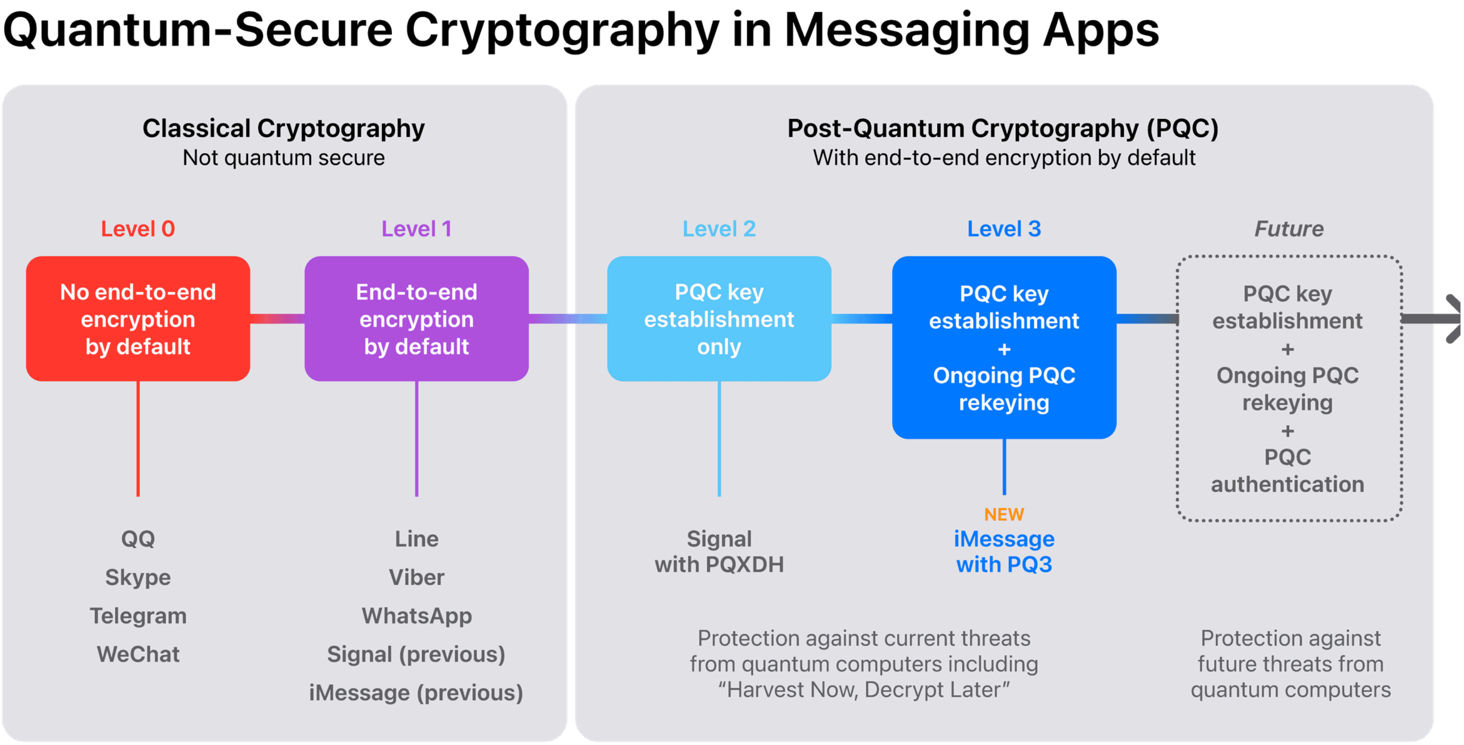
Blog

February 21, 2024

iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

Posted by Apple Security Engineering and Architecture (SEAR)

Today we are announcing the most significant cryptographic security upgrade in iMessage history with the introduction of PQ3, a groundbreaking post-quantum cryptographic protocol that advances the state of the art of end-to-end secure messaging. With compromise-resilient encryption and extensive defenses against even highly sophisticated quantum attacks, PQ3 is the first messaging protocol to reach what we call Level 3 security — providing protocol protections that surpass those in all other widely deployed messaging apps. To our knowledge, PQ3 has the strongest security properties of any at-scale messaging protocol in the world.


Quantum-Secure Cryptography in Messaging Apps
Classical Cryptography
Not quantum secure
Post-Quantum Cryptography (PQC)
With end-to-end encryption by default
Level 0
No end-to-end encryption by default
Level 1
End-to-end encryption by default
Level 2
PQC key establishment only
Level 3
PQC key establishment + Ongoing PQC rekeying
Future
PQC key establishment + Ongoing PQC rekeying + PQC authentication
QQ
Skype
Telegram
WeChat
Line
Viber
WhatsApp
Signal (previous)
iMessage (previous)
Signal with PQXDH
NEW
iMessage with PQ3
Protection against current threats from quantum computers including "Harvest Now, Decrypt Later"
Protection against future threats from quantum computers

Note: This comparison evaluates only the cryptographic aspect of messaging security, and therefore focuses on end-to-end encryption and quantum security. Such a comparison doesn't include automatic key verification, which we believe is a critical protection for modern messaging apps. As of the time of this writing, only iMessage and WhatsApp provide automatic key verification. The iMessage implementation, called Contact Key Verification, is the state of the art – it provides the broadest automatic protections and applies across all of a user's devices.

When iMessage launched in 2011, it was the first widely available messaging app to provide end-to-end encryption by default, and we have significantly upgraded its cryptography over the years. We most recently strengthened the iMessage cryptographic protocol in 2019 by switching from RSA to Elliptic Curve cryptography (ECC), and by protecting encryption keys on device with the Secure Enclave, making them significantly harder to extract from a device even for the most sophisticated adversaries. That protocol update went even further with an additional layer of defense: a periodic rekey mechanism to provide cryptographic self-healing even in the extremely unlikely case that a key ever became compromised. Each of these advances were formally verified by symbolic evaluation, a best practice that provides strong assurances of the security of cryptographic protocols.

Historically, messaging platforms have used classical public key cryptography, such as RSA, Elliptic Curve signatures, and Diffie-Hellman key exchange, to establish secure end-to-end encrypted connections between devices. All these algorithms are based on difficult mathematical problems that have long been considered too computationally intensive for computers to solve, even when accounting for Moore's law. However, the rise of quantum computing threatens to change the equation. A sufficiently powerful quantum computer could solve these classical mathematical problems in fundamentally different ways, and therefore — in theory — do so fast enough to threaten the security of end-to-end encrypted communications.

Although quantum computers with this capability don't exist yet, extremely well-resourced attackers can already prepare for their possible arrival by taking advantage of the steep decrease in modern data storage costs. The premise is simple: such attackers can collect large amounts of today's encrypted data and file it all away for future reference. Even though they can't decrypt any of this data today, they can retain it until they acquire a quantum computer that can decrypt it in the future, an attack scenario known as *Harvest Now, Decrypt Later*.

To mitigate risks from future quantum computers, the cryptographic community has been working on post-quantum cryptography (PQC): new public key algorithms that provide the building blocks for quantum-secure protocols but don't require a quantum computer to run — that is, protocols that can run on the classical, non-quantum computers we're all using today, but that will remain secure from known threats posed by future quantum computers.

To reason through how various messaging applications mitigate attacks, it's helpful to place them along a spectrum of security properties. There's no standard comparison to employ for this purpose, so we lay out our own simple, coarse-grained progression of messaging security levels in the image at the top of this post: we start on the left with classical cryptography and progress towards quantum security, which addresses current and future threats from quantum computers. Most existing messaging apps fall either into Level 0 — no end-to-end encryption by default and no quantum security — or Level 1 — with end-to-end encryption by default, but with no quantum security. A few months ago, Signal added support for the PQXDH protocol, becoming the first large-scale messaging app to introduce post-quantum security in the initial key establishment. This is a welcome and critical step that, by our scale, elevated Signal from Level 1 to Level 2 security.

At Level 2, the application of post-quantum cryptography is limited to the initial key establishment, providing quantum security only if the conversation key material is never compromised. But today's sophisticated adversaries already have incentives to compromise encryption keys, because doing so gives them the ability to decrypt messages protected by those keys for as long as the keys don't change. To best protect end-to-end encrypted messaging, the post-quantum keys need to change on an ongoing basis to place an upper bound on how much of a conversation can be exposed by any single, point-in-time key compromise — both now and with future quantum computers. Therefore, we believe messaging protocols should go even further and attain Level 3 security, where post-quantum cryptography is used to secure both the initial key establishment and the ongoing message exchange, with the ability to rapidly and automatically restore the cryptographic security of a conversation even if a given key becomes compromised.

iMessage now meets this goal with a new cryptographic protocol that we call PQ3, offering the strongest protection against quantum attacks and becoming the only widely available messaging service to reach Level 3 security. Support for PQ3 will start to roll out with the public releases of iOS 17.4, iPadOS 17.4, macOS 14.4, and watchOS 10.4, and is already in the corresponding developer preview and beta releases. iMessage conversations between devices that support PQ3 are automatically ramping up to the post-quantum encryption protocol. As we gain operational experience with PQ3 at the massive global scale of iMessage, it will fully replace the existing protocol within all supported conversations this year.

Designing PQ3

More than simply replacing an existing algorithm with a new one, we rebuilt the iMessage cryptographic protocol from the ground up to advance the state of the art in end-to-end encryption, and to deliver on the following requirements:

- Introduce post-quantum cryptography from the start of a conversation, so that all communication is protected from current and future adversaries.
- Mitigate the impact of key compromises by limiting how many past and future messages can be decrypted with a single compromised key.
- Use a hybrid design to combine new post-quantum algorithms with current Elliptic Curve algorithms, ensuring that PQ3 can can never be less safe than the existing classical protocol.
- Amortize message size to avoid excessive additional overhead from the added security.
- Use formal verification methods to provide strong security assurances for the new protocol.

PQ3 introduces a new post-quantum encryption key in the set of public keys each device generates locally and transmits to Apple servers as part of iMessage registration. For this application, we chose to use Kyber post-quantum public keys, an algorithm that received close scrutiny from the global cryptography community, and was selected by NIST as the Module Lattice-based Key Encapsulation Mechanism standard, or ML-KEM. This enables sender devices to obtain a receiver's public keys and generate post-quantum encryption keys for the very first message, even if the receiver is offline. We refer to this as initial key establishment.

We then include — within conversations — a periodic post-quantum rekeying mechanism that has the ability to self-heal from key compromise and protect future messages. In PQ3, the new keys sent along with the conversation are used to create fresh message encryption keys that can't be computed from past ones, thereby bringing the conversation back to a secure state even if previous keys were extracted or compromised by an adversary. PQ3 is the first large scale cryptographic messaging protocol to introduce this novel post-quantum rekeying property.

PQ3 employs a hybrid design that combines Elliptic Curve cryptography with post-quantum encryption both during the initial key establishment and during rekeying. Thus, the new cryptography is purely additive, and defeating PQ3 security requires defeating both the existing, classical ECC cryptography and the new post-quantum primitives. It also means the protocol benefits from all the experience we accumulated from deploying the ECC protocol and its implementations.

Rekeying in PQ3 involves transmitting fresh public key material in-band with the encrypted messages that devices are exchanging. A new public key based on Elliptic Curve Diffie-Hellman (ECDH) is transmitted inline with every response. The post-quantum key used by PQ3 has a significantly larger wire size than the existing protocol, so to meet our message size requirement we designed the quantum-secure rekeying to happen periodically rather than with every message. To determine whether a new post-quantum key is transmitted, PQ3 uses a rekeying condition that aims to balance the average size of messages on the wire, preserve the user experience in limited connectivity scenarios, and keep the global volume of messages within the capacity of our server infrastructure. Should the need arise, future software updates can increase the rekeying frequency in a way that's backward-compatible with all devices that support PQ3.

With PQ3, iMessage continues to rely on classical cryptographic algorithms to authenticate the sender and verify the Contact Key Verification account key, because these mechanisms can't be attacked retroactively with future quantum computers. To attempt to insert themselves in the middle of an iMessage conversation, an adversary would require a quantum computer capable of breaking one of the authentication keys before or at the time the communication takes place. In other words, these attacks cannot be performed in a *Harvest Now, Decrypt Later* scenario — they require the existence of a quantum computer capable of performing the attacks contemporaneously with the communication being attacked. We believe any such capability is still many years away, but as the threat of quantum computers evolves, we will continue to assess the need for post-quantum authentication to thwart such attacks.

A formally proven protocol

Our final requirement for iMessage PQ3 is formal verification — a mathematical proof of the intended security properties of the protocol. PQ3 received extensive review from Apple's own multi-disciplinary teams in Security Engineering and Architecture (SEAR) as well as from some of the world's foremost experts in cryptography. This includes a team led by Professor David Basin, head of the Information Security Group at ETH Zürich and one of the inventors of Tamarin — a leading security protocol verification tool that was also used to evaluate PQ3 — as well as Professor Douglas Stebila from the University of Waterloo, who has performed extensive research on post-quantum security for internet protocols. Each took a different but complementary approach, using different mathematical models to demonstrate that as long as the underlying cryptographic algorithms remain secure, so does PQ3. Finally, a leading third-party security consultancy supplemented our internal implementation review with an independent assessment of the PQ3 source code, which found no security issues.

In the first mathematical analysis, Security analysis of the iMessage PQ3 protocol, Professor Douglas Stebila focused on so-called game-based proofs. This technique, also known as reduction, defines a series of "games" or logical statements to show that the protocol is at least as strong as the algorithms that underpin it. Stebila's analysis shows that PQ3 provides confidentiality even in the presence of some key compromises against both classical and quantum adversaries, in both the initial key establishment and the ongoing rekeying phase of the protocol. The analysis decomposes the many layers of key derivations down to the message keys and proves that, for an attacker, they are indistinguishable from random noise. Through an extensive demonstration that considers different attack paths for classical and quantum attackers in the proofs, Stebila shows that the keys used for PQ3 are secure as long as either the Elliptic Curve Diffie-Hellman problem remains hard or the Kyber post-quantum KEM remains secure.

> *The iMessage PQ3 protocol is a well-designed cryptographic protocol for secure messaging that uses state-of-the-art techniques for end-to-end encrypted communication. In my analysis using the reductionist security methodology, I confirmed that the PQ3 protocol provides post-quantum confidentiality, which can give users confidence in the privacy of their communication even in the face of potential improvements in quantum computing technology. —Professor Douglas Stebila*

In the second evaluation, A Formal Analysis of the iMessage PQ3 Messaging Protocol, Prof. David Basin, Felix Linker, and Dr. Ralf Sasse at ETH Zürich use a method called symbolic evaluation. As highlighted in the paper's abstract, this analysis includes a detailed formal model of the iMessage PQ3 protocol, a precise specification of its fine-grained security properties, and machine-checked proofs using the state-of-the-art symbolic Tamarin prover. The evaluation yielded a fine-grained analysis of the secrecy properties of PQ3, proving that "in the absence of the sender or recipient being compromised, all keys and messages transmitted are secret" and that "compromises can be tolerated in a well-defined sense where the effect of the compromise on the secrecy of data is limited in time and effect," which confirms that PQ3 meets our goals.

> *We provide a mathematical model of PQ3 as well as prove its secrecy and authenticity properties using a verification tool for machine-checked security proofs. We prove the properties even when the protocol operates in the presence of very strong adversaries who can corrupt parties or possess quantum computers and therefore defeat classical cryptography. PQ3 goes beyond Signal with regards to post-quantum defenses. In PQ3, a post-quantum secure algorithm is part of the ratcheting and used repeatedly, rather than only once in the initialization as in Signal. Our verification provides a very high degree of assurance that the protocol as designed functions securely, even in the post-quantum world. —Professor David Basin*

Diving into the details

Because we know PQ3 will be of intense interest to security researchers and engineers as well as the cryptographic community, this blog post is really two posts in one. Up to now, we laid out our design goals, outlined how PQ3 meets them, and explained how we verified our confidence in the protocol with independent assessments. If you'd like to understand more detail about the cryptographic underpinnings, the remainder of the post is a deeper dive into how we constructed the PQ3 protocol.

Post-quantum key establishment

iMessage allows a user to register multiple devices on the same account. Each device generates its own set of encryption keys, and the private keys are never exported to any external system. The associated public keys are registered with Apple's Identity Directory Service (IDS) to enable users to message each other using a simple identifier: email address or phone number. When a user sends a message from one of their devices, all of their other devices and all of the recipient's devices receive the message. The messages are exchanged through pair-wise sessions established between the sending device and each receiving device. The same message is encrypted successively to each receiving device, with keys uniquely derived for each session. For the rest of this description, we will focus on a single device-to-device session.

Because the receiving device might not be online when the conversation is established, the first message in a session is encrypted using the public encryption keys registered with the IDS server.

Each device with PQ3 registers two public encryption keys and replaces them regularly with fresh ones:

1. A post-quantum Kyber-1024 key encapsulation public key
2. A classical P-256 Elliptic Curve key agreement public key

These encryption keys are signed with ECDSA using a P-256 authentication key generated by the device's Secure Enclave, along with a timestamp used to limit their validity. The device authentication public key is itself signed by the Contact Key Verification account key, along with some attributes such as the supported cryptographic protocol version. This process allows the sender to verify that the recipient device's public encryption keys were uploaded by the intended recipient, and it guards against downgrade attacks.

When Alice's device instantiates a new session with Bob's device, her device queries the IDS server for the key bundle associated with Bob's device. The subset of the key bundle that contains the device's authentication key and versioning information is validated using Contact Key Verification. The device then validates the signature covering the encryption keys and timestamps, which attests that the keys are valid and have not expired.

Alice's device can then use the two public encryption keys to share two symmetric keys with Bob. The first symmetric key is computed through an ECDH key exchange that combines an ephemeral encryption key from Alice with Bob's registered P-256 public key. The second symmetric key is obtained from a Kyber key encapsulation with Bob's post-quantum public key.

To combine these two symmetric keys, we first extract their entropy by invoking HKDF-SHA384-Extract twice — once for each of the keys. The resulting 48-byte secret is further combined with a domain separation string and session information — which includes the user's identifiers, the public keys used in the key exchange, and the encapsulated secret — by invoking HKDF-SHA384-Extract again to derive the session's initial keying state. This combination ensures that the initial session state cannot be derived without knowing both of the shared secrets, meaning an attacker would need to break both algorithms to recover the resulting secret, thus satisfying our hybrid security requirement.

Post-quantum rekeying

Ongoing rekeying of the cryptographic session is designed such that keys used to encrypt past and future messages cannot be recomputed even by a powerful hypothetical attacker who is able to extract the cryptographic state of the device at a given point in time. The protocol generates a new unique key for each message, which periodically includes new entropy that is not deterministically derived from the current state of the conversation, effectively providing self-healing properties to the protocol. Our rekeying approach is modeled after ratcheting, a technique that consists of deriving a new session key from other keys and ensuring the cryptographic state always moves forward in one direction. PQ3 combines three ratchets to achieve post-quantum encryption.

The first ratchet, called the symmetric ratchet, protects older messages in a conversation to achieve forward secrecy. For every message, we derive a per-message encryption key from the current session key. The current session key itself is then further derived into a new session key, ratcheting the state forward. Each message key is deleted as soon as a corresponding message is decrypted, which prevents older harvested ciphertexts from being decrypted by an adversary who is able to compromise the device at a later time, and provides protection against replayed messages. This process uses 256-bit keys and intermediate values, and HKDF-SHA384 as a derivation function, which provides protection against both classical and quantum computers.

The second ratchet, called the ECDH ratchet, protects future messages by updating the session with fresh entropy from an Elliptic Curve key agreement, ensuring that an adversary loses the ability to decrypt new messages even if they had compromised past session keys — a property called post-compromise security. The ECDH-based ratchet has a symmetrical flow: the private key of the outgoing ratchet public key from the sender is used with the last public key received from the recipient to establish a new shared secret between sender and receiver, which is then mixed into the session's key material. The new PQ3 protocol for iMessage uses NIST P-256 Elliptic Curve keys to perform this ratchet, which imposes only a small 32-byte overhead on each message.

Because the second ratchet uses classical cryptography, PQ3 also adds a conditionally executed Kyber KEM-based ratchet. This third ratchet complements the ECDH-based ratchet to provide post-compromise security against *Harvest Now, Decrypt*

Later quantum attacks as well.

The use of a post-quantum ratchet can cause significant network overhead compared to an ECDH-based ratchet at the same security level. The post-quantum KEM requires sending both a public key and an encapsulated secret instead of a single outgoing public key. In addition, the underlying mathematical structure for quantum security requires significantly larger parameter sizes for public keys and encapsulated keys compared to Elliptic Curves.

To limit the size overhead incurred by frequent rekeying while preserving a high level of security, the post-quantum KEM is instantiated with Kyber-768. Unlike the IDS-registered public keys used for the initial key establishment, ratcheting public keys are used only once to encapsulate a shared secret to the receiver, significantly limiting the impact of the compromise of a single key. However, while a 32-byte ECDH-based ratchet overhead is acceptable on every message, the post-quantum KEM ratchet increases the message size by more than 2 kilobytes. To avoid visible delays in message delivery when device connectivity is limited, this ratchet needs to be amortized over multiple messages.

We therefore implemented an adaptive post-quantum rekeying criterion that takes into account the number of outgoing messages, the time elapsed since last rekeying, and current connectivity conditions. At launch, this means the post-quantum ratchet is performed approximately every 50 messages, but the criterion is bounded such that rekeying is always guaranteed to occur at least once every 7 days. And as we mentioned earlier, as the threat of quantum computers and infrastructure capacity evolves over time, future software updates can increase the rekeying frequency while preserving full backward compatibility.

Completing the public key ratchets, whether based on ECDH or Kyber, requires sending and receiving a message. Although users may not immediately reply to a message, iMessage includes encrypted delivery receipts that allow devices to rapidly complete the ratchet even without a reply from the recipient, as long as the device is online. This technique avoids delays in the rekeying process and helps support strong post-compromise recovery.

Similar to the initial session key establishment, the secrets established through the three ratchets are all combined with an evolving session key using HKDF-SHA384 through sequential calls to the Extract function. At the end of this process, we obtain a final message key, which can now be used to encrypt the payload.

Padding and encryption

To avoid leaking information about the message size, PQ3 adds padding to the message before encryption. This padding is implemented with the Padmé heuristic, which specifically limits the information leakage of ciphertexts with maximum length M to a practical optimum of O(log log M) bits. This is comparable to padding to a power of two but results in a lower overhead of at most 12 percent and even lower for larger payloads. This approach strikes an excellent balance between privacy and efficiency, and preserves the user experience in limited device connectivity scenarios.

The padded payload is encrypted with AES-CTR using a 256-bit encryption key and initialization vector, both derived from the message key. While public key algorithms require fundamental changes to achieve quantum security, symmetric cryptography algorithms like the AES block cipher only require doubling the key size to maintain their level of security against quantum computers.

Authentication

Each message is individually signed with ECDSA using the elliptic curve P-256 device authentication key protected by the Secure Enclave. The receiving device verifies the mapping between the sender’s identifier (email address or phone number) and the public key used for signature verification. If both users have enabled Contact Key Verification and verified each other’s account key, the device verifies that the device authentication keys are present in the Key Transparency log and that the corresponding account key matches the account key stored in the user’s iCloud Keychain.

The device’s authentication key is generated by the Secure Enclave and never exposed to the rest of the device, which helps prevent extraction of the private key even if the Application Processor is completely compromised. If an attacker were to compromise the Application Processor, they might be able to use the Secure Enclave to sign arbitrary messages. But after the device recovers from the compromise through a reboot or a software update, they would no longer be able to impersonate the user. This approach offers stronger guarantees than other messaging protocols where the authentication key is sometimes shared between devices or where the authentication takes place only at the beginning of the session.

The message signature covers a wide range of fields, including the unique identifiers of the users and their push notification tokens, the encrypted payload, authenticated data, a ratchet-derived message key indicator that binds the signature to a unique location in the ratchet, and any public key information used in the protocol. The inclusion of these fields in the signature guarantees that the message can only be used in the context intended by the sender, and all the fields are exhaustively documented in the research papers from Stebila, Basin, and collaborators.

Conclusion

End-to-end encrypted messaging has seen a tremendous amount of innovation in recent years, including significant advances in post-quantum cryptography from Signal’s PQXDH protocol and in key transparency from WhatsApp’s Auditable Key Directory. Building on its pioneering legacy as the first widely available messaging app to provide end-to-end encryption by default, iMessage has continued to deliver advanced protections that surpass existing systems. iMessage Contact Key Verification is the most sophisticated key transparency system for messaging deployed at scale, and is the current global state of the art for automatic key verification. And the new PQ3 cryptographic protocol for iMessage combines post-quantum initial key establishment with three ongoing ratchets for self-healing against key compromise, defining the global state of the art for protecting messages against *Harvest Now, Decrypt Later* attacks and future quantum computers.

Copyright © 2024 Apple Inc. All rights reserved.