HKDS

The Hierarchal Key Distribution System

A fast, compact, high-security key distribution system designed for the 21st century.

HKDS

The Hierarchal Key Distribution System

Welcome to the Future

All over the world, there is an ongoing migration from traditional forms of paper currency, to electronic payment systems. Consumers and businesses both benefit from the reliability, security and convenience of these new systems, and electronic payments are replacing paper currency as the primary from of consumer payment.

The advantages of electronic payments have caused businesses to reduce their reliance on traditional currency, governments are using electronic payments for everything from disbursing social benefits and income tax refunds, to collecting fines and transit fares. This transition to electronic currency has provided economic empowerment to more people than ever before, and now accounts for more than 70 percent of consumer spending worldwide.

In order to meet these challenges, of rapidly expanding scale and increasing complexity of electronic payment systems, the financial services industry relies on the cryptographic community to provide secure systems, that protect these sensitive transactions, and enable the secure transfer of currency over public networks, while using remote devices whose physical security cannot be guaranteed. As more people, more businesses, banks, and governments are adopting electronic payments as their primary means of doing commerce, ever greater demands have been placed upon the infrastructure required to process these financial transactions.

One of the primary tools used by financial institutions to process electronic payments are distributed key systems, used by point-of-sale devices to encrypt secret PIN information that verifies the account holder. The preeminent protocol in use today, is the Distributed Unique Key Per Transaction protocol, or DUKPT.

DUKPT was originally developed by VISA, as part of the first encrypted payment systems, and has been used industry-wide for this purpose for nearly forty years. It is also a computationally expensive and difficult to scale solution, and its use has proved increasingly expensive and problematic as these payment systems continue to grow. What we present here is a new solution, a modern replacement for this aging key management scheme, one with superior security properties, unlimited scalability, unparalleled performance, and that provides a clear pathway to future increases in security requirements as we enter the age of new threats posed by quantum computers.

High Security

HKDS is built on the Keccak functions SHAKE and KMAC. Both SHAKE and KMAC are FIPS approved and part of the NIST SHA3 standard. Keccak is one of the most well-studied families of cryptographic primitives in use today. The Keccak team is comprised of some of the most senior and respected cryptologists in the world, including Joan Daemen, the co-author of Rijndael, the AES standard. There are dozens of cryptanalysis papers on the Keccak design, both by the Keccak team, and third-party cryptanalysts. Keccak was selected as the NIST SHA3 standard after nearly 4 years of intense study by the world's cryptographic community, making it one of the most studied, best documented and most thoroughly cryptanalyzed cryptographic primitives in modern times.

SHAKE is a part of the NIST standard as an extendable output function, a pseudo random function that strongly resembles a random oracle. It is suitable for generation of pseudo-random keying material, key derivation, hashing, and as a key-stream generator in a stream cipher construction, all uses of which are thoroughly documented, extensively analyzed and proven in the Keccak documentation.

KMAC, the message authentication code generator, is also NIST approved, and widely accepted within the cryptographic community as the successor to older functions like HMAC. Nearly half of all entries in the ongoing NIST Post Quantum asymmetric cipher competition use SHAKE internally as a pseudo-random function, demonstrating it is preferred by today's top cryptologists over older pseudo-random generators, and signaling its wide acceptance within the cryptographic community. Further, quantum-based attacks have not to date, found any serious reductions in the security of either the SHAKE or KMAC constructions, aside from Grover's algorithm, which affects most symmetric primitives, and can be countered by doubling the size of the key.

Because of the emergence of large-scale quantum computers, expected within this decade, the use of 128-bit keys will soon become obsolete. All industries will be forced to change to 256-bit keys in order to maintain the security of their systems and comply with new laws, regulations, and standards imposed on the industry. The path for this migration in distributed key management systems has been proposed to be DUKPT using AES-256, but that represents a doubling of computational expense over DUKPT AES-128, and an enormous increase of infrastructure would be required for that transition. Though AES is widely accepted in the industry, it is also more than twenty years old, and in the context of DUKPT, and particularly when using 256-bit keys, is a very inefficient and costly key management solution.

HKDS is extremely efficient, it has no expensive key schedule, does not require a complex scheme of cascading key derivations, and changes to the security from 128 to 256-bit keys, require only a small penalty in performance. In fact, HKDS can use up to 512-bit keys, and should quantum computers and attacks on symmetric cryptosystems ever advance to the point where 512-bit keys are required, HKDS can easily accommodate that change.

The HKDS master key, has two keys, the base derivation key (BDK), which is used to generate embedded device keys for point-of-sale devices, and can be shared with POS device manufacturers, and the secret token key (STK), known only to the transaction processing institution, providing a separation of secrets, whereby the key cache generation function used by the client and server, requires both the embedded key, and the secret token to generate the key cache, and only the owner of the master key, can recreate the key-cache and retrieve a transaction key.

The base STK is combined with the device's manufacturer and device identity strings, and a token counter, and used to key the PRF, which generates an irreversible secret token unique to that device and token request, so that each device, receives a different secret token, every time a token is requested. That token is combined with the embedded device key on the client, to key an instance of the pseudo-random function, that generates a set of transaction keys, that are cryptographically unique across all devices that use that master key, every single time.

This token is encrypted and sent to the client at periodic intervals, when the clients key-cache is exhausted (the key cache is 42 keys with 128-bit security, 34 keys with 256-bit security, but these cache sizes can be tuned to larger or smaller cache sizes). This injects new entropy into the PRF, each time the key cache is created, and the base token key STK, can be changed at any time, without any intervention required on the client device, and no interruption in service. This injection of new entropy is an efficient and secure process, that only need run periodically, and has only a very small impact on the performance of the system, just two additional messages once every cache cycle.

Contrast this system to DUKPT, which could feasibly be using the same base derivation key, for tens of thousands of devices, producing 100s of millions of keys from that one base key, without an injection of new entropy, and that any mistakes made in the application of that system, or a compromise of the base derivation key, could constitute wide-scale compromises to the entire network. That HKDS can be scaled to millions of devices per master key, that it can safely produce many millions of transaction keys on each device, that new entropy is constantly added, and that the keys can be changed at any time, without costly recalls or upgrades, distinguishes HKDS as a vastly superior key management system.

HKDS also employs a sophisticated message authentication scheme, using KMAC in an encrypt-then-MAC configuration. Token exchanges are always authenticated, and per the specification, messages can be optionally authenticated as the recommended use. In this capacity, benchmarks show that KMAC used in place of the traditional HMAC(SHA2) authentication mechanism, delivers a substantial performance advantage, while using a powerful authentication protocol, that re-uses the same permutation as SHAKE, creating a smaller and more efficient software footprint.

Scalability and Performance

Many DUKPT implementations now use AES-128 as the primary pseudo random function (PRF) to mix cryptographic keys and derive new ones. Prior to the use of AES, DES the Digital Encryption Standard and a stronger form of DES; Triple DES were used. Triple DES is still in use today in legacy systems, but it is widely regarded as an outdated and a potentially insecure cipher, and the industry has been steadily moving to replace 3DES with AES. The problem with using AES in these systems, is primarily one of cost; DUKPT uses a complex and expensive series of cascading key derivations, which rise in computational cost as the number of keys generated increases. AES has an expensive key schedule, a function of the cipher that turns a small input key into a much larger set of round-keys required by the transformation function. Each time AES is used in this complex scheme of successive derivations, the cipher must be re-keyed. On the server that decrypts a message from a client, this is a computationally expensive operation. Using the AES form of DUKPT as outlined in ANSI X924-3, can require as many as 18 AES-128 re-keys and calls to the rounds function to derive just one transaction key. There is also a 256-bit AES implementation, one that the industry will soon need to migrate to, as threats from quantum computers that effectively half the key size, will make traffic generated with AES-128 readable, and the AES-128 cipher obsolete. The AES-256 form of DUKPT uses exactly double the number of expensive cipher re-keys and calls to the AES transformation function, requiring as many as 36 of these operations to derive a single transaction key on the server.

HKDS does not use a key schedule, it does not require these expensive re-keying operations, and it does not require the complex cascading derivation operations of DUKPT, it works in a completely different way, one that is far more efficient, and more secure than DUKPT. HKDS is a 2-key system, one that uses an embedded key to encrypt and decrypt secret tokens, generated by the server from a base secret token key, and which combined with the embedded key, keys a PRF to generate a key cache; a set of transaction keys stored on the client device, that can be efficiently recreated on the server. This token injection system is periodic, once the client's key cache is exhausted, it requests a new

token, that is derived from a master token key, encrypted, and sent to the client to generate a new key cache.

There are many advantages to this system; embedded device keys on the point-of-sale devices, will never require updating. This system can generate 10s of millions of transaction keys safely and securely, without the expense of replacing terminal devices or localized re-keying of client devices. The base secret token key can be replaced at any time, without the need to modify the remote device, such that if it is suspected that a master key has been compromised, that key can be replaced, without any direct intervention with, or replacement of remote devices. New entropy is constantly being injected into the system, unlike DUKPT, which derives as many as half a million keys per device, on many different devices, all derived from the same base key, such that if that master key were compromised, all past messages on every device could become readable. HKDS works in a way that each device, receives a unique and irreversible token key, each time that token key is updated. Theoretically, using a single master key, more than 65 thousand manufacturers, each with up to 4 billion devices, with each device producing more than 2 billion transaction keys, will produce different transaction keys across all devices, without a single collision in the key-space.

Perhaps the most impressive characteristics of HKDS though, is its performance. Our benchmarks, which measure the complete set of transactions including token generation, encryption/decryption, authentication, and key cache generation, have shown a phenomenal difference in performance characteristics over DUKPT. On point-of-sale devices, the code is smaller, uses less memory, and operations like PIN encryption are more than a hundred times faster than DUKPT. On the server side, where performance is critical and represents the greatest costs in these transactions, HKDS is a full 4 times faster than DUKPT using AES-128, and 7 times faster than DUKPT using AES-256. This is when using embedded AES-NI instructions in an optimized C implementation of DUKPT versus a C software of HKDS with no SIMD speedups. Clearly DUKPT is no match for HKDS, and the reader should further understand, that Keccak instructions embedded on CPUs similar to the AES-NI, could offer a near doubling of performance advantage.

Competitive Advantage

The owner of this technology will possess a set of powerful competitive advantages over other actors in the industry. This technology allows an acquiring bank to perform transactions between point-of-sale devices and their servers, at a small fraction of the cost of the current methods, and those savings, in infrastructure, utilities, maintenance and personal, can be used to increase

profit share by drastically reducing costs through this more modern, secure, and efficient method.

Not only will they own a technology that can vastly outperform the cryptographic functions used by their competitors, but existing infrastructure, can be repurposed for other financial applications, like online transactions processing, crypto currency, or to expand capacity without having to invest in additional hardware; costs savings that could reduce the impact of expensive upgrades to meet increasing demands for many years.

The owner of this technology will also be able to license its use, and given the tremendous performance advantages, FIPS compliance, and the ease in which this system can be integrated to replace older technologies, it is certain to become the new standard in distributed key management technology. This means the owner will be able to control how and who uses this technology, generate large yearly revenues from licensing, while maintaining a competitive edge by controlling who can and cannot use the technology.

DUKPT is not FIPS compliant, it does not follow FIPS recommendations, nor use a FIPS approved RBG for key derivations. HKDS uses SHAKE and KMAC exclusively, both FIPS approved, and backed by extensive cryptanalysis. Standardizing DUKPT is still ongoing in some forums like ISO after nearly 40 years of use, because of its design, and that it does not use approved mechanisms, whereas HKDS uses components from the NIST SHA3 standard, and worldwide standardization efforts which have so far proven difficult for DUKPT, should be far simpler using this new mechanism.

HKDS is not limited to point-of-sale devices, it can generate keys that can be used in VPNs, such as those connecting financial institutions, by generating enough keying material to key an authenticated stream cipher, used to encrypt communication channels between institutions. HKDS can also be used in other terminal-based devices such as ATMs, lottery terminals, secure access terminals, or anywhere a high-security communications channel is required.

Asymmetric cryptography will always be the ‘weak link’ in our secure communications channels, far more susceptible to increases in computing power and mathematical breakthroughs than symmetric cryptography, as evidenced by the imminent failure of RSA and elliptic curve cryptography that will soon be made obsolete by quantum computers. Symmetric cryptography however, if applied with a sufficiently strong cipher, and large enough key size, could feasibly never be broken. Requiring systems that need true long-term security, security that must be unbroken for a lifetime, should be symmetric cryptography based. We believe HKDS can fill that vacuum, by providing a fast, scalable, efficient, high-security key management system.