From Zero to Hero: The impetus for quantum proof (PQC/QPC) mutual authentication

Christopher S. Batt, Toronto, Canada.

Revised January 3, 2023

Abstract

The issues of internet privacy, digital identity, authentication, and authorization remain unsolved problems, especially for industrial and distributed IoT environments. Despite the plethora of security products available from commercial and open-source vendors, attack footprints are not being reduced. The complexity of such solutions continues to grow, while overall protection and compliance figures, according to industry data sources, should be alarming to us.

The slow adoption of a zero-trust mindset and appropriate methodologies for overhaul has allowed legacy frameworks and network infrastructure to remain in place, while economic and human decision-making have postponed difficult conversations and admissions about the efficacy of current protections for data and critical infrastructure across the vast systems we aim to protect. Server/peer validation processes (such as Certification Authorities) and the lazy interpretations of the definition of end-to-end TLS security (involving intermediary termination) are problematic at best. This is highly correlated within environments where device peering or IoT scenarios cannot assume the presence of TLS infrastructure in the typical client-server model, and devices may be dynamically provisioned or compromised in the wild.

Lack of standards, convenience and convention always come at a cost, and do not lend themselves to embodying best practices. In light of the new NIST standardization process for Post-Quantum Cryptography (PQC), rationale for going beyond the TLS 1.3 Standard will be examined. The rising popularity and benefits of trust-less authentication in distributed networks and embedded assets, along with its application in user-based authentication on the Web, will also be evaluated. Solutions will be proposed in theory and in practice.

We aim to outline the specific problems, principles, and solutions that may address the challenge. All situations are unique, but there is a market for improvement, there are common requirements, and there will be demands for higher standards of professionalism in the encryption and authentication fields.

A move toward better solutions may begin with an aspect of which we are all well-aware. A safer and more authoritative (from the user's standpoint) login and authentication regime is required which addresses advances in quantum computing and NIST standards. It must lend itself to retrofitting existing system architectures and ad-hoc implementations. It should demand minimal overhead and conventional disruption, as to be the most relevant, practical, and ideal scheme for accomplishing a new standard of safety and privacy.

Likewise, the service boundaries and internal communications between systems must be re-examined to determine how to best protect infrastructure and devices from current and future threats. For industrial and banking scenarios, a zero-tolerance stance on weaknesses should be adopted, and forthright discussions must be brought to light, if we are to safeguard our foundational architecture.

Although technical in nature, we will focus on the sensibilities of some parts of a solution, rather than just the theory, implementation, or mathematics applied. A cursory awareness of asymmetric and symmetric encryption primitives and internet transport techniques should suffice to illuminate the ideas emerging from this work. Authorization, the granting of permissions to specific assets and actions, will be covered separately.[1]

As a result of modern breaches and the growing attack surface of public and private systems, it is progress to realize that we are quickly losing our trust. What comes next? In December 2022, the President of The United States signed into law a PQC initiative to implement a roadmap for adoption of strong cryptography for the Budget and Management Office, including many governmental systems.
https://www.infosecurity-magazine.com/news/biden-quantum-cybersecurity-law/

In 2023 many advances will be made, but it is a race against time, and amidst well-funded adversaries.

Problem Statement

To guarantee safe and appropriate measures of identity, permissions, data privacy, and integrity, a fundamental paradigm must be valued, applied, properly implemented, and empirically enforced. These four aspects are deeply connected, and are a continuous lifecycle throughout the design and operation of an ecosystem. Current authentication best practices and software implementations are limited to trusting large corporations and small web platform operators with managing information and secrets, including financial credentials. Large entities, though they may sport better security budgets, are aggregators of sensitive data and therefore become heavier targets for attack; they harbor a wealth of vital information about individuals' identities, and critically, their connections, motivations, and behaviors. In addition, we face the ubiquitous problem of password hygiene, inconsistent transport layer encryption, and identity assertion (such as TLS and x509 certificates for mutual authentication of devices). We place our trust upon, and tout these standards as the best available[2], but incur unacceptable costs in the form of attacks, dollars, operational overhead, and configuration/implementation risk. The status quo does not ultimately solve an individual's need for credential protection or strict end-to-end

[1] Out of scope is the design and implementation of authorization schemes and policy enforcement, which depend on the implementation of organization-based specifics and nuances. Role-based (RBAC) and Attribute-based (ABAC) Access Controls are provided by a combination of application, storage, networking, and other infrastructure designs. Security considerations should be fundamental to the authorization regime, and require assessment and auditing far beyond any authentication/identity management.

[2] Availability of next-generation (quantum resistant cryptography) protections, proprietary solutions for identity management, and strict end-to-end encryption exist beyond commonly found standards, but are now emerging. They are often used in private ad-hoc systems, peer-to-peer networks and anonymized services. As such, they are subject to implementation problems, the real stigma of ‘don't roll your own security/cryptography’, and state pressure or interference.

encryption, as we adapt to an environment where every human will require digital access and protections for their identity. We recognize the inroads Web3 is making in bringing key-based and encrypted credentials to the forefront, and will applaud its wider adoption, along with, to a lesser degree, Apple's noble first attempt with PassKeys to eradicate traditional passwords in favor of public keys. Centralized stores of credentials remain an infrastructural challenge, regardless of the form of those credentials.

API Security

Microservices and public/private APIs are especially vulnerable to security issues due to their vast connectivity with many parties, some from the Internet, in browsers and apps, and from internal (erroneously trusted) system components. These all form an ecosystem where trust needs to be efficient, and yet is often implicit, to the peril of users, their data, and the integrity of services as a whole. It is not uncommon for API keys to be passed unencrypted to browsers, back to other services again, or to be embedded in code and shared.

In our view, it is unacceptable to provide account-level API keys to the client or allow those keys to be passed unencrypted via headers between systems, even if assumed to be in the same 'secure' datacenter. Today, we rely so heavily on API connectivity that a new level of care and professionalism needs to be invested in the service of security among our systems. TLS (SSL, HTTPS) and implicit trust are not enough. Without securing the data path between client, gateway, servers, serverless functions, and microservices, APIs represent a threat to the Internet, our privacy, and security that must be addressed even before breaches that prompt customers to beg for improvement and remedies to the systemic problem. The past methods of securing APIs have been lacking since their inception, and solutions exist today to make cost-effective and performant inroads to solving the issues.

Background

Trade-offs remain an issue, pitting one set of objectives and priorities against security, resilience, integrity, and performance. Gilman and Barth state that "In building and designing such systems, we have found frustration in the pace of progress toward solving some of the more fundamental security problems plaguing our industry. We'd very much like to see the industry move more aggressively toward building the types of systems which strive to solve these problems."

Zero-trust is modeled after the fact that security is not inherent inside some defined network territory, behind a firewall, or *inside* as opposed to *outside*. Every transaction and actor/device and service on that device is treated as hostile, and therefore authentication and authorization policies need to be dynamic and multi-faceted, often relying on behavior-based analytics and monitoring that employs some level of artificial intelligence.

At the very outset, mathematically provable transactions and the safeguarding of credentials are essential, if a standard or cooperative of systems is going to be built on a solid foundation. Trust must be reassessed from first-principles, leaving nothing to chance in favor of convenience or compatibility. CloudFlare is doing just that, by transparently implementing quantum resistant certificates as an extension to TLS 1.3. However, we will see that TLS coverage across the data path is a weak link in the chain of secrecy and integrity, and this shortcoming

resides in the implementation of individual systems at large.

John Kindervag researched, authored, and implemented the seminal work in the realm of zero-trust networks and systems, which revolve around guarantees and the realistic assumptions that someone or something is always lurking around the corner waiting to compromise, interfere with, or steal something– even from inside a federated space with a well-defined perimeter. Much needed guarantees are objective and impartial. They don't favor one organizational department or individual's needs or preferences over another. As such, they cannot be partially devised or implemented and remain labeled as guarantees. However, trade-offs will inevitably crowd out the purest of ideals. No existing system (or even a green-field software project) exists in a vacuum. Real world engineering gets in the way of essential truths. It has been said that "battle planning is essential, but the plans are worthless".

Some insist, if only security policies were *set in design*, rather than being piled on top of common threats as they emerge, we would all be much more secure. What is the essential model to follow, if cost-effectiveness were to incorporate the dire human cost of loss of security, privacy, and the threat to life and liberty?

Consider the accounting department and comptroller's job to gate-keep and monitor every penny that transacts within, and outside, each unit in an organization. This is accomplished by a zero-trust methodology, in fact. We ask questions first, demanding proof of identity and intention, not after writing blank checks. The idea that a transaction remains unchallenged and untracked from the start would not be up for debate in the design of the financial affairs of a team and for an entity as a whole. This proactive enforcement must be followed up by transactional monitoring and ledgers, to ensure the execution matches the plan and the rationale for taking any action can be traced back to its event source. Consistently applying such methods prevents misaligned rules and gaps in implementation. Actions can be measured and justified precisely through the money path, and such must be the future of the data and security path.

We are falling further behind in progress as compared with adversaries, according to number, severity, and dollar cost of breaches and attacks. Over recent years, the coverage of known and unknown threats, in terms of systemic preparation and compliance, has fallen as a percentage to somewhere in the 81% range (Gartner 2021). A shocking percentage of ransomware demands are paid to criminal organizations, thus funding vastly harmful illegal and terrorist activities (according to other industry surveys and studies, probably under-reported). Perhaps the rise of Corporate Social Responsibility (CSR) will not tolerate such decision-making that supports criminals and such illicit activities and extortion much longer.

The deficiency *could be* blamed upon lack of resources, difficulty of adapting legacy systems, or the lag in adopting patches for emerging threats, for an IT organization within a company that inadequately funds it. But how do we decide *reasonable* funding and resources applied to the problem(s)? The human costs are becoming far reaching, more expensive, and harder to quantify, when considering the cost of financial, political, and social confidentiality in just about any part of the world’s interwoven economy.

Risk analysis (often triggered and incentivized post-incident) weighs the responsibility of the organization to

allow *acceptable* losses for stakeholders, versus qualitative and quantitative costs of preventing such risks– as may reasonably be expected to surface in practice. However, this predictive/reactionary stance aims to subjectively address outcomes and measure injury, where no such thing is actually possible or acceptable. The goal revolves around the cost-benefit of deploying resources and imposing change in an organization. It does little to acknowledge the human costs, scientific facts nor impose the most restrictive bottom-line protections we deserve as global citizens in this heavily connected and network-reliant world.

It is becoming more evident that better security philosophies (leading to actual practice) require an enhanced meritocracy rather than best-guess damage control equations. Decisions around acceptable losses for individuals whose lives may be vastly impacted, even by suffering or death, cannot be made by corporate actors. Professionalism in engineering must evolve and impose remedies in unsafe regimes, in theory, in practice, and must be vocal about it in every organization.

Mathematical, scientific, effective, and auditable guarantees are demanded and evoked by today's security landscape. There is no academic argument to supplant this assertion; only claims of practicality insisted upon by arguably negligent organizations who have their own missteps to defend. Those who attempt to rationalize their resistance to change will be obsolete and damaged in the process. If they remain steadfast in incompetence, hopefully they will also go down with their ships. Resistance to higher standards should be regarded with the ultimate skepticism and suspicion.

Complexity, often barely palatable as it is, also continues its creep at the expense of security. CrowdStrike imparts that "Complexity is the attacker's friend...Truly effective endpoint protection must provide the highest possible level of security, yet be simple to use. Complexity strains teams and processes, introducing security gaps that increase the risk of reduced productivity and harm to an organization’s reputation."

There has been exponential growth in the deployment of interdependent software components. Both diverging and converging sources of data and decision-making intelligence has led to a void of comprehension and authority in numerous domains, especially where privacy and enforcement are concerned. The deficiency of reasoning by individuals regarding undisclosed security vulnerabilities has resulted in cover-ups, to be certain, at every level of society. The argument of high costs to correct the lack of understanding of a system falls short. Malicious actors can make a profit by learning to understand important systems and their shortcomings better than the operators of those ecosystems. Why should the stewards of infrastructure be unable to profit from the *prevention* of such dangers? How long can we naively assume that current standards will not irreparably affect each of us?

"The reality for security today is that security leaders have too many tools. [We] found in the 2020 CISO Effectiveness Survey that 78% of CISOs have 16 or more tools in their cybersecurity vendor portfolio; 12% have 46 or more. Having too many security vendors results in complex security operations and increased security headcount.", says Gartner.

Describing the future requires imagined change, designed for infrastructure and workloads appropriate to that

time, and based upon where we find the present state of our industry. It necessarily requires outing the elephants in the room, when it comes to common shortcuts and trade-offs that are deemed acceptable today, but will not be tomorrow. If *we* do not think from a position of idealism, perfection, and elegance for ease of adoption, when and who will do it? Certainly it cannot be solved as an afterthought, patched on top of present dangers, subject to cost and resource constraints of the day.

Solution

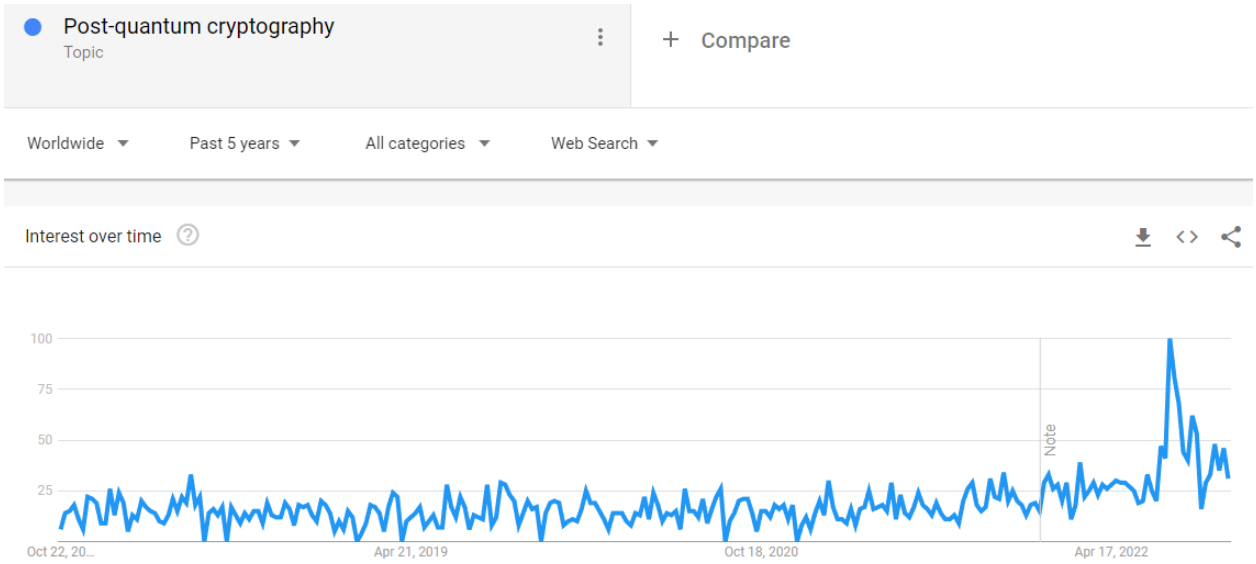
Zero-trust

The book *Zero Trust Networks* describes the model we will employ as such: "In this model, nothing is taken for granted, and every single access request—whether it be made by a client in a coffee shop or a server in the datacenter—is rigorously checked and proven to be authorized. Adopting this model practically eliminates lateral movement, VPN headaches, and centralized firewall management overhead. It is a very different model indeed; one that we believe represents the future of network and infrastructure security design."

Standards and industry-wide, open methodologies are the successful results of the right types of thinking over the years that the global network has developed. Even proprietary interests have embraced a more collective ecosystem when valuing the safety of the masses as the best scenario for protecting the individual. Outcomes have improved from the days before SSL (now TLS), when the wild west relied most on the immaturity and disorganization of hackers and government actors in relation to cyber-intelligence. At least three-letter organizations were not seen, then, as direct threats to corporate and individual interests in proportion to the exposure they had to the Internet. Today, individual and business entities are more sensitive, motivated by the increase in hacking activities, cyber-crime, and insulation even from their *own governments*.

The growing concern, being presently addressed by NIST, is that of quantum computers' future ability to compromise the algorithms currently employed for all our encryption, especially asymmetric (RSA and others), which is the backbone of Internet-based 'secure server' technology. Such compromises will render the mechanisms and content of global communication bare to inspection, malicious manipulation, and extortion.

In July 2022, further standardization of Quantum Resistant Cryptography (QRC) was announced, bringing about a call-to-action for organizations, CIOs, CTOs, and engineers alike. It did little to raise fanfare, as seen in the Google Trends graph below, which shows evidence that this field remains a niche interest at best, for now.



Libraries and simple methodologies for implementing asymmetric key exchange algorithms are available and performant today; they far surpass RSA computation speeds, especially in dynamic key generation, and can be straightforwardly implemented in server-side systems, browser-based, IoT, and native mobile platforms.

What if some low-hanging fruit were seized *now* by better thinking, exercising strong opinions, and discarding legacy habituation?

Figure 1 outlines some of the potential changes in how authentication can be done, and how trends are moving toward a discernible improvement to some of these maladies.

Figure 1: Before, Now, and Tomorrow (changes on the horizon)

BEFORE	NOW	TOMORROW
HTTP (unencrypted)	HTTPS (SSL>TLS encrypted)	END-TO-END encryption and digital signatures
INFRASTRUCTURE TRUST	AUTHORITY TRUST (CA)	TRUSTLESS (zero-trust)
HOSTS (distributed, decentralized)	PLATFORMS (cloud, centralized)	PEERS / EDGE / IoT (cloud, decentralized)
RESPONSIBILITY	DELEGATION	RESPONSIBILITY
ANONYMITY (unregulated)	AUTHORITY (federation)	INDIVIDUALITY (identity)
Web 1.0	Web 2.0	Web3, Industry 4.0
Enterprise (narrow)	Web technologies (wide)	FinTech (enforced)
Servers (disparate)	API/Serverless (common)	P2P/IoT (mutual)

NOTE: some of the departures from earlier styles and sensibility of architecture have been in the wrong direction, and we are now course-correcting, sometimes intentionally. Some of the inevitable return to sanity has been inexorably driven by distributed governance or *grassroots* development, and has often been hard won against the resistance of large industry players. Standards have often followed only when progressive changes have led to improvements.

The Approach

This paper proposes the adoption of a method that systematically addresses the challenges of today, tomorrow, and into the AI and quantum age. We are determined to maintain essential and calculable guarantees codified within it, which can be implemented efficiently, deeply, and transparently to avoid the common security risks associated with authentication of systems and Web-facing applications having dispersed users. This is a general and versatile approach, yet one created with practicality and calculable wins as top priorities. It aims to guide us toward the direction(s) we must seek. Zero-trust requires turning the traditional security model inside out. Adaptation and band-aids are often insufficient for required outcomes, and in these cases an overhaul is appropriate. We must effect real change, rather than paying mere lip-service to our shared needs. We must authentically uphold our obligations to private, public, human, and regulatory stakeholders.

The Identity Management Process

This process describes a vastly different view of *credentials* and *authority*, which cannot be fully explained here. This paper may serve as an introduction. The term "Your Keys, Your Responsibility" greatly applies. We advocate a significant shift in the concept of credential custody, and who/how authentication is achieved. We aim to minimize vulnerability to phishing attempts. We must re-examine the traditional practice of blindly entering our passphrases into any web form that *seems* legitimate. One concept in our ideology is that we must *never transmit passphrases/keys to any party*. Deterministic key generation and asymmetric encryption can solve this need for us, so that we never need to send our credentials across the wire. Further, only intended parties may read what is meant for them, when data is safeguarded using asymmetric encryption. An additional benefit of this method is that no other party or service needs to bear responsibility for seeing or storing our passwords, either encrypted or in plaintext.

This implies we might seek two other characteristics as a crucial design of the process:

1. Demanding the identity of the service or peer (the device, server or attacker). We impose and reap the benefits of *mutual* authentication with public key pinning.
2. Isolating the key generation and storing (in memory only) from the login context (other party's page, code, and cookies). This can be performed (optionally) by a local static resource the user/device *owns*, rather than being provided (and possibly altered in transit) by a third party.

The user or device generates and protects their own private keys using a passphrase, while freely publishing their public ones. PQC/QRC encryption techniques allow anyone else to provide a secret message *only discernible to the private key holder themselves*. This enables a simple proof of identity without prior arrangement between

strangers; only the holder of the private keys can prove they can decrypt a message meant only for them. Then, a coordinated secret can be shared among peers to reliably and flexibly exchange symmetrically encrypted (AES) messages of any length.

The user or agent acting on their behalf attains all authorization rights by possessing a deterministic key pair derived from secret credentials such as a passphrase or other proprietary string of data, along with a two-factor authentication string attained out-of-band through email, sms, or QR code. As per the zero-trust philosophy, the user/agent is solely responsible for maintaining these credentials, and need only trust their interface environment. Credentials can be exclusively entered in a trusted login form (to be especially thorough), not one provided by a potential attacker via the network. Local device code processes and protects the generated keys (in memory only), and never sends any sensitive bytes across the wire or to another device. Thus there is no reliance on trusted Transport Layer Security (TLS) anywhere in the global data path.[3]

Digital Signatures

Digital signatures were included in our initial handshake transaction during the proof-of-concept stage, which used RSA asymmetric keys. It was a reasonable component to implement. A trustworthy digital signature proved the signer's possession of the private key, and verified the handshake's data at the same time. The notable accomplishment of reducing network hops from 3 down to only 1 was a welcome result, allowing the request itself to assert identity, and the response the same. The initial theoretical design required a response from the peer, then a subsequent request containing a token (guarantee) to prove the decryptability of the prior response. With the inclusion of a signed handshake, this proof was self-contained, and did not even need an HTTP response, such as in IoT use cases which can use alternative protocols. This allowed for interesting possibilities with User Datagram Protocol (UDP) transmissions and multicast, a highly relevant use that has historically begged for better authentication and encryption solutions.

However, in the subsequent revisions of the protocol which incorporated QRC keys, digital signatures were not efficacious, requiring another library and separate algorithm from the CRYSTALS-Kyber key exchange mechanism. Therefore, the inherent properties of AES key decryption and the included message encrypted/decrypted with that AES key serves as a suitable replacement for digital signatures. This generation of the handshake proves possession of the private key by the sender, and the peer's response mirrors this assertion of identity. AES-GCM tags and serialized messages within a session serve to verify the authenticity of each message received from a peer, and ensure integrity of its contents in an AES context.

[3]This roughly models the techniques employed by TLS (formerly SSL) or Transport Layer Security, however it operates at the application layer, and does not require complex networking infrastructure to remain integrous throughout the data journey; it provides pure end-to-end secrecy (even allowing secret messages between end-user devices) without requiring trust in the infrastructure. It allows for the real-time creation and revocation of free certificates (keys). TLS remains an effective way to minimize interference and DOS via interruption or manipulation of data integrity, however it is not required to ensure mutual secrecy and positive authentication.

Certificate (Key) Pinning

Public certificate lookup and pinning use a highly distributed and cacheable model. A pinset may be cached in a local device, available as a service's REST API, and even served by CDN. Connecting to a traditional shell host via SSH for the first time is a comparable process. The expected public signature may be verified easily in a browser or other device, and can prompt user confirmation and intervention if applicable. This is superior to the opaque process of verifying TLS certificates with the tell-tale green padlock that aims to assure us of trust. We warn the reader the padlock does not provide all the *assumed* characteristics of a 'secure' connection. Public Key Infrastructure (PKI), in the publicly managed context, imposes the trust of the user upon Certification Authorities (CA) and infrastructure/cloud providers (CSPs). These so-called authorities can, have before, and will again be compromised. We believe that private PKI (internal authority and certificate management) or an explicit pinning and verification scenario removes an area of great concern, where unfounded trust is assumed by users and devices. This is the most dangerous form of trust.

This provides a significant departure from the status quo implementations and solutions utilized in production today. The assumption is made that TLS is compromised or terminated at some location, such as when traffic passes through a proxy, Content Distribution Network (CDN), or sometimes a local network router. A single TLS certificate with end-to-end (user to application) encryption does not exist in today's distributed and cloud/edge environments. It is left to the originators and final consumers of data to prove integrity mathematically, including secrecy of all protocols and data in play among them. This process is almost exclusively left undone in most private and public systems, due to ignorance and misunderstanding of engineers and users, who have little or no training in system security and cryptography.

To design and implement an alternative method to PKI, we require a reliable and safe mechanism for the user (who can't be expected to remember a 256 byte binary string) to establish their deterministic private and public keys *anywhere* with only a passphrase associated exclusively with the peer or service with which they wish to prove their identity. Keys are unique to each peer relationship, so one peer cannot correlate that identity in another peer-to-peer relationship. In addition, only public keys are stored, so no actual credentials can be leaked. They are never transmitted or stored by either peer. We achieve uniqueness/masking of the public key by using a 2FA token unique to each service or peer.

The Mutual Authentication Handshake

A handshake establishes a session and proves identities between two peers. It creates agreement on cryptographic keys for secret exchange of messages. The handshake payloads, delivered via an HTTP request URL and response body, standardize and simplify the mutual and cooperative process by which peers accomplish the authentication goals and manage state securely between them.

The protocol classifies neither peer as authoritative, so there is no notion of server and client in the relationship; peers are equals. Though one party technically makes an authentication request, and the other may be a service responding to it, this is where the similarity to client-server models ends. Both parties to a handshake provide an

assertion of their identities and prove them to the other, whether single requests or persistent connections (WebSockets) are used. Any type of peer (service, API, or user/device) expects the same rigorous mutual handshake. The ability to *login* (which is by definition simply to be authentically recognized subsequently by a peer) is not achieved due to the presence of a sensitive database of credentials that must be maintained and checked in a central (and potentially vulnerable, high-value) location. Traditional lookups provide significant bait for adversaries. Pinned public identities can tolerate a manageable degree of breach, and metadata can be itself autonomously encrypted by each peer.

This lookup is akin to a Certification Authority, yet it is as anonymous, trusted or untrusted, and as ephemeral as each peer desires. It is a QRC public key corresponding to a user's credentials for that service or peer only. The public key cannot be distinguished alongside another, even when the same credentials (username and passphrase) are used with another service, or when keys are rotated using the same credentials.

Encrypted message passing is straightforward. The message payload consists of an AES initialization vector (IV) and encrypted unstructured bytes. We stipulate only that it be fully encrypted by the AES key, and be accompanied by no extraneous data subject to manipulation. Additional data may be passed in the query string, but its privacy or authenticity cannot be verified. Serial ID validation (to prevent replay attacks and to correct network delivery ordering errors) is prefixed to the bytes before encryption, and the AES-GCM tag goes at the end. No other metadata is currently required in this methodology, offering an agnostic messaging channel for peers to communicate events and exchange ad-hoc data. This allows end-to-end (E2E) encryption and versatility for message serialization of any type.

The payload components are base64 encoded (with slashes replaced by periods) and appended to the base URL or are passed in the request/response body, delimited by slashes:

1. Sender Public Key
2. CRYSTALS-Kyber QRC Key Exchange
3. Message Payload(s)

Binary data was argued against, because for CORS rules, some MIME types trigger a preflight request. Further, raw bytes cannot be simply passed in the URL path, disrupting the uniformity of the payload encoding across various protocols and use cases. By encapsulating the handshake in the URL path, WebSocket connections can be efficiently authenticated even *before* accepting the connection and performing HTTP/WebSocket upgrade.

When the peer or API service has authenticated the caller, it sends a response body or establishes the WebSocket upgrade, and sends back the same scheme of components, thus achieving true mutual authentication.

Session Management and Extensibility

Sessions, state, and message composition (such as serialization methods and additional E2E encryption) are managed among peers, and can be implemented as desired. Sessions can span multiple requests, if implemented as such, or be contained to unique WebSocket connections. Either way, this protocol allows the flexibility to

design implementation details around a framework and robust algorithms that ensure a solid foundation for authenticated identities and encrypted communication channels, whether peer-to-peer (P2P) or on managed services such as KeySigna.com.

Comparison to JSON Web Tokens (JWT)

Although of seemingly similar purpose/design origins, our methodology differs from JWT in some significant structural ways:

New Methodology	JWT
Valid for single-use, verified sequence only	Validity is assumed for a long period
Stored in encapsulated memory or Web worker	Stored in insecure LocalStorage/cookies
Credentials isolated to trusted encryption layer (optional high-security mode)	Credentials provided to potentially compromised code/CSRF pages
Isolated from the network	Passed over the wire as header/cookie
Revocable	Irrevocable
Stateful / Sessions	Stateless
User Agent generated	Server generated

Tokens, as used in cookies and headers, present significant problems akin to the issue of passing passwords across the wire. They are just abstracted behind a layer of false assumptions, such as the integrity of third-party cookies, the proper management and secrecy of tokens, and the inherent trust that is placed in an authorization artifact that is liberally passed around the network. Between the client's browser, API gateways, services and microservices, JWTs are subject to misuse in many ways. While they do solve some problems, and tighten the attack surface for the authentication interface itself, they open other doors and defer responsibilities, perhaps where they ought not be passed.

Updates, Benefits, and Other Issues to Follow

Some other benefits and issues for discussion, along with posted test results and benchmarks will be published on our site at keysigna.com, include but are not limited to:

Perfect Forward Secrecy	End-to-End Encryption	Observability
No-Breach Credentials	Single-request Handshakes	Mutual Authentication
New Standards	Zero-trust	CDN/Edge Deployment
Privacy Shields	Peer-to-Peer Validation	5G/IoT Applications

Data/Auth. Sovereignty Embedded Microprocessors Cloud Functions

Conclusion

Our formal methodology has been crafted, based on ideas that we have considered to address these problems, since at least as early as 2008. Some of the concepts and implementations are roughly mirrored in the TLS 1.3 Standard of 2018, but they came a full decade after the marked need was identified in the wild by this and other Internet development professionals. We must not wait another 10 years for the prescient issues to be solved, which are evident in today's vital networked security landscape.

The growing need for 5G, IoT, and embedded solutions that run and maintain integrity in vastly diverse and distributed environments, while respecting resources, latency, and cost, are approached here in a practical manner and the described methodology solves numerous of the concerns presently held in the industry.

We request comments and welcome testers and open discussion of the problems and solutions that we must face, to evolve as professionals in an industry where 'security is everyone's responsibility'.

Please contact: `secure@keysigna.com`

References

Hidden Abstraction Layer Papers, Christopher S. Batt, 2008-10 [unpublished].

Zero Trust Networks, Evan Gilman and Doug Barth, 2017. O'Reilly Media Inc.

5 Critical Capabilities for Modern Endpoint Protection, CrowdStrike, Inc., 2022.

Top Security and Risk Trends for 2021, Gartner, 2021.

