Andy Manoske Follow
Product @ Hashicorp. Amplifier @ Amplify Partners. Loves live music and dead languages. Comically terrible at parallel parking.
Dec 18, 2017 · 14 min read

The Next Decade of Crypto

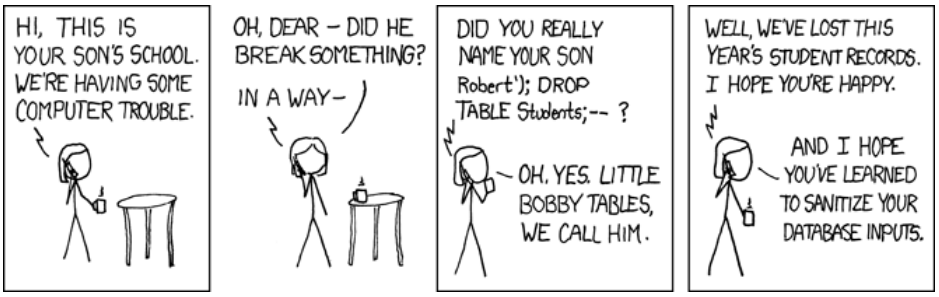
Cryptography Trends and Technology in the 2020's

The 2010's have been a landmark decade for cryptography. From the rise of Bitcoin and blockchain technologies to the democratization of encryption and key management with software like Intel's AES-NI libraries, encryption has been thrust into the mainstream as a tool for dealing with a host of major computing problems—both within security and without.

As the decade comes to an end, the security and cryptography community stands at the beginning of a new series of opportunities and challenges. New adversaries and seismic shifts in infrastructure technology over the next ten years pose significant challenges to encryption technology. But enabled by advances in software and economic cryptographic computing architectures, the 2020's will see cryptography protecting global data on a scale unseen in human history.

How we got here (2007–2017)

Before we start looking at the next ten years ahead, we need to look back at the last decade to fully appreciate the huge shifts in cryptographic technology that have taken place to give us a strong foundation for the next set of big leaps ahead.



In order to respond to the surging epidemic of stack overflow attacks, modern computers use TPM chips for protecting applications in-memory. These chips have also enabled powerful, economic cryptographic computing, and modern protocols like SSL routinely use more powerful cryptography than "military-grade" crypto in the previous decade.

For much of the last 10 years cryptography has been by a radical advance in economical computing power. Since the Nehalem chipset premiered on first generation of Intel's Core i* processors, most modern computers have contained a dedicated crypto sub-processor called a **Trusted Platform Module** (or TPM) to handle low-level authentication of components like an operating system and isolate memory between applications.

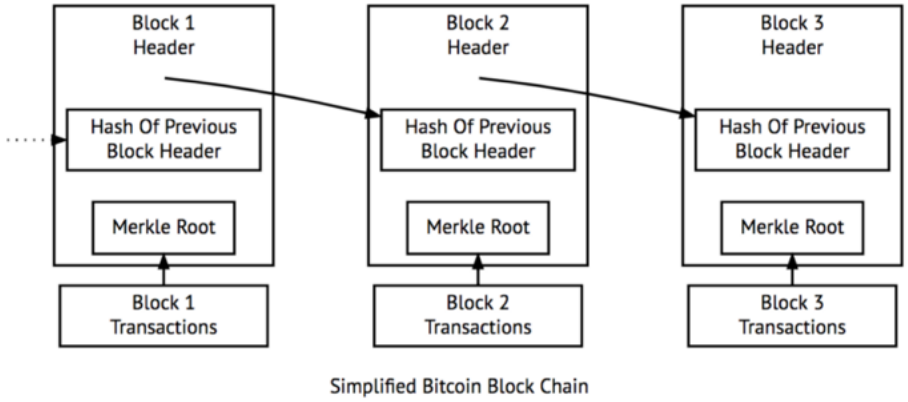
TPM chips rose to prominence out of the surging rise of complex code-based attacks like **buffer stack overflow** attacks wherein attackers exploited vulnerabilities in source code to "ram" data beyond the allotted memory for an application and force a computer to unwittingly execute instructions or divulge sensitive information (such as passwords or PII datar) from other running applications.

In response, TPMs use hardware-encoded cryptography modules to allow operating systems and applications to "wrap" the memory around applications. If a stack overflow attack were to occur, any information accidentally divulged by the computer would be encrypted ciphertext—yielding no advantage to the attacker even in a successful attack.

TPMs similarly provide the capability to certify applications and operating systems that are approved by an OS manufacturer by quickly decrypting and certifying public key cryptography certificates. Both of these usecases are enabled by powerful, low-level cryptography instructions written into modern processors that—in addition to satisfying the above usecases—have also been drafted into supporting general purpose cryptography across systems.

The result of the popularization of TPMs is that the modern consumer grade computer contains a processor that can quickly encrypt and decrypt most popular cryptography schemes faster than dedicated crypto processors in the previous two decades.

Such a hardware baseline has enabled the creation of powerful open source cryptography libraries such as Golang's crypto and x-crypto libraries and OpenSSL, and today in 2017 we find it normal to use cryptography that is an order of magnitude more powerful than the minimum level of cryptography required to protect Secret level military secrets in the early 00's.



Satoshi Nakamoto's 2008 paper on Bitcoin thrust secure multi-party computing into the forefront of popular technology and has disrupted how we think about trusted computing

Just under a decade ago, another major advance rocked the world of cryptography and computing that we are only now starting to appreciate. In 2008, an unknown internet user (or users—the vote is still very out on this one) wrote a technical whitepaper that described a secure, decentralized peer to peer protocol for transferring money. Called B**itcoin,** this currency would provide a transfer protocol and monetary instrument whose supply was automatically updated relative to its transaction volume.

Bitcoin's security lies in its use of cryptographic mathematics—hence the name "**cryptocurrency**." Instead of using a third party certificate authority to verify the authenticity of a computing node that hosted and managed the currency, bitcoin's **blockchain** protocol relies on a *trust no*

one model wherein a common ledger is maintained and certified purely on consensus and the verification of transactions via public-key cryptography.

While bitcoin itself is a topic worthy of its own blog post (or book), its popularization of untrusted distributed computing technology has brought what was once an arcane area of secure computing to the forefront of mainstream technology.

These advances, and others, have prepared us to make massive leaps forward in the next decade.

1–3 Years: Protecting Multi-Cloud Infrastructure

In the 2020's, most enterprise computing workloads will be distributed across multiple cloud services. Supporting this diverse "Multicloud" infrastructure is complicated and seriously disrupts traditional cryptographic infrastructures around key management.

In the 2010's, the world of infrastructure technology was rocked by the rise of public cloud computing. According to Gartner, public cloud operators such as Amazon AWS and Microsoft Azure operate in a $209 billion global market with an almost 20% year over year growth rate.

This massive, surging market (for comparison: the world spends and order of magnitude more on public services and software every year than it does on producing and exporting coffee) is driven by businesses

moving production workloads to the cloud to fulfill demanding performance and availability requirements.

In a world where most people on earth have a mobile computer more powerful than a high performance gaming computer from the previous decade, the cloud is necessary to spread compute, storage, and networking in order to enable stable, high performance applications.

But the "cloud" isn't just one vendor. As Azure and GCP gain steam, the world is increasingly looking for ways to distribute workloads across different public cloud providers in order to maximize data availability and performance while minimizing cost.

Availability is key here, as in some geographies new regulations around cryptography and data sovereignty ensure that specific domestic vendors gain major strategic advantages in hosting infrastructure for local users. This is especially true in China, where 40% of the world's daily internet users reside.

Such a diversification in public cloud resources has led to the rise of **Multicloud Computing:** the ability to manage a single, performant infrastructure across clouds. Multicloud has serious ramifications for all aspects of infrastructure technology. But for cryptography, it is severely disruptive.

In a multicloud world, already complex concepts of key management and key storage get much more complicated. If I'm having AWS, GCP, and Azure users encrypt and decrypt data, where am I storing the keys for those transactions? How do securely handle that encryption and decryption in a way that attackers can't intercept either keys or plaintext in cross-cloud communication? And, simply put, how do I even roll out computing infrastructure that can host *any* kind of performant data encryption in this environment?

These challenges have already had significant repercussions for the market for software secrets management—a nearly $8 billion industry that with a sizable 18% CAGR over the next four years.

Industry incumbents like CyberArk have turned to acquisition to deal with this disruption in key management form factor, acquiring newer players like Conjur in an attempt to support centralized secrets management across diverse cloud infrastructures using a traditional, monolithic key management and key storage architecture.

Other open source software such as Square's KeyWhiz, Docker Swarm, and HashiCorp's Vault (*Full Disclosure: I'm the product manager for Vault*) have simply done away with traditional key management in favor of other multicloud-friendly approaches to protecting secrets in flight and at rest with encryption.

For example, rather than relying on a personally managed KMIP server to process, distribute, and encrypt keys, Vault automatically self-manages the encryption for its key value stores without user input and employs Shamir's Key Sharing Algorithm to compose and distribute access to that infrastructure. Docker Swarm uses a distributed key model wherein critical data on nodes (regardless of that node's location) is encrypted with a key shared among its partners in a network.

The disruption of multicloud computing will have significant consequences for traditional secrets management. But as the 2020's gets fully underway, the economic and technology trends that wrought multicloud will seriously disrupt cryptography as a whole.

3–5 Years: Protecting New Data Architectures

Global availability and flexibility of compute and data resources are driving the world's move to multicloud. But by the mid 2020's, these forces may drive us into a world where significant compute and data resources operate on large, publicly available resources that require serious changes to how we think about distributed computing and cryptography.

Multicloud has been driven by a need to globally distribute data to provide performance and availability to an ever-increasing set of computing resources worldwide.

These forces have ensured that by the beginning of the 2020's multicloud will become the norm. And by the middle of the decade, momentum from these same forces will likely drive two other areas of distributed computing into the mainstream: Secure Multi-Party Computing (or SMPC) and serverless architectures.

Let's first address the more well-known of these two topics: serverless. **Serverless architectures** refer to a new standard of computing wherein some isolated tasks are performed without contextual information of where that task is performed or what resources it may run on.

Less technically, serverless computing jobs are "one-offs" like shuffling a deck of cards or arranging items in a database. These tasks don't necessarily require contextual information about the system that is running them, and they can be composed in a way that they can theoretically be run anywhere.

This is extremely advantageous in a multicloud world from a price economics and data availability perspective, and when properly instrumented serverless solutions can even be used to solve large, complex problems that can be uniquely addressed with divide and conquer-style solutions like dynamic programming.

Like shooting stars in a meteor shower, serverless tasks are ephemeral and seemingly fire "randomly" compared to traditional computing operations. Using traditional key management is infeasible in these high performance, complex environments.

In the mid-2020's, it's very likely that integrating serverless computing infrastructures will become a normal part of an organization's software engineering practices. This will likely yield major benefits in being able to quickly develop and maintain code for attacking distributed computing problems in multicloud architecture, and serverless will have a huge impact on how we think about virtualization and form factors like containers.

Serverless will also have a major impact on cryptography. While multicloud may complicate traditional key management, serverless takes that complication to the extreme. You at least *know* where you're spinning up a server in a multicloud architecture; with serverless I know little to nothing about the environment, timing, or location within which my computing will occur.

As a result, cryptographic systems for serverless computing simply cannot rely on a monolithic key management architecture. Some

measure of distributed and automated key management is necessary to manage cryptography the ephemeral serverless consoles that fire off like shooting stars to perform tasks seemingly at the random then disappear into the night.

Many of the open source cyrptography projects I mentioned in discussing multicloud can operate within a serverless model. However dealing with the ephemeral nature of serverless jobs is still a challenge, and will yield competitive consequences on new cryptographic systems' ability to operate large, vibrant serverless infrastructures.

With multicloud becoming the norm, secure and distributed large data sets will likely need to be maintained in order to provide performant but secure availability to multicloud resources. This is a unique opportunity for SMPC, particularly novel cryptographic solutions like blockchains.

Serverless focuses heavily on the ability to move computing tasks around a global multicloud infrastructure. But what about moving the data itself?

This is a much more complicated task, as while moving a serverless job from a node on AWS to AliCloud may require minimal bandwidth or space changes (assuming the use of a common serverless architecture), moving a hundred terrabyte or petabyte-scale dataset between two clouds may be technically or economically unfeasible.

There is an alternative to the approach of shuttling data between clouds to maximize performance: make the data available everywhere.

Globally available data presents a fairly obvious security challenge: how do I protect data access and availability of that data is literally everywhere. This is a problem, as most data online in the 2020's will either likely "touch" or physically contain sensitive data such as Personally-Identifiable Information (or PII) that are subject to strict government and regulatory compliance (GRC) regulations.

Because of the rise of major data breaches in the 2010's, may governments like the EU have chosen to implement strict regulations focused on protecting PII data with advanced encryption. Examples of these regulations include the EU's GDPR which restricts the sovreignty of data that does not comply with minimal cryptography and infosec standards, and the PRC's Draft Cryptography Law (国家商用密码管理办公室中华人民共和国密码法) which require the use of mandated encryption for protecting PII data.

GDPR and the Draft Cryptography Law are very serious regulations. Strict fines and possibly even jail time can be levied against companies and corporate officers who fail to comply with these standards. Both GDPR and the PRC's new crypto laws begin being enforced in mid-2018, and by the mid-2010s they will likely reach corporate GRC ubiquity like PCI-DSS and SOX.

In a multicloud world, globally-available data will require a system that allows the use of consensually-agreed upon cryptography and new secure computing standards—the likes of which we hithertoo never faced in computing.

Enter Secure Multiparty Computing.

Secure Multiparty Computing (or **SMPC**) is a field of computing and cryptography focused on preserving security and privacy of a data set. In its classic form, SMPC focuses solely on the preservation and isolation of different actors on the same data: a classic multi-tenancy model of cloud computing where multiple agents may draw data from the same public resovoir but do so in an isolated way so that their inputs to that computation are (in addition to being performant) private.

But SMPC has expanded in the late 2010's to also cover a different approach to this problem. Rather than just protecting individual nodes or tenants accessing data, what if we decide to work on sensitive data that needs to be encyrpted?

Basically: what if we just encrypt *everything*?

Uniform encryption throughout public data provides unique benefits to tenant isolation and allows computing to occur on large, sensitive data sets. But it also is a similarly unique challenge to traditional cryptography. Symmetric key systems like AES would likely not be able to handle the encrypt/decrypt workflow of constantly decrypting data to run analytical operations on them.

And if you think key management is complicated in a serverless world, key management in a world where every server doesn't trust each other (and can't agree on a common key management arbitrator) is nothing short of a nightmare.

SPMC gives us two potential answers to this problem, both of which are still early in their development today

The first is **homomorphic encryption,** a novel style of encryption where plaintext computation could be run on encrypted data without having to arduously decrypt and re-encrypt that data set.

For example, a homomorphic encryption system could expose a SSL-sheathed API to run data science tasks on globally-available PII data while ensuring that data remains encrypted. In this way, the tenant and the data source are both isolated from other data-accessors, and the data itself never leaves its protective encrypted state.

Homomorphic systems like Enigma exist today to address some of these tasks. But the field is in its commercial infancy, and general purpose homomorphic encryption is still outside the scope of most commercial applications at the time of this writing.

The alternative to purely homormorphic encryption is something much more familiar: Bitcoin-style blockchains.

Blockchain technology like Bitcoin already operate in a security model that complies with globally-available PII data: nodes don't trust each other in accessing that data and cryptography needs to certify and protect access to how those nodes drive consensus.

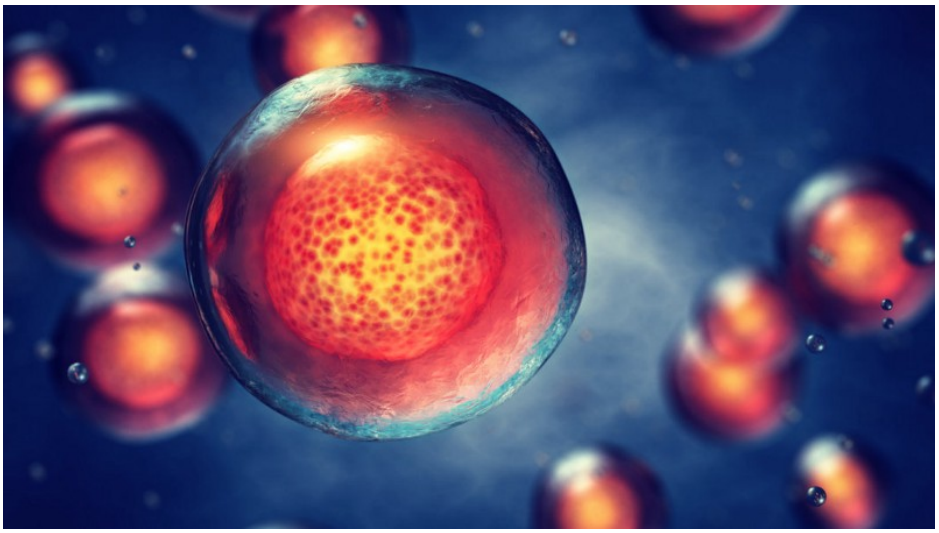
But while Bitcoin's "mining" step is focused on incentivizing the deployment of a cryptocurrency network, the same cryptographic technology that powers its Proof of Work algorithm could theoretically power a system to operate gated access to a cryptographically-secured "encrypted ledger."

This necessitates some fundamental changes to how most cryptocurrency blockchains are designed, as integrating security controls like secure logging and RBAC are usually outside the scope of the single-purpose mining and verification operations of blockchains like Bitcoin and even more fully-featured blockchainns like Etherium.

Homomorphic encryption and blockchain technology are by no means the only SMPC technologies in development. Neither are they mutually exclusive; Enigma uses blockchains to handle aspects of its underlying node-consensus protocol.

But they are fundamentally different approaches to cryptography than symmetric key or even traditional public key crypto. And this use of new cryptography architectures will become an even more salient topic as we begin to see the rise of completely new computing architectures at the end of the end of the next decade.

5–10+ Years: Disruption from New Computing Architectures



New computing architectures like quantum computers and biocomputers provide unique benefits and challenges that will disrupt how we think about encryption and information security as a whole

Since the beginning of modern Computer Science in World War 2, cryptography has been one of the primary usecases of performance computing.

For example, Alan Turing's Crypto Bombe was built to decrypt German Enigma messages and helped give the mathematic and technical foundations for the field. Advances in digital computing during the late 20th century were heavily funded by government agencies like the NSA and the CIA, largely out of a desire to create powerful cryptographic decryption and encryption engines during the Cold War.

Cryptography in the early 21st century is no exception. TPM chips that have allowed us to employ high performance cryptography without dedicated cryptographic hardware are largely the result of US DoD research into secure computing technology. Early research into public key cryptography used in blockchains like Bitcoin began as early as the 1970s, wherein organizations like the UK's GCHQ and the US NSA engaged number theory mathematicians to help design the first PKI systems.

This lock step of advances in computing being driven (or pulled) by cryptography will continue into the next decade and beyond. And while digital computing will likely persist at least for the next century, there are other alternatives to the electrical digital systems we use today that

are starting to leave the research phase now and likely will have some—if not very limited—commercial-scale application by the end of the next decade.

The most popular of these new computing architectures is **quantum computing**, or simply **QC**. Rather than relying on electrical gates to generate binary states, QC systems focus on the use of qubits that can hold multiple simultaneous states and enable novel approaches to computation that constitute a seismic shift in computing power not unlike our transition from Turing's mechanical computers to modern digital circuits.

QC heralds a number of major challenges and opportunities in cryptography that merit their own blog post. But most pressing among these are its invalidation of most modern PKI systems by making it much easier to search for the factors of large prime numbers. The consequences of this are dramatic given that common algorithms we use to protect online information like SSL rely on PKI mathematics.

While commercially available QC systems that can run algorithms like Shor's Algorithm to attack PKI may not be in scope for the next decade, they certainly are opportunities for first-world governments with advanced technology and resources.

As a result, long-standing cryptographic standards like the US' FIPS 140–2 are being currently revised to incorporate **Post-Quantum Cryptography** that can resist known QC-powered cryptanalysis via algorithms like Shor's. FIPS' post quantum cryptography regulations will enter the "draft" phase in the mid-2020's, and by the end of the decade their codification into the standard will likely herald the end of using traditional PKI such as RSA and ECDSA.

Less well-known but just as equally important to computing is the rise of **biocomputers**—computing systems that run on chemical processes within bio-engineered cells rather than in electrical circuits. Biocomputers provide a unique benefit in that, unlike QC systems, they could be relatively cheap to manufacture and much of the basic science around their manufacture and is already well known. They are also able to use digital computing mathematics and foundations, thus allowing

modern cryptographers an already robust toolkit to employ on their novel approach to computing.

Self-replicating biocomputers similarly provide unique challenges and opportunities for cryptography. Biocomputers are designed to work in tandem, distributing tasks to "divide and conquer" natively across their surface of cells and self-replicating during that process in order to scale "networking" and "storage" of compute tasks between cells. This makes them very valuable for brute force attacking certain types of cryptographic mathematics, and their continued development may warrant serious changes to the approved algorithms in regulations like FIPS.

Alan Turing once said that "we can only see a short distance ahead, but we know that there is great work to be done there." The very same could be said about cryptography at the end of the next decade. While the specifics of what will happen are left to time, it's clear that major advances in computing have ensured that the next ten years will see significant, disruptive change in how organizations (and even society) think about protecting data with encryption.

Note: this article also appeared on LinkedIn.