Uçtan uca şifreleme, gerçek anlamıyla açıklama

Sağlayıcılar E2EE dediklerinde ne söylerler ve neyi söylemezler. Reklam ambalajı olmadan mekanizmanın ve sınırlarının didaktik bir açıklaması.

Açık olalım: WhatsApp mesajlarınızın uçtan uca şifrelendiğini söyler. Bu doğru — ve yeterli değil. Yedekleme ek bir şifreleme olmadan iCloud veya Google Drive'a giderse, şifreleme kendi telefonunuzda bozulur. Operasyonel soru şifrelenmiş olup olmadığı değil, anahtarların nerede bulunduğudur.

Şifrelemenin gerçekte ne anlama geldiği

Bir mesajı şifrelemek, onu anahtar adı verilen belirli bir bilgiye sahip olmayan herkes için gürültü gibi görünen bir şeye dönüştürmek demektir. İşlem gönderenin cihazında yapılır ve doğru anahtarla alıcının cihazında geri alınır. Arada, mesaj bariz bir anlamı olmayan bir bayt dizisi olarak seyahat eder. Basit fikir budur. Makalenin geri kalanı, duruma göre onu gerçek bir garantiye veya bir pazarlama etiketine dönüştüren nüanslarla ilgilenmektedir.

Uçtan uca sıfatı — İngilizce *end-to-end*, kısaltması E2EE — bir kesinlik ekler. Şifreleme, bir ara sunucunun onu okuyup teslim edebilmesi için yapılmaz. Sadece iki ucun — gönderenin cihazı ve alıcının cihazı — anahtara sahip olması için yapılır. Mesajın geçtiği herhangi bir sunucu gürültüyü görür, mesajı değil. Bu, içeriğin bir sunucudan diğerine şifreli olarak seyahat ettiği ancak geçtiği her sunucunun onu yeniden iletmek için deşifre ettiği ve geçici olarak açık metni geri kazandığı *transit halindeki* şifreleme ile olan teknik farktır.

Paylaşılan sır paradoksu

Bariz bir sorun var. İki kişinin birbirleri arasında mesajları şifreleyip deşifre edebilmesi için her ikisinin de aynı anahtara ihtiyacı vardır. Ancak, birbirlerine gönderdikleri her şey, tanımı gereği, birisinin dinliyor olabileceği bir kanaldan geçiyorsa bu anahtar üzerinde nasıl anlaşırlar? Anahtarı daha sonra kullanacakları aynı kanalda kararlaştırmak imkansız görünüyor: eğer saldırgan anahtarı kararlaştırılırken duyarsa, sonraki her şeyi deşifre edebilecektir. Onlarca yıl boyunca klasik kriptografi bunu zor yoldan çözdü: anahtarlar kullanılmaya başlanmadan önce fiziksel karşılaşmalarda şahsen teslim edilirdi. Büyükelçiler, ceketlerinin astarına dikilmiş anahtar çantaları taşırlardı.

Günümüz e-postasında bu çözüm ölçeklenemez. Eğer şifreli olarak iletişim kurmayı amaçladığımız her kişinin evine fiziksel olarak gitmek zorunda olsaydık, kimseyle konuşamazdık. Kriptografi topluluğu tarafından elli yıl önce ortaya atılan soru şuydu: birbirini tanımayan ve sadece halka açık bir kanalı paylaşan iki kişinin, bu halka açık kanalda, kanalı dinleyen hiç kimsenin bilemeyeceği bir sır üzerinde anlaşmaları mümkün müdür?

Diffie-Hellman'ın zarafeti

1976'da Whitfield Diffie ve Martin Hellman adında iki matematikçi görünüşte imkansız bir şeyi kanıtladılar: sadece halka açık bir kanal üzerinden konuşan — herkesin söyledikleri her şeyi duyabileceği bir kanal — iki kişinin, herhangi bir dinleyicinin keşfetmesi mümkün olmadan gizli bir şifre üzerinde anlaşabileceğini. Büyü gibi geliyor. Değil: bu matematik. O zamandan beri bilinen adıyla Diffie-Hellman anahtar değişimi, internetin hemen hemen tüm şifreli iletişiminin temelidir ve yarım asırlık yoğun kullanım ve küresel akademik inceleme onun sağlamlığını teyit etmektedir. Görsel sezgiyi veya matematiği görmek isteyen okumaya devam edebilir. Çalıştığına güvenmeyi tercih edenler de makalenin akışını bozmadan devam edebilir.

Bunu bir görüntüde canlandırmak isteyenler için renklerle bilinen bir benzetme vardır. Alice ve Bruno'nun, onları dinleyen Eva'nın önünde temel bir renk — diyelim ki sarı — üzerinde açıkça anlaştıklarını hayal edin. Her biri özel olarak ikinci bir gizli renk seçer ve sırrını sarı ile karıştırır. Alice özel bir turuncu elde eder; Bruno özel bir yeşil elde eder. Sonuçları Eva'nın önünde birbirleriyle değiştirirler. Şimdi her biri alınan rengi kendi sırrıyla karıştırır ve her ikisi de aynı nihai renge ulaşır, çünkü karıştırma sırası önemli değildir. Eva sarıyı ve iki ara karışımı gördü ama sırları görmedi; sırlardan biri olmadan nihai renge ulaşamaz. Gerçek matematik, renkleri modüler gruplarda veya eliptik eğrilerdeki üs almalarla değiştirir, ancak fikir aynıdır: paylaşılan sır, kanaldaki hiç kimse tarafından yeniden inşa edilemeden halka açık bir şekilde inşa edilir.

Aritmetikte, mekanizmayı görmeyi tercih edenler için: Alice gizli bir a sayısı seçer, Bruno b'yi seçer. Kanal üzerinden açıkça g^a ve g^b değerlerini değiştirirler. Alice $(g^b)^a$ değerini hesaplar ve Bruno $(g^a)^b$ değerini hesaplar; her ikisi de aynı g^{ab} değerine ulaşır. Eva g, g^a ve g^b değerlerinin kanaldan geçtiğini görür, ancak g^a'dan a'yı geri kazanmak — ayrık logaritma problemi olarak adlandırılır — g uygun bir matematiksel grupta seçildiğinde evrenin yaşından çok daha büyük astronomik bir hesaplama süresi gerektirir.

Bunu küçük sayılarla kontrol etmek isteyenler için. Diffie-Hellman değişimi, matematiği elle yapabilecek kadar küçük rakamlarla baştan sona incelenebilir. Aritmetiğe girmeyi tercih etmeyen herkes, makalenin konusundan kopmadan bu bloğu atlayabilir; mekanizmanın adım adım nasıl çalıştığını görmek isteyenler burada bulacaktır. **Genel kurallar**, herkesin okuyabileceği: bir asal sayı $p = 11$ (gerçek Diffie-Hellman'da yaklaşık üç yüz basamaklıdır; işlemler bir sayfaya sığsın diye on bir kullanıyoruz), bir taban $g = 2$ ve tüm aritmetiğin *modulo p* ile yapıldığı

kuralı — hesaplarsınız, *p*'ye bölersiniz ve kalanı tutarsınız, tıpkı onu geçince sıfıra dönen on bir konumlu bir saat gibi. **Özel seçimler**, her biri bir tane ve asla paylaşılmaz: Alice $a = 4$'ü seçer. Bruno $b = 7$'yi seçer.

1. Adım Alice $2^4 = 16$'yı, ardından 16 mod 11 = **5**'i hesaplar. Beşi gönderir. Eva bunu not eder.

2. Adım Bruno $2^7 = 128$'i, ardından 128 mod 11 = **7**'yi hesaplar. Yediyi gönderir. Eva da bunu not eder. İki gönderimden sonra Eva'nın not defterinde dört veri parçası bulunur: $p = 11$, $g = 2$, A = 5, B = 7. Alice ve Bruno'nun türetmek üzere olduğu — ve Eva'nın yeniden oluşturamayacağı — paylaşılan sayı eksiktir.

3. Adım Alice, Bruno'nun kendisine gönderdiği yediyi alır ve bunu kendi özel üssü olan $a = 4$'e yükseltir. $7^4 = 2401$ ile uğraşmaktan kaçınmak için, her adımda modulo uygulanarak parça parça hesaplanır:

$7^2 = 49$

49 mod 11 = 5

$7^4 = (7^2)^2 = 5^2 = 25$

25 mod 11 = 3

Alice **3** sayısını elde eder.

4. Adım Bruno, Alice'in kendisine gönderdiği beşi alır ve bunu kendi özel üssü olan $b = 7$'ye yükseltir. Yine parça parça:

$5^2 = 25 \bmod 11 = 3$

$5^4 = (5^2)^2 = 3^2 = 9 \bmod 11 = 9$

$5^6 = 5^4 \times 5^2 = 9 \times 3 = 27 \bmod 11 = 5$

Son olarak $5^7 = 5^6 \times 5 = 5 \times 5 = 25 \bmod 11 =$ **3**.

Bruno da **3** elde eder.

Her ikisi de paralel çalışarak aynı sayıya, 3'e ulaştı. İkisi de özel üslerini hiçbir zaman göndermedi. Alice $b = 7$ olduğunu bilmiyor; Bruno $a = 4$ olduğunu bilmiyor. Her biri diğerinin gönderdiği genel değeri kendi özel üssüyle birleştirerek kullandı ve aynı varış noktasında buluştular. **Neden aynı sayıya ulaşıyorlar?** Her birinin hesapladığı şey: Alice, $(g^b)^a = 2^{7\times4} = 2^{28} \bmod 11$. Bruno, $(g^a)^b = 2^{4\times7} = 2^{28} \bmod 11$. Aynı miktardır çünkü üslerin çarpım sırası önemli değildir ($7 \times 4 = 4 \times 7$). Her biri farklı bir yoldan aynı varış noktasına geldi.

Peki ya Eva? Not defterinde $p = 11$, $g = 2$, A = 5, B = 7 var ve 3'ü bulmak istiyor. Onu hesaplamak için a veya b'yi bilmesi gerekir — ama ikisi de kanaldan geçmemiştir. Tek çaresi kendine şunu sormaktır: «hangi a üssü için $2^a \bmod 11 = 5$ olur?». Bu kadar küçük bir p ile 0, 1, 2, 3, 4...'ü deneyebilir ve bir dakikadan kısa sürede bulabilir. Ancak 11'i üç yüz basamaklı bir asal sayı ile değiştirdiğinizde, olası üsler uzayı gözlemlenebilir evrendeki atomlardan daha fazla elemana sahip olur. **Şu anda insanlığın bildiği hiçbir algoritma bu uzayı milyarlarca yıldan daha kısa bir sürede kat edemez.** Buna *ayrık logaritma problemi* denir: ileri doğru kolaydır, geriye doğru hesaplanması imkansızdır. Ve Eva tüm konuşmayı harfi harfine takip etmiş olsa bile şifrelemenin direnmesinin nedeni budur.

Üç basit bileşen —saat aritmetiği, üs alma ve çarpmanın değişme özelliği ($a \cdot b = b \cdot a$)— bir araya gelerek insanlığın yarısının özel iletişimleri için her gün güvendiği bir protokol oluşturur. Bu üç parçanın hiçbiri tek başına özel görünmez. Belirleyici olan montajdır.

Diffie-Hellman'dan Signal protokolüne

Bugün profesyonel mesajlaşma uygulamalarının kullandığı uçtan uca şifreleme, hemen hemen istisnasız, Diffie-Hellman değişiminin zarif ve sertleştirilmiş bir versiyonuna dayanmaktadır. 2013 ile 2016 yılları arasında Trevor Perrin ve Moxie Marlinspike tarafından tasarlanan Signal protokolü referanstır. İki ana fikri birleştirir. Birincisi, iki cihaz arasındaki ilk paylaşılan sırrı üreten eliptik eğrilerdeki (X25519) anahtar değişimidir. İkincisi, Double Ratchet — çift dişli mekanizması — olarak adlandırılan ve anahtarları her mesajla otomatik olarak yenileyen sistemdir; böylece cihazın bugün ele geçirilmesi geçmiş mesajların deşifre edilmesine izin vermez, dişli döndürüldükten sonraki gelecek mesajların da.

Zig'de, iki cihaz arasında paylaşılan sırrı üreten X25519 değişimi, standart kütüphaneyi kullanarak altı satıra sığar:

```
const std = @import("std");
const X25519 = std.crypto.dh.X25519;

// Alicia y Bruno generan cada uno un par (privada, pública).
const par_alicia = X25519.KeyPair.generate(io);
const par_bruno  = X25519.KeyPair.generate(io);

// Cada parte recibe la clave pública de la otra y deriva el mismo secreto.
const secreto_alicia = X25519.scalarmult(par_alicia.secret_key, par_bruno.public_key) catch unreachable;
const secreto_bruno  = X25519.scalarmult(par_bruno.secret_key,  par_alicia.public_key) catch unreachable;
// secreto_alicia == secreto_bruno  (32 bytes)
```

O altı satırda olanlar: Genel anahtarlar açıkça seyahat eder. Özel anahtarlar ilgili cihazdan asla çıkmaz. Her bir taraf, kendi özel anahtarından ve diğer tarafın genel anahtarından, kanaldaki hiç kimsenin geri kazanamayacağı aynı otuz iki baytlık sırrı türetir. Bu sır daha sonra değiş tokuş edilen mesajları şifrelemek için bir tohum (seed) görevi görür. Signal protokolünün Double Ratchet'ı, bu materyale sürekli bir rotasyon ekler, böylece bir anlık ele geçirilme görüşmenin geri kalanını tehlikeye atmaz.

Peki `std.crypto.dh.X25519`'un içinde tam olarak ne var? Gizli bir sihir yok. Bunlar Zig'in kendi standart kütüphanesinde bütünüyle okunabilen iki kısa fonksiyondur. İlki özel anahtardan genel anahtarı türetir — değişimin «g^a»sı:

```
pub fn recoverPublicKey(secret_key: [secret_length]u8) IdentityElementError![public_length]u8 {
    const q = try Curve.basePoint.clampedMul(secret_key);
    return q.toBytes();
}
```

Makalenin diliyle: özel anahtar Curve25519 eğrisinin temel noktası ile «çarpılır» — temel aritmetik anlamında değil, eliptik anlamda — ve sonuç otuz iki bayta serileştirilir. `clampedMul` işlemi, bu skaler çarpımın sertleştirilmiş (güçlendirilmiş) versiyonudur: kriptografi topluluğunun yıllar içinde bilinen saldırı ailelerine karşı koymak için eklediği güvenlik önlemlerini içerir. İki satırlık fonksiyon gövdesi.

İkinci fonksiyon sizin özel anahtarınızı diğer tarafın size gönderdiği genel anahtar ile birleştirir. Bu, ikinizin de hiçbir zaman iletmediği otuz iki baytlık paylaşılan sırrı üreten değişimin «$(g^b)^a$»sıdır:

```
pub fn scalarmult(secret_key: [secret_length]u8, public_key: [public_length]u8) IdentityElementError![shared_length]u8 {
    const q = try Curve.fromBytes(public_key).clampedMul(secret_key);
    return q.toBytes();
}
```

İki satır daha. Alınan genel anahtar eğri üzerinde bir nokta olarak yorumlanır ve kişinin kendi özel anahtarıyla «çarpılır». Eğri işleminin değişme özelliği nedeniyle — sayısal örnekte gördüğümüz üslerin çarpımının değişme özelliğine benzer şekilde — her iki taraf da aynı serileştirilmiş noktaya ulaşır: tam olarak makalenin bahsettiği paylaşılan sır.

Hepsi bu kadar. Bir uygulamada sihir gibi görünen şey, gerçekte her biri üç satırlık iki fonksiyondur. Teknik karmaşıklık, aynı standart kütüphanede daha aşağılarda yazılı olan, uluslararası kriptografi topluluğu tarafından onlarca yıldır gözden geçirilen ve harf harf okumak isteyen herkese açık olan tek bir işlemde, `clampedMul`'da toplanmıştır. Ne uygulamamızda ne de Zig'in standart kütüphanesinde kara kutu (black box) yoktur. İnsanın anlayabileceği, içine dalmak istediği hızı seçebileceği açık kaynak kodu vardır.

Uçtan uca şifrelemenin neyi koruduğu

E2EE'nin iyi koruduğu şey, doğru bir uygulama varsayarsak, mesajın transit halindeki içeriğidir. Şifrelenmiş verileri alan ve yeniden ileten bir ara sunucu, anlaşılmaz bayt dizileri görecektir. Kabloya, yönlendiriciye (router), wifi erişim noktasına erişimi olan bir saldırgan da aynısını görecektir. Trafiğin kopyalarını saklayan bir hizmet sağlayıcı, onu daha sonra okuyamayacaktır. Hizmet operatörüne içeriği teslim etmesini emreden bir Hükümet, sunucunun ilk başta sahip olduğu aynı anlaşılmaz baytları alacaktır.

Bu, pratik terimlerle çok şey demektir. Opak bir zarfın içine mektup yazmakla bir kartpostala yazmak arasındaki farktır. Her ikisi de ulaşır. Sadece biri içeriği postacıya karşı korur.

Uçtan uca şifrelemenin neyi korumadığı

Bunu aynı derecede iyi bilmekte fayda var. E2EE metadataları (üst verileri) korumaz: sunucu hala kullanıcı A'nın kullanıcı B'ye ne zaman, hangi sıklıkta ve nereden veri gönderdiğini, ne dediğini bilmese bile bilmektedir. Bu metadatalar, daha önce *Şifrelemek özel olmak değildir* makalesinde tartıştığımız gibi, genellikle içerikten daha fazla şey açığa çıkarır. Birinin bir Cuma günü saat 22:00'de boşanma konusunda uzmanlaşmış bir hukuk bürosunu otuz dakika boyunca aradığını bilmek, aramanın içeriğinin asla anlatmadığı bir hikaye anlatır. Bu, bir kişinin bir onkoloji kliniğine birkaç kez girip çıktığını görmekle aynı durumdur: ne olduğunu hayal etmek için içeride konuşulanlardan hiçbirini duymaya gerek yoktur. Tek bir izole metadata bir şey ifade etmeyebilir; ancak birbiriyle kesiştirilen birkaç tanesi gerçeğe çok benzeyen bir şeyler çizer. E2EE uçları korumaz: alıcının cihazı kötü amaçlı bir program tarafından ele geçirilmişse, mesaj o alıcı için normal şekilde deşifre edilir ve kötü amaçlı program onu okur. E2EE, muhatabın kimliğine karşı tek başına koruma sağlamaz: eğer Alice, Bruno ile konuştuğuna inanıyorsa ancak bir saldırgan başlangıçta araya girmişse (bir *man in the middle*) ve protokol bağımsız doğrulama içermiyorsa, iki taraf da birbirleriyle konuştuklarını sanarak saldırganla konuşur.

Belirsizlik olmadan formüle edilmeye değer dördüncü bir şey daha var. E2EE, onu sunduğunu iddia eden bir sağlayıcının, ayrıca mesajın şifrelenmemiş bir kopyasını kendi sistemlerinde saklamasını engellemez. «Mesajlarım uçtan uca şifrelenmiştir» ifadesi ile «sağlayıcı içeriğimi saklamıyor» ifadesi aynı değildir. Bir uygulama, ikinciyi ihlal ederken birinciyi yerine getirebilir; bunu 2018'den beri basın manşetlerinde defalarca gördük. Kullanıcı, istemcinin kodu doğrulanabilir olmadığı sürece, uzman incelemesi olmadan bir durumu diğerinden ayırt edecek teknik bir yola sahip değildir. Genel kamuoyunda en çok bilinen vaka: WhatsApp mesajları transit halindeyken uçtan uca şifreler, ancak kullanıcı iCloud veya Google Drive'da ek şifreleme olmadan yedeklemeyi etkinleştirirse, bu kopya üçüncü bir tarafın altyapısında okunabilir olarak saklanır ve şifreleme kullanıcının kendi ucunda bozulur.

Operatörün duymak istemediği soru

Uçtan uca şifrelediğini iddia eden bir uygulama, teknik olarak anahtarlarla ilgili üç şeyden birini yapabilir:

1. **Anahtarlar sadece cihazlarda bulunur.** Sadece kullanıcıların cihazlarında oluşturulur ve bulunur; operatör onları bilmez ve saklamaz. Bu en uygun durumdur.
2. **Operatör isterse erişebilir.** Operatör kullanıcıların anahtarlarına sahiptir (veya istediği gibi oluşturabilir) ve bunları veritabanlarında saklar. İsterse veya zorlanırsa içeriği okuyabilir. Bu, çoğu «bulut» hizmeti için geçerlidir.
3. **Operatör tasarım gereği erişemez ancak erişimi kontrol eder.** Operatör anahtarlara sahip değildir ancak onları üreten uygulamanın kontrolüne sahiptir. Zorlanırsa, şifrelemeden önce anahtarları veya içeriği yakalayan kötü niyetli bir güncelleme gönderebilir. Bu, birçok ticari E2EE hizmeti için geçerlidir.

Bu nedenle operasyonel soru, bir şeyin şifreli olup olmadığı değil, cihaza ve anahtarları yöneten yazılıma kimin sahip olduğudur. Solo2'de anahtarlar yalnızca «Kasanızda» (parolanızla şifrelenmiş IndexedDB) bulunur ve yazılım doğrulanabilir açık kaynaktır.

Profesyonel okuyucu için

Uçtan uca şifreleme dijital egemenlik için bir araçtır. Ancak her araç gibi etkinliği onu tutan ele ve dayandığı zemine bağlıdır.

1. Kriptografik anahtarlar nerede oluşturulur ve fiziksel olarak nerede bulunurlar? Eğer operatör onlara erişebiliyorsa (geçici olarak bile olsa, kurtarma kisvesi altında bile olsa), E2EE sadece nominaldir.
2. Görüşmenin kurulması sırasında ortadaki adam saldırısını (man-in-the-middle) önleyen muhatabın bağımsız bir doğrulaması (güvenlik numaraları, QR kodları, bant dışı karşılaştırma) var mıdır?
3. İstemcinin kodu denetlenebilir mi — açık, yayınlanmış, yeniden üretilebilir — yoksa istemcinin gerçekte ne yaptığı konusunda sağlayıcının sözüne güvenmeyi mi gerektirir?
4. Hizmet hangi metadataları üretir ve ne kadar süreyle saklar? İçerik opak olsa bile, metadatalar hassas bilgilerin büyük bir bölümünü yeniden oluşturabilir.

Bu dört soru ileri düzey teknik bilgi istemez; herhangi bir dürüst operatörün genel belgelerinde cevaplayabileceği bilgileri isterler. Cevabın kalitesi ve kesinliği, cevabın kendisi kadar ürün hakkında da çok şey söyler.

Uçtan uca şifreleme, doğru yapıldığında, çağdaş kriptografinin günlük pratiğe sunduğu en ince yapılardan biridir. Orijinal fikir —iki kişinin halka açık môt kanalda bir sır üzerinde anlaşabilmesi— 1976'da Whitfield Diffie ve Martin Hellman'a aittir; yarım yüzyıl sonra hala bunun sonuçlarını yaşıyoruz. Ancak, her teknik vaatte olduğu gibi, değeri etikete değil, gerçek yerine getirilmesine bağlıdır. Dürüst bir profesyonelin sorusu «şifreli mi?» değil, «anahtarlar kimde?» sorusudur. Cevapların farklı sonuçları vardır. Bunları bilmekte fayda var.

Kaynaklar ve ek okumalar

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, Kasım 1976. Açık anahtarlı şifrelemenin temel makalesi.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, Open Whisper Systems tarafından herkese açık şartname, 2016 revizyonu. Signal protokolünün ve endüstriyel türevlerinin temeli.
- RFC 7748 — Elliptic Curves for Security (IETF, Ocak 2016). Modern anahtar değişimlerinde kullanılan X25519 ve X448 eğrilerinin normatif özellikleri.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Anahtar değişimi ve kimlik doğrulamalı şifreleme protokolleri üzerine bölümler.
- Avrupa dijital kimlik çerçevesi (eIDAS 2) üzerine 2024/1183 sayılı Tüzük (AB) — muhatabın bağımsız doğrulamasının kurumsal destek kazandığı ve nominal ile gerçek şifreleme arasındaki ayrımın farklı yasal sonuçlara sahip olduğu çerçeveler oluşturur.

← ÖncekiKill switch ve kurumsal ele geçirmeSonraki →Bir güven sinyali olarak iş modeli

Son okumalar

- Analiz · 18 Mayıs 2026 Gerçek vs. görünür gizlilik: kendinize sormanız gereken sorular
- Analiz · 18 Mayıs 2026 Profesyonel bir uygulama olarak self-hosting
- Kavram · 18 Mayıs 2026 24 kelime: kriptografik kimlik nedir

Bu makaleyi ihtiyacınız olan her yere yanınızda götürün.

↓ Markdown ↓ Düz metin ↓ PDF

Dosya cihazınıza indirilecektir. Oradan kaydedebilir, Solo2'ye aktarabilir veya istediğiniz yerde paylaşabilirsiniz. Cuadernos hedef noktaya sizin yerinize karar vermez.

Mühür mumu · SHA-256 7b1b7201eceadb86303a7732fb17959ca362bdd66c05ec77778fa5624d351468

Cuadernos Lacre · Menzuri Gestión S.L. yayını ·
R.Eugenio tarafından yazıldı · Solo2 ekibi tarafından düzenlendi.