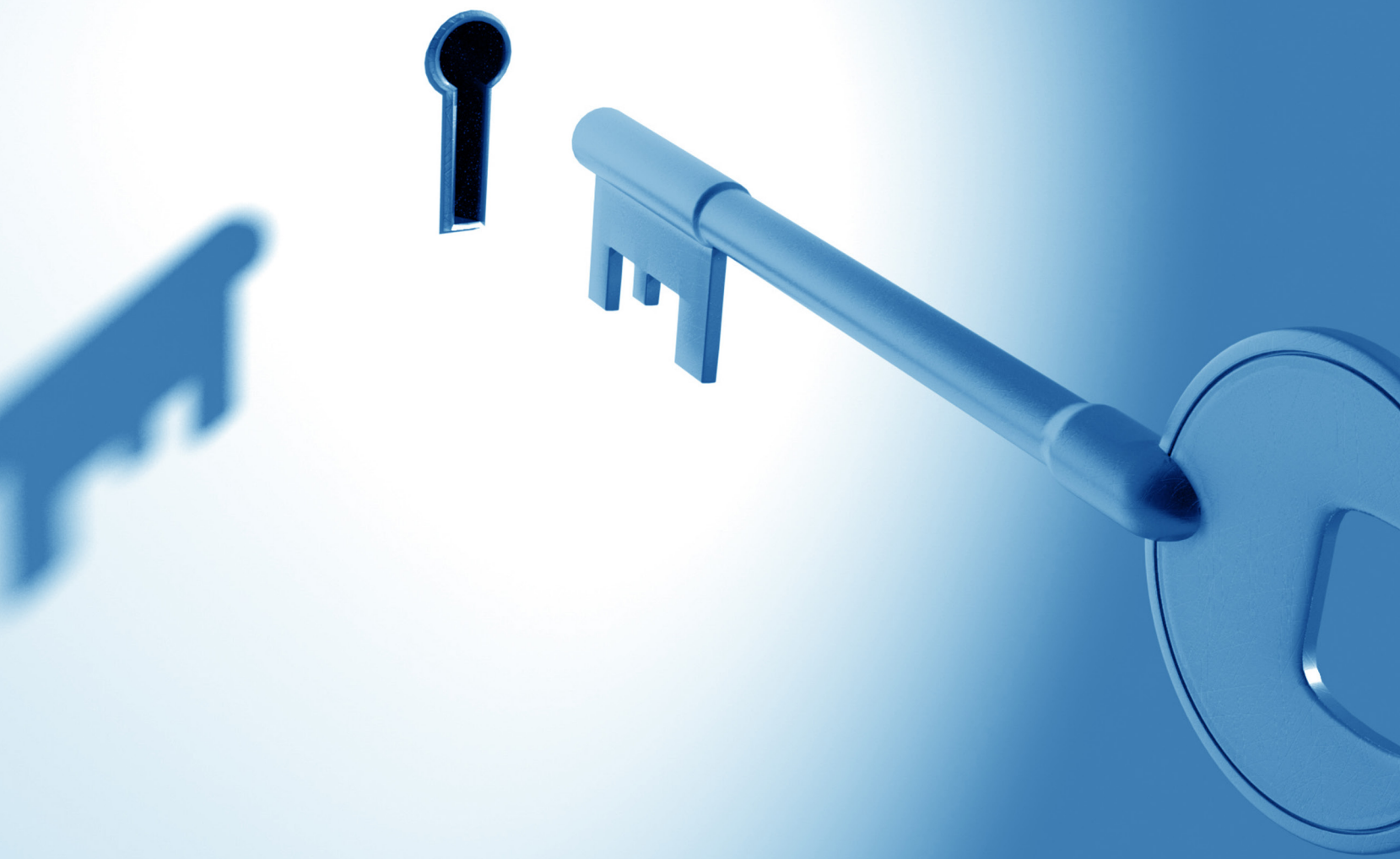
Nubeva | White Paper

It's All About the Keys

New Symmetric Key Intercept Unlocks Modern TLS



NUBEVA

Introduction

Network monitoring, network security and compliance all require full packet inspection. With each passing week, more data is encrypted and that encryption is getting stronger. New methods of encryption, new modes of securing data-in-motion and new computing architectures create stresses on legacy decryption methods upon which the monitoring, security and compliance systems rely. As more systems go blind to newly and more strongly encrypted traffic, the ability to effectively secure enterprise endpoint, cloud and datacenter environments wears thin. Legacy systems have reached a point of diminishing returns where they are increasingly expensive to implement and operate while also becoming increasingly ineffective to decrypt, inspect, detect, monitor and prevent modern threats. A new solution, a new methodology to deliver complete, decrypted visibility is required to restore the capabilities of existing systems and to enable full visibility into new environments and data encrypted in new ways.

> “87% of CIOs believe their security defenses are less effective [when] they cannot inspect encrypted network traffic for attacks. A new solution is therefore required if organizations are to take advantage of the benefits of encryption, yet ensure they are not subject to this new type of threat.”

Encryption: 2020’s Double-edged Sword, TechRadar, Dec. 2019[1]

Pretending that anomaly detection systems, header, log and digest systems are sufficient is a disservice to security and increases risk for an enterprise. While these systems are good at identifying potential problem areas, they lack the detailed visibility required for detailed determination or inspection. These are important and necessary systems for security and performance monitoring. However, while necessary, they are not sufficient for security, compliance and performance maintenance.

There is no substitute for inspection and security of data in motion. Security, compliance and DevOps professionals alike understand that full, decrypted packet visibility is required for deep packet inspection (DPI) capabilities that enable security and monitoring services like intrusion prevention systems (IPS), intrusion detection systems (IDS), application performance monitoring (APM), data loss prevention (DLP), forensics, root-cause analysis and compliance.

Back to Basics

For these systems and the analysts who run them to see all the way into packet traffic, the traffic must first be decrypted. In order to decrypt the traffic you must have the encryption key(s). To get the keys there are many, many legacy approaches. Man-in-the-middle (MITM) pretends to be one of the original endpoints and participates as a proxy in the TLS handshake giving it access to the ciphertext and cleartext packet traffic. Passive approaches replay the TLS handshake with pre-loaded certificate and/or public-private key pairs in order to derive the encryption keys and then decrypt the traffic for inspection. Early termination methods intercept the traffic before it hits its intended destination and decrypt it there for inspection. Application and debug code shims are one-off developer and QA approaches that are delicate operations suitable for troubleshooting an application in a dev environment but not for production and enterprise scale operations.

What all of these approaches have in common is the fact that they replay, duplicate or simulate the full TLS handshake process in order to derive the key that is used to actually encrypt and decrypt the packet traffic. While there are many keys created by the TLS process (something we'll explore later in this paper) only the final, symmetric key is used for the bulk encryption and decryption process that turns cleartext into ciphertext and back. But even here there are challenges posed by modern encryption and privacy standards that are actively thwarting visibility.

In order to provide full TLS visibility, it is the final, symmetric encryption keys that are required (along with the original packet streams and a decryptor function, of course). The ability to obtain symmetric keys is eliminated when perfect forward secrecy (PFS) is enabled. PFS prevents the ability to use pre-configured public/private keys or to derive these from certificate exchanges. Tools and visibility methods that rely on such exchanges or public/private key sharing have gone blind to PFS encrypted traffic. In TLS 1.2, PFS is nearly everywhere. Furthermore, PFS is mandated in TLS 1.3 which was ratified and adopted in 2018 and is being rapidly adopted now[2].

Inline solutions relying on session termination, state maintenance and session re-initiation (e.g. forward-proxies, firewalls and secure web gateways) are incredibly inefficient and come with a very high performance cost — as high as 99% according to NSS Labs[3]. Traffic inspection is still blind to 3rd party and pinned certificate traffic.

It all nets out to the fact that new encryption provides better privacy and protection for data in motion while blinding legacy decryption and the subsequent inspection and monitoring processes that rely on decrypted packet visibility.

The ability to obtain symmetric keys is eliminated when perfect forward secrecy (PFS) is enabled. PFS prevents the ability to use pre-configured public/private keys or to derive these from certificate exchanges.

A new methodology for decryption is required. The only solution is one that is able to discover the final, symmetric encryption keys at the source. This method should work for inline/active and out-of-band/passive systems, and net new/greenfield environments. This new method should enable visibility, inspection and monitoring systems to:

- See more than they can today
- Perform better than they can today
- Simplify visibility solutions, management and processes

This new method should:

- Cost less than legacy solutions
- Protect the investments already made in security and monitoring solutions
- Be future proof so that new encryption methods that emerge may still be visible

This new method is Nubeva Symmetric Key Intercept. It's all about the keys.

The Challenge Faced by Legacy Decryption Methods

Today there are two primary methods that are used by most legacy network visibility and decryption solutions. There are problems with each of these mechanisms. The two methods are:

1. Active/inline MITM — Man in the Middle
2. Passive/out-of-band key regeneration through handshake replay

There are other approaches as well, like using early termination in a load balancer or using code shims inside applications to inspect or mirror traffic from the host after the host completes its decryption of the packets in motion. However, the shim approach is more of a debug tool likely to be used by DevOps for application development and troubleshooting. It does not scale for network level inspection and monitoring. It requires knowledge and maintenance of shims for each and every application as well as constant tending. It is not a viable solution for inspection, security, monitoring and compliance.

All of these legacy approaches were designed for an era when RSA key exchange was the norm. They have evolved somewhat, but the patchwork approach that dealt with all of the incremental improvements to encryption started to fail when Diffie-Hellman ciphers and perfect forward secrecy were introduced. Modern encryption is engineered by design to thwart legacy decryption capabilities. TLS 1.3 specifically disallows MITM that is not always active and participating in the TLS ClientHello as an endpoint. There is no session disengagement allowed in TLS 1.3 effectively turning some legacy MITM systems into session relays and chokepoints. Furthermore, MITM devices must support TLS 1.3 in order to keep the sessions alive. Many MITM devices will downgrade a TLS 1.3 session to TLS 1.2 in order to make its decrypt functions work. However, TLS 1.3 has implemented specific protections against downgrade attacks while still allowing some legitimate downgrades.

MITM and Downgrading to TLS 1.2

TLS 1.3 contains two protections in its specification that protect against downgrade attacks. Any MITM device must support TLS 1.3 in order to prevent itself from looking like a MITM attack. First, both the TLS client and TLS server send a Finished message which contains a MAC over all previous handshake messages. In this way, both the TLS client and TLS server can see that the originally negotiated parameters have not been modified in the middle. A legitimate MITM decryption and inspection device should be participating in the session as the TLS client or TLS server (depending on whether the traffic is outbound or inbound) in order to send a properly formatted finished message and prevent the session from dropping or alerts being fired. Second, in the event that a TLS server receives a downgrade request, then it writes the last eight bytes of its ServerHello. random as "DOWNGRD" [or 44 4F 57 4E 47 52 44 01 for a TLS 1.2 request]. While a TLS server supporting 1.3 can accept a 1.2 request, the TLS client will check the ServerHello for the "DOWNGRD" value and compare it to what it originally asked for in its ClientHello. If it asked to use TLS 1.3 in its ClientHello and gets back a ServerHello containing "DOWNGRD" then it will know that the session has been intercepted and changed and it will drop the session. MITM devices that intercept and change the session traffic will not work in a TLS 1.3 world. Instead they have to fully proxy the traffic, fully terminate, maintain state and then re-establish the connection once their inspection, detection and monitoring tasks are done. While this works, it is substantially less efficient and more costly in terms of compute resources, throughput and latency.[4]

Certificate exchange is encrypted. Certificate pinning disables outbound inspection. With the explosion of SaaS software for end-user use (e.g. Office 365, Dropbox etc.) and 3rd party APIs that all use certificate pinning to secure their communications, organizations are forced to trust 3rd parties and forced to forego their right and responsibility to inspect their traffic.

All of the legacy approaches suffer at cloud scale and are unable to handle modern compute architectures. At cloud scale the volume of traffic is increasing. Transaction volume is skyrocketing due to microservice and elastic compute resources like containers, Kubernetes pods and VM scale sets — whether deployed in datacenters or in the public cloud. Transaction volume is also exacerbated by the ephemerality of TLS sessions and keys where PFS is in play. Shortening session length from TLS configurations, the explosive use of microservices, serverless computing and containers coming into and out of existence according to dynamic scaling events creates an exponential growth of transactions. Each transaction and microtransaction is encrypted. Each must be handled by the legacy approaches. All of this drives up cost and drives down performance of legacy-based systems.

As more traffic is being encrypted there is further interruption to visibility systems and solutions. The industry is making moves to block MITM inspection by design. Specifically, certificate pinning — mentioned above — is used by Google, Microsoft, and nearly all SaaS solutions. Pinning is specifically designed to disable certificate rewriting and connection proxying. Even workarounds like custom CA pushing into client browsers is in the process of being blocked by the big vendors in the industry. It is already standard practice for many in the mobile application space. All of this is evidence that large solution providers are actively engaged in preventing a 3rd party — like an enterprise — from claiming they are them — like rewriting a certificate so that MITM will work.

The industry is making moves to block MITM inspection by design.

Finally, new security and encryption protocols are emerging that are designed to thwart existing legacy inspection. In an age when QUIC and DNS over HTTPS (just to name two) are coming into their own, legacy proxy, secure web gateways, MITM and passive systems are completely out of their depth.

What is important to understand, though, is that all of these new protocols use symmetric encryption keys to perform their final bulk encryption / decryption.

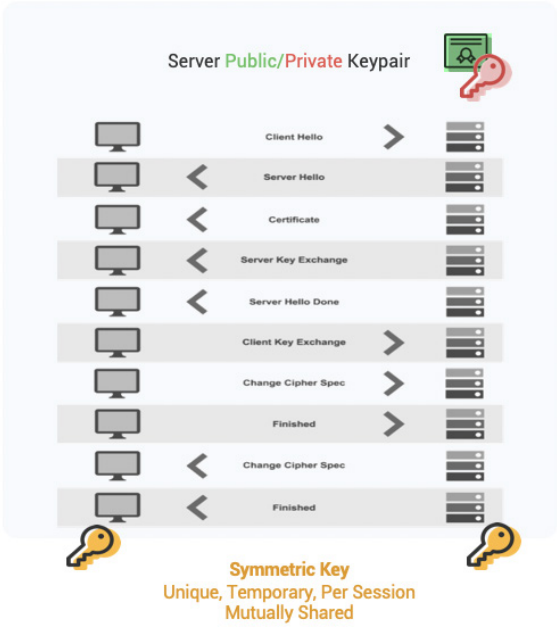
The Cost of Handshakes and Key Exchange in Legacy Systems

At their core, legacy methods for decrypted visibility all participate in an encryption / decryption process where they either directly engage in or replay the TLS handshake, key exchange and final, bulk-encryption key derivation process. This is the most resource and computationally intensive portion of the encryption / decryption process. The challenge faced by legacy solutions for inspection, visibility and performance is that they either intercept or replay the handshake, key exchange and key derivation process, sometimes several times, just to be able to recreate the final encryption / decryption keys. These final keys are then used to decrypt the packets which are then sent to the inspection processes, mirrored to security and performance systems or saved off as cleartext pcaps for later review.

While the original TLS client to server connection requires this process for modern security, only the final encryption keys are required for decryption. Therefore, the methods that require one or more replays/re-handshakes of the entire process become terribly bogged down from a performance and efficiency standpoint.

Running a large number of sessions that negotiate key selection requires complex state machines that cause a substantial performance drag for inline decryption, inspection, prevention and filtering systems. These solutions end up having to increase their bypass filters to accommodate modern encryption, 3rd party calls and certificate pinning. Meanwhile, passive / out-of-band decryption is simply withering away due to PFS as it is impossible to replay sessions and regenerate the final, symmetric TLS keys needed to access mirrored or stored packets. Increasing swaths of compute and network architectures are invisible to inspection.

STEP 1: HANDSHAKE
Authenticate – Negotiate – Key Exchange



Symmetric Key
Unique, Temporary, Per Session
Mutually Shared

STEP 2: SECURE COMMUNICATIONS
Encrypted Data Exchange – Client to/from Server

HW Accelerated, Efficient Symmetric Encryption/Decryption

via Symmetric Key

SYMMETRIC KEY
Required for each session's encrypt and decrypt

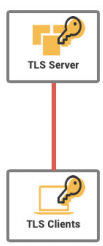
SYMMETRIC KEYS
Are different than Server Public/Private Key

HANDSHAKE COMPUTE COST
VERY HIGH vs pure symmetric encrypt/decrypt

What if There Were Another Way?

What if you could bypass all the replay, regeneration, termination-inspect-re-encrypt cycles? What if decryption could skip the TLS handshake all together but still have access to the final, symmetric keys? What if there was a way to have final symmetric keys when PFS and pinned certificates are enforced, with minimal impact on performance?

- The result would be orders of magnitude faster
- The result would decouple packets from the final, symmetric keys that encrypt them
- The result would allow decrypted visibility into inaccessible areas like pinned traffic



FACT
Symmetric Keys Exist
In Servers' and Clients' Memory
During The Session

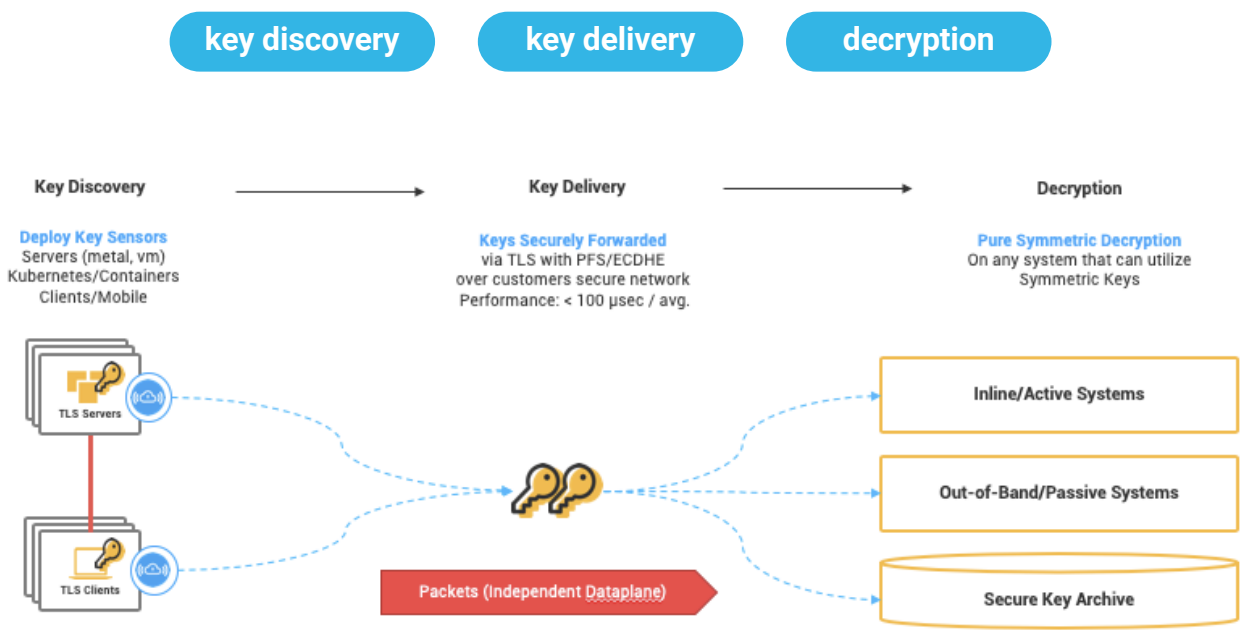
IDEA
Capture And **Reuse**
Symmetric Keys
To Fuel Decryption Systems

There is another way that delivers all of these results. That method is Symmetric Key Intercept and it has been perfected and made scalable by Nubeva. Nubeva Symmetric Key Intercept enables the discovery, extraction and reuse of TLS symmetric keys from TLS server and/or TLS client memory in real-time via a suite of endpoint microservices and agents. Nubeva does this independent of protocol or session type, without changes to code or libraries and without certificates/server keys (PKI/KMS).

Symmetric Key Intercept, or SKI, is the ability to discover the final symmetric encryption keys from active TLS client or TLS server memory. By eliminating the need for handshake, key exchange, and key derivation replay, SKI enhances and enables all network security infrastructure while opening up new areas for inspection.

Understanding Symmetric Key Intercept

SKI has three core parts:



Key discovery is done by read-only key sensors that are deployed on TLS clients and/or TLS servers. Key sensors must reside on at least one of the participating TLS endpoints in order to discover the symmetric encryption keys from active memory. SKI uses a set of signatures that indicate memory locations of keys as well as hunt-rules to discover where symmetric keys are located in memory during a TLS handshake. Using a read-only sensor that is present on either the TLS client or TLS server, the SKI method identifies and retrieves the final symmetric keys and session identifiers.

Key delivery is incredibly fast, completely secure and happens over the customer's own network. Key discovery and delivery are so fast that symmetric keys are discovered and delivered in less than 100 µs (i.e. 1 microsecond is .001 milliseconds). That is faster than the first packet arrives at a destination such as a next-generation firewall or IPS system at typical network speeds. Optionally, keys may be delivered to a key depot for key aggregation, buffering and control. Symmetric key depoting happens in a customer's environment (cloud or data center) and is important to enable parallel scaling and simultaneous multi-use.

Once keys are available, decryption is actually a commodity process that is fast and accurate. The fact is that many systems already have decryption capabilities built in and the decryption is very fast. The performance drag happens when the replay, key exchange and derivation processes have to happen just to arrive at the symmetric key. Nubeva eliminates the need for the entire performance crushing process and retrieves the final, symmetric keys directly. Nubeva's Symmetric Key Intercept delivers the symmetric encryption keys to the decryption systems that are able to use symmetric keys. This bypasses all the exception handling, tinkering and performance degradation typically experienced by those systems when decryption is enabled.

SKI creates a fundamentally new key-plane architecture to support, enhance and enable decrypted visibility in modern environments and for modern ciphers. The management of key discovery rules, key delivery destinations and automated sensor deployment (even in elastic environments) is handled by the Nubeva controller which also includes robust reporting and analytics on performance, sensor locations, keys discovered and triggers for automatic or conditional key intercept.

Nubeva eliminates the need for the entire performance crushing process and retrieves the final, symmetric keys directly.

The Advantages of Symmetric Key Intercept

The SKI method lets inline, active and passive network traffic analysis solutions:

- **See more** traffic like pinned certificate and application packets, 3rd party API and SaaS traffic, PFS, container and Kubernetes traffic.
- **Increase performance** by 8x for inline systems and 2x for out-of-band solutions by eliminating slow and expensive MITM, early TLS termination and TLS handshake mechanisms to get at the symmetric encryption keys. Deliver symmetric keys to systems before the first packet even arrives and enable decrypted visibility at line speed for the first time ever.
- **Simplify solutions** by eliminating all hand-crafted exception rules and constant tinkering. Eliminate the need for public-private key and certificate management. One solution for any environment without needing any application, library or environment code changes.

Augment, Enhance and Offload Inline Systems

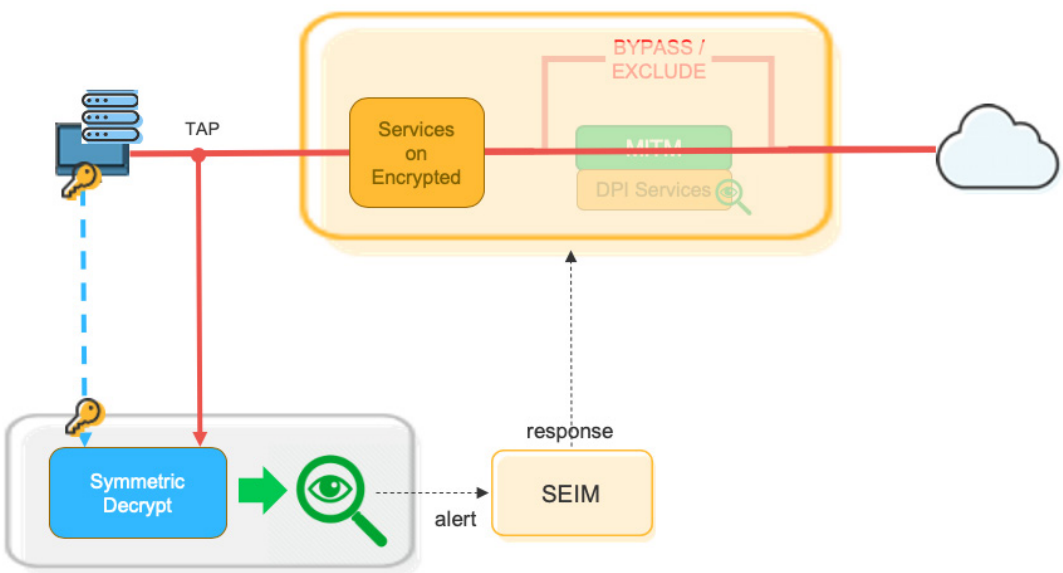
SKI augments inline systems by allowing traffic you want to inspect to be tapped at the inline system and sent to your inspection tools. SKI provides the final session secrets, the final symmetric encryption keys that it has discovered, to your inspection and detection tools.

SKI enhances inline systems by allowing them to inspect traffic that was previously inaccessible to them. Traffic like that encrypted by certificate pinning (e.g. to external platforms like Dropbox, Office 365 or GDrive) is invisible and passed through inline systems. Symmetric Key Intercept capability discovers the final encryption keys from the clients and delivers these keys to your decryption, inspection and detection systems.

SKI offloads inline systems and restores them to peak performance by allowing you to perform out-of-band decryption and inspection while keeping firewalls, secure web gateways and application delivery controllers running fast and performing their core duties like URL filtering and load balancing.

Inbound Inline

Inbound inline inspection is where traffic coming into your environment from the internet is inspected. Specifically for inbound inline inspection, SKI restores the efficiency of inline systems set up to inspect this incoming traffic. SKI allows these systems to offload decryption and deep packet inspection to out-of-band tools that can now see all the traffic to which they were previously blind, while keeping their internal filtering, white-and-black listing capabilities running at peak performance.

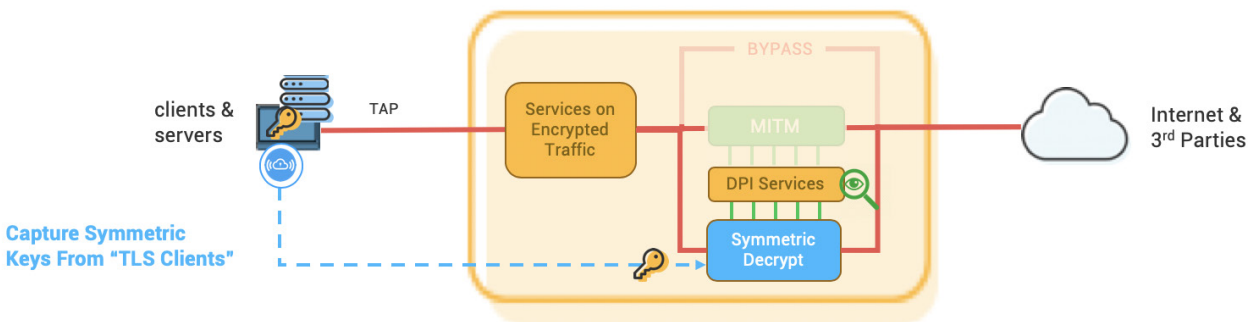


Inline systems could even be updated to accept delivery of symmetric keys to their internal decryptors; enabling them to skip the performance-crushing termination and state maintenance and re-encryption cycles they have to manage today.

SKI simplifies the management of inline inbound inspection systems and reduces the complexity of managing those systems. Inline inbound inspection requires that all certificate and certificate authority and PKI infrastructure be kept up to date, shared correctly among inspection, MITM and proxy devices. This is an incredibly delicate and fragile process that is prone to oversight and simple but severe mistakes. Moving to SKI eliminates the need to manage and share certificates across all network chokepoints just to ensure visibility continues. A host-based Symmetric Key Intercept solution is able to discover and deliver final symmetric keys (session secrets) directly to decryption tools without requiring session and certificate replay.

Outbound Inline

For outbound inline, SKI enables all the benefits of decryption and eliminates the outbound decryption blind spots caused by certificate pinning and 3rd party sessions. Because you don't own the certificate infrastructure for outbound calls, it is impossible for inline systems to gain decrypted visibility to that packet traffic with a forward proxy, MITM solution. Unfortunately, with TLS 1.3, MITM is largely eliminated as an option. Session termination is expensive in terms of compute resources, system cost and complexity and is impractical to architect in dynamic and distributed environments like modern data centers and the public cloud.

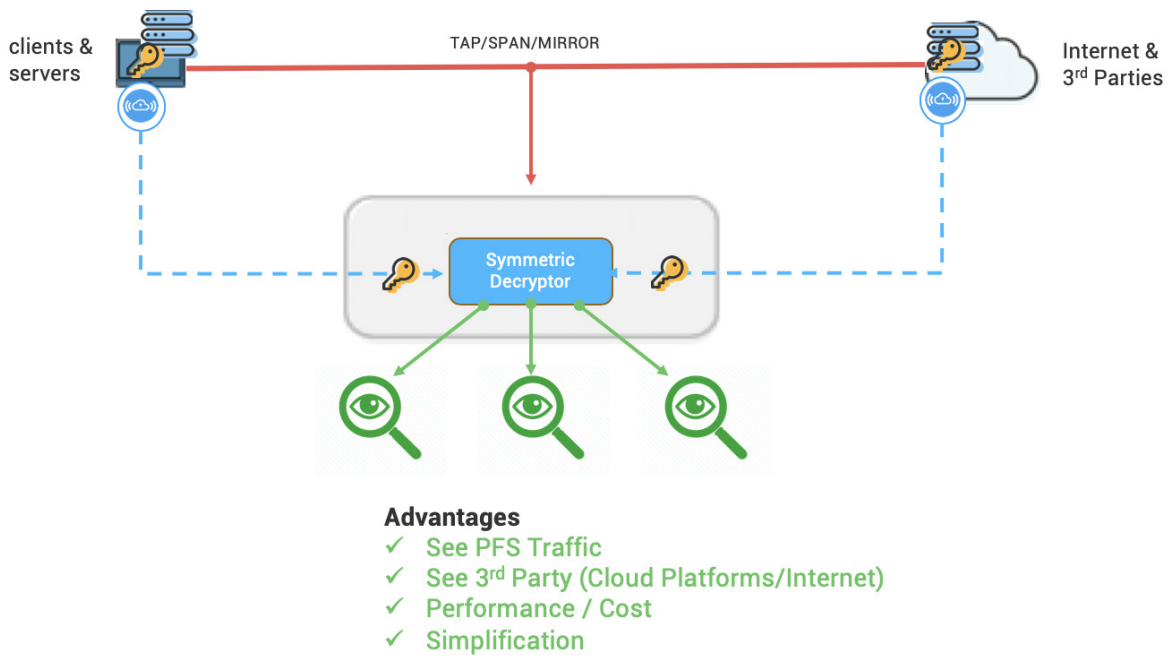


Enable Pure Symmetric Decryption Onboard

Restore and Enhance Out-Of-Band Decryption

SKI restores visibility to out-of-band detection, inspection and forensics tools. SKI has no need to interrupt or be in the middle of a session in order to get decryption keys. This method discovers final, symmetric encryption keys from system memory of either the TLS client or TLS server. SKI obtains the final encryption keys and delivers them to security tools with their own decryptors.

Nubeva uses Symmetric Key Intercept decryption to restore and enhance full passive, out-of-band decryption for TLS 1.2 with PFS and TLS 1.3. Nubeva decouples symmetric key discovery from the final act of decryption, which enables true out-of-band decryption with absolutely zero impact on network performance. What's more, with Nubeva users introduce no risks from passing decrypted content over the wire or downgrading encryption levels. The key discovery sensor identifies the final symmetric key during the TLS handshake and securely provides it to a decryptor — either Nubeva's decryptor or your own. With the Nubeva store-and-forward system, the ephemerality of modern encryption keys can be extended for the purposes of regulatory inspection, forensics, detection or any other purpose. Final encryption keys can be preserved as long as needed or flushed as rapidly as desired. Packet mirrors, taps or brokers — including Nubeva's own mirror capability, create copies of the fully encrypted packet streams and send them to decryptors and security tools. This achieves truly passive, out-of-band decryption. Multiple decryptors receiving the same mirrored packets can each retrieve the same symmetric key corresponding to that packet. This architecture allows unlimited parallel, balanced processing across multiple tools.



How Inspection and Detection Tools are Deployed

Passive, out-of-band decryption can happen with Nubeva's Symmetric Key Intercept approach. IDS, DPI, APT and monitoring tools are often deployed out of band, doing their work on copies of network packets. This out of band, passive tool deployment allows inspection, detection, monitoring and forensics to proceed without impacting network performance or introducing latency. In an active, inline deployment of tools, prevention is possible since network traffic is effectively sequestered while it is being tested and inspected. Simply having out-of-band tools is not the same as doing out-of-band decryption.

Let's see why that matters.

Simply having out-of-band tools is not the same as doing out-of-band decryption.

These are the top three reasons performing decryption out-of-band is more important than decrypting inline and only doing out-of-band inspection.

1. **PFS obsoletes legacy out-of-band decryption.** Older, out-of-band decryption solutions that relied on certificates to recreate encryption keys are obsolete. With forward secrecy only the TLS client and TLS server have the final encryption keys. Older decryption systems relied upon their ability to recreate encryption keys in order to decrypt sessions. That ability has evaporated with PFS in TLS 1.2 and 1.3.

Solving for forward secrecy. The only out-of-band solution is to discover and retrieve the keys from the client or server with a process called Symmetric Key Intercept. Symmetric Key Intercept effectively decouples key discovery from packet capture/mirroring from decryption.

2. **Inline decryption is not tolerated in modern architectures.** Inline decryption requires that it terminate the TLS sessions in order to have access to the final encryption keys. A legacy network edge or gateway setup would perform a terminate — decrypt — inspect — reencrypt — forward process. If you read closely, this is exactly what legacy packet brokers describe. They decrypt in-line and then mirror the decrypted packets to out-of-band tools for inspection and detection while re-encrypting the original session traffic and sending it on. Not only does this process introduce risk, it is also incredibly inefficient. This process simply crushes packet broker and firewall performance (the traditional locations for inline decryption). NSS Labs discovered SSL decryption degraded the firewall performance by as much as 80 percent and reduced transactions per second by 92 percent. Furthermore, edge-based decryption makes no sense in modern architectures. There simply is no edge anymore. Or, said another way, everything is an edge and dropping expensive (and 92% inefficient) hardware chokepoints everywhere is not only silly, it is prohibitively expensive. Relying solely on ingress traffic from your own network is an example of a devastatingly incomplete security solution architecture. It ignores pinned certificates. It ignores all East-West traffic. It ignores all API traffic to 3rd party services and clouds.

Solving for modern architectures. The only practical solution is a modern out-of-band decryption solution that decouples key discovery from packet mirroring from the actual act of decryption. A Symmetric Key Intercept decryption solution does exactly this and is the answer to bring back true, out of band decryption even in modern architectures.

3. There's no middle in modern and elastic environments. This is not the only problem faced by inline decryption. As the in-line vendors are finding out, there is no middle in modern architectures like clouds, Kubernetes and microservices compute environments. Modern compute environments are highly elastic and enable resources to spin up, execute and spin down in microseconds. Inside Kubernetes clusters the entire concept of networking is a completely different animal with fixed IP addressing being non-existent. This means solutions that rely on pre-known IPs and locations are completely unable to work in these kinds of environments. ignoring the problems doesn't make them go away. Pretending like East-West visibility, inter- and intra-Kubernetes and container traffic visibility is unimportant is simply bad advice. Pretending like decrypted visibility only at an ingress edge is sufficient is wrong-headed.

   Legacy solutions that pretend their increasingly shrinking footprint is all you really need are doing the IT and Security and DevOps communities a huge disservice.

Nubeva Sensor

Solving the modern middle. The only practical solution is a Symmetric Key Intercept solution that is able to run in modern compute environments. Fortunately, the Nubeva sensor is such a solution. It is available as a Kubernetes DaemonSet, container based read-only sensor (not an agent) and as a windows or native Linux service. This solution is fully modernized to understand, identify and automatically learn and recognize TLS sessions wherever they occur, not just on TLS servers. The decoupling of final, ephemeral, PFS key discovery from packet mirroring and decryption enables security teams and tools like IDS, DPI, APT and anomaly detection processes to regain full decrypted visibility to packet traffic wherever it comes from. This includes all PFS, all pinned certificates, all Diffie-Hellman based ciphers, all TLS 1.2 and TLS 1.3.

The reality is that fully passive, out-of-band decryption is available, at scale today. Nubeva is the only fully open, out of band, complete PFS decryption and mirroring solution that handles all architectures and elastic environments. Nubeva doesn't pull a "fast one" by talking about out-of-band tools. Instead, our patented Symmetric Key Intercept technology decouples key discovery from packet mirroring and decryption.

Nubeva doesn't pull a "fast one" by talking about out-of-band tools.

This approach lets enterprises extend the ephemerality of modern encryption keys long enough to use them for security inspection and detection, then remove the keys for good. Enterprises can separately mirror the fully encrypted packet streams to any number of inspection and detection solutions. There, the packet streams may be decrypted by the Nubeva decryptor or a firm's own decryptor. The ability to retrieve the discovered symmetric keys via API or via Nubeva's decryptor means that massive parallel scaling is not only possible, it is affordable. Indeed in modern elastic environments, this is the only solution that makes sense.

Nubeva Decryptor

Nubeva also provides its own, software-based decryptor that buffers incoming packet streams and matches them up with the correct symmetric keys to create massively scalable, enterprise grade decryption. Decrypted packets can then be securely delivered to inspection, detection and monitoring tools.

Open and Expand Visibility into Previously Inaccessible Network Spaces

SKI enables security tools and teams to see into areas they've never seen before. SKI's host-based, read-only software sensors not only discover final encryption keys, they also acquire packet traffic from:

- East — West: Inside Kubernetes clusters. Inspect packet traffic from inside the ephemeral networking environment of Kubernetes clusters.
- East — West: Intra VPC / Region traffic. Scale to inspect traffic between domains where chokepoints are impractical or prohibitively expensive.
- East — West Inter VPC / Region traffic. Easily scale to enable decryption of 100Gb/s E-W network nodes without sacrificing performance or breaking the bank.
- Inspect cloud infrastructure calls.
- See inside 3rd party API calls.

Extend the ephemerality of forward secrecy to enable better forensics, root-cause analysis, compliance reviews and application performance monitoring. Symmetric Key Intercept architecture enables discovered keys to be:

- Sent directly to tools for use in decryption and inspection.
- Depot'd in a secure, air-gapped key depot and preserved for future use, whether that is for five minutes, five days or indefinitely.

Advantages of Nubeva Symmetric Key Intercept

	Inline	Passive	Greenfield
See More	Pinned Certs 3rd Party Traffic	PFS 3rd Party Services	Compute environments like Kubernetes, Containers, 3rd party API calls
Performance	Systems run 8x faster by offloading or eliminating MITM mechanisms for decryption	Systems run 2x faster by eliminating the need for handshake replay to decrypt	High speed scaling and parallel processing that pushes performance off the charts
Simplicity	Eliminate all hand-crafted exception rules and constant tinkering to keep systems working	Reduce and eliminate the necessity for certificate and key management	One solution for any environment. No code changes, no application changes, no library changes required.

Demystifying Keys

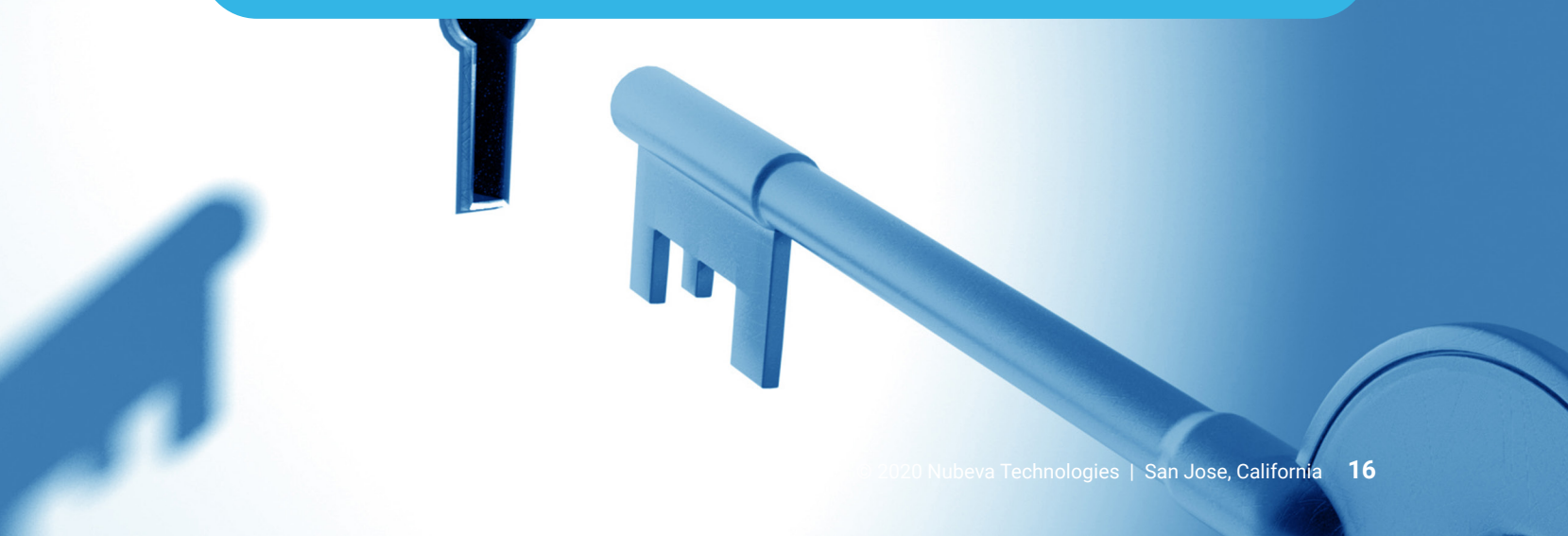
Encryption keys are an ambiguous topic. There are many different kinds of keys that are used in different parts of the encryption and TLS connection, validation and identification processes. It's important to understand the different keys and why symmetric key intercept is such a disrupting process.

The actual data being transported is encrypted with one or more symmetric keys during the session. The TLS client and TLS server both determine the same, unique symmetric key to use based either on a pre-shared key or a key-agreement protocol like Diffie-Hellman.

According to NIST[5] there are many different types of cryptographic keys that have different purposes. In this paper we are concerned only with three of the keys in the encryption / decryption process. These keys are:

- **Private authentication key:** A private authentication key is the private key of an asymmetric-key (public-key) key pair that is used with a public-key algorithm to provide assurance of the identity of an entity (i.e., identity authentication) when establishing an authenticated communication session or authorization to perform some action.
- **Public authentication key:** A public authentication key is the public key of an asymmetric key (public-key) key pair that is used with a public-key algorithm to provide assurance of the identity of an entity (i.e., identity authentication) when establishing an authenticated communication session or authorization to perform some action.
- **Symmetric data-encryption key:** These keys are used with symmetric-key algorithms to apply confidentiality protection to data (i.e., encrypt plaintext data). The same key is also used to remove the confidentiality protection (i.e., decrypt the ciphertext data). Note that for authenticated-encryption modes of operation for a symmetric key algorithm, a single key is used for both source authentication and encryption.

For a complete list of the keys involved in the encryption / decryption and secure traffic transport process see the NIST document from May, 2020 "Recommendation for Key Management" (PDF).



Conclusion

Decrypted TLS visibility in the modern age is really all about the symmetric keys. Because modern encryption standards render legacy decryption methods impotent, the symmetric keys are required to restore, enable and open visibility.

Symmetric Key Intercept is able to deliver final symmetric encryption keys to systems, tools and processes. Many decryption systems have the ability to decrypt using symmetric keys. Those systems need to first be able to receive the symmetric keys and then use them to perform their decryption services. Any tool that is able to receive final symmetric keys for use in its own internal decryptor can receive keys from Nubeva via secure API calls. Any tool that is able to read symmetric keys from a file on that tool can also perform fast, symmetric encryption since Nubeva is able to write keys to a file. Any system that is able to perform symmetric decryption if-only for the keys, has the potential to be helped by Nubeva's SKI approach. Users are those who desire to implement Nubeva in order to provide decrypted network visibility to their own systems and Open Source tools can start using Symmetric Key Intercept today. Nubeva has reference implementation and AWS quick starts available for open source tools like Zeek, Moloch, Suricata and others. Enterprises and security teams using these tools are able to immediately expand the operating footprint of their tools and enable them to see and inspect any encrypted content in the public cloud, private cloud or data center. Open source tools that inspect network packet traffic typically allow the Nubeva decryptor to be installed directly onto the tool.

Security and monitoring tool manufacturers have a deep interest in gaining access to decrypted packet streams. Tools that deliver IDS, APT detection, threat hunting, anomaly detection and performance monitoring based on their ability to process and inspect fully decrypted packet traffic will require a symmetric key intercept approach to remain viable moving into the future.

Nubeva's Symmetric Key Intercept methodology for delivering decrypted visibility reduces cost, protects your existing investment and future-proofs your security and monitoring infrastructure.

Reduce Cost

- Price performance increase and investment protection of your existing tools
- Achieve better price performance by allowing your existing products to last longer by restoring their ability to inspect and monitor traffic
- Achieve better price performance by reducing the number of products you must purchase in order to achieve the same end result

Protect Your Existing Investment

- Extend the ROI on your existing solutions, teams and processes by allowing them to work again instead of going blind to modern encrypted traffic
- Restore functionality and performance of your inline and out-of-band systems

Future Proof Your Solutions

- SKI focuses on the final symmetric keys regardless of the processes, protocols and technologies used to generate them
- Remove exception tinkering and corner cases
- Create one common solution across all applications and implementations

The Nubeva Value

Nubeva technology eliminates the need for MITM, handshake replay and key regeneration just to obtain the final encryption keys that are actually used to decrypt network traffic. Nubeva completely changes the game for modern decrypted visibility by focusing on final, symmetric keys and retrieving them directly from memory. Nubeva eliminates the need for slow and insecure mechanisms like handshake replay or TLS termination to achieve visibility. Nubeva enhances and further enables all network security infrastructure and opens new areas for inspection.

Sources and Citations

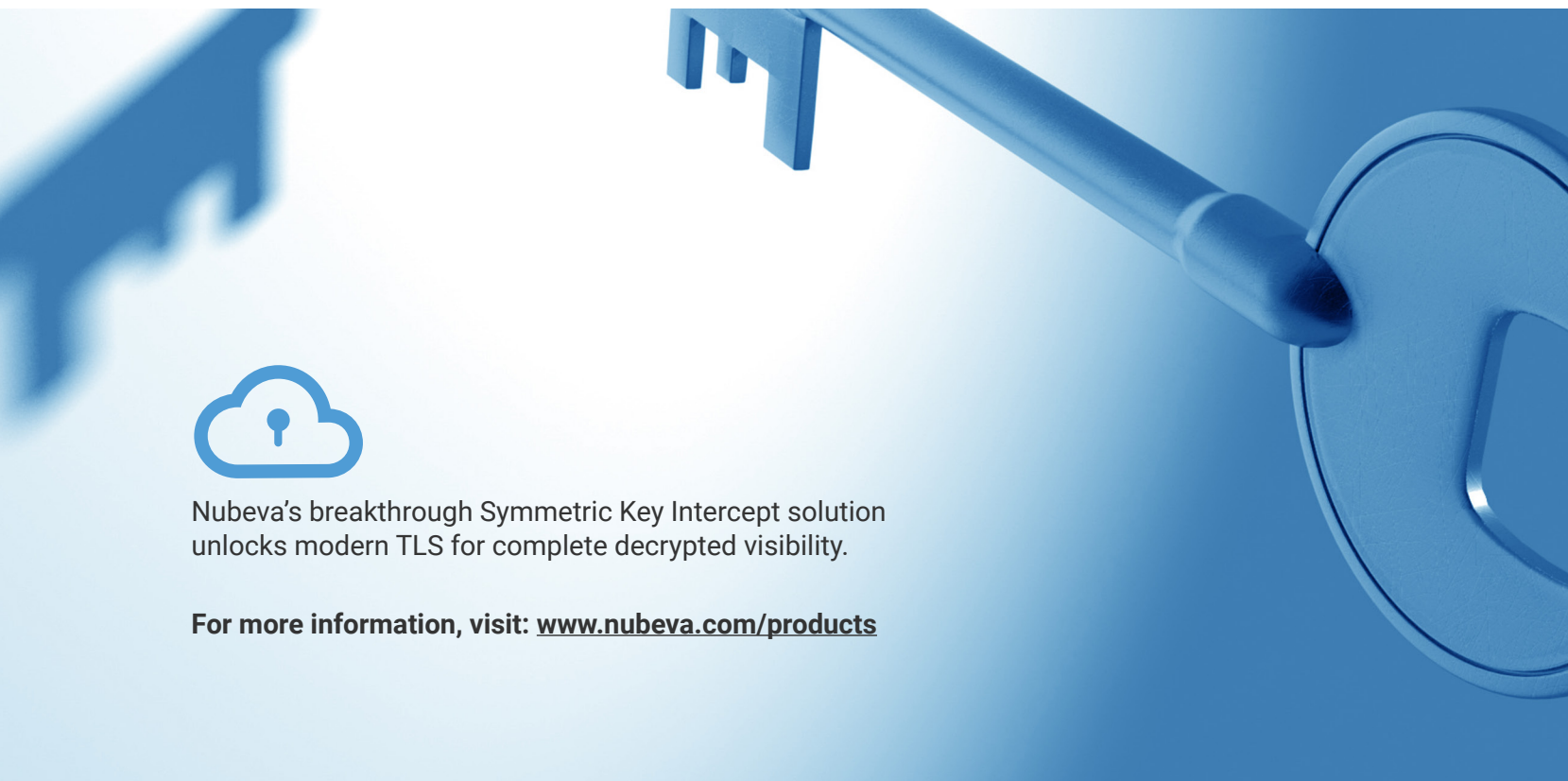
[1]https://www.techradar.com/news/encryption-2020s-double-edged-sword

[2]https://www.ietf.org/blog/tls13-adoption/?

[3]https://www.globenewswire.com/news-release/2018/07/24/1541279/0/en/NSS-Labs-Expands-2018-NGFW-Group-Test-with-SSL-TLS-Security-and-Performance-Test-Reports.html There was a 92% drop in the average connection rate of the tested products, connection degradation ranged from 84% to 99%. Latency in the average application response time of the tested products increased by 672%; latency ranged from 99% to 2,910%. There was a 60% drop in the average throughput of the tested products, throughput degradation ranged from 13% to 95%.

[4]More detail on how TLS 1.3 handles downgrades can be found on The Blog Of A Gypsy Engineer https://blog.gypsyengineer.com/en/security/how-does-tls-1-3-protect-against-downgrade-attacks.html

[5]NIST SP 800-57 Part 1 Rev 5, "Recommendation for Key Management: Part 1 — General", May 2020. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf pp 30-32



Nubeva's breakthrough Symmetric Key Intercept solution unlocks modern TLS for complete decrypted visibility.

For more information, visit: www.nubeva.com/products