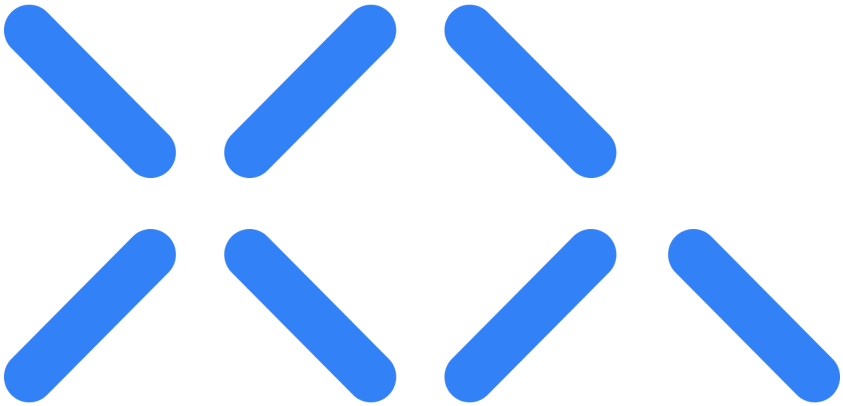


XQ Message

Zero-Trust Data Protection Platform

xqmsg.co

Technology Overview

(Updated Jan 2022)

Overview

XQ provides frictionless zero-trust data protection. Especially for systems that require data integrity and confidentiality, XQ has been integrated into data lakes, digital twins, IoT, and critical infrastructure in the cloud, on-prem, or hybrid solutions. Protecting your data means you can be confident that your data is safe, wherever it may travel.

The XQ platform delivers data provenance, residency, compliance, and logging and visualizing throughout its life.

Introduction

Data is growing exponentially as networking and computational systems impact every aspect of the economy from how services are delivered and products are manufactured to how we live. Not surprisingly, as the amount of data increases so do cyberattacks from familiar identity theft and ransomware to new state-of-the-art data lake exfiltration. Unfortunately, traditional approaches to protecting data don't work with state-of-the-art computing processes or networks. To address the need to protect data that is either moving across networks or is accessed by multiple software processes, XQ Message has developed a new form of data protection that is based on a Zero Trust Architecture.

XQ Zero Trust Data Protection

XQ implements a Zero Trust security model in which encryption keys are generated on edge systems (which can be a mobile app, server program or even a new 5G IoT gateway). Data is then encrypted using crypto-agile techniques. The encrypted data is wrapped with a metatag which serves as a pointer to the policies set by the data owner and context of the edge encrypting application. The policies and keys for access and authorization are sent to a key cache.

XQ data protection compliments existing security controls. XQ only handles policy-based key management and never the data itself. The XQ backend never touches the data nor knows anything about the edge devices except identity and authorization. Every

customer has their own XQ key cache that can be cloud-hosted or run on a physical server.

With XQ, every data transaction can have a different key. This key rotation significantly improves traditional encryption, where a single key is used to protect large data blocks. Because each data transaction has its own key, tracking and logging capabilities embedded within XQ monitor any attempt to access the data. When a security interaction occurs with the data, the identities, time, location, and event type are logged. This logging provides several benefits. First, data provenance for each transaction is produced through a complete chain of custody. Second, data residency is established by geolocating each transaction. Third, compliance by logging every interaction. These logs can be fed into any system view webhooks, API calls, or log exports.

XQ is ideal for data transfer and storage for or from data lakes, digital twins, IoT, and critical infrastructure in the cloud, on-prem, or hybrid solutions.

Use case examples include moving data from IoT sensors to Smart Energy and mobile phones to Transportation systems. Systems that require zero-trust security for provenance, data integrity, or data residency should implement XQ.

Exfiltration Alerts

This automatic logging helps security teams meet compliance requirements and detect data exfiltration attempts instantly. When XQ data is kept encrypted at rest, any attempt to access it by unauthorized or fake identities is immediately reported and flagged. Each piece of data or data lake entry is encrypted and tracked with a separate key and access policies. Applications with privileged access never have total data access, so a compromised entity does not grant the adversary complete access to the data. Since keys are stored separately, once the data is exfiltrated, the keys can be deleted, and the exfiltrated data remains encrypted indefinitely. This approach also prevents ransomware attackers from threatening to publish data in double-dipping extortion schemes forcing the victim to repeatedly pay to keep the attackers from posting their trade secrets and clients' personally identifiable data to the web.

Dynamic Policing

XQ focuses on protecting data and not the device or network. This provides powerful new flexibility to data authorization rights and policies. It is a completely new kind of granular data control.

XQ empowers data owners to set authorization rights for data as it moves from the edge to the cloud or between cloud systems.

For instance, an entity onboarding a vendor and providing that vendor access to certain data can time box or revoke access to the data at the end of the project. All-access to that data is tracked and logged. A new dataset can be provisioned for the vendor for the next project.

XQ Technology Innovations

Crypto-Agile

XQ's patented design streams entropy in the form of Quantum Random Numbers (QRNG) to edge systems such as a mobile phone, server, or IoT Gateway, where the entropy is used to generate a local encryption key. That key encrypts the data using any crypto algorithm whatsoever. The key is then posted to XQ's key cache with retrieval policies such as authorization rights. XQ's crypto-agile architecture enables solution architects and software developers to select the best algorithm for their projects.

Embeddable

XQ does not require current technology stacks to be uprooted and replaced but instead is an Encryption as a Service (EaaS) solution that is layered on top of existing security controls to enhance security via more robust encryption and secure key management. Users and applications encrypt data on edge devices either via XQ's applications or via XQ's APIs and SDKs.

Zero Trust Access Control

Only software programs authorized for the data or message access can receive the decryption keys. As a result, any device or software process that is not a valid recipient will not be able to retrieve keys, even if the application possesses an XQ token. The application may have the token but will not be able to get the encryption key from the server during the token validation process. During token validation, authorized applications and policies are validated to gain access to the key. Therefore, encryption keys are only retrievable by valid recipients.

Data Geo-Tracking

XQ's applications also provide the ability to geofence data. All data transmissions are tracked by their status and the IP address of the accessing entity. If the IP address is outside the specified geographical area, the key requester will be deemed invalid, and the key will not be delivered.

Data Revocation and Suspension

XQ also provides applications with the ability to revoke or suspend access to data. Revocation takes two forms: first, a user or application can be removed as an authorized recipient of the data. This will revoke access by that one identity, but not to other authorized identities. Second, XQ allows the deletion of the key, which will indefinitely invalidate any requests for that key. In the context of a message, if a message is sent accidentally or the recipient is deemed no longer valid, the sender can revoke the message. If the recipient uses the token to try to retrieve the encryption key, it will not be possible due to the previously revoked message.

Suspending access is a temporary function that can be taken as a precautionary measure, especially in the case of unusual access logs. For example, if an XQ-enabled edge client begins making key requests from a geographical area outside the expected range, access to those keys could be suspended for that device until a proper investigation can be carried out to determine whether the behavior was malicious or benign.

Unique XQ Data Protection Applications

Protecting Data From Different IoT Sensors Using A Shared Wireless Gateway
XQ can be used to ensure different IoT sensors using the same WiFi/5G gateway can only be accessed by the authorized software. That data is homogenized, and context is added via metatags.

Separating Network Traffic On A Shared Satellite Link
XQ can be used to ensure that enterprise data traveling across shared satellite links can only be accessed by authorized systems. This creates a virtualized private network over Satellite communications.

Ensuring Video Conference Stays Within Geo-Fences For Regulated Enterprises
XQ's Geo-tracking can be used to ensure video conference participants are only communicating from authorized locations.

Encrypting Web Chat Sessions For Regulated Medical Entities
XQ enables regulated entities to uniquely log and protect regulated medical and financial client chat sessions.

Detecting Email Credential Theft For Sensitive Messages
XQ enables senders to ensure their sensitive information has only been accessed at one location.

Launching A Secure Chat From Within A Public Chat Service
XQ enables users of iMessage or Whatsapp to launch a secure chat session by transmitting an invite to other participants.

Creating Private Channel In A Public CBRS Network
XQ enables enterprises using CBRS networks to protect their communications from unauthorized eavesdropping even when on the same RF channel.

Protecting Smart City Data Lakes From Unauthorized Access
XQ enables Smart City operators to enforce different access policies within data lakes by using unique metatags and encryption keys for various fields.

Protecting Data On Web Forms From Server Skimmers
XQ enables websites to protect from data by encrypting the data at the user's browser (edge system) and then directly transmitting it to an email server.

API-Based Policy Enforcement

XQ can be embedded into any application or edge device that is internet-facing. Quantum random numbers can be generated via the XQ API and streamed to the embedded execution environment of an application or device, where it is used to generate a local encryption key. That key encrypts the data inside the application or device environment. The key is then posted to XQ's backend server with retrieval policies such as identity and expiration. The application will need to be registered in the XQ portal in order to receive an application ID associated with the XQ account. The application ID can then be used to make REST calls to the XQ API to manage encryption policies and keys. It is important to note that the SDKs are used to create encryption, while the APIs facilitate key and policy management.

API Functions:

1. Identity authorization
2. Supply entropy to be used as a key or to seed an encryption key
3. Store and retrieve keys
4. Validate token authorization
5. Manage data policies

Endpoints:

1. QRNG server which serves quantum entropy
2. Validation server where keys and policies are stored
3. Subscription server where users are stored

Policies:

1. Who
    a. Which identities can decrypt the data
2. When
    a. For how long can the identity access the data
3. Where
    a. From where can the identity access the data
4. Dynamic
    a. Contextual information that the end application may add
        i. Example: meta-tagging for search

Logging:

The following items are logged by default for each security transaction:

1. Application
2. Accessing identity
3. Geolocation
4. Event
5. Timestamp
6. Alert level

Recommended Vendor Access Policies:

1. It is recommended that entities contractually require vendors to keep data at rest encrypted
2. All data accessed by vendors should be through an XQ encrypted gateway
3. Vendor data access should be revoked post project
4. Vendor data usage should be monitored while in their custody

Encryption and Decryption Process

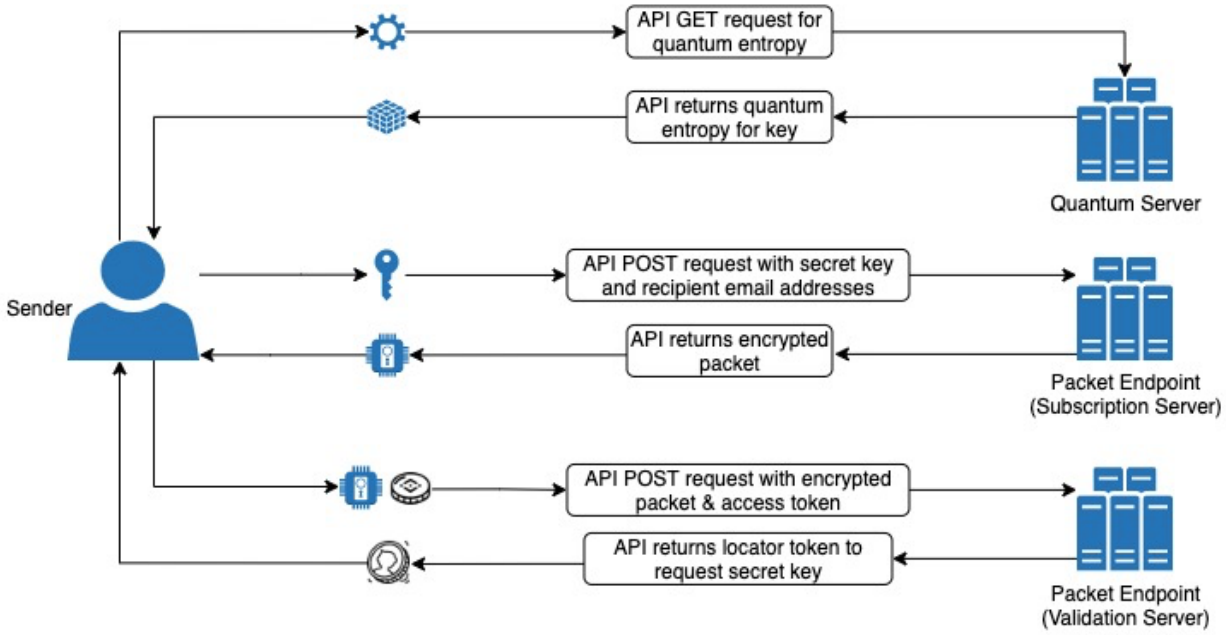
The following outlines the typical encryption process, specifically in the context of encrypting a message within a text file. Please note that the description is intended to outline in detail the encryption and decryption process, which is universal in nature and follows the same process for various implementations.

Encryption Process

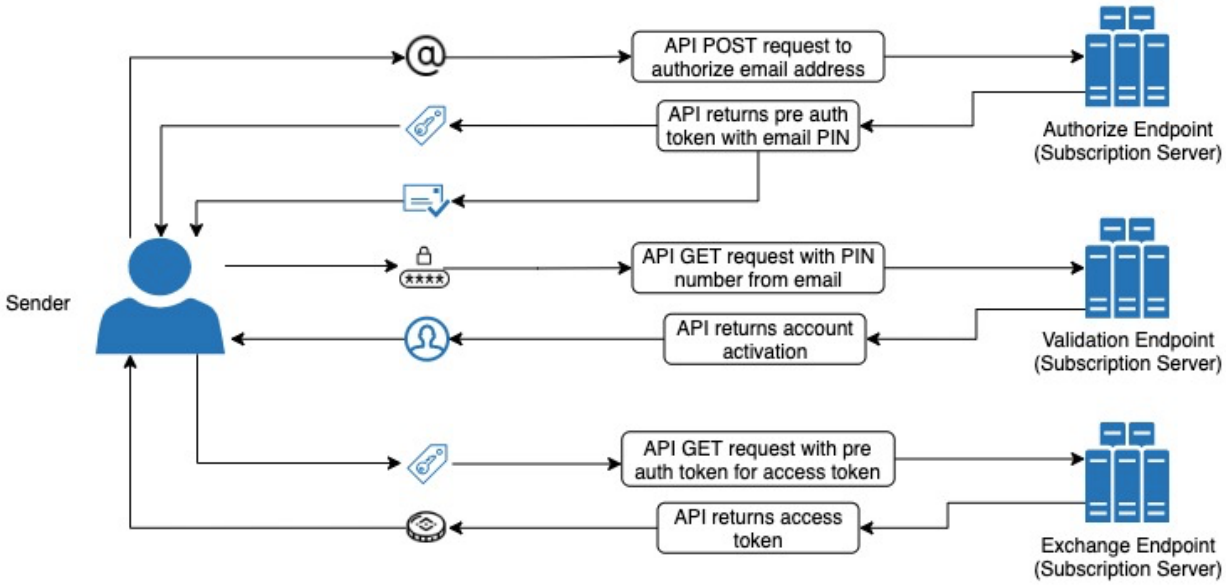
XQ's encryption process begins with retrieving entropy from any source the customer desires. After receiving the entropy, the encryption is configured using a selected cryptographic library, such as OpenSSL, AES, or OTP - XQ is crypto agile and therefore compatible with any cryptographic algorithm. Next, the accessing entity (in this case, a user with an email address) is validated via the subscription server, returning a pre-auth token with a PIN code to be confirmed by the user. The user, in this case, is verified via confirmation email with the PIN code; however, it can also occur via an API call with an associated PIN from email and pre-auth token. Once the user is verified and authorized, the pre-auth token is submitted to the subscription server to receive an access token. Once the access token is returned, the key is then submitted to the server with the access token and the recipient address. An encrypted packet is received then submitted along with the same access token to the validation server. The packet is verified, and a locator token is returned, which will be used by the recipient to request the encryption key to decrypt the data.

The flow chart diagram below outlines the process as mentioned above in two separate flows for more precise understanding: one for the encryption phase where the quantum key is fetched and the second for the user verification and authorization.

Sender Key Submission Flow



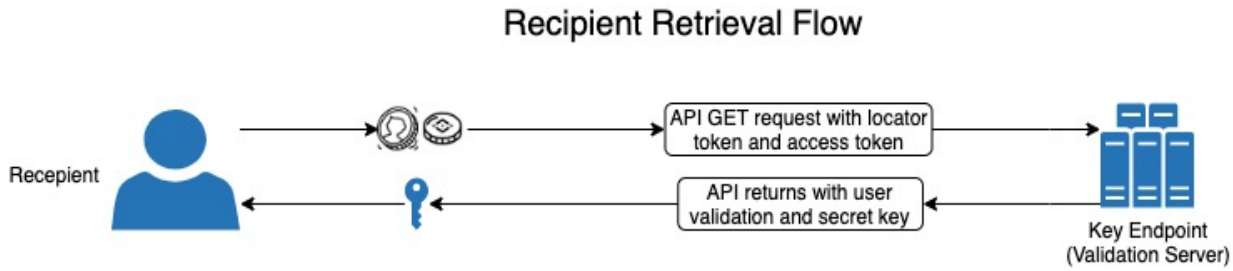
Sender/Recipient Authentication Flow



Decryption Process

Once the encrypted data is transmitted to its destination, the receiving end will receive the encrypted data along with the XQ access token intended for the specific recipient. The recipient then makes a request to the XQ backend and passes the access token and a locator token. The XQ backend verifies that the access token received is valid and verifies that the user who sent the token is a valid user and allowed to view the message. Once the user is validated as a recipient, the backend sends the encryption key to that client. Finally, the client uses the encryption key to decrypt the data.

The diagram below outlines the aforementioned process that occurs between the recipient and the XQ backend.



Implementation Scenarios

Using Existing Encryption

How an enterprise integrates XQ into its current tech stack and environment is entirely at the organization's discretion. XQ is optimally used for implementing a Zero Trust Architecture to manage access data and encryption key management. Either the XQ SDKs can be used for encryption, or the XQ API can be used for key management and to handle the encryption independently. Customers can opt to use their own cryptographic standard alongside XQ, such as AES, OTP or any NIST algorithm. All that is required is configuring the cryptographic library to be compatible with XQ clients so that data can be decrypted.

For example, when using XQ's email extension, once an email is encrypted, the payload is packaged in a certain format that XQ extensions can understand - this format is a link, such as https://*xqmsg.net?encrypted_message*. When the recipient receives the message, the link is recognized as an encrypted XQ message, and the extensions are

able to decrypt the text. If the particular link is not detected, the message is not decrypted.

Therefore, to integrate XQ, an enterprise can build their own extensions entirely and is not required to use XQ's native extensions, but instead can use XQ solely to manage the keys via the API. Once the data is received, the extension will detect a specifically configured parameter. When the parameter is detected, the XQ token is extracted and is then used to request the XQ endpoint to retrieve the key. The key is then received, and the data is decrypted in a reverse manner that it was encrypted.

Using AES 256 bit encryption

Advanced Encryption Standard (AES) is a symmetric key cipher that uses the same secret key for both encryption and decryption and requires that both the sender and receiver have a copy of the key. XQ is crypto agile and is compatible with various encryption ciphers and, in this case, can provide secure key management. XQ can generate stronger keys and manage the keys, as with AES 256 bit, there is still an encryption key that needs to be sent to the recipient. For standard edge encryption, devices generate a standard random key that is a number which can eventually be guessed by knowing the processor that was used to generate the number. XQ supplies a verified quantum random number seeded encryption key via pulling quantum entropy from an XQ server. This is a much more secure key. Current encryption algoruthms are still used but with stronger keys. XQ then manages the keys when it's time to transmit the encrypted data to the recipient. As a result, the recipient does not need to provide the key but instead provides the token received from XQ. The recipient then uses the token to retrieve the key and decrypt the message.

NIST Cybersecurity Compatibility

XQ is crypto-agile, and as a result, the edge encryption algorithm can be switched on any device and can change to meet the needs of the data and destination platform. XQ supports the use of new NIST-approved quantum-resistant algorithms. Furthermore, there is a multitude of various cryptographic libraries that XQ can be layered upon (OpenSSL, NaCl, CryptoJS, etc.) and subsequently seeded with QRNG entropy from any source the customer desires.

On-Premise Deployment

Customers can have local instances of XQ's platform which live in any cloud or on-prem environment. These local instances ensures data provenance for regulated entities. With a local instance of XQ and associated quantum entropy pool, when data hits the gateway to be encrypted, that gateway receives entropy from the local instance supplying the quantum entropy pool.

The process is as follows:

1. Gateway will first pick up quantum entropy to generate a key and then encrypt the data using that new key.
2. Once the data is encrypted with the key, the application will send the key to XQ servers along with the recipients who are authorized to retrieve the key.
3. After XQ servers receive a package of metatags and the encryption key, a token is generated. The encrypted data can be transmitted with that token or through a control channel.
4. On the recieving side, when decrypting the data, the recipient will extract the token, use the token to make a request to XQ endpoint, which will will return the decryption key if policies and identity permit.

Interoperability & Holistic View

A massive benefit of XQ is that the platform creates a frictionless exchange between disparate systems while adding context to data, especially from the edge.

Consider two systems - water management and traffic management in the same municipality. The two systems have data in completely different formats, yet when they are edge encrypted using XQ SDKs, or the XQ Zero-Trust Gateway, each data transaction now has the same format and is furthermore ”vectorized” with contextual metatags. EG - where the data came from and what its purpose is. This “reformatting” allows for a homogenous view across systems into the security state of the entire data landscape for the municipality. That data can then be uniformly ingested and then decrypted and operated on per data format.

XQ Name Server

XQ can be used either as a SaaS product where XQ holds the keys and policies or as an on-prem deployment. With the XQ Name Server, no matter which XQ deployment holds the rights to an encrypted piece of data, the data knows where to look up its validation server.

This means you can have two entities, each with their own on-prem deployment of XQ, exchanging secure messages while retaining data provenance and control.

Remote XQ deployments must register their location with the XQ Name Server to enable this feature.

Please contact XQ for direction on how to implement this feature.

Identity Agnostic

The XQ platform is identity system agnostic. It supports multiple systems currently and is easily adaptable for new systems. Currently, supported identity systems include:
IP+ Security Key, Internal XQ Auth, Auth0, OKTA, phone number.

KMS agnostic

The XQ platform is Key Management System agnostic. It supports multiple systems currently and is easily adaptable for new systems. Currently, supported KMS systems include MYSQL and Cassandra. Future systems on the roadmap include AWS KMS Azure KMS.

Entropy agnostic

The XQ platform is entropy source agnostic. Currently, XQ supplies Quantum Random Entropy through our API. This can be configured to use any digital entropy source desired.

How to Get Started

XQ is free and easy to get started with XQ's add-on applications for email, mobile messaging, and Slack are free and available to download and install across Gmail, Outlook, Slack, IOS, and Android. This means common applications used for daily communications across employees and customers can be secured and encrypted immediately. In addition, access to the XQ API and SDK is available through the XQ portal online.

The process begins at https://manage.xqmsg.com/signup where a user sets up an XQ account. Setting up an account will provide access to the dashboard UI. The dashboard gives real time activity monitoring of messages sent via the add on applications, ability to provision user accounts, verified endpoint tracking, ability to encrypt and decrypt files, access to audit tools to export logs and encryption keys, and access to developer resources such as the API and SDKs.

Application Downloads:

Google Chrome extension:
https://chrome.xqmsg.com/install

Google Gmail:
https://gsuite.google.com/marketplace/app/xq_secure_email/293580994869

Microsoft Outlook:
https://appsource.microsoft.com/en-us/product/office/WA200000090

Slack:
https://xqmsg.com/product/product_xq_slack_app.php

Apple iOS mobile app:
https://apps.apple.com/us/app/xq-msg/id1479922405

Android mobile app:
https://play.google.com/store/apps/details?id=com.xqmsg.xqmessage

Word Press Secure Forms
https://wordpress.org/plugins/xq-secure-form/

API and SDKs

The first step to using the XQ API is generating an API key. This can be completed by creating your first application at https://manage.xqmsg.com/applications or by selecting “Applications” under the “Developer” section of the portal.

The API key will be bound to your created application, and data from the API use will and the encryption activity associated with the API key will be shown in the “Monitor” section of the XQ portal.

The API is accessed through RESTful calls as sampled in the SDKs provided on XQ’s Github page: https://github.com/xqmsg

The following tutorials are available with step by step walkthroughs:

Encryption API Tutorial:
Start the tutorial

Decryption API Tutorial:
Start the tutorial

You can also find guided installation tutorials here.

XQ Applications

Platform:
The XQ application platform, which supports the APIs and administration tools, is fully deployable from Dockerized Containers for cloud and on-prem. These deployments provide control over entropy source, the identity system, the KMS system as well as monitored event types and webhooks.

Main portal:
https://manage.xqmsg.com/login
This is where users are able to manage the various XQ applications as well as the XQ API and SDK. The portal also provides an overview of all users, activities, and messages. The portal can also be used to encrypt and decrypt messages or files and provide access to audit tools.

SDKs:
https://github.com/xqmsg
These SDKs add functionality to the XQ API endpoints and accelerate the integration of XQ into an application.

Zero-Trust Gateway:
Point to Multipoint cloud-managed VLAN and IP encryption. Functions as an IP VPN replacement. The Gateway is a breakthrough in data protection as it allows you to secure data and then have that data travel over any series of network links that are out of your control while still keeping your data safe and controlled. In a data protection revolution, the Gateway provides software-defined IP-based protection, and software-defined VLAN channel data protection and routing.

Google Chrome extension:
https://chrome.xqmsg.com/install
The browser extension provides a shortcut to the XQ portal as well as a shortcut to create an encrypted message. This also loads additional features for composing an email message when used in conjunction with the GSuite app (it is recommended to install both).

Google Gmail:
https://gsuite.google.com/marketplace/app/xq_secure_email/293580994869

Microsoft Outlook:
https://appsource.microsoft.com/en-us/product/office/WA200000090
The email add-ons provide the ability to encrypt and decrypt email messages and attachments natively within each respective email client UI.

Secure Forms:
https://wordpress.org/plugins/xq-secure-form/
This Wordpress plugin embeds XQ's zero trust protection into Wordpress site forms directly from the client browser. It will then deliver the encrypted data either by email or to a backend system.

Slack:
https://xqmsg.com/product/product_xq_slack_app.php
The Slack app provides the ability to encrypt and decrypt both direct messages and messages within a public channel natively within the Slack UI via a keyboard command.

Apple iOS mobile app:
https://apps.apple.com/us/app/xq-msg/id1479922405

Android mobile app:
https://play.google.com/store/apps/details?id=com.xqmsg.xqmessage
The mobile apps serve as an overlay for text messages in both types of smartphone operating systems and provide the ability to encrypt and decrypt messages natively.

Edge Environment Requirements

Hardware baselined against XQ Zerotrust GAteway and per transaction key rotation.

a. Source Device
    i. A standard machine running Ubuntu 18.04 LTS / Ubuntu 20.04 LTS, with a NIC speed of 1000 Mbps and connectivity to the devices listed below as well as external access.

b. Encryption/Decryption Device (Qty: 2)
    i. Minimum Requirements (Target/Expected speeds of up to 250 Mbps)

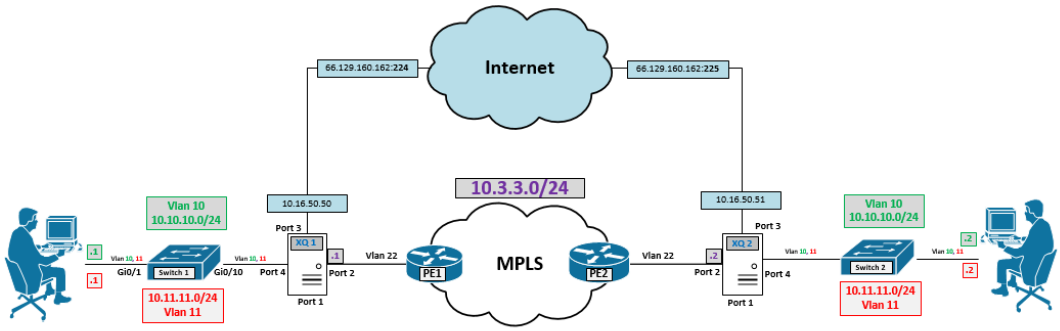
CPU	2.0Ghz Intel Celeron or similar
RAM	4 GB
Disk Space	2 GB
NIC Speed	1,000 Mbps (1 Gbps)
Operating System	Ubuntu 18.04 LTS / Ubuntu 20.04 LTS

    ii. Recommended Requirements (Target/Expected speeds of 700 Mbps and above)

CPU	2.4Ghz Intel i7 or Higher
RAM	8gb or Higher
Disk Space	2gb
NIC Speed	10,000Mbps (10Gbps)
Operating System	Ubuntu 18.04 LTS / Ubuntu 20.04 LTS

c. Destination Device
    i. A standard machine running Ubuntu 18.04 LTS / Ubuntu 20.04 LTS, with a NIC speed of 1000 Mbps and connectivity to the devices listed below as well as external access.

Example configuration

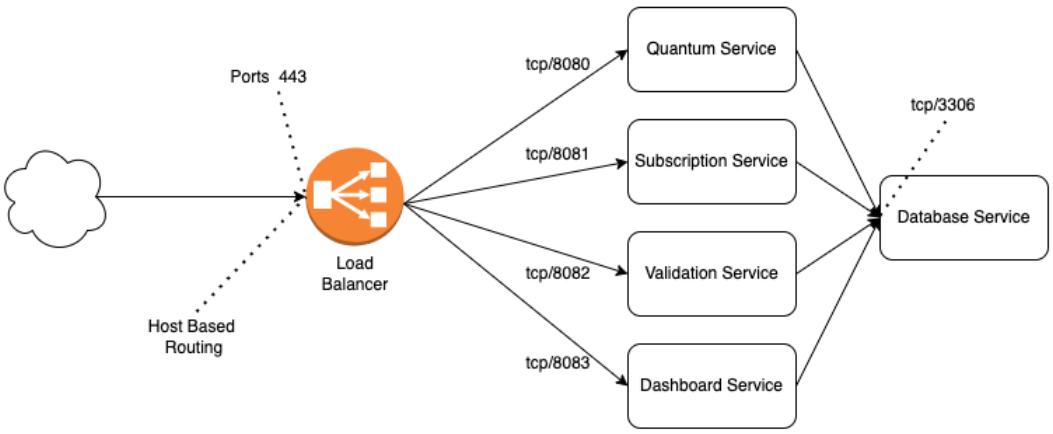


* The IP addresses and ports identified in the image are just for documentation purposes and don't need to be set as identified in the image.

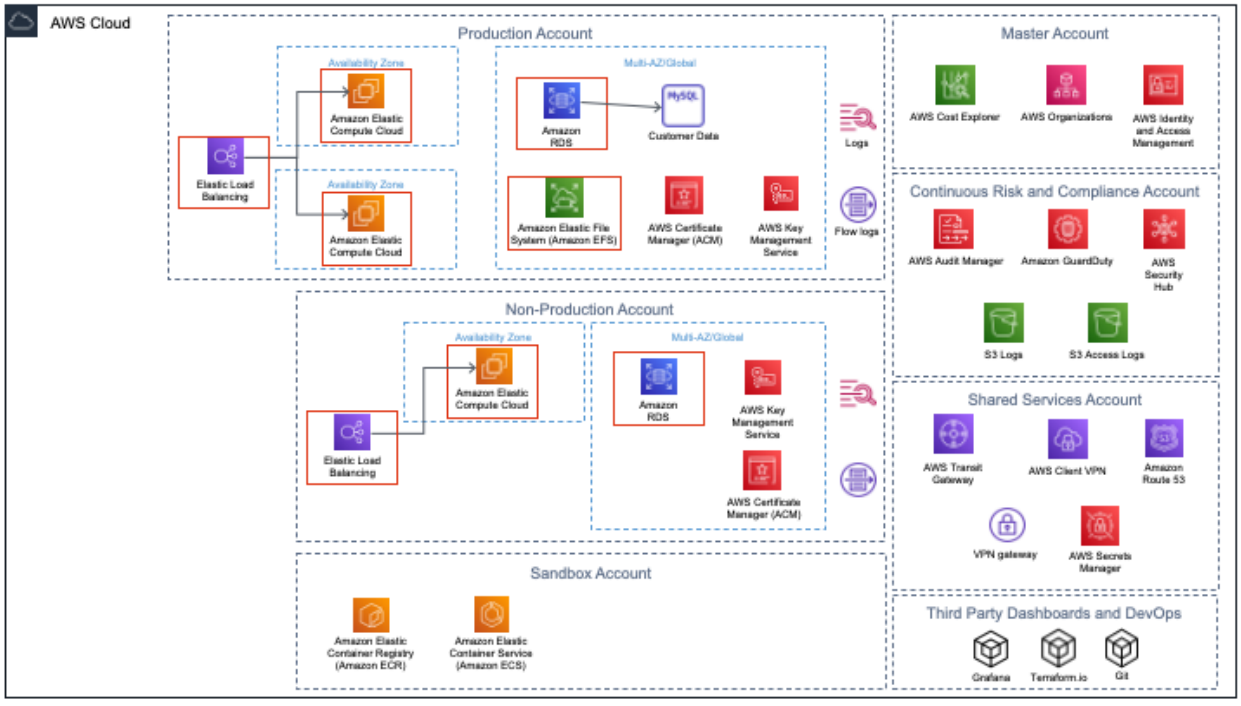
* The switches within the image can be removed for simplicity in the POC stage but it is documented to display a real-world setup.

XQ Deployment Architecture

Note: Port numbers are illustrative and may not map onto the actual deployment.



XQ AWS Deployment



Competitive Chart

ATTRIBUTE	COMPETITOR	XQ	IBM	PaloAlto	Perimeter 81	Google
Multi-Cloud	Edge to any cloud system	✓	X	✓	✓	X
Data Tracking	Data can be tracked everywhere it might go	✓	X	X	X	X
Audit & Compliance	Product supports audit and compliance	✓	✓	✓	✓	✓
Zero Trust Data	Authenticated ID, constant verification, crypto agile	✓	X	X	X	X
Cost Effective	Site license options	✓	X	X	X	X
Distributed	Most Security systems are centralized	✓	X	X	X	X

FAQ

- *Explain how XQ intakes structured and unstructured data*

XQ does not intake data. It encrypts the data at the edge and stores the key and policies associated with that data.

The output of XQ is an encryption format that is structured, homogenized, and vectorized data.

- *Business continuity considerations*

For XQ Saas Application Integrations: Disaster Recovery Plan, Operations Plan, and Vulnerability Plan are available upon request.

- Mobility Considerations - Online / Offline access

Decryption requires local or internet-based access to the validation server.

- Accountability Considerations - Key Rotations / Key Generation
Keys are not session or ID based. Key rotation happens at the edge and can occur on a per-transaction basis.