Methods

Mouser's technology & solutions ezine

Volume 1, Issue 2

Data Security

MOUSER ELECTRONICS

In This Issue

Executive Editor

Deborah S. Ray

Contributing Authors

Majeed Ahmad
Michael Camp
JPaul Carpenter
Barry Manz
Paul Pickering
Andrew Plato
Rudy Ramos
Raymond Yin

Technical Contributors

JPaul Carpenter
Paul Golata
Rudy Ramos

Production

IEEE GlobalSpec

With Special Thanks

Kevin Hess
Sr. Vice President, Marketing

Russell Rasor
Vice President, Supplier Marketing

Jack Johnston, Director
Marketing Communication

Raymond Yin, Director
Technical Content

Angelique O'Rourke
Media Manager

Welcome from the Editor

Having grown up in the 1980s, I vividly remember watching *War Games* at the theater and being enthralled by the tale of a high school kid accessing a military supercomputer programmed to predict possible outcomes of nuclear war. The movie was the first I'd ever heard of "backdoor passwords" (or computer passwords of any sort, for that matter) and was the first notion I ever had that computers were penetrable and the data inside, valuable.

The concept of data security dates back a few hundred years to early bookkeeping and record keeping, which is often considered the first form of corporate documentation. The recording of data required materials and resources that were scarce and time-consuming to develop. In some cases, scriveners copied documents word for word onto pages that were hand-stitched together and bound inside hand-carved covers made of cardboard, wood, or leather. Sometimes the covers were further adorned with additional inking, gold inlay, or carvings—which were visible indicators of the value of the data it protected.

The most valuable information would be stored under lock and key and guarded with other physical security measures as well. Even in pre-computing times, data security could mean protecting data from physical destruction, but it increasingly meant protecting against many nuances of theft, corruption, and exploitation, as well as ensuring access only by authorized personnel. In today's data security terms, these same aspects are described as a data security triad of protecting confidentiality, integrity, and availability.

A number of factors makes accomplishing the security triad a significant challenge: Embedded systems, especially those that are a part of the Internet of Things, are a particular challenge because of their many end-nodes, their interconnection with other systems, and their inclusion in mobile devices using wireless networks. Today, data security means securing devices that are on the move used by a myriad of people consuming data that's transferred along wireless networks.

"'Shall we play a game?' no longer refers to *War Games*…. It also describes today's hackers."

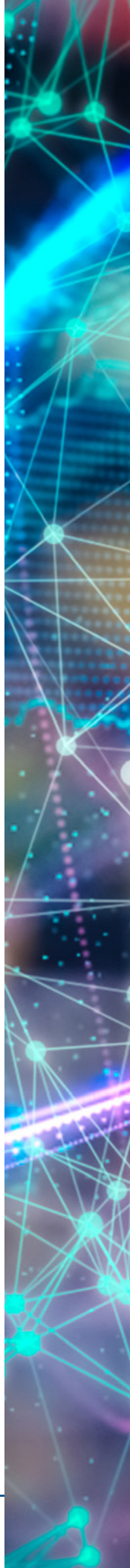
Perhaps the biggest challenge, though, is not in the technological advances that brought us to this juncture, but the cybercriminals themselves. "Shall we play a game?" no longer just refers to the WOPR computer's invitation in *War Games*—haltingly pronounced in digital monotone, of course—where the game of Global Thermal Nuclear War was both the genius and purpose of the computer's programming. It also describes today's hackers.

In September 2017, Mouser Electronics in partnership with the cybersecurity professionals at Anitian Corporation hosted the highly-successful webinar, "Data Security: Think Like a Hacker," which aimed to help embedded systems designers better understand how hackers discover, penetrate, and exploit security vulnerabilities and address complex security challenges. This issue of *Methods* aims to complement the webinar and provide additional insights into securing embedded systems at the electronics hardware level.

Securing data is perhaps the most critical computing imperative of our time. Join us in this quest to be part of the data security solution. ●

Deborah S. Ray

Executive Editor,
Mouser Electronics



Foreword

"Just because you're paranoid doesn't mean they aren't after you."

Joseph Heller, *Catch-22*

In the late 19th century, Dutch linguist and cryptologist Auguste Kerckhoffs noted that secure communications systems should not require secrecy in their design. Writing in *le Journal des Sciences Militaires* (Journal of Military Science), Kerckhoffs suggested that these systems should be able to fall into enemy hands without compromising communications as long as the encryption keys remain secret—a concept known today as Kerckhoffs' principle.

Years later, this general idea was posed more succinctly as "the enemy knows the system" by Claude Shannon, who is considered the father of information theory. Today, system developers face a more extreme version of these warnings: The enemy not only knows the system, but probably also knows its security vulnerabilities better than the developers themselves.

A little paranoia is a healthy thing when it comes to system security, particularly with the Internet of Things (IoT). IoT applications require widespread distribution of their associated devices, which completely violates classic security tenets that call for compartmentalization and physical isolation. Notwithstanding Shannon's maxim and Kerckhoffs' principle, conventional security policies do rely on secrecy with the reasonable expectation that it's always better when fewer people know the system and when less is known about it. The IoT, of course, does not work that way. Gaining physical possession of IoT devices is not only a simple matter, but it is also usually expected, even required, to serve the functions of the overall IoT application. Once in possession of the physical system, sophisticated hackers can take their time prying secrets from unprotected devices.

In a severe test of Kerckhoffs' principle, IoT security relies not on hiding the design of security algorithms, but on the secrecy of the keys that underlie basic security mechanisms such as authentication and encryption. Here's where security ICs (integrated circuits) play a role. By storing keys and certificates in protected memory, these devices protect the secrets while in storage. Still, secure storage is only one element in the classic data-security foundation that calls for protecting data at rest, in transit, and in use. More advanced security ICs meet all three by combining secure storage with on-chip encryption accelerators. This approach keeps secret data from ever leaving the security chip where it could be vulnerable in transit on exposed buses between different chips or in use during algorithm execution on a less secure IC. As a result, these devices can support the root of trust required to protect IoT networks from unauthorized intrusion and protect IoT applications from corrupted data streams.

Of course, logistics and reality spoil this pristine vision. Security requires more than maintaining security on the chip. It requires tight control in loading the secret data onto the chip in the first place—a process that can present more than just technical concerns. The same vulnerabilities in social engineering, human behavior, and simple accidents that have exposed secrets in other domains await similar missteps in provisioning the secret keys and certificates onto security ICs. As a result, protecting this secret data requires parallel efforts in the supply chain to ensure secure key generation and provisioning. After device deployment, the ability to manage certificates, keys, and security upgrades will rise in importance as hackers double their efforts with more subtle attack methods.

Mouser Electronics considers the need for IoT security an urgent Call to Action for manufacturers, distributors, engineers, and users, as it's time to not only address the many vulnerabilities, but also get ahead of them. Ultimately, IoT security requires a little bit of ongoing paranoia—and even an ability to thrive in an environment where vulnerability is the norm. Indeed, it's less a matter of paranoia than simply the practical recognition that different kinds of threats can come at any time from any source. By combining awareness with robust design built around security ICs, however, developers can demonstrate that their IoT applications are just too much trouble to hack. ●

Raymond Yin

Director of Technical Content
Mouser Electronics

Stats:
2017 Data Breaches

By Rudy Ramos, Mouser Electronics

A 2017 report from Lloyd's of London stated that damages resulting from a major cyber-attack could lead to losses as high as 121 billion dollars—an amount that's comparable to the economic damage caused by Hurricane Katrina in 2005. The report went on to state that a security attack that causes a widely-used cloud service provider to fail could cost upwards of $50 billion.

The reality is that in the information age we live in, no one is entirely immune to cybercrime including big business. Today's world is more interconnected than ever, and while most see its advantages, the increased connectivity also brings an increased risk of theft, fraud, and unauthorized access to sensitive data. The proliferation of wireless protocols and emerging technologies like the Internet of Things (IoT), Industrial Internet of Things (IIoT), connected cars, and smart devices and wearables all exacerbate the risk.

Recent Security Breaches

According to the Identity Force website, 2016 saw a 40 percent increase in reported data breaches from the previous year. At the same time, Yahoo! also announced the largest data breach in history in 2016, affecting more than one billion accounts.

So far in 2017, there have been over thirty "major" data security breaches. I emphasize major because these security breaches represent only the worst ones and do not account for the countless others that happen on a daily basis that go unreported. Every business sector has been affected by these security breaches. Even companies that tout being "the best cybersecurity consultants in the world" have made the hack list this year. The irony is that easy-to-implement steps—such as implementing two-factor

authentication—were not taken, which allowed hackers to access a company's data using a single password.

It's almost like companies have become complacent or desensitized to the constant onslaught of data security threats…or perhaps have become complacent in that they've not yet been breached.

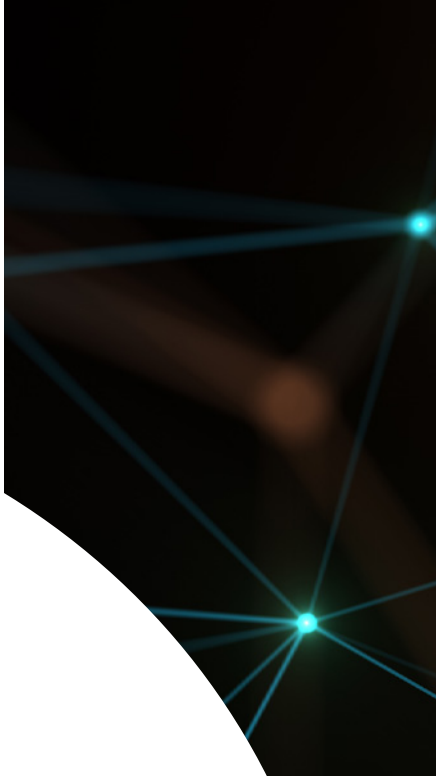
Potential Sources and Points of Entry for Security Threats

- Being a Small/Mid-sized Business
- Bots & Synthetic Bots
- Cloud Services
- Corporate Data on Personal Devices
- DDoS Attacks
- Hardware
- Improper System Configuration
- Inadequate Security Technology
- Increasingly Compromised Web
- Insider Threats
- Lack of Encryption
- Machine Learning-Enabled Attacks
- Malware & Mobile Malware
- Outdated Security Software
- Shortage of Security Experts
- Social Media
- Sophisticated Ransomware
- Third-party Entry

Who's Been Compromised?

As of the time of this writing, Identity Force reports these major 2017 security breaches:

- America's JobLink
- Arby's
- Blue Cross Blue Shield / Anthem
- Bronx Lebanon Hospital Center
- Brooks Brothers
- California Association of Realtors
- Chipotle
- Deep Root Analytics
- Deloitte
- Disqus
- DocuSign
- Dun & Bradstreet
- E-Sports (ESEA)
- Experian
- FAFSA: IRS Data Retrieval Tool
- Gmail
- Hyatt Hotels
- InterContinental Hotels Group
- Kmart
- OneLogin
- Online Spambot
- River City Media
- Sabre Hospitality Solutions
- Saks Fifth Avenue
- Sonic
- SVR Tracking
- TalentPen
- TigerSwan
- U.S. Securities and Exchange Commission (SEC)
- UNC Health
- University of Oklahoma
- Verizon
- Verifone
- Washington State University
- Whole Foods Market
- Xbox 360 ISO and PSP ISO
- Yahoo!



Communicating Security Risk to Executive Leadership

By Andrew Plato, Anitian

"I don't get it!" said the CEO as he dropped the 300-page report on the conference table. Something was very wrong.

It was 2010 and my team had just completed a large, enterprise risk assessment for a financial services company. We followed a traditional assessment methodology and delivered a robust report filled with worksheets, diagrams, charts, graphs, and detailed explanations of risk…none of which made a bit of sense to the executive leadership. The CEO threw the report down on the table and dismissed all our work.

Risk management is a complex and highly nuanced aspect of information security. It is also largely inaccessible to executives who are not immersed in the language, philosophy, and theories of risk. Risk management is also the cornerstone of any good security program. Executive leadership needs to understand the risk the business faces if they are ever going to make informed decisions.

We need a new way to communicate risk to executive leadership. Fortunately, there are steps you can take to improve your risk management communication. However, to understand these steps, we must understand the problem with current risk management techniques.

The Challenges of Communicating Risk

After the aforementioned incident, we reflected on the whole risk assessment process. We had numerous conversations with industry peers and clients. We cataloged our notes and identified a number of trends. Business leaders were frustrated with risk management. Common complaints we heard, included:

- "Why does it take so long?"
- "I thought we had security controls in place to take care of this stuff?"
- "How do we fix these problems?"
- "What do these risk numbers mean? Are we in danger or not?"
- "This is just busywork to keep the regulators happy."

When we analyze how risk assessments are conducted, we identified the following challenges with current risk management techniques.

Challenge 1: Assigning Value is Difficult

In a 2012 article in NetworkWorld, Richard Stiennon describes some of the problems with assigning value to IT assets. Stiennon argues that IT assets have extremely volatile value. Moreover, how do you assign value to something like a single email? The value of

any one email can dramatically fluctuate over time and based on its content.

Stiennon goes on to describe how risk assessment efforts often devolve into "protect everything" efforts, which is equally impossible. There are ample examples of organizations that had massive budgets for security controls and risk management, but failed to actually prevent attacks. This suggests that such measures are ultimately ineffective since they do not adequately manage risk.

Modern IT environments are incredibly complex and volatile. When you consider all the possible IT assets, such as mobile devices, apps, data, and networks, the list of assets is huge, even for a small company. To compound this problem is the impossibility of assembling the people in an organization who are able to properly value IT assets, particularly when such activities would be viewed as wasting time. The ultimate problem is that traditional risk assessment methodologies are extremely dependent upon reliable valuations. Therefore, we need a new approach to value, that refocuses on threats over asset valuation.

Challenge 2: Risk and Security Language is Incomprehensible to Leadership

Language not only affects comprehension, but acceptance. When people do not understand the language of risk assessments, they are not likely to accept the conclusions. Consider this definition for Defined Evaluation Activities from the OCTAVE risk methodology:

Implementing defined evaluation activities helps to institutionalize the evaluation process in the organization, ensuring some level of consistency in the application of the process. It also provides a basis upon which the activities can be tailored to fit the needs of a particular business line or group.

While the concept of defined evaluation activities may be relevant to a risk assessor, this is a foreign language to executives. The challenge here is that most people (especially executives) are not immersed in the vocabulary of risk management. The terminology of risk is foreign and misunderstood. Most leaders do not understand, for example, that risk is not a problem, but a measurement of the potential of a threat to cause harm. As such, we need simplified language if we want executives to both understand and accept risk assessment information.

Challenge 3: Numbers Can Deceive

IT security risk assessments are comprised of complex and often disparate data types. Vulnerabilities

can range from trivial to wildly complex. Impact rankings are dynamic and highly variable. Probability values are, at best, guess work since there are limited statistics on breaches and attacks. Moreover, as mentioned earlier, assessing the value of an IT asset is nearly impossible.

IT security risk assessments are ultimately subjective efforts. That is to say, most of the data points that go into a risk assessment are the result of an educated guess from an assessor (or team of assessors). Probability is a good example of this problem. What is the probability of a single server getting hacked? The factors that comprise that are, to say the least, complex. While there are published statistics, those statistics make a lot of assumptions. Published statistics are also based on reported incidents, which comprise a minuscule fraction of the total number of attacks.

At best, any numerical value assigned to that probability is an informed guess. That guess is also dependent upon the assessors' skill and experience in information security. According to Bruce Schneier, less experienced assessors are likely to overestimate sensational threats and underestimate the seriousness of less interesting threats.

Many risk assessors are fond of using complex equations and metrics in their risk management reports. This gives the assessment the illusion of accuracy and attempts to disguise the subjective nature of the assessment. In fact, risk assessment numbers can be very misleading. If an IT asset has a risk ranking of 61, what does that mean? Is that good, or bad? Also, if that number was derived from other numbers, which at best were guesses, then the final risk ranking is also a guess.

Numbers skew how people view risk. It makes risk too impersonal and unnecessarily sterile. Risk is a human issue and people relate to risk in a human way. As such, we need a method of communicating risk that does not rely on meaningless numbers and equations. We need to use simplified, plain language to describe risk if we want executives to understand it.

Challenge 4: Risk Data Gets Stale Quickly

The threat landscape of information technology is volatile. The vulnerabilities, tactics, tools, and motivations of attackers are constantly changing and evolving. Couple this with the rapid pace of change in IT, and you have a target that is not just moving, but moving very quickly.

Unfortunately, current risk assessment practices are extremely time consuming. NIST and OCTAVE estimate two to three months of full-time work for an assessor to complete a comprehensive risk assessment. The time participants must commit to these assessments is onerous as well. Anecdotally, we know of companies that need 10 to 12 months to complete their organizational risk assessment. The complex worksheets and data matrices consume immense amounts of time. Moreover, risk assessment processes can easily devolve into a melee of competing opinions and statements. The process lacks focus and therefore consumes more time.

Any risk data that is over 90 days old is stale. This adds another layer of complexity to communicating risk to leadership. Risk assessors must be able to complete assessments quickly so the intelligence from those risk assessments is timely.

Talk Like an Executive

Communicating risk to executives necessitates an understanding of how leaders view risk as well as how they consume information. While there are many different types of leaders, there are some basic steps you can take to improve your odds of communicating the complexities of risk to leadership.

Use Emotional Words Sparingly

Risk is an emotional thing for everybody. We all rely on our instincts to evaluate risk and determine our tolerances. Unfortunately, this instinct is

easily skewed when people do not understand the landscape of the risk. When people do not understand risk, fear takes over and decisions become quick and irrational. Furthermore, emotionalism can lead to an overemphasis of improbable sensational threats, while ignoring more serious (and likely) boring ones.

As such, risk assessments must downplay sensationalism without entirely discounting the inherently instinctual nature of risk evaluation. One strategy for reducing sensationalism is to avoid aggressive, fear words like “war,” “terror,” and “catastrophe,” while embracing most positive, secure words such as “safe,” “protect,” and “enable.”

Unfortunately, the IT security industry is very fond of promoting and exaggerating sensationalist threats. Recent stories about hacking cars and medical device implants are good examples of threats that are interesting and have terrifying consequences, but not very probable in the real world. To avoid sensationalism, focus exclusively on probable threats, rather than merely possible ones. Probable threats are those that have a reasonable chance of happening to the organization and causing significant damage.

Deliver Intelligence, Not Data

Risk assessments are ultimately subjective assessments that are full of ambiguity. While most leaders are comfortable with “shades of gray” they need to be able to see there is a way through all the grayness to something better.

Where many risk management efforts fail is when they try to present data, rather than intelligence. Intelligence is data that has been rendered down into insight and action. Leaders want (and need) intelligence; they do not want binders full of data. Data gets ignored; intelligence gets attention.

The way to do this is quite simple. Explain what the data says, not what it is. Executives look to security professionals who can tell them what all the data means. Interpret the data and give a definitive assessment of what it means. The point of collecting data is to support intelligence, not replace it.

Communicate in the Now

How you express risk, is just as important as what you communicate. Consider these phrases:

- We will have to implement security controls if we want to protect data.
- We should have implemented controls to protect our customer data.
- Data compromise is a serious threat to our

business. We must implement security controls to reduce this threat.

Those sentences are respectively, future, past, and present tense. Notice how they read? The first one sounds like optional and a little like a threat. The second sounds like a compliant. The final example is in present tense. It puts a threat as the subject initially. Moreover, the implementation of controls is a response to the threat. The control is not the subject.

Present tense is more definitive. It does not have the attitude and weakness of past and future tense. Those tenses should be reserved for times when you genuinely need to express something as in the past or future. Otherwise, all risk should be stated as if it is a risk right this very moment.

Present tense also can focus risk conversations correctly. Security practitioners are fond of making new security controls and technologies the focus of their risk communication, with threats being the reason for implementing a new control. Notice that in the first two examples above, controls are the subject and the threat is the object.

This is the exact opposite of how executives view risk. For executives, the threat is the focus of risk and controls are a way to reduce the threat. Controls are dependent upon threats, not the other way around. Forcing yourself to use present tense will also force you to reorganize your communication to place threat and risk at the center of the discussion, and controls as a dependence.

How to Improve the Risk Conversation

To improve the risk conversation, we must begin with the basics and build a new approach to risk assessment. In this section, we will explore six tangible steps you can take with executives to improve your risk assessment efforts and make risk conversations more effective.

Step 1: Agree on Six Words

Communicating risk to leadership begins with perhaps the most troublesome challenge: The word "risk." Ask a room of 20 people to define the word risk and you are likely to get 20 different definitions. Many people conflate risk with threat or vulnerability. This leads to a misunderstanding of the risk assessment process and what risk really is.

Therefore, the first step to communicating risk to executive leadership is to ensure everybody can agree upon the meaning of six words:

- **Threat**: Something bad that could happen.
- **Vulnerability**: A weakness that could let a threat happen.
- **Control**: A protection that helps fix vulnerabilities and stop threats from happening.
- **Impact**: How bad it will be if a threat happens.
- **Probability**: How likely is a threat to happen?
- **Risk**: An assessment of a threat based upon the vulnerabilities, controls, impact, and probability that are associated to it.

Once everybody agrees to these simple definitions, it becomes significantly less difficult to communicate risk. It is important to keep this list to these six words. If you add any more words, it will make people lose interest. Also, notice that the definitions above are simple. This ensures maximum comprehension among a diverse audience of people.

It is vital that risk assessors understand that most people do not care about the nuance of risk management. Complex language and constructs make risk management more confusing and inaccessible. To communicate to leadership effectively, use simple, plain language.

Step 2: Establish a Lens

If you want to communicate something complex, it must be broken down into pieces an audience can understand. Risk is a big, complex issue that is difficult to understand even for skilled security practitioners. Risk data demands structures that organize

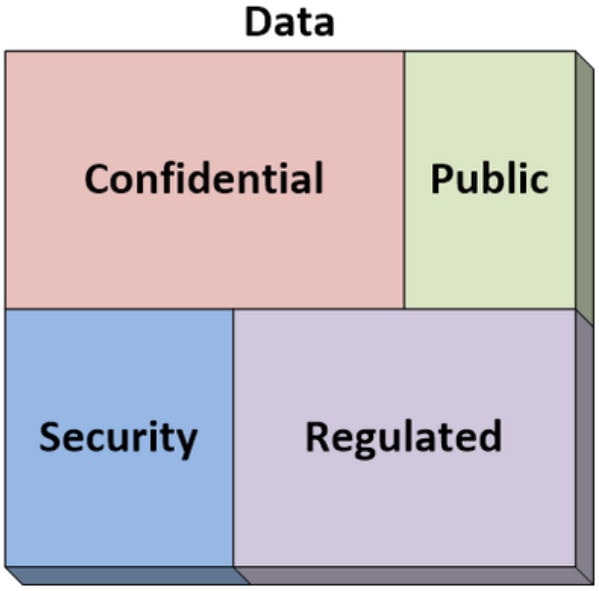


Figure 1: A lens is way to breakdown a larger whole into manageable chunks. (Source: Anitian)

and categorize data for easier comprehension.

A *lens* is a method of categorizing threat data to make it more comprehensible **(Figure 1)**. A lens can be any attributes that define an environment, like data, system, or application types. The most common lens we use is data type, which are the various types of data present in an environment. For most companies, there are only a few types, such as confidential data, regulated data, security data, public data, and so forth.

A lens, such as data type, not only helps organize threats, it also aids the process of analyzing threats. A lens forces the definition and explanation of a threat into the context of its lens. For example, the threat of malware infection is no longer some vague possibility; it is a malware infection that could threaten confidential data, or regulated data. In the context of a data type lens, the threat of malware becomes easier to evaluate and discuss.

A lens also provides a more efficient way to present risk intelligence to leadership. If leadership is concerned about attacks against confidential data, risk intelligence can be quickly organized to show the threats that are relevant to this data type.

The process of creating and enforcing a lens has a very useful impact on the risk analysis process. It is also difficult to do. Risk assessors must constantly reinforce the lens and continue to frame discussions into the lens. This requires discipline, but the benefits to communicating risk are immeasurable.

Step 3: Express Security Issues in Terms of Threat

Executive leadership always wants to know what could go wrong. The whole point of risk assessment is to deliver risk intelligence. Security practitioners often express security as a problem. Moreover, phrases like “best practices” or “regulatory requirement” can feel like an imperative to an executive and may elicit an automatic rejection.

On the other hand, when security is framed as a response to a threat, it becomes about the organization improving and protecting itself. This is language that is more comforting to executives, who now can understand the security controls in context of what they can do. Consider these two examples:

Undesirable: *Our network is insecure. We do not have strong authentication or intrusion detection systems leaving us vulnerable to attacks. Best practices state we must implement these technologies. We are also required to implement them to meet regulatory requirements.*

Preferred: *An attack and theft of our confidential data is a very*

tangible threat to the business. Stronger authentication and intrusion detection technologies would reduce the risk of these threats, as well as meeting important regulatory requirements.

Framing the issue in the context of a defined threat to the business makes the implementation of security controls a dependency on a specific threat. In the undesirable example, it begins with a problem and makes regulations and best practices a dependency of the problem. This is an also an example of "flipping the object," and extremely valuable communication tactic. That is, turn the object of a conversation into the subject. In the preferred example, the subject is a threat and the object is a solution. This is a much more natural expression of risk.

Step 4: Get Data, Put it in the Backseat

Executives need to know that when they are looking at charts and graphs, there is real, valid data behind that analysis. As previously discussed, risk evaluations are inherently subjective. Using real security data goes gives weight to those subjective evaluations. Vulnerability and penetration testing data are ideal for risk assessments. They can provide a snapshot of the technical security of an organization. Configuration analysis is another valuable data point. Specifically, reviews of firewalls, routers, switches, and system hardening efforts. A review of these technical controls can provide some extremely valuable insight into the overall security of the organization.

Technical data such as this is not perfect. Vulnerability data can be skewed and configuration reviews biased. To avoid these biases, engage a third-party assessor to conduct technical reviews. Penetration testing, should always come from an external, unbiased source. However, risk assessors must possess the skills to interpret this data, or engage people who can interpret the data.

Technical data on its own does not tell the full story, nor is it what executives want. What it provides is a foundation for the intelligence that executives do want. The best technical evidence is therefore in the background, providing support and confidence.

Step 5: Simplify Impact and Probability

Impact and probability are key components to risk. They also are difficult for non-security people (such as executives) to fully understand. Both of these measurements demand both a scale and context to make them more meaningful.

Scale is especially important to probability assessments. Without a timeframe, virtually anything is probable. Therefore, all probability must be bounded with a specific time period. Ideally, this should be no more than 12 to 24 months. This gives probability a frame of reference that non-secure people can understand. It also helps aid in the analysis process, as the assessor can evaluate a threat in a controlled timeframe.

Impact evaluations have a similar problem. Impact is a compound assessment with a variable range. In other words, it is very complex and can quickly confuse executives. First, there are multiple forms of impact: Financial, operational, and reputational are the most common. Second, what constitutes a "high" or "low" impact depends on what is being analyzed.

For example, consider the threat of regulatory non-compliance if systems are not configured correctly. The impact in this case is compounded. There is a financial impact; fines could be levied. There is an operational impact; systems could have to be taken offline to be fixed. There is also a reputational impact; other organizations may not want to do business with a non-compliant entity (which also exacerbates the financial impact). Not all of these impact values are the same.

As you can see, impact can quickly become a fantastically complex evaluation, which will just confuse executive leaders. Therefore, it is best to simplify both probability and impact into overall rankings. When assessing impact, it is okay to consider all the possible types of impact, but these need to be condensed into a single impact statement. Probability works the same way. There may be many levels and complexities to probability, but these also demand simplification.

What does need to be explained is what constitutes "high" or "low" impact and probability. **Table 1** shows a good example of how to do this. **Table 2** extends the matrix with an impact statement.

These two charts greatly simplify what probability and impact mean

to the organization. This is a scale that executive leadership can quickly understand and digest. These charts lack some of the detail and precision of traditional risk assessment, but they more than compensate for that in their ability to communicate the nature of risk.

Step 6: Embrace Simplicity and Brevity in Reporting

Lastly, good risk intelligence needs to be condensed and simplified. Ideally, risk intelligence should be a simplified as much as possible without losing too much resolution. Consider the example in **Table 3** of a malware infection threat.

This is another example of why condensation, simplification, and brevity all work in favor of communicating risk. The more complex risk intelligence is, the more likely it will be ignored. People simply lack the ability to comprehend the vast nuance and complexity of risk. Executives, who are not immersed in the daily details of IT security, are not going to read hundreds of pages of risk analysis and worksheets. What they need is risk reduced down to the basic, core components.

The Malware Infection threat is very clearly cross-referenced against the vulnerabilities in the organization. However, these vulnerabilities are extremely simplified. They are a distillation of what the assessor discovered about the organization. Moreover, this chart eliminates a description of the controls in place for the benefit of brevity. Instead, the chart presents recommended remediation steps. These recommendations are written in actionable, present tense language.

Table 1: A Probability Matrix example.

Metric	Description
Certain	<95% likelihood of occurrence within the next 12 months
High	50–95% likelihood of occurrence within the next 12 months
Medium	20–49% likelihood of occurrence within the next 12 months
Low	1–20% likelihood of occurrence within the next 12 months
Negligible	>1% likelihood of occurrence within the next 12 months

Table 2: Impact example.

Metric	Description
Critical	Catastrophic effect on the Data Asset.
High	Serious impact on the Data Asset's functionality .
Medium	Threat may cause some intermittent impact on the Data Asset, but would not lead to extended problems.
Low	Impact on the Data Asset is small and limited. Would not cause any disruption in core functions.
Negligible	Data Asset remains functional for the business with no noticeable slowness or downtime.

Table 3: Malware Infection threat example.

Threat	Vulnerabilities	Recommendation	Impact	Probability	Risk
Malware Infection	• Outdated anti-virus • Lack of anti-virus on 36% of servers • 32 high ranked vulnerabilities on in-scope systems • Lack of virus scanning at the network layer	• Endpoint antivirus must be installed on all hosts. • All endpoint antivirus must be updated daily. • All systems must have new patches applied within 30 days of release. • Company must deploy a more robust patch management platform. • Implement a core firewall that can perform virus scanning at the network layer	H	C	H

Another strategy is to condense risk data into an intelligence report. Consider the example in **Table 4**.

In this summary, risk is categorized into five focus areas. Each of these categories is assigned an overall risk rating, based on the summation of threats that comprise that risk category. A description then summarizes the risk. Notice the description does not have all the answers, this would be best left for an Action Plan. However, it also does not only focus on problems. It points out areas where there are good controls.

This type of summary is a good way to open a conversation about risk with executive leadership. It is accessible, written in business language, and definitive.

Conclusion

The key to making risk communication work is simplification. Risk is a very complex concept. It is difficult for anybody to understand, let alone executives. The emotional nature of risk can also cloud judgment, which can lead to bad decisions.

Simplicity and brevity cut right to the issue. The shorter and more succinct risk intelligence is, the more likely executives will not only understand it, but accept it and do something about it. ●

Table 4: Business Risk Intelligence summary.

Issues	Severity	Description
Regulatory Risk	High	Company faces extensive HIPAA regulatory risk due to significant non-compliance, both in technical information security and privacy matters, and in general business process requirements.
Legal Risk	Medium	The global security risks throughout the IT infrastructure expose the Company to potential risk of lawsuits from patients and their employees if PHI is stolen or corrupted.
Reputational Risk	High	Insufficient controls protecting ePHI exposes Company to a high degree of Reputational Risk. Enforcement actions resulting from a failing OCR HIPAA assessment also have a high potential for negative reputational impact.
Financial Risk	Medium	The Company's IT environment is not aligned with most security best practices, increasing the likelihood of a security breach. This includes the potential for fines due to regulatory compliance violations and lawsuits from data owners (patients).
Operational Risk	Low	The Company is at some risk from technical issues, such as the uncertainty of whether an Internet outage would cause significant interruption of business. However, there is good redundancy in the environment.

Mforum

Mouser asked suppliers invested in data security to answer the following question:

Everybody's definition of "secure products" and "security built-in" are different. What does "security built-in" mean for your company's hardware products?

Here's what Maxim Integrated, Microsemi, NXP Semiconductors, and Renesas Electronics had to say:

Maxim Integrated

Gregory Guez, Executive Director, Micros/Security/Software Business Unit
Scott Jones, Managing Director, Micros/Security/Software Business Unit

Maxim has been active in the embedded security market for over 25 years, initially building products to protect one of the most critical and challenging security applications—credit card payment systems. That same expertise has subsequently been applied to products serving many other markets. For example, Maxim secure ICs ensure medical consumables are genuine so that patient safety is not compromised. They provide assurance that battery packs are charged properly and won't explode, protecting consumers from harm. They validate sensor data so that control systems receive trusted information and can make correct decisions.

For Maxim, "security built-in" means that we provide products that allow our customers to develop an immutable and reliable "root of trust." Software can be modified, while the hardware ROM in our chips cannot. The most secure solutions are based on an unalterable ROM that can be unequivocally trusted. Once that root of trust is established at the base level using Maxim secure ICs, additional firmware, software, or other applications can be added. That level of trust can be extended upward because each layer inherits a trust level from the layer below. Taking shortcuts that only consider security at the firmware or application layer provides numerous ways for hackers to attack such a system and subvert intended operation by modifying the underlying hardware, firmware, or software.

Security is a complex topic, and threats are constantly changing and escalating. Companies are often overwhelmed with the array of choices and the challenges of implementing a truly secure system, frequently giving up and doing nothing or implementing something that is trivial to defeat. Although Maxim's security products are industry-leading and provide the highest possible level of protection against all types of threats, they are also designed to be easy to implement, without requiring our customers to have deep levels of security expertise.

Our customers can rely on us to be their trusted advisor on security. Our broad portfolio of secure ICs ensures that the right level of security is applied for a given application, whether retrofitting existing products or addressing new developments. And the ever-escalating levels of attacks are met with Maxim's constantly increasing protection levels, like our new secure authentication products that implement a physically unclonable function (PUF). The new PUF devices continue our 25-year history of ensuring security is built-in at the base level, providing that critical root of trust that every secure system must have.

Microsemi

Ted Marena, Director
FPGA Marketing

Microsemi's FPGA products group considers security a non-negotiable feature. For us, built-in security means we address three security areas: Design (IP) security, data security, and secure hardware. Our FPGA products include class-leading features for all three of these:

Design Security

By design security we are talking about making sure the IP and keys that are stored in our devices are not only encrypted but also protected from differential power analysis (DPA). You can see numerous YouTube videos of hackers with a $500 electromagnetic probe extracting the key from devices. The probe needs only to be near a device without DPA countermeasures, and the key and your IP will become known. Microsemi's low density (IGLOO2 and SmartFusion2) and mid-range density FPGAs (PolarFire) have built-in DPA countermeasures. If you use a device that does not have DPA countermeasures, your design is susceptible to DPA attacks.

Data Security

Is the data coming into and out of your devices and boards secure? This is data security. Microsemi offers a class-leading cryptographic processor (Athena TeraFire EXP-5200B) in every PolarFire FPGA device. Not only does this Athena core implement numerous data encryption protocols, it also incorporates DPA countermeasures, so the data stream cannot be read by hackers. Every PolarFire FPGA also includes secure flash to store keys and precious data, a physically unclonable function (PUF), which is a unique device identifier and more security IP blocks for a complete data security solution.

Secure Hardware

Are you concerned about your design being overbuilt or cloned? Microsemi has a secure production programming solution (SPPS) that prevents this. All you have to do is use a Microsemi IGLOO2, SmartFusion2, or PolarFire FPGA in your design and enroll in our SPPS. Once this is done, your product will not able to be cloned or overbuilt. The SPPS leverages Microsemi FPGA's unique ID for each device (the flash key storage) and matches these to encrypted bitstreams. Only the number of devices you request to be programmed will ever be.

If you want to build security into your product contact Microsemi. We offer class-leading security that you can use in your next design.

NXP

Donnie Garcia, Solutions Architect for Secure Transactions

The broadest meaning of "security built-in" comes down to the guideline that to most effectively protect the data and services provided by a product, the security architecture must be considered in all phases of the design, from concept to end-of life. The security is "built in." This is the best strategy to ensure that the security policies of a device will be met.

For devices like NXP's i.MX RT series, "security built-in" also means that the implementation of security functions is rooted in the technology provided by the product itself. This is essential to providing a secure solution to the end consumers of embedded designs. A great example to illustrate this is the implementation of a secure boot: At the hardware level, the

i.MX RT crossover processor is designed to have explicit control of the boot process to ensure that only trusted firmware is executed. There are dedicated fuses for cryptographic keys, hashes of key material, and rollback protection. Within the i.MX RT is immutable firmware to check the authenticity of application code and protect the application code with cryptography. This ROM firmware performs the key management and leverages the hardware cryptographic accelerators inside the chip. The i.MX RT product enables the use of the high assurance boot (HAB) with tools to perform the code signing and provisioning to end devices. Because this security technology is built in, it reduces the effort required by the embedded developer, and it provides greater protection from physical attacks. From the start, having the capability to perform a secure boot ensures that the services provided by the application software are trusted.

Furthermore, built into the i.MX RT are capabilities related to maintaining security once the trusted application software is running. There are hardware firewall capabilities for protecting input-output peripherals, memory, and debug interfaces from attack. The Central Security Unit allows application developers the ability to build containers based on the state of the Arm® Cortex®-M7 processor which protect against unwanted accesses. This hardware is the basis for maintaining the principle of least privilege for the system.

Semiconductor manufacturers, OEMs, and cloud service provider all play a necessary role to address security. For its part, NXP products integrate the technology needed so that the built-in capabilities enable the successful integration of security functions for end consumers.

Renesas

Mark Schaeffer, Sr. Product Marketing Manager, Security Solutions, Synergy IoT Platform Business Division

In today's environment, security is a core product feature that must be integrated into multiple aspects of your product, much like safety and quality has traditionally been. This is especially important for IoT devices which are increasingly impacting every aspect of our lives. Solutions must provide protection against a variety of attacks/vulnerabilities such as malware, IP theft/product cloning, identity theft (compromising a person or device's identity), eavesdropping, data theft, authorized modification or replaying of data, etc. However, all security solutions typically have the same basic core components which include:

- Generation and secure storage of cryptographic keys, many of which must be protected in hardware
- Cryptographic algorithm implementation (e.g. symmetrical & asymmetrical encryption, hashing, random number generation)
- Provisioning identity (e.g. Certificates) to keys
- Mechanisms to isolate secure code and data from unsecure / general purpose code and data
- Mechanisms to ensure software/firmware is authentic and unmodified
- Security Protocols

Renesas Synergy™ is an embedded platform that includes:

- A Synergy MCU with hardware security features that includes protecting keys and sensitive code and data, a cryptographic engine which is lower power, faster, and requires less memory footprint than porting a software library, plus a true random number generator
- Synergy Software Package (SSP) libraries that provide security features such as TLS and key management, integrated with a core embedded development environment that includes hardware drivers, an RTOS, and development tools
- Security solutions with partners that provide identity/certification provisioning, device management, secure wireless connectivity, connectors to cloud providers, and other cloud solutions

The rapid development of increasingly connected IoT devices makes it imperative to take advantage of a platform such as Synergy to ensure timely deployment of your IoT devices while ensuring that you efficiently and effectively meet the security requirements of your environment. It is important to ensure that the foundation of your security solution (keys and firmware) is protected in hardware (e.g. the Synergy MCU) as most attacks on IoT devices are remote attacks on the software stacks. Once you have established a strong hardware based root-of-trust, you can use the additional components and solutions provided by Renesas and partners to complete building a solution that your customers can trust.

Secure Embedded Systems with More Economical Hardware

Hardware-based security solutions are quickly evolving to offer much-needed protection for embedded systems, joining the connectivity bandwagon while opening new entry points for intruders.

By Majeed Ahmad for Mouser Electronics

Embedded systems have so far relied on the "security by obscurity" approach while creating software patches here and there in order to address specific security threats. But things are quickly changing as an increasing number of embedded devices are becoming connected while opening up new entry points for hackers and intruders.

The real and present danger is turning embedded system designs upside down with the popularity of the Internet of Things (IoT), machine-to-machine (M2M) communications, and remotely-controlled industrial systems.

According to a recent study from the Barr Group, nearly half of the embedded systems surveyed feature some form of connectivity. At the same time, however, a significant portion of engineers didn't place a high priority on securing their embedded designs.

Is securing embedded systems in a robust manner really a mystery? Security breaches in embedded systems are happening around the world. Take, for instance, the cyber attacks on a power grid station in Ukraine that affected energy supply to nearly 200,000 people.

Another example highlights data breaches in the U.S. hospitals where hackers injected malware in blood gas analyzers and X-ray equipment. Then, there is this well-publicized event in which car hackers were able to disable the brakes of a Jeep Cherokee by replacing the infotainment system's firmware with malicious code.

This article delves into the anatomy of hardware-based security solutions and shows how they are transforming the embedded system designs by making them safer. It will also debunk the common myths associated with hardware security solutions.

Why Hardware-based Security

The U.S. Defense Advanced Research Projects Agency (DARPA) has recently called for building security protections directly into hardware, a testament that the traditional ways to safeguard embedded systems have reached a crossroads.

The nature of security breaches in embedded systems keeps growing and changing the modus operandi. Software developers create patches to address a security threat only to find out that determined intruders have discovered a new loophole.

And while the list of security threats to embedded designs keeps expanding—from spoofing to tampering and from denial-of-service to elevation of privilege—it's simply not feasible to keep adding embedded software code.

Enter hardware-based security solutions that implement authentication and cryptography parts early in the design cycle. And instead of creating software patches now and then, they provide a more holistic solution to a diverse array of security challenges.

The quest to build security from scratch is centered on processors that facilitate a multitude of protection mechanisms. And these processor-based solutions cater to both device- and network-level security needs.

Here are some of the key benefits that hardware-centric solutions bring to embedded system designs.

Speed

A processor-centric hardware solution speeds up the security operations with cryptographic accelerators. The hardware-based cipher suite reduces compute time by orders of magnitude over software solutions, and that leads to significant performance improvements when compared to firmware-based solutions.

For example, take an embedded device such as an energy meter that implements a Transport Layer Security (TLS) mechanism to secure the link to the cloud. The hardening of TLS communication speeds up the authentication of the link between the embedded device and the cloud ecosystem. Hardening allows designers to eliminate software vulnerabilities by creating an additional hardware security layer.

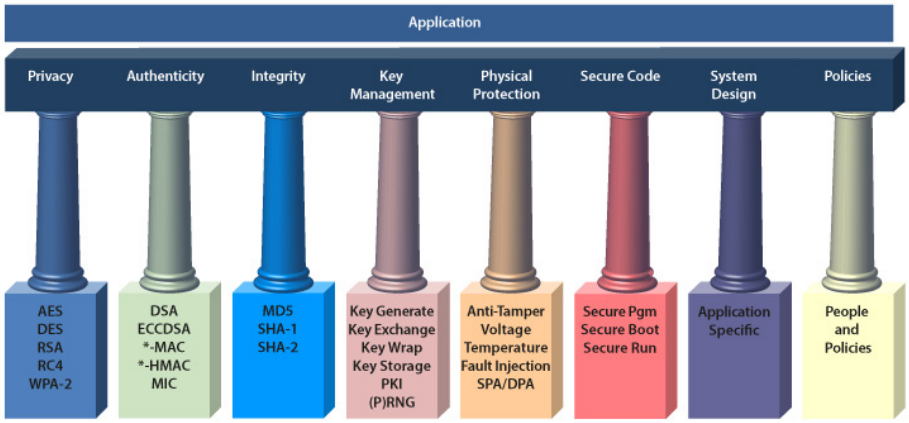


***Figure 1:** The ever-expanding threats to embedded systems demand a robust security framework. (Source: Microchip)*

Next, there are improvements in latency responsiveness as embedded applications can run security measures quickly.

Robustness

Hardware solutions are self-contained because the software is hardcoded into the chip. As a result, hardware-based security solutions are quickly emerging to serve reliability-conscious industries like automotive, industrial, medical, aerospace, etc.

Moreover, as embedded systems are becoming smaller, they have limited code space and power budgets to accommodate the increasingly complex software

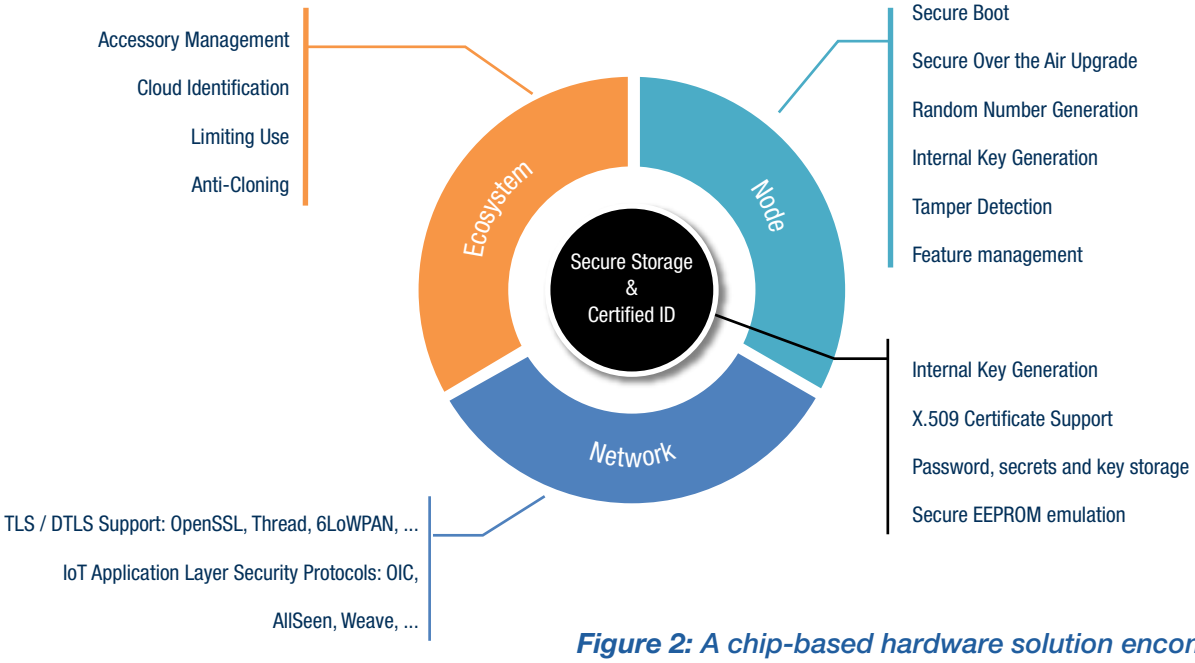


***Figure 2:** A chip-based hardware solution encompasses the entire root-of-trust to counter a vast array of security issues. (Source: Microchip)*

solutions where the amount of code keeps on growing.

On the other hand, security processors with embedded root-of-trust include a set of robust cryptographic functions—encryption and decryption, hashing, and cryptographic key generation—and all of these are performed with well-defined APIs.

And new security processors are mostly using the Elliptic Curve Cryptography (ECC) algorithms instead of employing the traditional RSA ciphers, which are expensive and consume more power due to larger key size.

Longevity

Unlike software solutions, which require regular updates with security patches, hardware security allows embedded designers to cope with a multitude of security threats for much longer periods of time without worrying about software updates.

The embedded systems like security cameras and industrial robots are up and running for longer periods of time. Likewise, there are physical barriers to periodic updates in embedded systems like turbines for hydroelectric dams.

The security by design mechanism solves this conundrum and brings the much-needed operational stability that embedded systems demand.

Putting Structure to Embedded Security

What hardware-based solutions do is bring structure to the security framework for embedded systems and help them counter threats of multiple dimensions.

Here is a sneak peek into how the multi-level hardware security streamlines the three major building blocks of embedded security:

Secure Boot

It's the first line of defense, which ensures that the firmware code and operating system in an embedded system are authentic. Secure boot safeguards against threats such as cloning, hacking and reverse engineering.

Software-based secure boot solutions, which use hash algorithms, are time-consuming and they burden the main MCU. On the other hand, hardware solutions are faster, cheaper and consume lower power.

Hardware solutions include MCUs with embedded non-volatile memory (eNVM) such as ROM, flash, etc. that's write-protected. Then, there are custom solutions like on-chip one-time programmable (OTP) memory that provide randomized key storage to prevent tampering.

Another approach entails the locking of flash during the manufacturing. And most security processors come with a built-in secure boot feature.

Device Identification and Keys

It's worth noting that there are three key pillars of embedded security: Authentication, encryption and secure data storage. But hackers rarely break into the authentication and encryption because it's a computationally intensive task.

Hackers and intruders usually break into areas where the keys are stored, so they don't have to decrypt any algorithms. Therefore, an embedded system is as strong as how securely its stores the keys and certificates.

Security processors protect against reading and modification of keys and certificates with encrypted data storage and secure communications.

While eNVM and OTP memory technologies facilitate secure key storage, security processors come equipped with unique keys and certificates. And that eliminates the need for adding secure keys to an embedded device during the manufacturing process.

Cryptographic Acceleration

The hardware acceleration for cryptographic algorithms such as ECC and Secure Hash Algorithm (SHA) can be implemented in a wide array of security requirements. It provides random number generators while speeding up the embedded system's boot time.

Hardware accelerators also help generate, verify, and certify public and private keys. What they do is accelerate the coding and decoding process for tasks like ECC cipher suites for mutual authentication and key agreement.

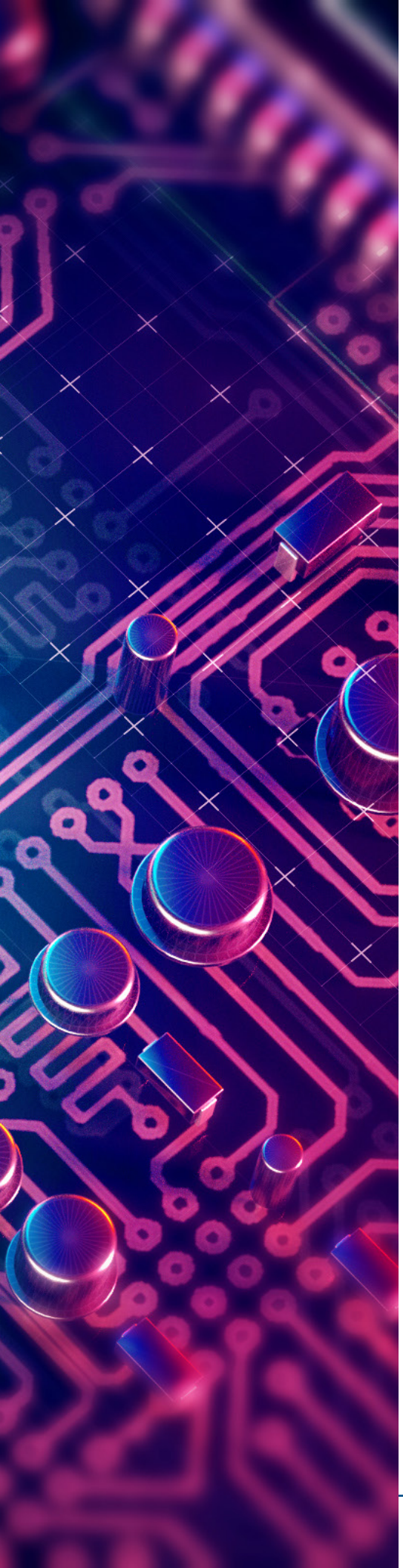
Hardware Security Stumbling Blocks

The common perception among embedded designers about hardware-based security is that it's a costly proposition. Indeed, a robust level of security demands processor overhead, and that puts upward pressure on the cost.

A case in point is a hardware secure module (HSM) that helps create keys and certificates for embedded systems using a secure database infrastructure. The keys are then programmed into embedded devices by connecting the HSM to automation equipment during the device manufacturing.

But it's an expensive solution that only high-end embedded devices produced at a larger scale can afford. Likewise, the solutions based on the Trusted Platform Module (TPM) technology, which has its roots in the computer industry, need to maintain a database of keys to provide provisioning services.

So TPM-based solutions are relatively expensive for low- to mid-end embedded systems produced at a smaller scale. Not surprisingly, therefore, security



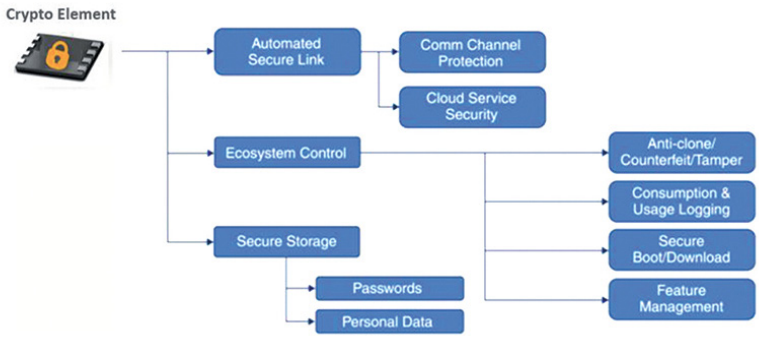


Figure 3: *A crypto element embeds the whole root-of-trust in a low-cost and low-power chip. (Source: Microchip)*

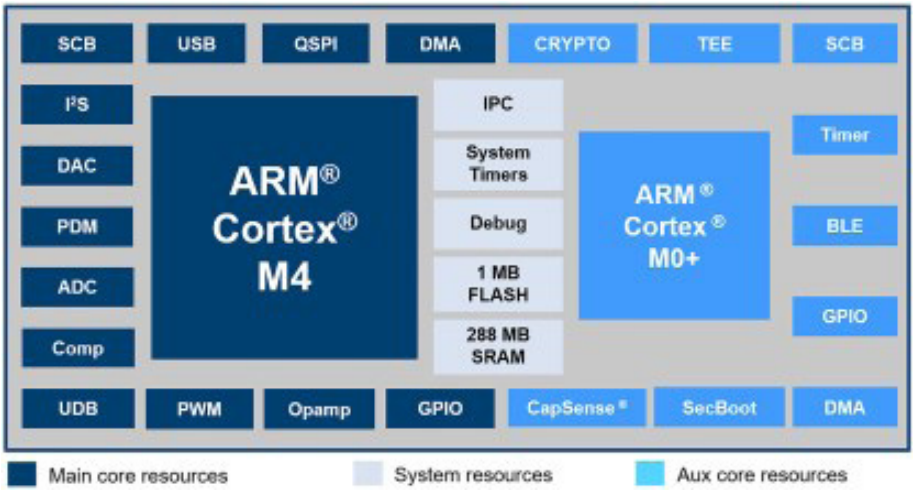


Figure 4: *Cypress Semiconductor has integrated secure boot, protected memory, and crypto accelerator around the Cortex®-M0+ core in its PSoC 6 microcontroller. (Source: Cypress Semiconductor)*

systems based on the HSM and TPM technologies are giving way to more economic solutions.

And that leaves us with two fundamental approaches in hardware-based security solutions, and each one of them is promising to facilitate embedded security in a more economical way.

Specialized Security Processors

A crypto element or secure element is a low-cost MCU that features cryptographic capabilities and secure memory and interfaces **(Figure 3)**. It is a specialized IC that eliminates the need for HSMs, secure rooms, and manufacturing audits.

A security co-processor is designed from the ground up to counter multiple security threats by catering to all three building blocks outlined in the above section. Furthermore, it offloads security tasks from the main MCU or MPU so they can focus on compute-intensive tasks like sensor hub management.

A security co-processor comes pre-loaded with unique keys and certificates. That simplifies the manufacturing process by effectively dealing with the key provisioning. Moreover, it ensures that encryption keys are secure throughout the manufacturing supply chain.

SoCs with Built-in Security Features

The second popular approach in hardware-based security is to integrate an MCU with cryptographic accelerators and secure memory and interfaces within a system-on-chip (SoC) device **(Figure 4)**.

The security subsystems in a multicore SoC embedded processor use the CPU cycles to accelerate authentication of code and applications. But it's a more flexible solution for integrating various protections in embedded designs.

For instance, to counter threats like tampering, peripherals can be configured to perform current and voltage sensing and clock monitoring. That, in turn, helps detect the unauthorized attempts when someone tries to open the enclosure.

Conclusion

Hardware systems, which build a solid security foundation for embedded systems, are far more difficult to spoof or hack than software solutions. And while software is increasingly becoming complex, new hardware solutions simplify the security obligations.

So the evolution from software to hardware is refining the design journey for engineers aiming to secure embedded systems. A new ecosystem for securing embedded devices is emerging while component prices are gradually coming down.

The emergence of new chip architectures is reshaping the hardware security solutions once afforded by only large OEMs with deep pockets. That also means that embedded system designs are going to heavily rely on the expertise of chipmakers.

In the final analysis, hardware security for embedded designs is no more a costly proposition. There are security co-processors that cost less than 50 cents for embedded applications. So you don't have to spend a fortune to secure your embedded system.

Inside the Trusted Zone

By Paul Pickering for Mouser Electronics

Among security best practices, system partitioning separates and isolates security-related hardware, software, and data in a "trusted zone" and tightly limits all attempts to access trusted content from outside the zone. Non-secure software cannot directly access secure resources. The Trusted Zone isolates security-related MCU hardware, software, and external devices (**Figure 1**).

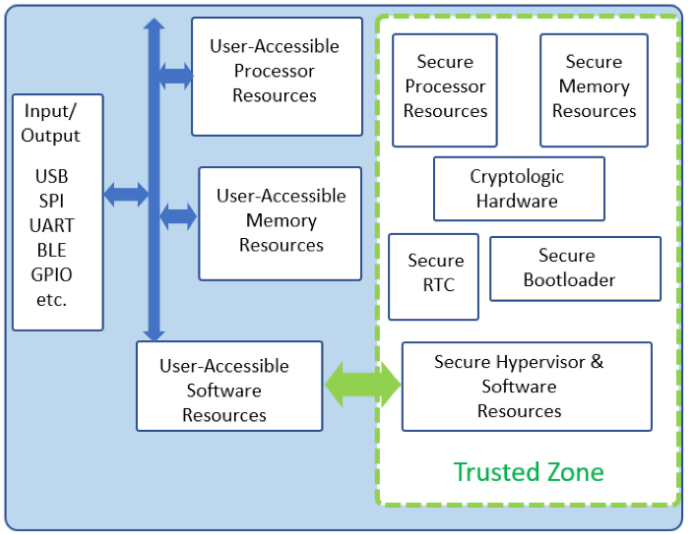


***Figure 1:** A generic secure embedded system architecture must isolate security-related MCU hardware, software, and external devices. (Source: Author)*

Inside the Trusted Zone

Secure Boot and Secure Download

If an attacker can fool an embedded system into accepting fake code as authentic, then all is lost. All code must be verified as trusted. During manufacturing, the factory generates a public/private key pair. The public key is stored in a secure location in the MCU, often in one-time programming (OTP) memory. When the MCU boots up at power-on or receives a download, it verifies the code against its key and rejects non-conforming code.

Hardware Accelerator

Implementing an AES or RSA algorithm is computationally intensive, so microcontrollers in secure embedded systems usually contain hardware accelerators to speed up common cryptographic operations. Specialized instructions in the MCU then access the accelerators to perform operations such as AES encryption and decryption.

A Secure Real-Time Clock (RTC)

A secure RTC guards against an attacker tampering with the clock settings in an attempt to disable system operation. The function is often combined into a supervisor device that continually checks the system voltage and switches to battery backup if primary power fails; if the battery backup voltage drops, the supervisor will signal a tamper event.

True Random-Number Generator (TRNG)

Random numbers are critical in secure systems to generate random cryptographic keys for secure data transmission. A software algorithm can produce a long pseudo-random sequence but it is deterministic and therefore vulnerable to attack, so a secure microcontroller should incorporate a hardware TRNG, which gives an unpredictable output. TRNGs have a colorful history—the numbers from one early online TRNG originated in the waxy blobs generated by a lava lamp—but modern implementations use thermal noise or the interaction between several free-running oscillators as sources of randomness.

Temperature and Voltage Monitoring

Excessive variations in system voltage and temperature are two common ways to mount a hardware attack. This feature detects such excursions and initiates an appropriate response, such as switching to backup battery power or zeroing out private keys. ●

Encryption: The Foundation for Embedded System Security

By Barry Manz for Mouser Electronics

A design engineer's goal is to start from nothing and work toward accomplishing something useful. In contrast, a hacker's mission is exactly the opposite: Stripping hardware and software of its clothes, finding a point of entry, taking control of its host, and extracting from it as much salable information as possible. Attackers first explore the most obvious and potentially easiest ports of entry: Where the system connects with the rest of the host platform, to peripheral devices, and outward to the Internet. Each one must be secured as effectively as possible, including USB and other removable media, SATA, FireWire, Ethernet and fieldbuses, wireless interfaces such as Bluetooth, Wi-Fi, and other radio networks, as well as any other wired or wireless interfaces. The multitude of possible ports of entry in embedded systems is why securing them is so critical and so challenging.

Encryption is a key component of embedded system security because it can ensure data is secure as it enters and exists the multitude of nodes it passes through. This article explores encryption as it relates to embedded systems and discusses encryption design considerations, as well as provides additional security best practices.

What is Encryption?

Encryption refers to scrambling data into unreadable code before it's transported between end-nodes, using highly-complex algorithms. Identification keys are then used to unscramble the data on the receiving end.

Many efficient, secure building blocks, such as the National Security Agency-approved Suite B cryptography, can be implemented with software,

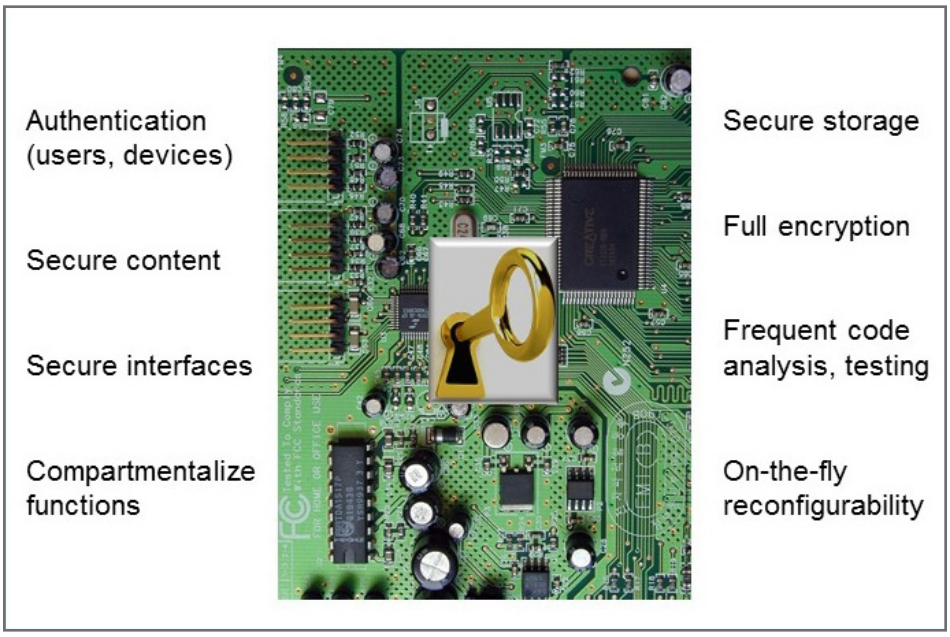


***Figure 1:** Securing an embedded system requires adhering to these rules, and others. (Source: Author)*

firmware, or hardware, and can often be obtained as open-source IP. Advanced Encryption Standard (AES) is a specification standard by the National Institute of Standards and Technology (NIST) for the security of data. AES is a widely recognized and adapted cryptographic module used in the U.S., Canada and worldwide by military, government, financial institutions, and organizations all around the world as the standard for encrypting and decrypting of data.

There are different degrees of AES hardware encryption—for example 128-bit, 192-bit, and 256-bit—with each key size providing an increased level of protection and complexity. Encryption effectiveness, though, also depends on the generation, distribution, and protection of keys, and it is often assumed that larger keys (e.g., moving from 128-b to 256-b AES encryption) provide added benefits. However, in practice this has a minimal effect because the brute force attacks required to break them on 128-b algorithms are extraordinarily difficult without massive computational resources. Consequently, the perception that higher-level encryption is "better" can produce a false sense of security, while hackers simply use other techniques for intrusion.

Encryption Design Considerations

Recurring questions when implementing security, especially algorithmic cryptographic processing, are whether the embedded system has the resources to handle this additional load, and how security will affect system costs and design time. The team's responsibility is to reach an equitable balance between achieving an acceptable level of security and the budgeted cost, taking into consideration the system's size, weight, and available power, as well as usability and the allocated development time.

This issue is a significant design concern today as the number of resource-limited embedded circuits is increasing exponentially with the deployment of industrial and consumer IoT networks. In many cases the cost of wireless-enabled sensors is so low that the addition of cryptographic processing could be the same as the device itself. Ironically, these devices are precisely those in which security is needed most, as nearly all connect to the Internet. However, various levels of security are available to designers of these tiny circuits.

For example, transceiver SoCs support one or more short-range wireless standards such as Bluetooth and ZigBee and thus include their inherent security features (if the designer fully enables them). Mid-priced 16-and 32-b microcontrollers increasingly provide AES encryption as well as code protection. It remains to be seen if in the future, silicon vendors can incorporate processing resources inexpensive enough for use in building the dirt-cheap IoT devices that will be used by the billions.

Every device and point of entry should be authenticated using

cryptographically-generated digital signatures. An unidentified key should produce a response and if it occurs repeatedly, it's probably an attack. This is hardly an exhaustive list and considerable resources are available on the Web that—after sorting the wheat from the chaff—can be extremely helpful.

Additional Security Design Best Practices

The following priorities appear frequently in the writings of the security community **(Figure 1)**.

Think Like a Hacker

View security from the perspective of the hacker. This mindset is a change from how engineers approach design but well worth the effort. The security community also stresses that designers strive for simplicity, as every bit of unnecessary code is a potential vulnerability.

Make Security Part of the Design Plan

Also high on the list is the need to continually reaffirm the design's security from conception through finished product rather than as a late-stage endeavor. Problems are easier, faster, and cheaper to fix the earlier they are discovered.

Analyze Source Code

Static Application Security Testing (SAST) tools for analyzing source code are invaluable, widely available, and often free. And as with every type electronic system, security efforts can only be accurately verified when tests subject the product to environments as close as possible to what they will experience in the "real-world."

Compartmentalize

Another rule is to compartmentalize, separating functions and minimizing their ability to expose information to each other or to the network. In the same vein, access to software including the operating system, firmware, and downloaded code should be regulated among different portions of the system during different stages of execution.

Use Identical Test Routines

Data at both ends of a transmission path should be verified using identical test routines because while bad data coming from a trusted device may be benign, it can also be the result of a hack. Every device and point of entry should be authenticated using cryptographically-generated digital signatures. An unidentified key should produce a response and if it occurs repeatedly, it's probably an attack.

Conclusion

Encryption is the foundation of embedded system security. Many efficient, secure building blocks implemented with software, firmware, and hardware; however, simply standard cryptographic algorithms cannot guarantee adequate security. Instead, encryption effectiveness is directly related to how well the algorithms are integrated. Every device and point of entry should be authenticated using cryptographically-generated digital signatures. Using encryption, along with other security best practices, will help ensure system and user data is secure. ●

WEBINAR

Think Like a Hacker

Now Available On Demand
Presented by Mouser Electronics in partnership with Anitian Corporation

When your job is to create and build embedded systems and products, intentionally breaking your creations seems wrong. But not to a hacker. They do not see your systems and products as elegant designs that solve problems. Hackers see your product as a means to an end. When we look back over the past 20 years of cybersecurity and data breaches, almost every incident has some type of exploitation of a vulnerability—perhaps a poorly designed API or a third-party component with outdated code. This is what cybercriminals, malware writers, and state-sponsored hackers obsess over: How to break what you build.

If you want to build more secure, more resilient embedded systems and products, you have to look at them the way a hacker does. This means identifying potential vulnerabilities, prioritizing security as a design requirement, and then integrating security into every dimension of system and product development. In partnership with Mouser Electronics, Anitian security experts will discuss how you can start looking at your embedded system designs and architecture the same way a hacker does. Attendees will learn how to improve development practices to integrate security at every stage.

Key Take-Aways

- Understand how systems get hacked
- Learn ten areas of embedded systems and products that hackers focus on
- Discover how to add security to the product development process

Presenter: Andrew Plato, CEO, Anitian

Andrew Plato is a 20-year veteran of the information security industry. In 1995 while working at Microsoft supporting one of the first Internet e-commerce sites, Andrew inadvertently hacked the site using a SQL injection style attack. This incident inspired Andrew to start Anitian with a mission to help people understand the complexities of protecting data and systems. Since then, Andrew has participated in thousands of security projects, authored numerous articles, and given hundreds of presentations on all matters of IT security. Andrew serves a brilliant team of security analysts and is committed to building collaborative, high-trust environments that value innovation, engagement, and accomplishment. Andrew›s forthright and pragmatic views on security, risk, and compliance have made him a highly sought speaker, author, and advisor.

Combatting Side-Channel Attacks

By Paul Pickering for Mouser Electronics

When we think of physical security of embedded systems, we most often think of securing the premises or securing against theft of mobile devices, for example. A subtype of physical system security, called side-channel attacks, pose unique security needs, however. *Side-channel attacks* are ones in which the hacker exploits a system based on information gained from the physical system, such as the power consumption, timing information, electromagnetic leaks, and cache access, to name a few. What's more, accessing, monitoring, and exploiting side-channel attacks often only require common laboratory equipment such as a PC, a temperature probe, and a digital oscilloscope.

Two characteristics make side-channel attacks difficult to combat:

- The attack might not disturb normal operation in any way, making it difficult to detect.

- The algorithms and source code for cryptographic methods are publically available and their implementation sequences are well understood,

making attacks relatively easy to implement.

This article explores three types of side-channel attacks—power attacks, timing attacks, electromagnetic emission attacks—and provides steps for protecting against them.

Common Side-Channel Attacks

A system's power, timing, and electromagnetic emissions all provide useful information for a hacker to exploit:

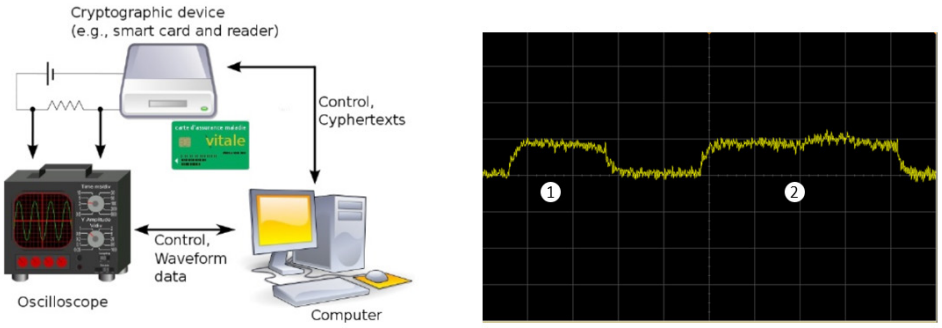


***Figure 1:** A differential power analysis uses commonly available test equipment to provide useful information about secure data. (Source: Wikipedia)*

Power Attacks

A *power attack* measures the power dissipation during cryptographic operations. Analysis can reveal what operations are performed in the chip, leading to conclusions about the secure information. In a differential power analysis, the attacker makes guesses about a secret or private key, collects the power signals related to the hypothesis, and correlates it with the actual power signal. **Figure 1**, for example, shows an attempt to decode RSA key bits using power analysis. The left peak (1) represents the changes in MCU power consumption while executing a step in the RSA algorithm without multiplication; the right peak (2) shows a step that involves multiplication, allowing to the attacker read bits 0 and 1. The stronger the correlation between the hypothesis and the actual measurement, the closer the guess. After multiple operations, the secret key can be deduced from statistical analysis.

Timing Attacks

A *timing attack* logs the number of clock cycles—i.e., the computation time—needed by the embedded system to provide a cryptographic result. The variation in computation time can provide information about the secure key's Hamming weight, a measure of how close the predicted key is to the real value. An attacker can start guessing key bits and observe which results show the strongest correlation between the predicted and actual times. With enough samples the attacker can recover the whole key.

Electromagnetic Emissions Attacks

An *electromagnetic emission attack* uses the electromagnetic radiation emitted from the chip for simple or differential analysis using a comparable approach to that of a power attack. The attack can target a specific area of a chip; the detector can even be mounted remotely, although stray RF interference and measurement errors limit the effectiveness of such a setup.

Defending Against Side-Channel Attacks

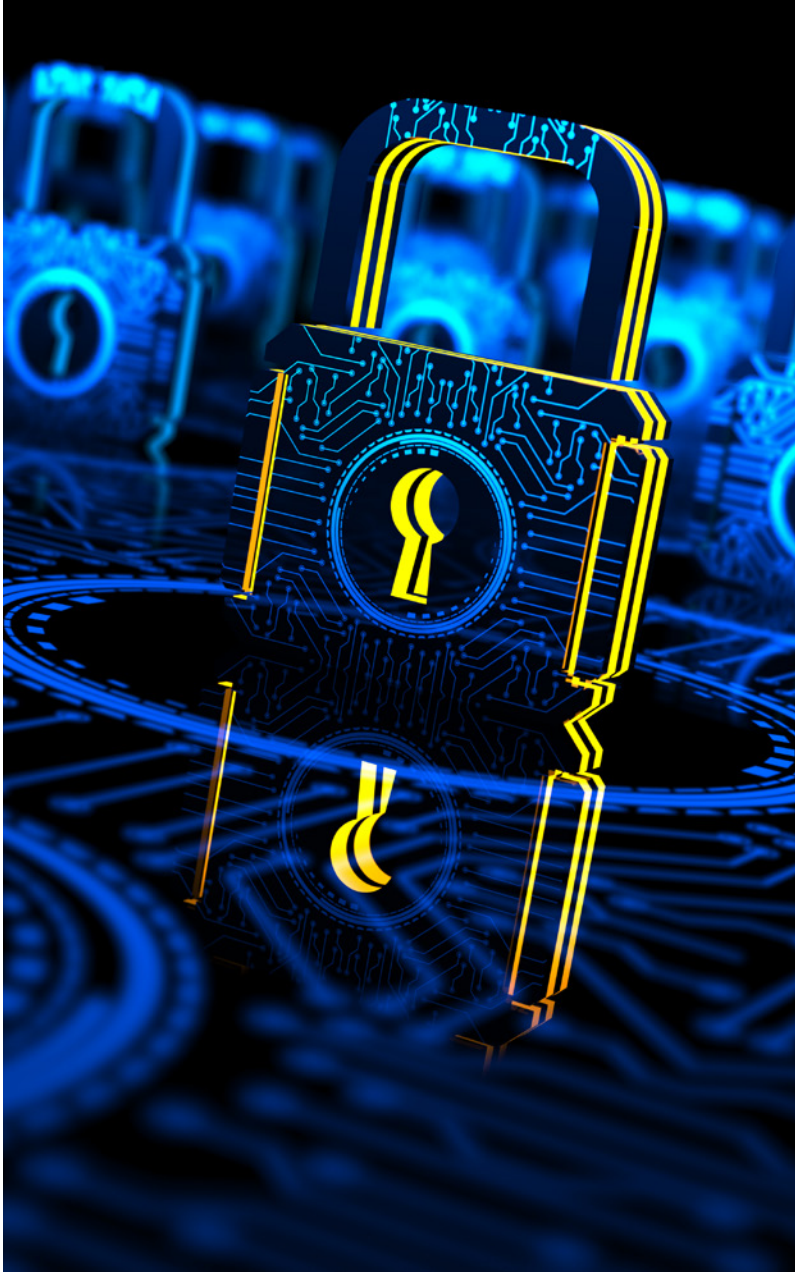
Since the operations needed to perform the encryption and its correlation to physical events—power consumption, emissions, or computation time—are well understood, the underlying defensive principle is to break, or at least minimize, the link between the physical effect and the cryptographic operation. The following techniques are commonly used:

- **Eliminating the release of information:** Understanding and eliminating information that is leaked from the system—and thus available for hackers to exploit—is a first step in combatting side-channel attacks. A common step here is to add shields and filters that lessen electromagnetic emissions.

- **Uncorrelating the Leakage:** Another method of defending against side-channel attacks is to uncorrelate the leakage from the underlying operation by, for example, inserting random elements into the algorithmic computation, or using code with a constant execution path regardless of the operation being performed.

- **Minimizing the magnitude of the relevant parameter:** Another method is minimizing the magnitude of the relevant parameter in all phases of operation and normalizing it so that it remains unchanged during the cryptographic operation. For example, adding shielding or internal metal layers are ways to reduce emissions, although they do increase product cost and size.

- **Adding noise to the channel:** Adding random signals or delays can help as well. Here, the more noise, the more measurement and analysis needs to be done to identify system information. Adding cryptographic code to cache is an example of adding noise, which in this case, prevents hackers from determining the frequency (or infrequency) of data access.



Conclusion

Side-channel attacks exploit information gained from the physical system, such as the power consumption, timing information, electromagnetic leaks, and cache access. These types of attacks are hard to combat because monitoring system information requires only common lab equipment, doesn't disturb normal operations, and applies source code that's commonly available. Power attacks, timing attacks, and electromagnetic emission attacks are common types of side-channel attacks, but risk can be mitigated by eliminating the release of information, minimizing leaks and parameters, and adding noise. ●

Security and Industrial IoT

By Michael Camp for Mouser Electronics

Since the introduction of the first programmable logic controller (PLC) in the late '60s, industry has largely depended on physical security and the point-to-point nature of connections like RS-485 to avoid the problems associated with data security. As serial connections are increasingly being replaced by local area and wireless networks, data that was once confined to the physical premises of a factory or plant now routinely travels around the world for processing, monitoring, or storage. With the advent of Industry 4.0, the secure perimeter around a company's assets is evolving from a physical one to virtual one.

The "factory floor" is no longer confined to the space under the roof, but now extends to materials yards, mine pits, oil and gas fields, and rail yards. Instead of numerous discrete locations operating independently, the enterprise functions as a virtual "factory" spanning cities, states and even hemispheres.

The Hacker's Rationale

It would be a mistake to assume that a hacker's motivation to compromise IoT devices is limited to creating a nuisance for the operator. A wide range of economic and political rationales may drive an individual to disrupt an industrial IoT (IIoT) network. An attacker could disable remote sensing devices, or inject spurious data with the intent of wasting company resources on dispatching a service crew to the location of the compromised device. Both utilities and energy production companies seek to lower maintenance costs through remote monitoring. Being forced to dispatch a service crew to a remote oilfield or substation places pressure on company resources.

Understanding how your company can be harmed by compromised devices and applications requires a certain amount of cynicism about people's intentions. We all need to take off our rose-colored glasses occasionally and think like a hacker. Protecting the MCU, data, physical systems, and the network are all important in protecting the IIoT.

Protecting Your IIoT

Security consultants have long advocated "layers of security" for both physical and data-based systems. The strategy still applies when you are protecting the remote IoT devices that are outside of your immediate control.

Protecting the MCU

Securing a device begins with securing the MCU and the boot process. Most of today's secure platforms are built

around the Trusted Platform Module (TPM) specifications published by the Trusted Computing Group (TCG). The TPM provides resources to the system that help with encryption, authentication and management of security keys. Keys are stored in dedicated, non-volatile memory on the TPM chip, which prevents code on the MCU from inspecting them.

A secure boot system would first execute microcode or non-readable code that performs a cryptographic hash on the boot code stored in flash. Only if the signature of the boot code matches that stored on the TPM would the boot process proceed. At this point the boot code is considered "trusted" because the contents are confirmed to be in a previously trusted state.

The boot code itself can perform similar checks on any additional software that is loaded, creating a "chain of trust" that holds if all the links in the chain are verified. A secure boot process makes tampering with the device software extremely difficult, if not impossible.

MCUs from several manufacturers also provide additional features to enable secure computing. The code security module (CSM) allows developers to secure on-chip memory by writing a 128-bit passkey to a specific location. Once the code is secured, the chip will permit instruction fetch operations on the secure addresses, but not data read or write operations. An option to permanently secure the memory is also available, after which the memory cannot be read regardless of the passkey entered.

Protecting the boot code and the MCU does not happen automatically. These are conscious decisions made by the device designers and may have consequences, such as a bug in the secure boot code that cannot be corrected once memory is permanently secured.

Protecting Data

A determined attacker with much to gain from a hack will gladly disassemble even the most rugged device to find its vulnerabilities. Shipping production hardware with debug interfaces is a great way to make the hacker's lives easier. It is well within the resources of hackers to purchase JTAG or SWD probes to read and write RAM, or re-flash on-chip memory. Peripheral TPM chips generally use an I2C or SPI interface directly to the MCU to prevent easy bus snooping, but are still vulnerable. The TPM hardware is delivered with a pre-programmed key that is never exposed to the rest of the system, and additional keys are typically derived from the factory installed key, and never leave the module itself.

Eliminating debug interfaces provides an additional deterrent to malicious tampering.

Protecting Physical Devices

It may be impractical to keep an eye on your IoT sensing devices 24/7, particularly if you have deployed thousands of them. For this reason, it makes sense for your devices to keep an eye on themselves, and let you know if something undesirable happens.

Desktop computers have included intrusion detection switches for years that are capable of reporting when the enclosure is opened, but software to actually take action on such an event is relatively rare. Cover open alarms are more common in a server environment where out of band management software is much more widely used.

The principle is easily extended to IoT devices, where reporting the open/closed state of a switch is fairly trivial. Adding software to send an immediate alarm when the enclosure is opened provides a minimal degree of assurance that the device has not been tampered with. At the very least, the operations team that monitors the alarm can disregard data received from the sensor that was interfered with.

Protecting the Network

The greatest vulnerability of any IoT network is its exposure to the public Internet. Simply adding an Ethernet connection or wireless transceiver makes the devices a target for botnet herders, ransomware schemes, and data skimmers.

By definition, disabling Internet access is not an option because there is no such thing as the _____ of Things. So, if having a network interface of some kind is such a

gaping security hole, how does Industry 4.0 make sure that the connection is used only for good, and not evil?

One approach may be to take advantage of out-of-band communications that are available on cellular radio modules. All cellular modems expose either a binary- or AT-command-interface that allows software to send and receive short messages (SMS). While the cellular infrastructure may use IP for both data and SMS, they are routed differently in each case, so correlating and SMS with a particular TCP/IP connection is difficult. If an IoT device does not allow incoming TCP connections, we can envision the following scenario.

Suppose that one of your IoT devices is reporting errors, so you want to log in remotely and check the device logs to diagnose the problem. Most current devices leave open a Telnet, SSH, or HTTP port, which allows an operations team to connect to it for diagnostic purposes. If we instead disable incoming TCP connections, there is no way to initiate a session with the device. In this case, a specially formatted (and encrypted) SMS is sent to the device, which responds by starting a TCP session to the IP address encoded in the SMS. This creates a kind of "reverse" tunnel over which to conduct subsequent exchanges.

A natural option for securing communications is to use SSL/TLS for establishing a secure channel between the operations center and the remote device. This is an ideal choice if the remote device has sufficient processing power to manage the encrypted link. An additional barrier to using this approach is that each device needs to be provisioned with a certificate and private key, which complicates the manufacturing process.

Regardless of bandwidth constraints of the device, it is good practice to ensure that only TCP ports that are actually used by the application are left open. Additionally, if the device absolutely has to include a management interface (web or SSH), do not provision the devices with the same administrator username and password from the factory. Instead, use a default password that is random, and include it on a sticker with the device. This puts additional burden on the installer to either make note of the device's password, or to change the password to something that is managed by the purchaser.

Randomized passwords may be perceived as less user-friendly than a fixed default, but it was exactly how the botnet that attacked *Krebs on Security* was formed—by scanning the Internet for devices with default passwords still active.

Conclusion

A company with thousands of remote sensors stands to lose much more than the inability to check a nanny-cam on the web. The more distributed a company's operations, the more potential vulnerabilities there are to exploit. Even the lowliest device can become a loose thread that can be picked at until your entire operation unravels. ●

Physical Security for Embedded Systems

By JPaul Carpenter, Mouser Electronics

When we think of embedded system security, we often focus on securing the data as it moves from end-node to end-node; however, two asepcts of embedded systems make them vulnerable to physical security breaches: Their portability, as in the case of mobile devices, as well as their connection to and place in smart systems. This article explores key concepts and solutions for physically security.

Physical security solutions aim to meet the following goals:

- Deter a potential threat
- Detect an actual threat
- Asses the detected threat
- Alert the end user or controller
- Respond to the threat

Premises with the potential for widespread detrimental effects of a security breach are particularly vulnerable, as are ones not secured by onsite security staff or that are otherwise monitored remotely. And for premises that have high levels of electromagnetic interference (EMI), the security challenges increase exponentially. In securing physical premises, we've seen dramatic improvements in all areas of premises security, especially in the latter four.

Deterring Potential Violators

The first line of defense for any security solution is deterring those who don't belong by installing physical barriers and communicating restrictions and potential consequences: Chain link fences, 20-gauge barbed wire, minimal lighting, signage, and sensors in areas most vulnerable to intrusion. Often, deterrents are layered to provide multiple security barriers between potential violators and the protected assets, such as fences, lighting, remote-access locks, and motion-detection sensors around individual assets, for example.

More recently, the focus shifted toward the methodology of deterring, while also striking a balance with aesthetics due to NIMBY (Not In My Back Yard), which refers to opposition to new development that's perceived as being too close to nearby residents. Most advancements have focused on increasing the difficulty of getting past the fence, increasing the visibility of intrusion, providing a two-way live communication system, and making the lighting solutions better. Another area of advancement is increasing the difficulty in accessing or obtaining a protected asset by putting multiple security barriers between potential perpetrators and the protected asset, such as fences, lighting, remote-access locks, and motion-detection sensors around individual assets, for example.

Cameras serve a couple of different functions in a security system, including being a deterrent just in their presence. And behind the scenes, large scale video monitoring systems allow security professionals to view thousands of people, track their entire route on premises, and analyze faces and walking gaits. For this use, camera systems are a good choice.

Detecting Violators

Detection is the next key element to any security design solution. This concept is clearly explained in signal detection theory (**Figure 1**), which identifies the idea that there is a stimulus or there isn't, and the system can correctly or incorrectly identify that fact. The opportunities for failure can be introduced from sensor and connection failure and contribute to the False Alarm result or the human who contributes to the Miss result.

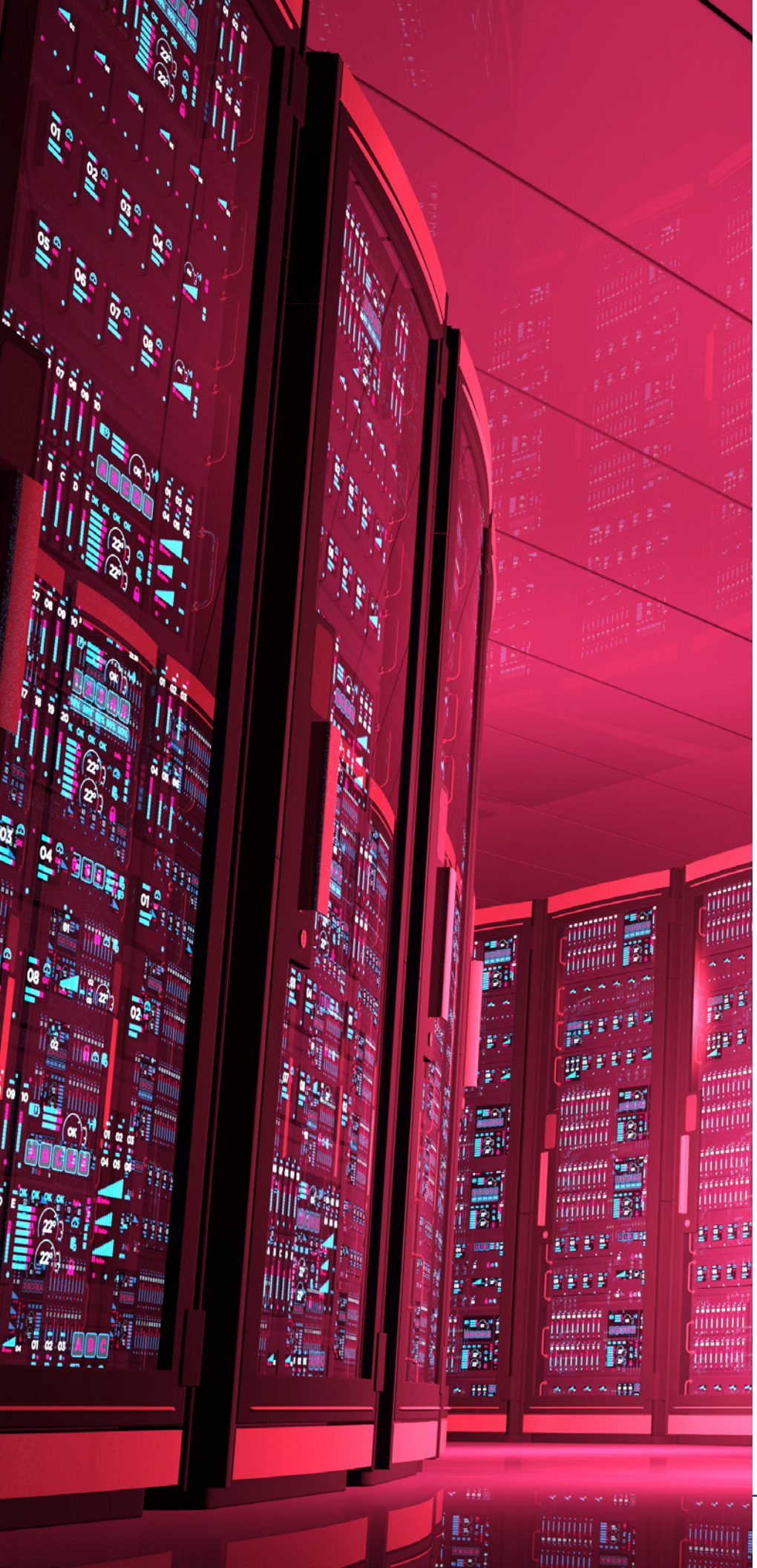
		State of the world	
		Signal	**Noise**
Response	**Yes**	HIT	FALSE ALARM
	No	MISS	CORRECT REJECTION

***Figure 1:** There are two successful outcomes and two opportunities for failure. (Source: Abdelhamid, et al. 2004)*

The oldest detection was done with living, breathing things. Whether animals or human, they all suffer from a few common problems: They all need to sleep, eat, take breaks, cannot see well in the dark, have hearing limitations, and can simply become distracted. This is where technology has come in, offering solutions that are more reliable, more sensitive, and faster to detect and alert. Support sensors such as PIR/occupancy sensors, radar, and beam break all still serve valid, essential functions and are becoming smaller and smarter serving to increase a building's overall intelligence.

Further, imagine if the security system were connected to other smart building components. What if the system could also detect potential threats from the lighting and occupancy sensors and incorporate it into its machine learning algorithm to detect abnormal use? This is now possible due to advances in technology specifically in machine learning; automated systems are designed to detect normal human activity patterns and adjust the system they are responsible for.

A common scenario is integrating lighting, thermostat, and security systems. The smart lighting and thermostat systems look for human activity patterns and become smarter, keeping people comfortable and rooms lit when there's activity. Now let's say that an irregular event occurs: The phone or power goes down, a vibration or glass-break sensor goes off, a motion or light sensor is tripped,



or a door sensor is activated. In an integrated security design, one sensor can heighten the reaction of the others and communicate to a centralized controller, which can then provide more information to help assess threat credibility.

But even technology-based detection solutions have downsides. In particular, they're not human and, as a result, don't always accurately determine acceptable and unacceptable access. Additionally, sensors can go bad, and wires can fail or corrode—all leading to false alarms. Various suppliers offer connectors with multi-feature benefits that enable them to handle high temperature and high vibration, as well as auditory and mechanical feedback that confirms good connection. Connection mismatch is avoided by the optional keyed connections that eliminate installer issues, and a triple lock insert can safeguard against high vibration and accidental disconnects.

Assessing the Detection

Assessing detection is a critical factor, as this phase determines threat credibility. Until recent years, humans have been at the core of threat assessment, but it has evolved into the comprehensive network of sensors and output that we have today.

However, sensor output in a security system produces a new

challenge: The sheer amount of data produced that needs to be collected and analyzed, as well as potentially stored for further analysis and insights. Historically, data collected from these sensors was rarely kept for any analytics purpose, but today's "smart" or "intelligent" systems use this data to predict and assess threats. Events that are different or out of place are ones we want to detect, but so often it is obscured by irrelevant data. With the help of artificial intelligence (AI), computers can now recognize acceptable or normal use and call attention to just the outliers, thereby only bringing credible threats to the human monitoring the system.

Alerting after Detected Threat

While it may seem that alerting the end user to a threat that's been verified as credible may be straight-forward, yet this phase has its challenges as well. Returning to the scenario: The phone line or power goes down, a vibration or glass-break sensor goes off, a motion or light sensor in your learning thermostat is tripped, and finally a door sensor is activated. At what point did this experience not fit the ordinary course of events? At what point does the alarm alert the user or company? Based on user preference and premises need, the system can alert them at a pre-determined point in the sequence.

Responding to a Threat

The final component of a security solution responding to a confirmed threat. Goals here include preventing furthing loss, recovering goods, and identify the perpetrator. Security systems provide the communication channels needed to respond to confirmed threats—such as automatic calls to 911 and the end user; however, the majority of response comes from security personnel and/or the police. Throughout the response step sensor data is collected, summarized, and relayed, to give the people responding as clear a picture as possible. This allows their responses to be as potent and safe as possible.

Conclusion

Premises security solutions aim to deter, detect, assess, alert, and respond. Premises with the potential for widespread detrimental effects of a security breach are particularly vulnerable, as are ones not secured by onsite security staff or that are otherwise monitored remotely. Deterring, detecting, assessing, alerting, and responding to threats are key functions of physical security solutions. In recent years. we've seen dramatic improvements in all areas of premises security, expecially in the latter four.