Qwyit™ - Hardware Use Cases

Introduction

There are several new markets for Qwyit™: IoT, AI, autonomous cars, mobile, computing, storage, health, entertainment, gov, etc. In every single one of these, there are no *native security capabilities.*

Native Hardware Encryption

Native defined: Embedded hardware security treating all data exactly the same – flow, registers, instruction sets; e.g., there is no processing difference between open data and secure data (authentication and encryption) – it is all handled *identically*. This means in the *base computing capability:* The CPU instruction set, ALU instructions, GPU, MPU (micro), etc. using the General Purpose Registers (GPRs). "Extensions" (such as Intel's SSE) where 'secure' data is handled differently than routine open data, while indeed on the chip, *are not native.*

There are two fatal problems without *native*:

1. Different handling introduces security vulnerabilities; attacks that can never be eliminated
2. Future networks will never have the required instant end-to-end security in their exponentially growing new connectivity; the networks will suffer today's *$Trillion cybercrime heritage*

The reason there are no *native security capabilities* is because every current authentication and encryption method is too complex to fit inside *native* processing, and too slow to be performed everywhere, end-to-end all the time.

> Obvious Solution: Embed a *native* authentication and encryption capability in all the chips; there would be no distinction between 'secure' and 'open': everything is processed the same. The required properties of this *native capability* are speed (operating within the latency of the data processing) and efficiency (able to fit and operate within the most minimal chip and system size/bandwidth/storage requirements).
>
>> *This is Qwyit™: A simple one-step authentication and encryption protocol that performs in one clock cycle and is <400 SLOC. These benchmarks are fast enough and small enough to meet* ***every data chip architecture****.* Qwyit™ security creates a unique one-time key that is mathematically one-way created by an authorized system authentication token, and performs encryption with that key in a single cycle – *this is identical to open data with no key*. *Performance of the system is identical either way.*
>
> The route to Qwyit™ proliferation is straightforward:
> 1. Analyze process – straightforward block/flow diagram of the data flow, end-to-end
> 2. Identify managing creation, transmission and storage interfaces/processors
> 3. Insert Qwyit™ as required into the identified chips and control instructions
>   a. The Application Control S/W (ACS) is then simply updated to use QwyitKey™, the Participant-Managed Authentication Key Distribution system, by inserting a *single* instruction: "Get Key", prior to sending any data

The Fatal Non-Native Problems

Side Channel Attacks

In computer security, a **side-channel attack** is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself (e.g. cryptanalysis and software bugs).

The knowledge gained [regardless of attack vector (EM, power, acoustic, differential fault, data remanence, etc.) *any and all*], is *by comparison* (more, less, the same as known output based on known input, etc.). The knowledge can be any meaningful distillation of brute force (generally a non-vulnerability) down to realtime loss (substantial vulnerability).

> Obvious solution: Secure everything and do it within the smallest measure of any working parts of the computer system, then there is nothing to compare.
>
> *This is Qwyit: every encryption is computationally identical (nothing but ALU Augend/Addend register storage/adds – and could be performed in the GPRs) and Qwyit performs within the smallest increment of computational use: 1 clock cycle. There are no channel attacks – side or otherwise. There are no other current security methods that are capable of this complete elimination of channel attacks – every other solution is a half-measure.*

Instant End-to-End Security

Without Native Security, all of the following – every single example Use Case – will never materialize properly, capably, safely. Without Qwyit™, they will all come to be in some fashion, infused with fraud, crime, catastrophe on a scale much grander than today's monetary cybercrime. These system examples demand instant, always-on, perfectly secret, everywhere security: Qwyit™

Use Cases:

- **IoT** – Here's an example of where we are today: Medical Devices That Are Vulnerable To Attacks; the conclusion for how to fix the problem? Become an expert device, protocol and data manager and then implement an expert-required security system that *won't be end-to-end protected.* (Not much of a solution, is it?!) This isn't a very good current status, and the future looks even worse: There are over 40 communication and data protocols. There will be 25+*Billion* devices in 2025. The security concern is the biggest challenge in IoT.

  Fortunately, it's not that hard to envision the solution: every device (no matter how low powered or computationally constrained) with embedded auth/encryption in the chipset; control hubs owning any defined 'network' and connectivity. End-to-end is assured, no performance degradation. There isn't any other way to 'improve' the current ill-suited, fatally flawed security complexity.

- **Artificial Intelligence (AI)** – There are several things about AI that has everybody worried, like job losses, taking over the world, etc. And there's plenty of reason for the fear, such as the way we train AI is fundamentally flawed. How do we get past all the fear and instill *trust* in these systems? We make sure to maintain ownership, control and management. These are provided by Authority and Authentication; and fundamental security of the systems is essential in maintaining the validity of those properties. Quantum Safe, perfectly secret (unbreakable, by *any* intelligence, human or artificial) are required, along with everywhere availability.

AI is the type of new system that will *demand* Dynamic Security – the ability to provide instant, on-the-fly new authentication and data encryption connections whenever/however the network demands. This isn't your grandmother's PKI.

- **Autonomous cars** – In 2015, two security researchers took over a Jeep and drove it around a parking lot. Nothing has been done since to eliminate this type of intrusion into the exponentially more complex autonomous car systems – and the severity of the consequences are astronomically higher: who will be responsible when an autonomous taxi full of teens leaving a Prom is driven off the road and violated?

  With several multi-access points (sensors, cameras, lidar/radar, AI self-driving chips, routine 'electronic' chips for fuel management, braking, etc.), following the current approach ('It's hard to hack these systems, so we're safe.') isn't just inadequate, it will become downright criminal. There's only one way to secure them all: everywhere/embedded native security.

- **Mobile** – WhatsApp is worth a $Bazillion – and there isn't a CDMA/GSM/etc. standard secure encrypted cellular phone. Plenty of specialized services, but no 'native' Apple/Samsung next-gen phone that is already 'security enabled'; they do this routinely ahead of service system upgrades (all the G's – 3, 4 ,5 and counting), but not security. The difference between VoIP and cellular is because of the ability to enable security at the application layer; cellular service encryption hasn't changed in years and it's all broken – and there's no phone-to-phone security.

  It's easy to see how the simple addition of native embedded security in the phone's communication chipset (the SOC, SIM, anywhere deemed applicable), in combination with the phone's contacts management operation, can easily deliver point-to-point communication security. The service type wouldn't be affected – and in just the same way that WhatsApp performs all kinds of smartphone media encryption, cellular service would provide the same.

- **Entertainment** – What does the intersection of a YouTuber, Social Media Influencer, Ex-Network Producer and Streaming Service New Show Director look like? We'll all know when they combine to determine their target market segments – and how to reach them, how to charge them, how to make money off them. If technology limits their imagination, we all lose. It is one thing to continually advance the viewing options and clarity of the presentation (QLED anyone?!), but what about directed service options for shared/single viewing, unique/varied show times, age-related controls – all requiring audit, authority, authentic and encrypted one-to-many, many-to-many, etc. and other unlimited distribution combinations, channels, capabilities, devices.

  Fairly easy to see what native/embedded/everywhere security here means: There are no technical limits to the imagination. Design a distribution model – whether particular to single content endeavors, or universal ones – and present them. There is a large current difference between movie and TV presentation budgets; unlimited technical distribution channels can change this type of production cost profile constraint. In the same way that 'book' media expansion and self-publishing has exponentially increased available written content, with everywhere security, all media can experience this same wonderful growth.

Conclusion

It's obvious how all these and other future use cases that we can somewhat imagine will be provided definitive security solutions if instant native is enabled in every piece of hardware.

But what's more important than what we *think the future will be*, is the assurance that instant will always be *instant*, and native will forever be *included* such that the new innovative wonders we haven't yet imagined won't have security issues: they'll have 100% security confidence. Qwyit™ delivers now and forever.