Mã hóa đầu cuối, giải thích thực sự

Những gì nhà cung cấp nói khi họ nói E2EE, và những gì họ không nói. Một lời giải thích mang tính giáo dục về cơ chế và các giới hạn của nó, không có vỏ bọc quảng cáo.

Nói cho rõ: WhatsApp nói rằng tin nhắn của bạn được mã hóa đầu cuối. Điều đó đúng — và điều đó là chưa đủ. Nếu bản sao lưu được chuyển đến iCloud hoặc Google Drive mà không có mã hóa bổ sung, thì việc mã hóa sẽ bị phá vỡ trên chính điện thoại của bạn. Câu hỏi vận hành không phải là nó có được mã hóa hay không, mà là các khóa nằm ở đâu.

Mã hóa thực sự có ý nghĩa gì

Mã hóa một tin nhắn có nghĩa là biến đổi nó thành một thứ trông giống như tiếng nhiễu đối với bất kỳ ai không sở hữu thông tin nhất định được gọi là khóa. Thao tác được thực hiện trên thiết bị của người gửi và với khóa chính xác, nó được hoàn tác trên thiết bị của người nhận. Ở giữa, tin nhắn di chuyển dưới dạng một chuỗi các byte không có ý nghĩa rõ ràng. Đó là ý tưởng đơn giản. Phần còn lại của bài viết đề cập đến những sắc thái biến nó, tùy từng trường hợp, thành một sự đảm bảo thực sự hoặc thành một nhãn dán tiếp thị.

Tính từ *đầu cuối* — tiếng Anh là *end-to-end*, viết tắt là E2EE — thêm vào một sự chính xác. Mã hóa không được thực hiện để một máy chủ trung gian có thể đọc và chuyển nó đi. Nó được thực hiện để chỉ có hai đầu — thiết bị của người gửi và thiết bị của người nhận — sở hữu khóa. Bất kỳ máy chủ nào mà tin nhắn đi qua đều thấy tiếng nhiễu, không phải tin nhắn. Đó là sự khác biệt kỹ thuật với mã hóa *đang chuyển tiếp* (in transit), nơi nội dung di chuyển được mã hóa từ máy chủ này sang máy chủ tiếp theo, nhưng mỗi máy chủ mà nó đi qua sẽ giải mã nó để gửi tiếp, khôi phục tạm thời văn bản rõ.

Nghịch lý của bí mật chung

Có một vấn đề rõ ràng. Để hai người có thể mã hóa và giải mã tin nhắn qua lại cho nhau, cả hai cần có cùng một khóa. Nhưng, làm thế nào họ thỏa thuận được khóa này nếu mọi thứ họ gửi cho nhau, theo định nghĩa, đều đi qua một kênh nơi ai đó có thể đang nghe lén? Thỏa thuận khóa trên chính kênh mà sau đó họ sẽ sử dụng nó dường như là không thể: nếu kẻ tấn công nghe thấy nó khi đang thỏa thuận, họ sẽ có thể giải mã mọi thứ sau đó. Trong nhiều thập kỷ, mật mã học cổ điển đã giải quyết vấn đề này theo cách khó khăn: các khóa được bàn giao trực tiếp, trước khi bắt đầu sử dụng, trong các cuộc gặp gỡ vật lý. Các đại sứ mang theo những chiếc cặp đựng khóa được khâu vào lớp lót áo khoác của họ.

Trong email đương đại, giải pháp đó không thể mở rộng. Nếu chúng ta phải đi trực tiếp đến nhà của mỗi người mà chúng ta định giao tiếp mã hóa cùng, chúng ta sẽ không bao giờ được nói chuyện với ai. Câu hỏi được cộng đồng mật mã đặt ra cách đây năm mươi năm là: liệu có thể để hai người không quen biết nhau và chỉ chia sẻ một kênh công khai thỏa thuận được, trên chính kênh công khai đó, một bí mật mà không ai nghe lén kênh có thể biết được?

Sự thanh lịch của Diffie-Hellman

Năm 1976, hai nhà toán học tên là Whitfield Diffie và Martin Hellman đã chứng minh một điều tưởng chừng như không thể: rằng hai người, chỉ nói chuyện qua một kênh công khai — một kênh mà bất kỳ ai cũng có thể nghe thấy mọi điều họ nói — có thể thỏa thuận một mật khẩu bí mật mà không một người nghe nào có thể khám phá ra. Nghe có vẻ giống như ma thuật. Nhưng không phải: đó là toán học. Trao đổi khóa Diffie-Hellman, như tên gọi của nó từ đó đến nay, là cơ sở cho hầu hết mọi giao tiếp mã hóa trên internet, và nửa thế kỷ sử dụng cường độ cao cũng như sự giám sát học thuật toàn cầu đã chứng thực sự vững chắc của nó. Ai muốn thấy trực giác hình ảnh hoặc toán học có thể tiếp tục đọc. Ai thích tin tưởng rằng nó hoạt động cũng có thể tiếp tục mà không làm mất mạch của bài viết.

Đối với những ai muốn hình dung nó qua một hình ảnh, có một phép so sánh phổ biến với màu sắc. Hãy tưởng tượng Alicia và Bruno công khai thỏa thuận một màu cơ bản — giả sử là màu vàng — trước mặt Eva, người đang nghe lén họ. Mỗi người chọn riêng một màu bí mật thứ hai và trộn bí mật của mình với màu vàng. Alicia có được một màu cam đặc thù; Bruno có được một màu xanh lá đặc thù. Họ trao đổi kết quả trước mặt Eva. Bây giờ mỗi người trộn màu nhận được với bí mật của chính mình, và cả hai đều đạt được cùng một màu cuối cùng, vì thứ tự trộn không quan trọng. Eva đã thấy màu vàng và hai hỗn hợp trung gian, nhưng không thấy các bí mật; nếu không có một trong các bí mật, cô ấy không thể có được màu cuối cùng. Toán học thực tế thay thế màu sắc bằng các phép lũy thừa trong các nhóm modulo hoặc đường cong elliptic, nhưng ý tưởng là như nhau: bí mật chung được xây dựng công khai mà không ai trong kênh có thể tái tạo lại được.

Trong số học, dành cho những ai thích xem cơ chế: Alicia chọn một số bí mật a, Bruno chọn b. Họ trao đổi g^a và g^b công khai trên kênh. Alicia tính $(g^b)^a$ và Bruno tính $(g^a)^b$; cả hai đều đạt được cùng một g^{ab}. Eva thấy g, g^a và g^b đi qua kênh, nhưng việc khôi phục a từ g^a — cái gọi là bài toán logarit rời rạc — đòi hỏi một thời gian tính toán thiên văn vượt xa tuổi của vũ trụ khi g được chọn trong một nhóm toán học phù hợp.

Dành cho bất cứ ai muốn kiểm tra nó bằng những con số nhỏ. Việc trao đổi Diffie-Hellman có thể được xem xét toàn bộ với các số liệu đủ nhỏ để làm phép toán bằng tay. Bất cứ ai không muốn đi sâu vào số học có thể bỏ qua khối này mà không làm mất mạch của bài viết; bất cứ ai muốn xem cơ chế hoạt động từng bước sẽ tìm thấy nó ở đây. **Các quy tắc công khai**, mà bất kỳ ai cũng có thể đọc: một số nguyên tố $p = 11$

(trong Diffie-Hellman thực tế, nó khoảng ba trăm chữ số; chúng tôi sử dụng số mười một để phép toán vừa trên một trang), một cơ số $g = 2$, và quy ước rằng tất cả các phép toán số học được thực hiện theo *modulo p* — bạn tính toán, chia cho p và giữ lại phần dư, giống như một chiếc đồng hồ mười một vị trí thiết lập lại về 0 khi vượt qua số mười. **Các lựa chọn riêng tư**, mỗi người một lựa chọn và không bao giờ chia sẻ: Alicia chọn $a = 4$. Bruno chọn $b = 7$.

Bước 1. Alicia tính $2^4 = 16$, sau đó $16 \bmod 11 = \mathbf{5}$. Cô ấy gửi số năm. Eva ghi lại nó.

Bước 2. Bruno tính $2^7 = 128$, sau đó $128 \bmod 11 = \mathbf{7}$. Anh ấy gửi số bảy. Eva cũng ghi lại nó. Sau hai lần truyền, cuốn sổ của Eva chứa bốn mẩu dữ liệu: $p = 11$, $g = 2$, A = 5, B = 7. Cô ấy đang thiếu con số chung mà Alicia và Bruno chuẩn bị suy ra — và là thứ mà Eva sẽ không thể tái tạo lại.

Bước 3. Alicia lấy số bảy mà Bruno gửi cho cô và nâng nó lên số mũ riêng tư của cô là $a = 4$. Để tránh xử lý số $7^4 = 2401$, nó được tính theo từng phần bằng cách áp dụng modulo ở mỗi bước:

$7^2 = 49$

$49 \bmod 11 = 5$

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

$25 \bmod 11 = 3$

Alicia nhận được số **3**.

Bước 4. Bruno lấy số năm mà Alicia gửi cho anh và nâng nó lên số mũ riêng tư của anh là $b = 7$. Một lần nữa theo từng phần:

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

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

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

Cuối cùng $5^7 = 5^6 \times 5 = 5 \times 5 = 25 \bmod 11 = \mathbf{3}$.

Bruno cũng nhận được **3**.

Cả hai đều đã đạt đến cùng một con số, 3, hoạt động song song. Không ai trong hai người gửi số mũ riêng tư của họ vào bất kỳ lúc nào. Alicia không biết rằng $b = 7$; Bruno không biết rằng $a = 4$. Mỗi người sử dụng giá trị công khai mà người kia gửi kết hợp với số mũ riêng tư của riêng họ và họ gặp nhau tại cùng một đích đến. **Tại sao họ lại đạt được cùng một con số?** Những gì mỗi người đã tính toán: Alicia, $(g^b)^a = 2^{7 \times 4} = 2^{28} \bmod 11$. Bruno, $(g^a)^b = 2^{4 \times 7} = 2^{28} \bmod 11$. Đó là cùng một số lượng vì thứ tự nhân các số mũ không quan trọng ($7 \times 4 = 4 \times 7$). Mỗi người đã đi qua một con đường khác nhau để đến cùng một đích.

Còn Eva thì sao? Cô ấy có trong sổ tay của mình $p = 11$, $g = 2$, A = 5, B = 7, và cô ấy muốn có số 3. Để tính được nó, cô ấy sẽ cần biết a hoặc b — nhưng cả hai đều không được truyền qua kênh. Cách duy nhất của cô ấy là tự hỏi: «đối với số mũ a nào thì $2^a \bmod 11 = 5$?». Với p nhỏ như vậy, cô ấy có thể thử 0, 1, 2, 3, 4... và tìm thấy nó trong vòng chưa đầy một phút. Nhưng khi thay thế số 11 bằng một số nguyên tố gồm ba trăm chữ số, không gian của các số mũ có thể có chứa nhiều phần tử hơn số nguyên tử trong vũ trụ quan sát được. **Hiện tại không có thuật toán nào được nhân loại biết đến có thể đi qua không gian đó trong thời gian ngắn hơn hàng tỷ năm.** Đây là cái gọi là *bài toán logarit rời rạc*: dễ theo chiều tiến, không thể tính toán theo chiều lùi. Và đó là lý do tại sao mã hóa vẫn chống cự được ngay cả khi Eva theo dõi toàn bộ cuộc trò chuyện qua từng chữ cái.

Ba thành phần đơn giản — số học đồng hồ, lũy thừa, và tính chất giao hoán của phép nhân ($a \cdot b = b \cdot a$) — kết hợp lại tạo ra một giao thức mà một nửa nhân loại phụ thuộc vào mỗi ngày cho các giao tiếp riêng tư của họ. Không có mảnh nào trong ba phần này, khi tách rời nhau, có vẻ đặc biệt. Điều quyết định là sự lắp ráp.

Từ Diffie-Hellman đến giao thức Signal

Mã hóa đầu cuối mà các ứng dụng nhắn tin chuyên nghiệp sử dụng ngày nay hầu như không có ngoại lệ, dựa trên một phiên bản thanh lịch và được tăng cường của trao đổi Diffie-Hellman. Giao thức Signal, được thiết kế bởi Trevor Perrin và Moxie Marlinspike từ năm 2013 đến 2016, là tài liệu tham khảo. Nó kết hợp hai ý tưởng chính. Thứ nhất, trao đổi khóa trong đường cong elliptic (X25519), tạo ra bí mật chung ban đầu giữa hai thiết bị. Thứ hai, cái gọi là Double Ratchet — bánh cóc kép —, tự động đổi mới các khóa với mỗi tin nhắn, sao cho việc xâm nhập thiết bị ngày hôm nay không cho phép giải mã các tin nhắn trong quá khứ, cũng như các tin nhắn trong tương lai sau khi bánh cóc đã được xoay.

Trong Zig, trao đổi X25519 tạo ra bí mật chung giữa hai thiết bị nằm gọn trong sáu dòng, sử dụng thư viện tiêu chuẩn:

```
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)
```

Điều gì xảy ra trong sáu dòng đó: Các khóa công khai di chuyển công khai. Các khóa bí mật không bao giờ rời khỏi thiết bị tương ứng. Mỗi bên rút ra, từ khóa bí mật của mình và khóa công khai của bên kia, cùng một bí mật ba mươi hai byte mà không ai trong kênh có thể khôi phục được. Bí mật đó sau này đóng vai trò là hạt giống (seed) để mã hóa các tin nhắn được trao đổi. Double Ratchet của giao thức Signal thêm một vòng xoay liên tục của vật liệu đó để việc xâm nhập tại một thời điểm không làm ảnh hưởng đến phần còn lại của cuộc hội thoại.

Và chính xác thì có gì bên trong `std.crypto.dh.X25519`? Không có phép thuật ẩn giấu nào cả. Chúng là hai hàm ngắn có thể được đọc toàn bộ trong chính thư viện tiêu chuẩn của Zig. Hàm đầu tiên suy ra khóa công khai từ khóa bí mật — «g^a» của quá trình trao đổi:

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

Theo ngôn ngữ của bài viết: khóa bí mật được «nhân» — theo nghĩa elip, không phải số học cơ bản — với điểm cơ sở của đường cong Curve25519 và kết quả được tuần tự hóa thành ba mươi hai byte. Phép toán `clampedMul` là phiên bản được tăng cường của phép nhân vô hướng đó: nó kết hợp các biện pháp bảo vệ mà cộng đồng mật mã đã bổ sung trong nhiều năm để chống lại các họ tấn công đã biết. Hai dòng thân hàm.

Hàm thứ hai kết hợp khóa bí mật của bạn với khóa công khai mà bên kia gửi cho bạn. Đó là «$(g^b)^a$» của quá trình trao đổi, thứ tạo ra bí mật chung dài ba mươi hai byte mà không ai trong hai bạn từng truyền đi:

```
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();
}
```

Thêm hai dòng nữa. Khóa công khai nhận được diễn giải như một điểm trên đường cong và được «nhân» với khóa bí mật của chính mình. Bằng tính chất giao hoán của phép toán trên đường cong — tương tự như tính chất giao hoán của phép nhân các số mũ mà chúng ta đã thấy trong ví dụ bằng số — cả hai bên đều đạt được cùng một điểm được tuần tự hóa: chính xác là bí mật chung mà bài viết nói đến.

Đó là tất cả. Điều trông giống như phép thuật trong một ứng dụng, trên thực tế, là hai hàm, mỗi hàm dài ba dòng. Độ phức tạp kỹ thuật tập trung trong một phép toán duy nhất, `clampedMul`, được viết ở phần sau trong cùng một thư viện tiêu chuẩn, được cộng đồng mật mã quốc tế xem xét trong nhiều thập kỷ và có sẵn cho bất kỳ ai muốn đọc nó từng chữ một. Không có hộp đen nào trong ứng dụng của chúng tôi hay trong thư viện tiêu chuẩn của Zig. Có mã nguồn mở mà con người có thể hiểu được, với việc lựa chọn tốc độ mà họ muốn đào sâu vào nó.

Mã hóa đầu cuối bảo vệ điều gì

Điều mà E2EE bảo vệ tốt, giả sử việc triển khai đúng đắn, là nội dung tin nhắn đang chuyển tiếp. Một máy chủ trung gian nhận và chuyển tiếp dữ liệu được mã hóa sẽ thấy một chuỗi các byte khó hiểu. Một kẻ tấn công có quyền truy cập vào cáp, bộ định tuyến, điểm truy cập wifi cũng sẽ thấy điều tương tự. Một nhà cung cấp dịch vụ lưu giữ các bản sao của lưu lượng truy cập sẽ không thể đọc được nó sau đó. Một Chính phủ ra lệnh cho nhà điều hành dịch vụ bàn giao nội dung sẽ nhận được cùng một chuỗi các byte khó hiểu mà máy chủ đã có lúc đầu.

Về mặt thực tế, điều này là rất nhiều. Đó là sự khác biệt giữa việc viết một lá thư bên trong một chiếc phong bì đục và việc viết nó trên một tấm bưu thiếp. Cả hai đều đến nơi. Nhưng chỉ có một cách bảo vệ được nội dung trước người đưa thư.

Mã hóa đầu cuối không bảo vệ điều gì

Điều này cũng rất đáng để biết. E2EE không bảo vệ siêu dữ liệu (metadata): máy chủ vẫn biết người dùng A gửi dữ liệu cho người dùng B, vào giờ nào, với tần suất bao nhiêu và từ đâu, mặc dù nó không biết họ nói gì. Những siêu dữ liệu này, như chúng tôi đã lập luận trong *Mã hóa không có nghĩa là riêng tư*, thường tiết lộ nhiều điều hơn cả nội dung. Việc biết rằng ai đó đã gọi cho một công ty luật chuyên về ly hôn vào tối thứ Sáu lúc 22:00 trong ba mươi phút kể một câu chuyện mà nội dung cuộc gọi không bao giờ kể. Đó cũng là tình huống giống như việc thấy một người ra vào nhiều lần một phòng khám ung bướu: không cần nghe bất cứ điều gì được nói bên trong để hình dung điều gì đang xảy ra. Một siêu dữ liệu riêng lẻ có thể không có ý nghĩa gì; nhưng nhiều dữ liệu đan chéo nhau sẽ vẽ nên một thứ gì đó quá giống với sự thật. E2EE không bảo vệ các đầu cuối: nếu thiết bị của người nhận bị xâm nhập bởi một chương trình độc hại, tin nhắn sẽ được giải mã bình thường cho người nhận đó và chương trình độc hại sẽ đọc nó. E2EE không tự bảo vệ chống lại danh tính của người đối thoại: nếu Alicia tin rằng mình đang nói chuyện với Bruno nhưng kẻ tấn công đã xen vào ngay từ đầu (một cuộc tấn công *man in the middle*) và giao thức không bao gồm xác minh độc lập, hai bên sẽ kết thúc bằng việc nói chuyện với kẻ xâm nhập trong khi nghĩ rằng họ đang nói chuyện với nhau.

Có một điều thứ tư đáng để phát biểu không mơ hồ. E2EE không ngăn cản một nhà cung cấp tự nhận là cung cấp nó giữ thêm một bản sao tin nhắn chưa mã hóa trong hệ thống của chính họ. Khẳng định «tin nhắn của tôi được mã hóa đầu cuối» và khẳng định «nhà cung cấp không lưu giữ nội dung của tôi» không phải là một. Một ứng dụng có thể thực hiện khẳng định thứ nhất trong khi vi phạm khẳng định thứ hai; chúng ta đã thấy điều này trên các tiêu đề báo chí lặp đi lặp lại kể từ năm 2018. Người dùng, trừ khi mã nguồn của ứng dụng khách có thể xác minh được, không có cách nào về kỹ thuật để phân biệt trường hợp này với trường hợp kia mà không có sự điều tra của chuyên gia. Trường hợp được công chúng biết đến nhiều nhất: WhatsApp mã hóa tin nhắn đầu cuối khi đang chuyển tiếp, nhưng nếu người dùng kích hoạt sao lưu trong iCloud hoặc Google Drive mà không có mã hóa bổ sung, bản sao đó sẽ được lưu trữ dưới dạng có thể đọc được trong cơ sở hạ tầng của bên thứ ba, và việc mã hóa bị phá vỡ tại chính đầu của người dùng.

Câu hỏi mà nhà điều hành không muốn nghe

Một ứng dụng tự nhận là mã hóa đầu cuối có thể, về mặt kỹ thuật, làm một trong ba điều liên quan đến các khóa:

1. **Các khóa chỉ nằm trên thiết bị.** Chúng được tạo ra và nằm duy nhất trên thiết bị của người dùng; nhà điều hành không biết và cũng không lưu trữ chúng. Đây là trường hợp tối ưu.
2. **Nhà điều hành có thể truy cập nếu họ muốn.** Nhà điều hành sở hữu các khóa của người dùng (hoặc có thể tạo chúng theo ý muốn) và lưu trữ chúng trong cơ sở dữ liệu của mình. Nếu muốn hoặc bị ép buộc, họ có thể đọc nội dung. Đây là trường hợp của hầu hết các dịch vụ «đám mây».
3. **Nhà điều hành không thể truy cập theo thiết kế, nhưng kiểm soát quyền truy cập.** Nhà điều hành không có các khóa, nhưng kiểm soát ứng dụng tạo ra chúng. Nếu bị ép buộc, họ có thể gửi một bản cập nhật độc hại để thu thập các khóa hoặc nội dung trước khi mã hóa. Đây là trường hợp của nhiều dịch vụ E2EE thương mại.

Do đó, câu hỏi vận hành không phải là liệu một thứ gì đó có được mã hóa hay không, mà là ai có quyền kiểm soát thiết bị và phần mềm quản lý các khóa. Trong Solo2, các khóa chỉ nằm trong «Hầm» của bạn (IndexedDB được mã hóa bằng mật khẩu của bạn) và phần mềm là mã nguồn mở có thể xác minh được.

Dành cho độc giả chuyên nghiệp

Mã hóa đầu cuối là một công cụ cho chủ quyền kỹ thuật số. Nhưng cũng như mọi công cụ, hiệu quả của nó phụ thuộc vào bàn tay cầm nó và nền đất mà nó tựa vào.

1. Các khóa mật mã được tạo ra ở đâu và chúng cư trú vật lý ở đâu? Nếu nhà điều hành có thể truy cập chúng (thậm chí tạm thời, ngay cả dưới vỏ bọc phục hồi), thì E2EE chỉ là danh nghĩa.
2. Có sự xác minh độc lập nào đối với người đối thoại (số bảo mật, mã QR, so sánh ngoài băng tần) để ngăn chặn cuộc tấn công man-in-the-middle trong quá trình thiết lập cuộc trò chuyện không?
3. Mã máy khách có thể kiểm toán được không — mở, được công bố, có thể tái tạo — hay nó yêu cầu phải tin vào lời của nhà cung cấp về những gì máy khách thực sự làm?
4. Dịch vụ tạo và giữ những siêu dữ liệu nào, và trong bao lâu? Ngay cả khi nội dung là không rõ ràng, siêu dữ liệu có thể tái tạo lại một phần lớn thông tin nhạy cảm.

Bốn câu hỏi này không yêu cầu thông tin kỹ thuật nâng cao; chúng yêu cầu thông tin mà bất kỳ nhà điều hành trung thực nào cũng có thể trả lời trong tài liệu công khai của họ. Chất lượng và độ chính xác của câu trả lời nói lên nhiều điều về sản phẩm cũng như chính câu trả lời.

Mã hóa đầu cuối, nếu được thực hiện tốt, là một trong những cấu trúc tinh vi nhất mà mật mã học đương đại đã mang lại cho thực tiễn hàng ngày. Ý tưởng ban đầu — hai người có thể thỏa thuận một bí mật trên một kênh công khai — thuộc về Whitfield Diffie và Martin Hellman, 1976; nửa thế kỷ sau chúng ta vẫn đang sống trong hệ quả của nó. Nhưng, như với bất kỳ lời hứa kỹ thuật nào, giá trị của nó phụ thuộc vào việc thực hiện thực tế, chứ không phải ở nhãn dán. Câu hỏi của người chuyên nghiệp trung thực không phải là «nó có được mã hóa không?», mà là «ai giữ chìa khóa?». Các câu trả lời có những hệ quả khác nhau. Nên biết chúng.

Nguồn tham khảo và đọc thêm

- Diffie, W.; Hellman, M. — *New Directions in Cryptography*, IEEE Transactions on Information Theory, tháng 11 năm 1976. Bài báo nền tảng của mật mã khóa công khai.
- Perrin, T.; Marlinspike, M. — *The Double Ratchet Algorithm*, đặc tả công khai của Open Whisper Systems, bản sửa đổi năm 2016. Cơ sở của giao thức Signal và các dẫn xuất công nghiệp của nó.
- RFC 7748 — Elliptic Curves for Security (IETF, tháng 1 năm 2016). Đặc tả quy chuẩn của các đường cong X25519 và X448 được sử dụng trong các trao đổi khóa hiện đại.
- Ferguson, N.; Schneier, B.; Kohno, T. — *Cryptography Engineering: Design Principles and Practical Applications* (Wiley, 2010). Các chương về trao đổi khóa và các giao thức mã hóa được xác thực.
- Quy định (EU) 2024/1183 về khuôn khổ nhận dạng kỹ thuật số châu Âu (eIDAS 2) — thiết lập các khuôn khổ nơi việc xác minh độc lập người đối thoại nhận được sự hỗ trợ về mặt thể chế, và nơi sự phân biệt giữa mã hóa danh nghĩa và thực tế có những hậu quả pháp lý khác nhau.

← TrướcKill switch và sự thâu tóm thể chếTiếp theo →Mô hình kinh doanh như một tín hiệu của sự tin cậy

Các bài đọc gần đây

- Phân tích · Ngày 18 tháng 5 năm 2026 Quyền riêng tư thực sự vs biểu hiện: những câu hỏi bạn nên tự đặt ra
- Phân tích · Ngày 18 tháng 5 năm 2026 Self-hosting như một thực hành chuyên nghiệp
- Khái niệm · Ngày 18 tháng 5 năm 2026 24 từ: danh tính mật mã là gì

Tải bài viết này để sử dụng ở bất cứ đâu bạn cần.

↓ Markdown ↓ Văn bản thuần túy ↓ PDF

Tệp sẽ được tải xuống thiết bị của bạn. Từ đó, bạn có thể lưu trữ, nhập vào Solo2 hoặc chia sẻ ở bất cứ đâu. Cuadernos không quyết định điểm đến thay cho bạn.

Dấu sáp · SHA-256 3950f4b23489eaa3e54802b9b5743bdfa66872e05a052cfe77cda3ced28eb05a

Cuadernos Lacre · Một ấn phẩm của Menzuri Gestión S.L. ·
viết bởi R.Eugenio · được biên tập bởi đội ngũ Solo2.