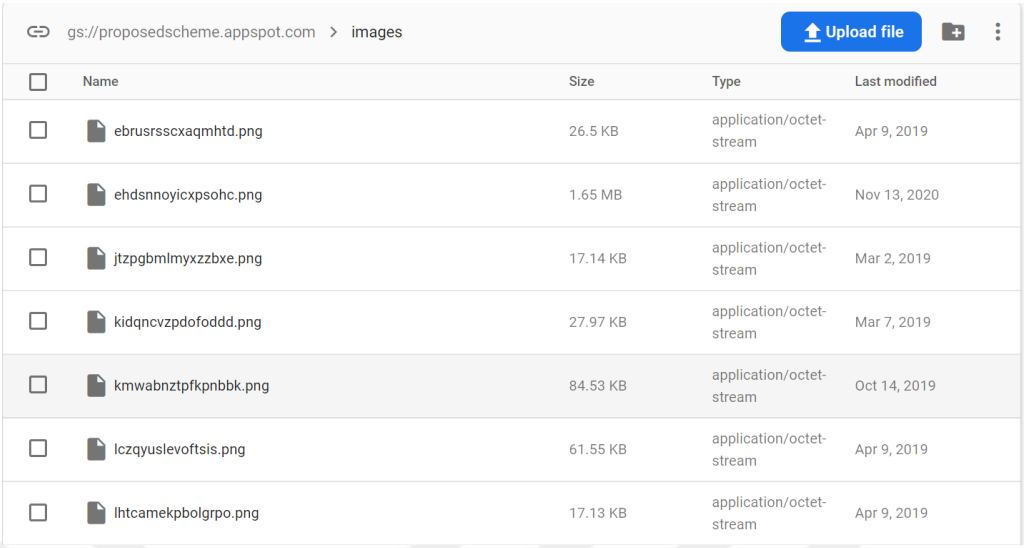


Şekil 7.3. Veritabanı Mesajlar Tablosu.

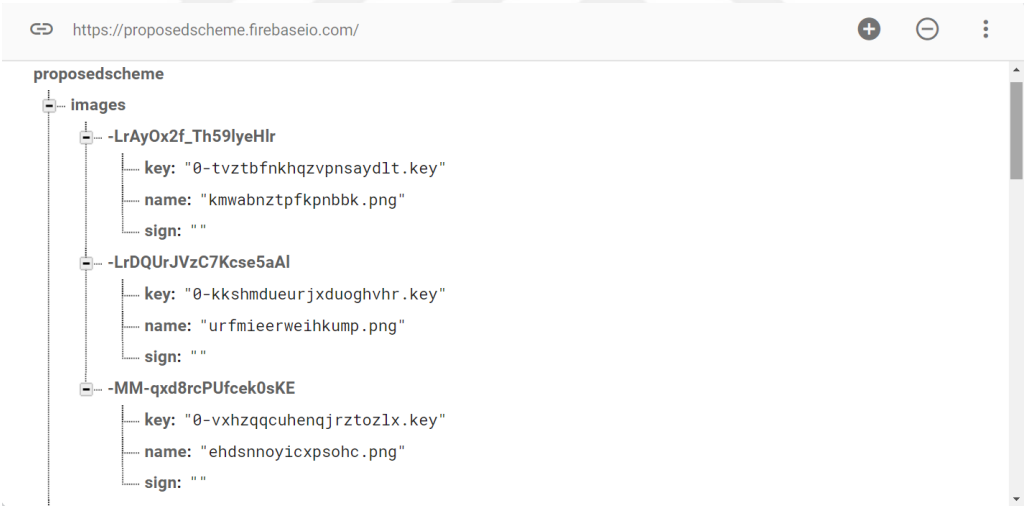
üzere, yayın merkezinden kullanıcılara gönderilen metin mesajları messages tablosunda tutulur. Mesajlar farklı anahtarlarla şifrelenmiş olabileceğinden, tabloda her mesaj için o mesajın hangi anahtar ile şifrelendiğini belirten bir anahtar bilgisi de bulunur.

Şekil 7.4'de görüldüğü üzere, veritabanı üzerindeki resimler şifrelenmiş olarak "images" klasöründe tutulur. Kullanıcı ekleme ve çıkarma işlemlerinde şema üzerinde kullanıcının pozisyonuna göre belirli bir anahtar kümesi sürekli güncellenmektedir. Bu anahtarların başında kök düğümde bulunan ortak gizli anahtar yer alır. Yayın merkezi mesaj iletimi esnasında kök düğümde bulunan güncel ortak gizli anahtarı kullanır.

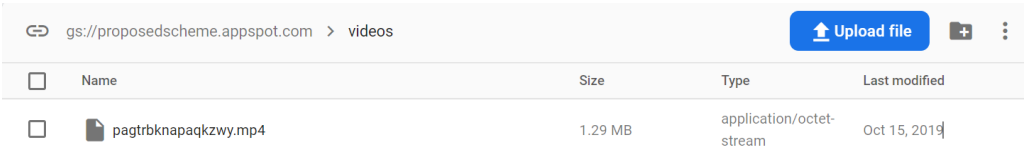
Ortak gizli anahtar sürekli güncellendiğinden veritabanında tutulan resimler birbirlerinden farklı anahtarlarla şifrelenmiş olabilirler. Bu nedenle, Şekil 7.5'de görüldüğü üzere, ayrıca bir "images" tablosu oluşturulmuştur. Bu tabloda her resim için, o resmin hangi anahtar ile şifrelendiğini belirten bir bilgi tutulur.



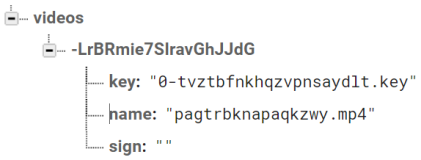
Şekil 7.4. Veritabanı Resimler Klasörü.



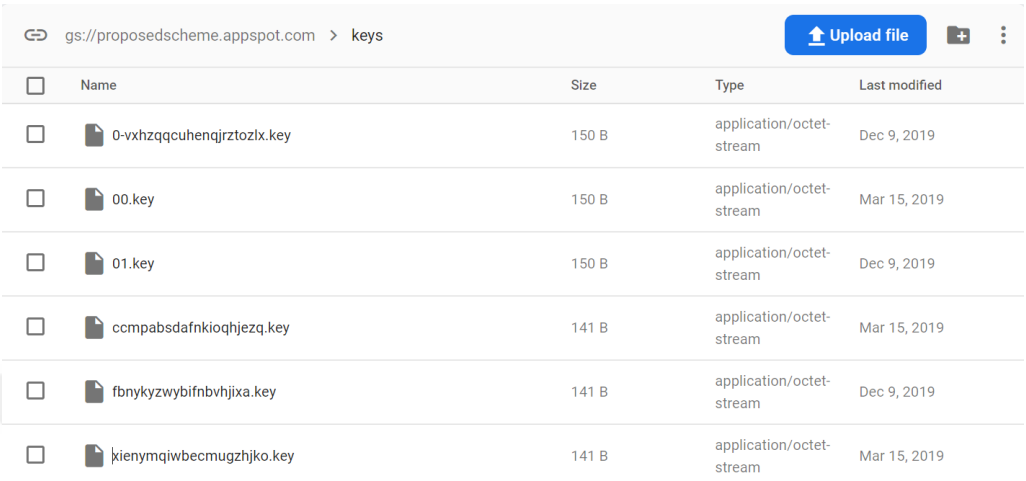
Şekil 7.5. Veritabanı Resimler Tablosu.



Şekil 7.6. Veritabanı Videolar Klasörü.



Şekil 7.7. Veritabanı Videolar Tablosu.

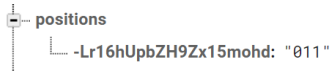


Şekil 7.8. Veritabanı Anahtarlar Klasörü.

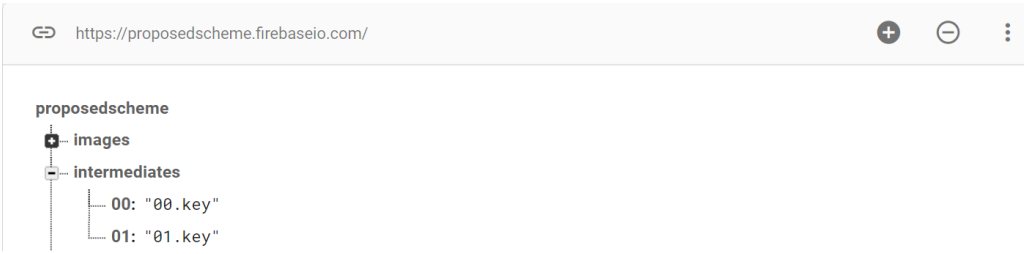
Şekil 7.6'de görüldüğü üzere, veritabanı üzerindeki videolar şifrelenmiş olarak "videos" klasöründe tutulur. Ortak gizli anahtar sürekli güncellendiğinden veritabanında tutulan videolar birbirlerinden farklı anahtarlarla şifrelenmiş olabilirler. Bu nedenle, Şekil 7.7'de görüldüğü üzere, ayrıca her videonun hangi anahtar ile şifrelendiğini belirten bir "videos" tablosu tutulur.

Şekil 7.8'de görüldüğü üzere, şema üzerinde bulunan tüm anahtar değerleri "keys" klasöründe tutulur. Bu anahtarlar kullanıcı anahtarları, aradüğüm anahtarları ve yayın merkezinde bulunan ortak gizli anahtardan oluşmaktadır.

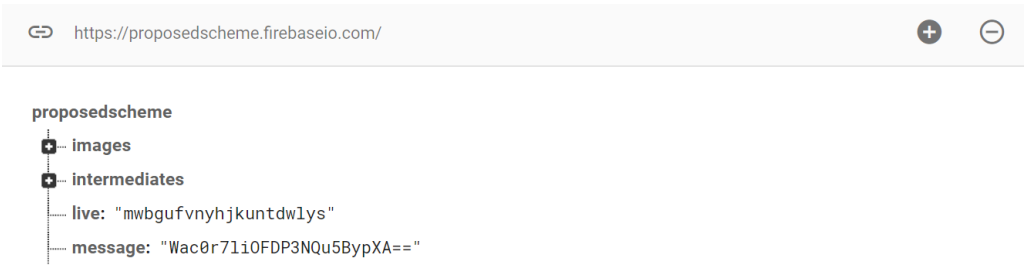
Şekil 7.9'de görüldüğü üzere, şema üzerinde yeni kullanıcıların katıldıklarında kullanabilecekleri pozisyon değerleri "positions" tablosunda tutulmaktadır. Bu tablodaki pozisyon değerleri ağaç derinliği arttıkça, güncellenerek artmaktadır. Şekil 7.1'deki resimde ağaç derinliği 2 dir. Bu nedenle 0 pozisyonu yayın merkezinde, 00 ve 01 pozisyonları aradüğümlerde, 000,001,010 pozisyonları ağaçta bulunan kullanıcılarda



Şekil 7.9. Veritabanı Pozisyonlar Tablosu.



Şekil 7.10. Veritabanı Aradüğüm Tablosu.



Şekil 7.11. Veritabanı Canlı Yayın ve Anlık Mesaj Alanı.

bulunur. 011 pozisyonu yeni bir kullanıcı için müsaittir. Bu pozisyon değeri yeni bir kullanıcının için kullanıldığı takdirde ağaç derinliği otomatik olarak bir artarak 3 olacaktır. Mevcut aradüğüm anahtarları güncellenecek ve yeni aradüğümler oluşturulacaktır. Ayrıca mevcut kullanıcıların pozisyonları güncellenecek ve kullanılabilecek yeni pozisyon değerleri "positions" tablosuna eklenecektir.

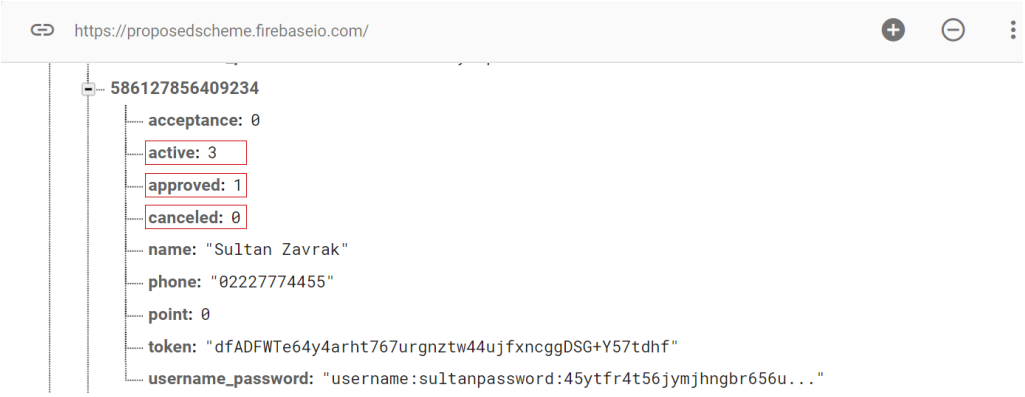
Şekil 7.10'de görüldüğü üzere, "intermediates" tablosunda aradüğüm pozisyonları ve bu pozisyonların anahtar bilgileri tutulur. Ara düğüm anahtarları "keys" klasörü içerisinde yer alır.

Şekil 7.11'de görüldüğü üzere, şemada bulunan live alanı, canlı yayın bilgisini tutar. Kullanıcılar bu bilgiye erişerek yayın merkezinin başlattığı canlı yayına erişebilir.

"message" alanı ise yayın merkezi tarafından anlık iletilen mesajların tutulduğu alandır. Bu mesajlarda geçmiş verilere erişilmez. "message" alanında son aktarılan mesajın güncel ortak gizli anahtarla şifrelenmiş hali yer alır.

7.1.2. Dağıtık Model

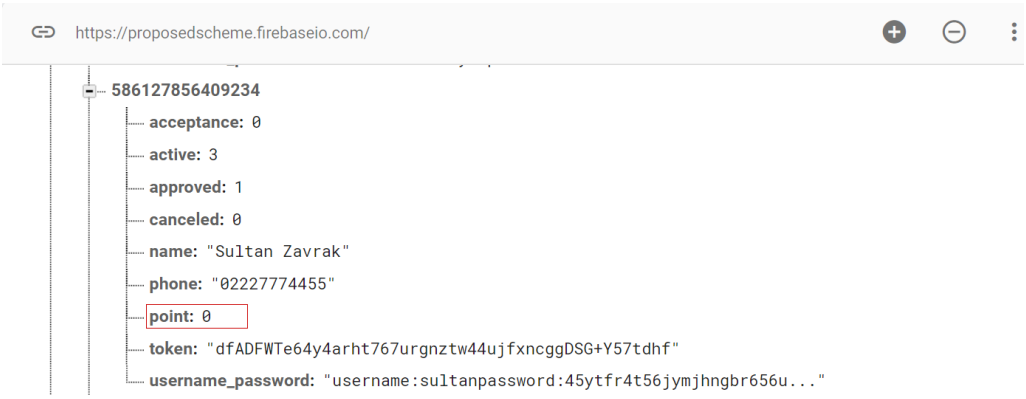
TMAD şemasının dağıtık bir hale dönüştürülmesi için blok zinciri teknolojisinin fikir birliği yöntemi şema için uyarlanmıştır. Bu nedenle veritabanı üzerinde bazı eklemeler yapılmıştır. Öncelikle "users" tablosunda her yeni eklenen kullanıcı için 4 yeni satır



Şekil 7.12. Veritabanı Kullanıcılar Tablosu - Eklemeler.

eklenmiştir. Şekil 7.12'da görüldüğü üzere, bu satırların ilki şemaya yeni bir kullanıcı eklendiğinde, o esnada şemada aktif durumda olan kullanıcıların sayısının belirlendiği "active" alanıdır. Eklenmesi gereken diğer iki alan "approved" ve "canceled" alanlarıdır. Yeni bir kullanıcının şemaya katılması için aktif kullanıcılardan bir onay istenir. Onaylayan kullanıcıların sayısı "approved" alanına, onaylamayan kullanıcıların sayısı ise "canceled" alanına yazılır.

Her bir aktif kullanıcı kendisine gelen talebi onaylarsa "approved" alanı, onaylamazsa "canceled" alanı bir arttırılır. "active" alanında belirlenen aktif kullanıcı sayısının %51'i bu talebe onay verirse bir diğer ifade ile "approved" alanındaki sayı "active" alanındaki sayının %51 ine ulaşırsa, yeni bir kullanıcının şemaya katılması için gerekli atama ve anahtar güncelleme işlemleri gerçekleştirilir. Kullanıcının şemaya alınması işleminin ardından "users" tablosunda bulunan "active", "approved" ve "canceled" alanları silinir. Eğer katılım için gereken %51 sağlanmazsa kullanıcının talebi reddedilir. Bir mevcut kullanıcının şemadan ayrılması için de yine benzer adımlar uygulanır. Eğer bir kullanıcının ekleme-çıkarma işlemi için aktif kullanıcıların tespiti yapılmaz ve onay işlemi tüm kullanıcılara gönderilirse, bu durumda şemadaki veri paylaşımını takip etmeyen kullanıcıların şemaya eklenen ya da şemadan ayrılan kullanıcılar için söz sahibi olmasını gerektirir. Bu durum zaten pasif durumundaki kullanıcıların herhangi bir işlemde bulunmamasına neden olur. Şemaya kullanıcı ekleme-çıkarma işlemi zorlaşır. Şekil 7.13'de görüldüğü üzere, "users" tablosuna eklenmesi gereken son satır her bir kullanıcının yayın merkezinden aktarılan veri paylaşımını takip edip etmediğinin tespitini sağlayan "point" alanıdır.



Şekil 7.13. Veritabanı Kullanıcılar Tablosu - Eklemeler 2.

Şekil 7.14. Veritabanı Ayarlar Tablosu - Eklemeler.

"users" tablosundaki "point" alanına benzer bir alan "settings" tablosunda tutulur. Şekil 7.14'de görüldüğü üzere, bu tablodaki "broadcast_point" alanı ise yayın merkezinin veri paylaşımının sayısını tutar. Eğer users tablosunda bulunan "point" değeri, "broadcast_point" değerine eşit ya da belirli bir aralıkla yakınsa o kullanıcı aktif bir kullanıcıdır.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Hüseyin BODUR

Doğum Tarihi ve Yeri : -

Yabancı Dili : İngilizce

Eposta : -

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Elek-Elekt. ve Bil.Müh.	Düzce Üniversitesi	2021
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2015
Lisans	Bilgisayar Müh.	Pamukkale Üniversitesi	2012

A. Uluslararası hakemli dergilerde yayımlanan makaleler :

A1. H. Bodur and R. Kara, "A Comparison on Broadcast Encryption Schemes: A New Broadcast Encryption Scheme," Advances in Electrical and Computer Engineering, c. 20, s. 4, ss. 69-80., Doi: 10.4316/AECE.2020.04009, 2020.

B. Uluslararası bilim toplantılarında sunulan ve bildiri kitaplarında basılan bildiriler :

B.1. H. Bodur and R. Kara, "Implementing Diffie-Hellman Key Exchange Method on Logical Key Hierarchy for Secure Broadcast Transmission," International Conference on Computational Intelligence and Communication Networks (CICN), ss. 17-17, 2017.

B.2. H. Bodur and R. Kara, "Logical Key Hierarchy Implementation in Cloud Computing," International Conference on Engineering Technology and Innovation (ICETI), ss. 175-179, 2017.

B.3. H. Bodur and R. Kara, "NFC Tabanlı Mobil Ödeme Sistemleri Için Yeni Bir Güvenli Kimlik Dogrulama Yaklasımı," ISCTurkey, ss. 60-64, 2016.

B.4. H. Bodur and R. Kara, "Secure SMS Encryption Using RSA Encryption Algorithm on Android Message Application," 3rd International Symposium on Innovative Technologies in Engineering and Science ISITES, ss. 161-170, 2015.

C. Ulusal hakemli dergilerde yayımlanan makaleler :

C.1. H. Bodur and R. Kara, "Yayın Sifreleme Semaları Üzerinde Bir Karsılastırma: Bir Yeni Yayın Sifreleme Seması," Düzce Üniversitesi Bilim ve Teknoloji Dergisi, c. 7, s. 1, ss. 861-871, Doi: 10.29130/dubited.509102, 2019.