



# DELIVERING IoT TRUST

## WHITE PAPER

Security, Safety, and Privacy in the  
Internet of Things

WRITTEN BY:

**Jothy Rosenberg**

Founder & CEO

FEBRUARY 2018

**DOVER**  
MICROSYSTEMS

When people write about securing the Internet of Things (IoT), it's typically all "doom and gloom." They describe the bad things that have happened, are happening, and could happen. They tell an increasingly scary story, and we all need look no further than the daily news cycle to prove their claims. Fortunately, it doesn't have to stay this way — and that's the story I tell in this paper.

- Jothy Rosenberg



## A Different Kind of IoT World

Imagine an IoT world where cyberattacks are virtually non-existent, where things that could hurt people can't and don't, where personally identifiable information is sacrosanct, and where confidential data never gets into the wrong hands. Imagine a safer and more trustworthy connected world—the kind of world the Internet of Things was supposed to deliver.

### **Imagine an IoT world where we could use the word "trust." And mean it.**

"Trust" is a powerful word. The Oxford dictionary defines it as: a firm belief in the reliability, truth, ability, or strength of someone or something. Banks talk about trust often; sometimes they even include the word in their name, wisely tapping into the human sentiment that trust is the opposite of fear. Security people, on the other hand, rarely use the word "trust." They paint a different picture. Former Secretary of Defense, Leon Panetta, warned of a cyberattack that could be as destructive as 9/11.\* Such an event would evaporate all trust in IoT, and propel people to disconnect every one of their "things" from the internet. That would be an enormous loss for us all.

So, how do we create this alternative IoT world where people can **trust** the Internet of Things and thereby realize its full potential?

**The answer is by establishing and ensuring the Trust Triad: security, safety, and privacy.**

\*U.S. Department of Defense, News Transcript of Remarks by Secretary Panetta on Cybersecurity to the Business Executives for National Security, New York City, October 11, 2012: <http://archive.defense.gov/transcripts/transcript.aspx?transcriptid=5136>

# Understanding the Trust Triad

First, let's drill down on each of the trust triad principles to understand what they mean. Later, I'll explain how we can enforce and guarantee them.

## Security

In the world of technology, the word "security" is typically shorthand for a combination of computing security and communications security.

In the context of the Internet of Things, **computing security** is about creating an impenetrable barrier between the edge nodes (the "things") and the outside world (the network). It aims to block network-based attacks that can subvert a device's processor by exploiting software weaknesses ("bugs") in the application or operating system running on that processor. Threats include buffer overflow attacks, control-flow hijacking, and code injection; these three classes of attack combined represent over 90% of today's network-based attacks. Unless and until we stop writing the preponderance of IoT device software in unsafe languages like C and C++, we will continue to see about 15 bugs per thousand lines of code. Coupled with this reality is the fact that processors are not equipped to do anything about these bugs and the attacks that prey on them. Our computing devices have processors with an architecture that dates back to the 1940s—an era when network intrusions could not even be imagined, never mind prevented. Conventional processors blindly execute whatever instructions they are presented with, even if those instructions have been exploited.

**Communications security** is about keeping communications secure via encryption. Encryption is a process that scrambles data according to a sophisticated mathematical algorithm, and allows only authorized parties with a matching key to unscramble or decrypt it. If the encrypted bits are delivered to the wrong individual, data is still secure because it will present as indecipherable without the correct key. Modern encryption, like AES (Advanced Encryption Standard), is an effective tool for protecting sensitive communications, but only if data is encrypted correctly and encryption keys are secured. Managing encryption keys is the biggest challenge with communications security; it usually depends on special hardware called a Trusted Platform Module (TPM) to create and store the keys, and sometimes also execute the encryption/decryption code.

*Security is shorthand for a combination of computing security and communications security.*

## Safety

In the cyber-physical world of IoT, the consequences of a system failure or malfunction can be deadly. With systems used for autonomous driving, weapon systems, medical devices, and all sorts of critical infrastructure (nuclear power stations, chemical factories, dams, and much more), people draw a natural correlation between safety and fear.

Unlike with computing and communications security where threats may be less immediate and tangible, safety concerns are easy to describe as a set of rules. Medical devices, for example, can be tampered with or hacked so that they deviate from their intended duties. Devices such as pacemakers can cause serious danger to human life if they are compromised because of faulty code or insufficient safeguards.

Safety is the first cousin of security, and violations of safety policies can be caused by either a hostile external agent or by human error in the programming or configuration of the device or the applications controlling it. It will take just one high-profile safety violation that results in injury or death to instill enough fear in users that the perceived risks of the Internet of Things outweigh the perceived benefits.

*Gartner predicts the Internet of Things will have nearly 21 billion devices connected to the internet by 2020.*

## Privacy


IoT privacy is about keeping confidential personal, corporate, and military information from being exfiltrated—that is, intercepted or stolen by an unauthorized party. Stories of exfiltration have made frequent headlines in recent years. In 2013, Target's consumer financial and credit card data was siphoned out of Point-of-Sale devices at check-out stations, resulting in 110 million stolen credit cards, not to mention consequent headaches and costs for consumers and their banks. More recently, Equifax lost detailed credit data on 145.5 million U.S. consumers, enabling the perpetrators to steal identities using the same data that financial institutions use to establish someone's identity in the first place.

With the Internet of Things slated to have nearly 21 billion devices connected to the internet by 2020—and with many of these devices handling sensitive data—privacy is a critical piece of the IoT Trust Triad.

As mentioned earlier, if data is correctly encrypted, it is useless to someone who is not in possession of the necessary decryption key. Attackers, however, attempt to exfiltrate data by bypassing encryption routines. In the normal course of computing, there are three main steps in the encryption/decryption data flow:

1. Data is sent to a machine encrypted.
2. Data is decrypted and processed on the machine.
3. Data is re-encrypted and sent to another machine or to storage.

An attacker can exploit a software vulnerability to subvert step 2 and send decrypted data over the network, skipping step 3 entirely. Encryption alone is not enough to ensure privacy.



*“Trust takes years to build and seconds to break.”*

- Anonymous



## Enforcing the Trust Triad

As a wise person once said, “Trust takes years to build and seconds to break.” To prevent that break in IoT, we need to guarantee that the Trust Triad is unassailably enforced and protected. CoreGuard™ from Dover Microsystems provides this guarantee.

CoreGuard is part software and part hardware.

**Micropolicies** are the software that define security, safety, and privacy rules. Micropolicies maintain metadata—descriptive information about data—for every piece of data and every instruction that is handled by the host processor. It is the combination of micropolicy rules and their associated metadata that gives CoreGuard the knowledge it needs to distinguish good instructions from bad. Micropolicies are expressed as a set of rules to be enforced. These rules allow formal verification to ensure that policies do exactly what they claim to do and nothing more. (Formal verification is an important formulation that creates the equivalent of a mathematical proof.)

**Policy enforcer** hardware is implemented as part of a processor’s silicon architecture to check every instruction for compliance with micropolicies. When an instruction violates any micropolicy, the policy enforcer blocks it from executing.

Because CoreGuard is controlled by hardware, it cannot be changed by someone or something from across the internet—and this is why the term “unassailable” can be used. Because micropolicies are software, they can be written, customized, mixed, matched, and updated to give a system the exact protection it needs.

Let’s walk through examples of how CoreGuard and its micropolicies enforce each pillar of the trust triad.

## Security

CoreGuard provides computing security by ensuring that the processor in the IoT device executes an application only as intended, without any deviations precipitated over the network by malicious actors. A key insight here is that today's processors are not equipped with sufficient information to know if they are executing software as the programmer intended, or doing what an attacker wants them to do. As mentioned earlier, processors cannot discern good instructions from bad. Take a buffer overflow, for example—the most common type of memory violation. If an instruction writes more data to a buffer than the buffer is designed to hold and the program is not designed to handle it, then the excess data will overflow into the adjacent buffer. Attackers probe the network for applications with buffers that can be overwritten, and then exploit those vulnerabilities to replace data in the adjacent buffer with data that changes the control flow of the program.

*Today's processors are not equipped with sufficient information to discern good instructions from bad.*

To reliably stop buffer overflows, a processor needs to know the start and end addresses in memory for each buffer. CoreGuard collects this information in the form of metadata that assigns the same color to both the buffer in which data resides and the pointer to that buffer. A micropolicy rule then dictates that a STORE instruction cannot write data

to a buffer unless the color of the buffer matches the color of the pointer. This ensures that data stays strictly within the bounds of its intended buffer, even if the programmer forgot to verify this in the code.

As for communications security, CoreGuard leverages a system's TPM to encrypt data that is leaving the chip (going to disk or the network); this enforces privacy, which we look at more closely later. Also, by guaranteeing computing security, CoreGuard can ensure that encryption is not bypassed by attacks that attempt to take control of the flow of a program.

## Safety

When we think of safety, what hits closer to home than a device implanted in someone's chest? A pacemaker's job is to ensure the heart beats properly by regulating its rhythm with a small electric pulses. If the heart stops responding to these pulses and the heart starts beating abnormally, a built-in defibrillator automatically jolts the heart with 800 volts to reset the heart's rhythm. An 800-volt jolt to a functioning heart, however, results in almost certain death. Therefore it is critical that under no circumstances—whether through hostile means, configuration error, or software error—does the defibrillator activate when the heart is beating regularly.

CoreGuard can guarantee that the pacemaker application software executes two routines in a prescribed order only. The first routine is called `isBeatingReg()`, and it identifies whether the heart is beating regularly. The second routine is called `fireDefib()`, and it fires the 800-volt defibrillator. Micropolicy rules can dictate that before `fireDefib()` can activate, the defibrillator `isBeatingReg()` routine must have been called *and* returned "No," and that the result of the call to `isBeatingReg()` cannot be modified.

## Privacy

“Confidential” is the term typically used to refer to private data—whether it be personal, corporate, or military—that requires explicit permission to access. Knowing which data is confidential is the role of the application developer who can declare specific variables and memory locations as “confidential.” CoreGuard can then track those confidential distinctions with its policy metadata.

Confidential data must never be stored on disk or sent across a network as plaintext. Additionally, if confidential data is combined in any way with non-confidential data, then the resulting data must also be considered confidential. Suppose we consider someone’s age to be a confidential piece of data, and suppose a birthdate is included in the individual’s personnel file. Today’s date is, of course, public data and not considered confidential. So how do we prevent a combination of these two pieces of data from delivering an unencrypted result that can be used to determine the person’s age? CoreGuard uses a mechanism called taint. When the processor receives an instruction to subtract today’s date (public) from the person’s birthdate (confidential), CoreGuard enforces a micropolicy rule stating that if an instruction

performs computation of both confidential and non-confidential data, then the result (“age” in this case) is “tainted” and therefore also marked as “confidential.”

### Golden Rules of Privacy

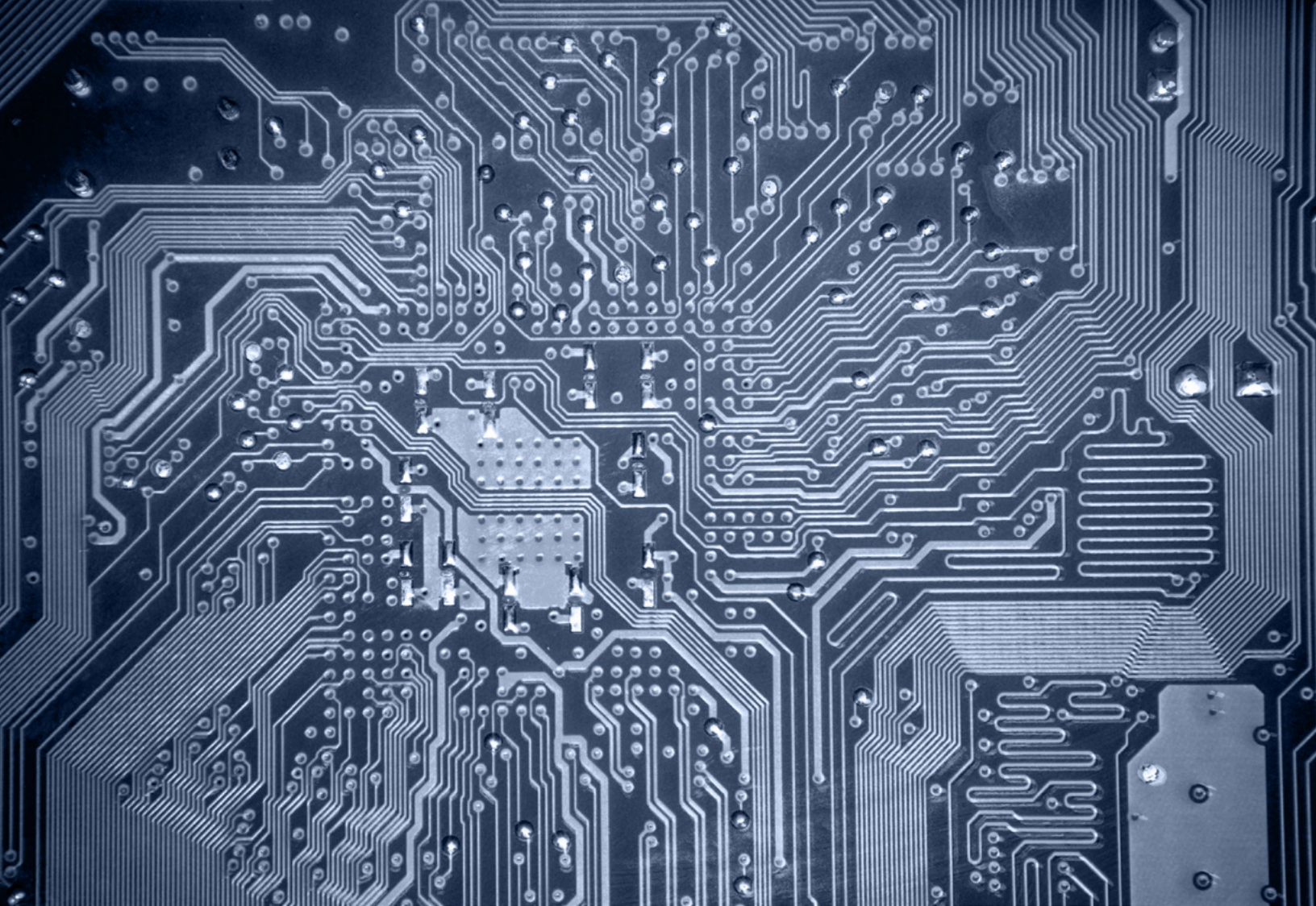
*Never let confidential data be stored on disk or leave the device unencrypted.*

*If non-confidential data combines with confidential data, the result is confidential data.*

CoreGuard also tracks locations in memory that correspond to writing data to the network (memory-mapped IO), and then prohibits writing tainted data to those locations. To prepare tainted data to be exported, CoreGuard uses the TPM’s encryption routine to remove the taint from the data. Only that trusted routine can remove taint, thus ensuring that only authorized individuals who possess the appropriate decryption key can see the confidential data.

## Empowering IoT to Deliver on Its Promises

Security, safety, and privacy are the three pillars that will support trust in the new world of IoT. We need to preserve and protect this triad at each and every node in the vast network of IoT devices. CoreGuard can do this with its flexible software micropolicies enforced by unassailable hardware. Nothing less can guarantee the safer and more trustworthy connected world that the Internet of Things can and should deliver.



**Learn More:** [info@dovermicrosystems.com](mailto:info@dovermicrosystems.com)

## About Dover Microsystems

Dover's lineage began in 2010 as the largest performer on the DARPA CRASH program. In 2015, Dover began incubation inside Draper before spinning out in 2017.

Based in Boston, Dover is the first company to bring real security, safety, and privacy enforcement to silicon. Dover's patented CoreGuard solution integrates with RISC processors to protect against cyberattacks, flawed software, and safety violations.



[www.dovermicrosystems.com](http://www.dovermicrosystems.com)