

Yeni Hata Toleranslı Konferans Anahtarı Anlaşma Protokolü

Orhan Ermiş¹, Emin Anarım², M. Ufuk Çağlayan¹

¹ Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

² Boğaziçi Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, İstanbul
orhan.ermis@boun.edu.tr, anarim@boun.edu.tr, caglayan@boun.edu.tr

Özet: Oturum anahtarları, kriptografide, bir oturuma katılan katılımcıların güvenli bir şekilde iletişim kurabilmesi için kullanılır. Bu anahtarlar, anahtar kurma protokolleri kullanarak elde edilir. Eğer oturumdaki katılımcı sayısı ikiden fazla ise o zaman, anahtar kurma protokolleri, konferans ya da grup anahtarı kurma protokolleri olarak adlandırılırlar. Konferans anahtarı kurma protokolleri: konferans anahtarı anlaşma protokolleri ve konferans anahtarı dağıtım protokolleri olmak üzere ikiye ayrılır. İlk tipteki protokollerde, konferans anahtarı oturumdaki bütün katılımcıların katılımıyla oluşturulur. Diğerinde ise konferans anahtarı, güvenilir merkezi bir otorite vasıtasıyla katılımcılara dağıtılır. Bu çalışmada, biz hata toleranslı ve ileri gizlilik ile güçlendirilmiş yeni bir konferans anahtar anlaşma protokolü sunacağız. Konferans anahtar anlaşma protokolleri güvenilir, hızlı ve hata toleranslı olmalıdır. Bu çalışmada, sunacağımız protokol şu ana kadar geliştirilmiş protokollere göre bilinen ataklara karşı daha fazla güvenlik sağlamanın yanında yine diğer protokollere kıyasla daha iyi performans sağlamaktadır.

A New Conference Key-Agreement Protocol

Anahtar Sözcükler: Konferans Anahtarı Anlaşma Protokolleri, ağ güvenliği, gizlilik, hata toleransı.

Abstract: In cryptography, session keys are used for providing a secure communication among participants of a session. The key is obtained by using key establishment protocols. The key establishment protocols are called as conference or group key establishment protocols if the number of participant in the session is more than two. The conference-key establishment can be categorized into two classes which are conference-key agreement protocols and conference-key distribution protocols. In the former one, all of the participants in the conference session agree on a conference key. In the latter one, the session key is distributed by trusted third party. In our study, we will propose a fault-tolerant conference-key agreement protocol with forward secrecy. Conference-key algorithm must be secure, efficient and fault-tolerant. We will show that our protocol provides security against known attacks of conference-key agreement protocols while improving the performance of the algorithm.

Keywords: Conference-key agreement protocols, network security, privacy, fault tolerant, forward secrecy.

1. Giriş

Herhangi bir ağın katılımcılarının güvenli bir şekilde birbiri ile iletişim kurabilmesi için şifreleme algoritmaları kullanmak gerekir. Şifreleme algoritmalarının çoklu katılımcılı ortamlarda kullanılmasındaki en büyük sorun şifreleme anahtarının belirlenen ağın katılımcısı olan her katılımcıya doğru bir şekilde aktarılmasıdır. Bu süreç kriptografide anahtar kurma protokolleri diye adlandırılır ve özellikle periyodik olarak şifreleme anahtarı güncelleyen sistemlerde hayati önem arz etmektedirler.

Anahtar kurma protokolleri: anahtar dağıtım ve anahtar anlaşma şeklinde iki gruba ayrılır. Bu sınıflandırma, şifreleme anahtarlarının merkezi ya da merkezi olmayan yöntemler ile yapılmasına göre belirlenir. Daha açık ifade edecek olursak, eğer belirlenen ağdaki katılımcıların hepsinin güvenilirliğine ikna olduğu üçüncü kişi (ya da kişiler) veya ağın içerisindeki bir katılımcı tarafından şifreleme anahtarı dağıtılıyorsa bu yöntem, anahtar dağıtım protokolleri şeklinde adlandırılır. Eğer

şifreleme anahtarı, ağın içerisindeki katılımcıların birbirlerine kendi oluşturdukları alt anahtarları yollayıp, yine katılımcılar daha önceden belirlenen fonksiyon ile hesaplanıyorsa buna anahtar anlaşma protokolleri denir. Anahtar anlaşma ve anahtar dağıtım protokolleri, eğer ağın içerisindeki katılımcı sayısı iki ve ikiden büyükse sırasıyla, konferans/grup anahtar anlaşma ve konferans/grup anahtar dağıtım protokolleri şeklinde adlandırılırlar.

Anahtar anlaşma protokolü, ilk olarak, Diffie ve Hellman tarafından bulunmuştur [1]. Bu çalışmada iki katılımcının herhangi bir merkezi otorite olmaksızın nasıl kendi anahtarlarını diğer katılımcıya açıklamadan ortak anahtarda karar kılınabileceği anlatılmıştır. Daha sonra, Ingemarsson, bu protokolü birden fazla katılımcının bulunduğu bir ağa uyarlayabilmek için ilk konferans anahtarı dağıtım protokolünü geliştirmiştir [2].

Konferans anahtarı anlaşma protokolleri, yukarıda da bahsettiğimiz gibi birden fazla katılımcının, hızlı ve güvenli bir şekilde oturum anahtarını hesaplamak için

kullanılır. Protokolün hızlı ve güvenli olabilmesi için aşağıdaki özellikleri sağlaması gerekir [14]:

- **Gizlilik:** Oturumun içerisindeki katılımcıların kendi aralarındaki iletişimin oturum dışındaki katılımcılar tarafından ele geçirilememesidir
- **Hata Toleransı:** Kötü niyetli katılımcıların, oturum anahtarı oluşturma esnasında, diğer katılımcıların yanlış hesaplamasının önüne geçilmesidir.
- **İleri Gizlilik:** Algoritmanın uzun süreli anahtar kullanımındaki güvenli olmamasıdır. İlk olarak Diffie, Oorschot ve diğerleri tarafından [3]'te ortaya çıkartılmıştır.
- **Raunt Sayısı:** katılımcıların alt anahtarları birbirlerine gönderirken kaç kez iletişim kurduklarını gösterir.
- **Mesaj uzunluğu:** Katılımcıların kendi ürettikleri alt anahtarları birbirlerine yolladıkları her mesajın uzunluğudur.

Bu çalışmada, geliştirdiğimiz yeni konferans anahtarı anlaşma protokolünü anlatacağız. Biz, bu yeni protokolda, daha önceden geliştirilen protokollere nazaran belli noktalarda daha güvenli hale getirmemizin yanında hem katılımcıların gönderdiği mesajların uzunluğunu düşürdük hem de raunt sayısını azalttık. Böylelikle herhangi çoklu katılımcılı bir ağ için katılımcıların, daha hızlı ve daha güvenli şekilde yeni anahtarı hesaplaması sağlanmış olduk. Ek olarak, katılımcı ağına yeni katılımcı eklendiğinde veya katılımcılardan herhangi bir tanesi oturumu terk etmek istediğinde, protokolün alt fonksiyonları sayesinde bütün süreci tekrarlamadan yeni konferans anahtarını oluşturması mümkün hale getirdik.

Bildirinin sonrakibölümünde, konferans anahtarı anlaşma protokolünü daha iyi anlatabilmemiz için gerekli tanımlamaları yapacağız. Daha sonraki bölümde ise önceki protokollere değineceğiz. Dördüncü bölümde ise yeni konferans anahtarı anlaşma protokolünden bahsedeceğiz. Beşinci bölüm, protokolün güvenlik ve performans analizlerini anlatacağımız bölüm olacak. Son olarak ise sonuç ve önerilerden bahsedeceğiz.

2. Genel Tanımlamalar

Bu bölümde konferans anahtarı anlaşma protokolleri ile ilgili genel tanımları yapacağız. Konferans ağındaki katılımcıların bulunduğu küme: $\mathcal{U} = \{U_1, U_2, \dots, U_n\}$, ile gösterilir ve kümenin eleman sayısı: $|\mathcal{U}| \geq 2$ olmalıdır. Bu liste Tzeng'in protokolü [7] hariç diğer anlatılan protokollerde dairesel liste şeklinde kullanılacaktır (ör: $U_{n+1} = U_1$). Katılımcı listesinin elemanı olan her bir katılımcının bu listeyi bildiği kabul edilir. Konferans anahtarı anlaşma protokollerindeki ortak parametreleri aşağıdaki gibidir [5]:

1. $p = 2q + 1$; p ve q büyük asal sayılar,
2. g , öyle ki $G_q = \{i^2 | i \in Z_q^*\}$ alt grubu için üreteç,
3. T , tekrar saldırıları için zaman damgası,
4. H , tek yönlü kıyım fonksiyonudur.
5. V_{ij} , U_i katılımcısının U_j katılımcısı hakkında gönderilen mesajlar sonunda yaptığı doğrulamanın sonucunu tutar, bu sonuca göre "hatalı" ya da "başarılı" değerlerini alır.
6. Uzun süreli özel anahtar: $x_i \in Z_q^* p$
7. Uzun süreli ortak anahtar: $y_i = g^{x_i} \text{ mod } p$ şeklinde ifade edilir.

Protokolün kurallarına harfiyen uyan katılımcılarına *iyi niyetli katılımcı* denir. İyi niyetli katılımcıların, konferans anahtarını doğru şekilde hesaplamasını önlemeye çalışan katılımcılara *isekötü niyetli katılımcı* denir.

3. Konferans Anahtarı Anlaşma Protokolleri

Önceden bahsettiğimiz gibi konferans anahtarı anlaşma protokolleri, öncelikli olarak Diffie ve Hellman tarafından iki katılımcı için geliştirilmiştir [1]. Daha sonra ise Ingemarsson, ikiden fazla katılımcı için konferans anahtarı dağıtım protokolü geliştirmiştir [2]. Bu çalışmalardan sonra ise en önemli uygulama, Burmester ve Desmedt tarafından geliştirilen kimlik doğrulamalı ve kimlik doğrulamasız konferans anahtarı dağıtım protokolleridir [4]. Bu çalışmanın anahtar dağıtım protokolü olarak adlandırılmasının nedeni uzun süreli anahtarların, konferans katılımcılarına bir merkezden dağıtılıyor olmasıdır. Yoksa o çalışmada anlatılan ilk kimlik doğrulamasız protokol yakın zamanda yapılmış bazı konferans anahtarı protokollerinin temelini oluşturmaktadır. Örnek olarak [10, 11]'deki çalışmalar verilebilir. Daha sonra ise Li ve Pieprzyk, gizli paylaşım protokollerini temel olarak bir konferans anahtarı anlaşma protokolü geliştirmişlerdir [6]. Konferans anahtarı anlaşma protokolleri için bir diğer önemli örnek ise Tzeng tarafından geliştirilmiş protokoldür [7]. Bu çalışma birçok çalışma için temel oluşturmuştur [8, 9, 10, 12] ve bu kadar önemli olmasının nedeni hata toleranslı protokoller ilk olarak Tzeng tarafından konferans anahtarı anlaşma protokollerine uygulanmış olmasıdır. Bu bölümün devamında, konferans anahtarı protokolleri ile ilgili iki önemli örnekten bahsedeceğiz.

3.1 Tzeng'in Konferans Anahtarı Anlaşma Protokolü

Tzeng tarafından geliştirilmiş olan protokol, (1) *gizli dağıtım ve teslim etme*, (2) *alt anahtar hesaplaması ve doğrulaması*, (3) *hata tespiti* ve (4) *konferans anahtarı hesaplama* şeklinde dört ana işlemden oluşur ve aşağıdaki gibidir [7]:

(1) Gizli dağıtım ve teslim etme: Başlangıçta katılımcı listesindeki her bir U_i katılımcısı için $R_i, K_i \in Z_q$ ve $S_i \in Z_q^*$ rastgele seçilir. U_i , n . dereceden (Z_q üzerinden), $h_i(x)$ polinomunu oluşturur, bu polinom,

$$(1 \leq j \leq n) \text{ için } (0, K_i) \text{ ve } (j, (y_j^{R_i} \bmod p) \bmod q)$$

noktalarından geçer. Daha sonra her bir U_i aşağıdakileri hesaplar ve yayımlar:

$$\omega_{ij} = h_i(n+j) \bmod q, 1 \leq j \leq n$$

$$\alpha_i = g^{R_i} \bmod p$$

$$\gamma_i = g^{S_i} \bmod p$$

$$\delta_i = S_i^{-1}(H(K_i, T) - \gamma_i x_i) \bmod q$$

Burada üretilen ω_{ij} değerini, katılımcı kendisi hariç diğer bütün katılımcılar için hesaplar ve $(T, \alpha_i, \gamma_i, \delta_i, \omega_{i,j})$ olarak, her $U_j (1 \leq j \leq n \text{ ve } i \neq j)$ için yayımlar.

(2) Alt anahtarların hesaplanması ve doğrulanması: Her bir katılımcı $\omega_{jl} (1 \leq l \leq n)$ ve α_j mesajlarını aldıktan sonra, kendi özel anahtarı x_i 'yi kullanarak n . dereceden ve $(1 \leq j \leq n)$ değerine göre $(n+l, \omega_{jl})$ ve $(i, (\alpha_j^{x_i} \bmod p) \bmod q)$

noktalarından geçen $h'_j(x)$ polinomunu hesaplar. Daha sonra bu polinomu kullanarak anahtar hesaplanır $K'_j = h'_j(0) \bmod q$. Diğer katılımcılardan gelen her mesaj için $g^{H(K'_j, M)} = y_j^{\gamma_j} \gamma_j^{\delta_j} \bmod p$ denklemi kullanılarak hata kontrolü yapılır. Eğer herhangi bir katılımcı için hata tespit edilirse, U_i katılımcısı hatalı mesaj gönderen U_j katılımcısı için $V_{ij} = \text{"hatalı"}$ mesajını yayımlar.

(3) Hata tespiti: Bir önceki adımda hata tespit edilmişse [7]'deki hata tespit (fault-tolerant) prosedürüne göre kontrol edilir ve hata tekrar ederse, o katılımcı katılımcılar listesinden çıkartılır. Daha sonra protokol 1. adımından tekrardan başlatılır.

(4) Konferans anahtarının hesaplanması: Farz edelim ki katılımcı listesinin son hali $\mathcal{U} = \{U_1, U_2, \dots, U_m\}$ şeklinde olsun. Bu durumda konferans anahtarı aşağıdaki şekilde hesaplanır:

$$K = (K'_1 + K'_2 + \dots + K'_m) \bmod q$$

3.2 Tseng'in Konferans Anahtarı Anlaşma Protokolü

Tseng tarafından geliştirilmiş olan protokol: (1) *kısa küreli anahtar dağıtımı*, (2) *alt anahtar dağıtımı*, (3) *Alt anahtar doğrulama ve konferans anahtarı hesaplama* şeklinde üç adımda gerçekleştirilir [11]. Bu protokolün Tseng'in protokolünden [7] farkı: iki adımda da karşılıklı iletişim olduğu için *hata tespit ve düzeltme* adımı ayrı bir prosedür şeklinde tanımlanmıştır ve iki adım için de uygulanabilir:

(1) Kısa süreli anahtarların dağıtımı: Her katılımcı U_i rastgele iki kısa süreli anahtar seçer $k_i, v_i \in Z_q^*$ ve aşağıdaki geçici sertifika değerlerini hesaplar ve (w_i, A_i, B_i, T) şeklinde yayımlar:

$$w_i = g^{k_i} \bmod p,$$

$$A_i = g^{v_i} \bmod p,$$

$$B_i = v_i^{-1}(H(w_i, T) - A_i x_i) \bmod q.$$

(2) Alt anahtar dağıtımı: Her bir katılımcı diğer katılımcılardan (w_j, A_j, B_j, T) mesajlarını aldıktan sonra $(1 \leq j \leq n \text{ ve } i \neq j \text{ için})$, $g^{H(w_j, T)} = y_j^{A_j} A_j^{B_j} \bmod p$ 'yi sağlayıp sağlamadığı kontrol edilir.

Daha sonra $w_j^q \bmod p \equiv 1$ ve $2 \leq w_j \leq p-1$ kullanılarak w_j 'nin gerçekten G_q alt grubu için tanımlı olup olmadığı kontrol edilir. Bu testlerden herhangi bir katılımcı için hatalı sonuç elde edilirse (ör. U_j) $V_{ij} = \text{"hatalı"}$ mesajı yayımlar ve hata tespit ve düzeltme prosedürü işleme konur. Aksi durumda, her katılımcı $(U_i \in \mathcal{U})$ rastgele bir $r_i \in Z_q^*$ seçer ve $(z_i, \alpha_i, \beta_i, \delta_i)$ değerlerini yayımlar:

$$z_i = (w_{i+1}/w_{i-1})^{k_i} \bmod p,$$

$$\alpha_i = g^{r_i} \bmod p,$$

$$\beta_i = (w_{i+1}/w_{i-1})^{r_i} \bmod p,$$

$$\delta_i = r_i + H(z_i, \alpha_i, \beta_i, T) k_i \bmod q.$$

(3) Alt anahtar doğrulama ve konferans anahtarı hesaplama: Her bir katılımcı $U_i, (z_j, \alpha_j, \beta_j, \delta_j)$ parametrelerini aldıktan sonra

$$g^{\delta_j} = \alpha_j w_j^{H(z_j, \alpha_j, \beta_j, T)} \bmod p \text{ ve } (w_{i+1}/w_{i-1})^{\delta_j} = \beta_j z_j^{H(z_j, \alpha_j, \beta_j, T)} \bmod p$$

kullanılarak doğrulanır. Yine bu kontrollerden herhangi birinde hata oluşması durumunda bir önceki adımdaki gibi hatalı katılımcıyı belirleyip hata tespit ve düzeltme prosedürü çağırılır. Aksi durumda alt anahtarların ve kısa süreli anahtarların dağıtımında herhangi bir sorun yok demektir ve her katılımcı U_i konferans anahtarını aşağıdaki gibi hesaplayabilir:

$$K = w_{i-1}^{n k_i} z_i^{n-1} z_{i+1}^{n-2} \dots z_{i-2} \bmod p \\ = g^{k_1 k_2 + k_2 k_3 + \dots + k_n k_1} \bmod p$$

Hata tespiti ve düzeltmesi: Protokolün ikinci ya da üçüncü adımında herhangi bir U_j kötü niyetli katılımcısı katılımcıların bulunduğu listeden çıkartılacaktır. Bu durumda U_j katılımcısının bir öncesindeki katılımcı U_{j-1} ve bir sonrasındaki katılımcı U_{j+1} , birinci adımda dağıtılmış olan (w_{j-2}, w_{j+1}) ve (w_{j-1}, w_{j+2}) çiftini kullanarak yeni $(z_{j-1}, \alpha_{j-1}, \beta_{j-1}, \delta_{j-1})$, $(z_{j+1}, \alpha_{j+1}, \beta_{j+1}, \delta_{j+1})$ değerlerini yayımlar. Eğer katılımcı sayısı ikiden büyük olursa yeni katılımcı listesine $\mathcal{U} = \{U'_1, U'_2, \dots, U'_m\}$ göre hesaplanır. Eğer katılımcı sayısı üçten küçükse bu hesaplama iptal edilir.

4. Yeni Konferans Anahtarı Anlaşma Protokolü

Bu bölümde, Tseng [7] ve Tseng [11]'deki protokollerin eksikliklerini göz önünde bulundurarak geliştirdiğimiz yeni protokolü açıklayacağız. Bu protokol (1) *alt anahtar dağıtımı*, (2) *alt anahtar doğrulanması*, (3) *hata tespiti ve düzeltmesi* ve (4) *konferans anahtarı hesaplama* olmak üzere dört bölümden oluşur:

(1) Alt anahtar dağıtımı: Her bir katılımcı U_i , iki adet rastgele kısa süreli anahtar seçer $k_i, r_i \in Z_q^*$ ve aşağıdakileri hesaplayıp, yayımlar $(T, \beta_i, \omega_i, \delta_i)$:

$$\alpha_i = g^{k_i} \bmod p$$

$$\beta_i = g^{r_i} \bmod p$$

$$\omega_i = ((\alpha_i y_{i+1}) / y_i) \bmod p$$

$$\delta_i = r_i^{-1} (H(T, \beta_i, \omega_i) - (x_{i+1} + k_i - x_i) \beta_i) \bmod q$$

(2) Alt anahtar doğrulama: Her bir katılımcı $(T, \beta_j, \omega_j, \delta_j)$, $1 \leq j \leq n$ mesajlarını aldıktan sonra gelen mesajların doğruluğunu $g^{H(T, \beta_j, \omega_j)} = \omega_j^{\beta_j} \beta_j^{\delta_j} \bmod p$, $\omega_j^q \bmod p \equiv 1$ ve $2 \leq \omega_j \leq p$ kullanarak kontrol eder. Eğer gelen değerlerde bir hataya rastlanırsa yine diğer protokollerde olduğu gibi hata parametrelerinin tutulduğu matriste ilgili değer $V_{ij} = \text{"hatalı"}$ olarak işaretlenir.

(3) Hata tespiti ve düzeltme: Eğer $V_{ij} = \text{"hatalı"}$ ile hatalı olduğu belirlenmiş U_j katılımcısının gönderdiği mesaj tekrar hatalı olarak işaretlenmişse, bu katılımcı katılımcılar listesinden çıkartılır ve ondan bir önceki katılımcı U_{j-1} tekrardan kısa süreli anahtarları rastgele seçer $k_{j-1}, r_{j-1} \in Z_q^*$ ve diğer katılımcılara yollayacağı mesajın parametrelerini tekrardan hesaplayıp, yayımlar $(T, \beta_{j-1}, \omega_{j-1}, \delta_{j-1})$. Eğer U_j hatalı değerler yayınlamadıysa, bu sefer aynı işlem U_i için uygulanır.

(4) Konferans anahtarı hesaplanması: Hata tespiti ve düzeltme adımının ardından eğer katılımcı listesi yeterli sayıda katılımcı barındırıyor ($|U| \geq 2$), konferans anahtarı listenin son hali $U = \{U'_1, U'_2, \dots, U'_m\}$ ve $m \leq n$ için aşağıdaki gibi hesaplanır:

$$K = \omega_1 \omega_2 \dots \omega_m \bmod p$$

$$= g^{k_1 + k_2 + \dots + k_m} \bmod p$$

5. Güvenlik ve Performans Analizi

Bu bölümde yeni sunduğumuz konferans anahtarı anlaşma protokolünün Tseng [7] ve Tseng protokollerine [11] göre güvenlik ve performans kıyaslamalarını yapacağız.

5.1 Güvenlik Analizi

Güvenlik analizi kısmında, öncelikli olarak konferans anahtarı anlaşma protokollerinin uygulanması esnasında karşılaşacağımız özel durumlardan bahsedeceğiz. Bunlar kısaca: *Yeni katılımcı ekleme ve katılımcıların sistemden ayrılması* şeklindekiye ayrabiliriz. İlk başta da belirttiğimiz gibi gizlilik konferans anahtarı anlaşma protokolleri için büyük önem arz etmektedir. Bu yüzden sisteme yeni bir katılımcı dâhil olduğunda o katılımcının eski iletişim kayıtlarına ulaşmaması gereken durumlar olabilir. Aynı durum tam tersi için de geçerlidir. Bir katılımcı, katılımcı listesinde çıkartıldığında, onun yeni iletişim kayıtlarına ulaşmaması istenir. Bu durumda

konferans anahtarının yeniden oluşturulması ya da değiştirilmesi istenir. Bu değişiklikler için aslında yapılacak işlem yukarıda bahsedilen protokollerin hata tespit ve düzeltme adımlarında kısmen anlatılmıştır.

Biraz daha detaylı anlatacak olursak, Tseng'in protokolünde [7] bahsedilene göre kötü niyetli bir katılımcının konferans anahtarının hesaplanmasını önlemeye çalıştığında, o katılımcı, katılımcıların listesinden çıkartılır ve konferans anahtarı anlaşma protokolü bütün iyi niyetli katılımcılar için en baştan çalıştırılır. Bu işlem bahsettiğimiz özel durumlar için de aynı şekilde olacaktır. Yani listeye yeni bir katılımcı eklenirse ya da kayıtlı bir katılımcı sistemi terk ederse, protokol yeniden çalıştırılıp yeni konferans anahtarı oluşturulacaktır.

Tseng'in protokolünde [11] bahsedilen ise biraz daha farklıdır. Yukarıda bahsedilen hatayı tespit ve düzeltme adımından yola çıkacak olursak. Bir katılımcı listeden çıktıktan sonra, çıkan katılımcının öncesindeki ve sonrasındaki katılımcılar protokolün ikinci adımını tekrardan hesaplar ve yeni parametreler yayımlar. Daha sonra dayeni anahtar hesaplanır. Yeni katılımcı eklendiğinde de yapılan işlem fazla değişmeyecektir. Katılımcı listesi $U' = \{U_1, U_2, \dots, U_n, U_{n+1}\}$ olarak değiştirilecektir. Bu durumda U_{n+1} katılımcısı protokolü baştan çalıştıracak, U_1 ve U_n katılımcıları ise ikinci adımı tekrardan hesaplayacaktır. Bu işlemin dezavantajı ise [13]'te tanımlanan saldırı ile belirtilmiştir. Bu saldırıya göre ilk adımda kısa süreli anahtarlar dağıtıldığı ve protokolün baştan çalışması haricinde değiştirilmediği için bu parametreler oturumdan ayrılan katılımcının elinde bulunur. Bu alt anahtarlar ile yeni anahtarı oluşturmak mümkündür. Yine aynı şekilde sisteme yeni dâhil olan katılımcı da eski anahtar oluşturulurken gönderilen mesajların büyük çoğunluğuna sahip olacağı için eski anahtarı oluşturabilir. Her iki durumda da gizlilik ihlal edilmiş olacaktır.

Bizim önerdiğimiz protokolda ise katılımcı ekleme şu şekilde işlemektedir: öncelikli olarak liste $U' = \{U_1, U_2, \dots, U_n, U_{n+1}\}$ şeklinde güncellenir. U_n ve U_{n+1} (yeni olduğu için) protokolü baştan çalıştırır, diğerleri ise sadece önceki anahtarın oluşturulması esnasında gönderdikleri mesajları tekrardan gönderirler ve her katılımcı anahtarı hesaplar. Bu durumda yeni katılımcı haricinde bir katılımcı daha kısa süreli anahtarları tekrar oluşturmuş olacağı için yeni katılımcının eski kayıtlara ulaşma imkânı yoktur. Sistemden ayrılma durumunda da sadece bir önceki katılımcı (yani U_i katılımcısı ayrılıyorsa U_{i-1}) protokolü baştan çalıştırır ve kısa süreli anahtarı baştan hesaplayıp, diğer katılımcılara dağıtır. Bu durumda, [5]'e göre sistemden ayrılan katılımcının yeni anahtarı ele geçirmesine imkân yoktur.

Bir diğer önemli konu da *ileri gizlilik* konusudur[3]. Eğer herhangi bir anahtar dağıtımı ya da anlaşması protokolünde, yeni anahtar ile ilgili bilgilerin dağıtılması tamamen uzun süreli anahtarlara bağlıysa, o protokol güvenli değildir. Örnek olarak Tzeng'in protokolünü [7] ele alabiliriz, bir katılımcıdan diğerlerine gönderilen mesajlar, gönderilen katılımcının ortak anahtarı ile şifrelenir. Bu durumda bir katılımcının ortak anahtarına karşılık gelen gizli anahtarını ele geçirildiği takdirde konferans anahtarının listenin dışındaki bir katılımcı tarafından hesaplanmasını engelleyecek bir durum yoktur. İlk olarak bu yeni özellik Tseng tarafından konferans anlaşma protokollerine uyarlanmıştır [9].

Bizim protokolümüz de [3]'te bahsedilen ileri gizlilik özelliğini sağlamaktadır. Konferans anahtarının hesaplandığı denklem kısa süreli anahtarlar üzerine oluşturulmuştur. Böylelikle hem gizlilik hem de ileri gizlilik özelliklerini sağladığı için diğer protokollerden daha güvenli bir yapısı vardır.

5.2 Performans Analizi

Bu bölümde, yukarıda bahsettiğimiz protokollerin çalışma performanslarını kıyaslayacağız. Konferans anahtarı anlaşma protokollerinin performansları kıyaslanırken iki konuya dikkat

Protokoller	Mesajların toplam uzunluğu (bit)	Raunt sayısı
Tzeng [7]	$5*(n-1) * k$	1
Tseng [11]	$8*k$	2
Yeni KAAP	$4*k$	1

Tablo 1: Protokollerin Performans Analizleri

edilmesi gerektiğinden bahsetmiştik. Bunlar mesaj uzunluğu ve raunt sayısıdır. Mesaj uzunluğu her iletişimde gönderilen mesajın uzunluğunu tutar, bu uzunluk arttıkça mesajı oluşturmak ve doğrulamak için harcanan zaman da artacaktır. Raunt sayısı ise protokol esnasında katılımcılar arasında kaç kere iletişim kurulduğudur. Raunt sayısının artmasıyla iletişimde gönderilen parametreler ve bu parametreleri doğrulamak için yapacağımız hesaplamalar artacaktır.

Daha basit bir şekilde ifade edebilmek için bir katılımcının konferans anlaşma protokolü sırasında

oluşturduğu her parametrenin uzunluğu k bit olsun. Bir katılımcının protokollerin çalışması esnasında gönderdikleri toplam mesaj uzunluğu ve raunt sayıları aşağıdaki Tablo 1'de verilmiştir.

Ek olarak özel durumlarda ve hatanın düzeltilmesi durumunda protokollerin kıyaslamasını yapabiliriz. Listeye yeni katılımcı eklenmesini dikkate

alırsak Tzeng'in protokolünde [7], bütün katılımcılar protokolü baştan çalıştırmak zorundadır. Tseng'in protokolünde [11] ise sadece iki katılımcı (U_n ve U_1) ikinci adımdan itibaren, yeni eklenen (U_{n+1}) de baştan itibaren protokolü çalıştırır. Bizim protokolümüzde ise sadece iki katılımcı (U_n ve U_{n+1}) baştan itibaren protokolü çalıştırır. Hata düzeltme ve katılımcının konferanstan ayrılması durumları da benzerlik göstermektedir. Tzeng'in protokolünde [7], bütün katılımcılar protokolü en baştan başlatacağıdır. Tseng'in protokolünde [11], dairesel listeye göre sadece ayrılan katılımcının (U_i) öncesindeki ve sonrasındaki katılımcılar (U_{i-1} ve U_{i+1}) ikinci adımdan itibaren protokolü çalıştıracaklardır. Bizim protokolümüzde ise yine dairesel listeye göre sadece ayrılan katılımcının (U_i) öncesindeki katılımcı (U_{i-1}) protokolü baştan çalıştıracaktır.

6. Sonuç ve Öneriler

Konferans anahtarı anlaşma protokollerini dağıtık ağlarda katılımcıların güvenli bir şekilde iletişim kurabilmesi için gerekli şifreleme anahtarını oluşturmak için kullanılır. Bu protokollerin hem güvenli hem de hızlı bir şekilde çalışması beklenmektedir. Çünkü olası bir hatada daha sonra katılımcıların veri alış verişi güvensiz bir ortamda gerçekleşecektir. Bu yüzden, biz daha önce tasarlanmış olan protokollerin güvenlik eksikliklerini ortadan kaldırıp, daha hızlı ve daha az bilgi göndererek konferans anahtarı oluşturulmasını sağladık.

Bu çalışmada öncelikli olarak protokolün matematiksel olarak ifade edilmesi ve diğer protokollere göre güvenlik ve performans kıyaslamalarını ifade ettik. Daha sonra çeşitli ağlara, gerçek zamanlı ve kıyaslamalı performansını gözlemlemek üzere uyarlanabilir.

7. Kaynaklar

[1]Diffie, W. ve Hellman, M. E., "New Directions in Cryptography", IEEE Transactions on Information Theory, 22: 644-654, (1976).

[2]Ingemarsson, I., Tang, D. and Wong, C.K., "A Conference Key Distribution System", IEEE Transactions on Information Theory, 28: 714-719 (1982).

[3]Diffie, W., van Oorschot, P. C. and Wiener, M. J., "Authentication and Authenticated Key Exchanges", Design, Codes and Cryptography, 2: 107-125 (1992).

[4]Burmester, M. and Desmedt, Y., "A Secure and Efficient Conference Key Distribution System (Extended Abstract)", Eurocrypt, Italy, (1994).

- [5] Boneh, D., "Decision Diffie-Hellman Problem", Proceedings of the Third International Symposium on Algorithmic Number Theory, USA, (1998).
- [6] Li, C.-H. and Pieprzyk, J., "Conference Key Agreement from Secret Sharing", ACISP, Australia, (1999).
- [7] Tzeng, W.-G., "A Secure Fault-Tolerant Conference-Key Agreement Protocol", IEEE Transactions on Computers, 51:373-379, (2002).
- [8] Shi, T., Guo, Y. ve Ma, J., "A Fault-Tolerant and Secure Multi-Conference-key Agreement Protocol", International Conference on Communications Circuits and Systems, China, (2004).
- [9] Tseng, Y.-M., "An Improved Conference-Key Agreement Protocol with Forward Secrecy", Informatica, Lith. Acad. Sci., 16:275-284 (2005).
- [10] Tseng, Y.-M., "A Robust Multi-Party Key Agreement Protocol Resistant to Malicious Participants", The Computer Journal, 48:480-487 (2005).
- [11] Tseng, Y.-M., "A communication efficient and fault-tolerant conference-key agreement protocol with forward secrecy", The Journal of Systems and Software, 80:1091-1101 (2007)
- [12] Huang, K.-H., Chung, Y.-F., Lee, H.-H., Lai, F. and Chen, T.-S., "A Conference Key Agreement Protocol with Fault Tolerant Capability", Computer Standards and Interfaces, 31:401-405 (2009).
- [13] Lee, S., Kim, J. and Hong S. J., "Security weakness of Tseng's fault-tolerant conference key agreement protocol", The Journal of Systems and Software, 82: 1163-1167 (2009).
- [14] Tzeng, W.-G., and Tzeng, Z.-J., "Round-Efficient Conference Key Agreement Protocols with Provable Security", ASIACRYPT, Japan (2000).