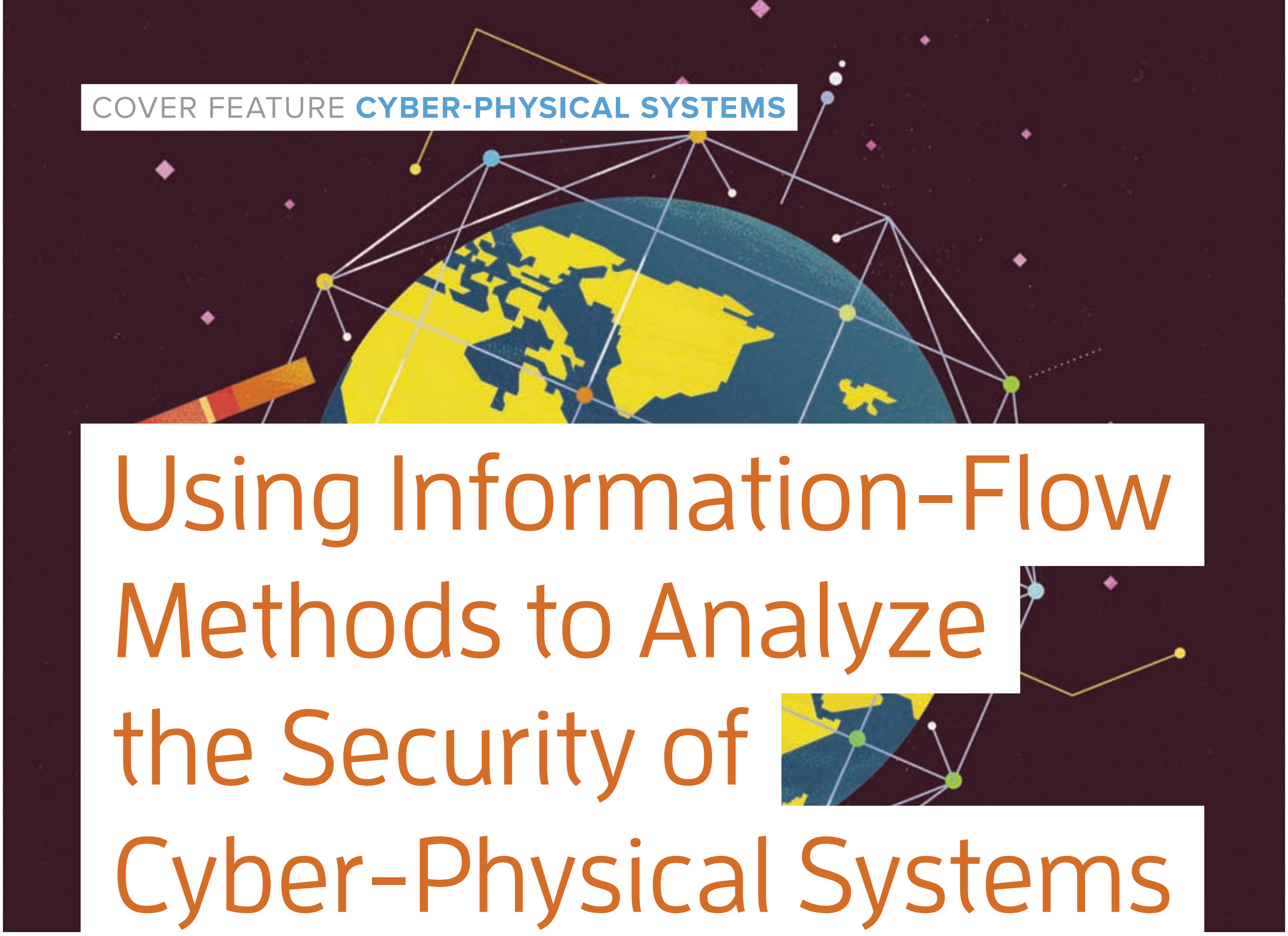
COVER FEATURE **CYBER-PHYSICAL SYSTEMS**



Using Information-Flow Methods to Analyze the Security of Cyber-Physical Systems

Gerry Howser, Kalamazoo College

Bruce McMillin, Missouri University of Science and Technology

Securing information flow is essential to methods that must ensure confidentiality, but information-flow disruption is equally important because it points to an integrity vulnerability. A proposed security model addresses both aspects, accounting for cyber-physical systems' unique confidentiality and integrity vulnerabilities.

Critical infrastructures that comprise computers, embedded devices, networks, and software systems are vital to day-to-day operations in modern life. Cyber-physical systems (CPSs) consisting of embedded computers and communication networks govern both physical manifestations and computations, which greatly affects how these two components interact with each other and the outside world.[1] Myriad complex issues surround CPS specification, design, correctness, stability, reliability, and security. A CPS's combined discrete computational and continuous physical nature compounds these issues—not only because semantics vary widely, but also because boundaries between the cyber and physical realms are blurred, creating a host of new security and privacy vulnerabilities. Critical infrastructure protection must address these issues to ensure the sound operation of CPSs, which is vital to a nation's economic and social stability.

We developed a security model that unites the computational and physical aspects of security, tailored to CPSs' unique confidentiality and integrity vulnerabilities. The Multiple Security Domain Nondeducibility (MSDND) model deals with information-flow security—how information moves among user groups within the security domains (SDs) that make up the system.[2] Information-flow security policies prevent information

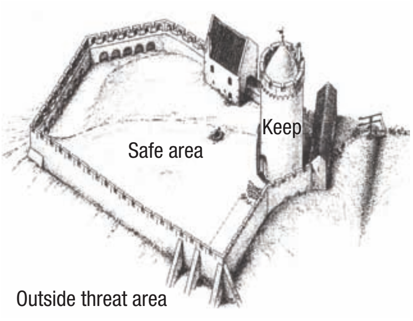


FIGURE 1. A medieval castle's model of security. Security mainly involved building walls to protect assets from outside threats. The most valuable assets were in the keep, with the highest protection level. (Castle image source: Turaida Museum Reserve, used with permission).

from flowing to users who are not authorized to receive it. However, preserving information-flow security in a CPS is challenging because the flow is inexorably intertwined among cyber and physical components. MSDND integrates the information flow contained in the physical components with the information flow contained in the computing components. Thus, it can model the complex SD characteristics of a CPS without definitions of high and low security designations, which other models require.

Security notions are often limited to restricting information flow from a high-security domain to some low-security domain, but information actually flows in both directions. Information flowing to someone who is not supposed to see it is a confidentiality violation. Correct information not flowing to someone who is supposed to see it is an integrity or availability attack. The security community understands information flow's dual nature, but efforts to incorporate that understanding are rare. Our model incorporates that understanding to thwart both confidentiality and integrity attacks on a CPS's cyber and physical parts.

SECURING A CYBER-PHYSICAL SYSTEM

Understanding information-flow security requires first understanding what it means to secure both the cyber and physical parts of a CPS and the information that flows between them.

Physical system security

People have been securing physical assets for centuries, and the basic process has changed little over time. Securing a goat herd, for example, requires accounting for assets (each animal) and controling access to the herd, so the administrator (the goats' owner) implements monitoring—watching the goats by day—and a defensive mechanism—a strong fence to keep the goats in and attackers out at night.

This method of using guards and fences reached its peak with the medieval castle, an example of which is shown in Figure 1. The thrust of security was to build better walls with highly secure areas inside; the attackers' main goal was to build better battering rams. At some point, attackers began to attempt to take the castle by stealth, and modern physical attacks were born.

Cybersystem security

Securing the electronic, computer, and communications systems that form the cyber portion of a CPS is almost as well understood as securing the physical portion. The past three or four decades have seen the creation of an excellent cybersecurity tool suite. The security community knows how to secure a cybersystem and understands the tradeoffs between security and accessibility. Techniques such as user names and passwords, when combined with common sense and encryption, work well to manage the asset use. Messages between cybersystems can be kept private with proper encryption.

Information-flow security

Too many people think that combining physical security and cybersecurity is sufficient to secure a CPS. It is not. A CPS typically leaks information when it operates normally. For example, many modern cars automatically unlock when the correct key fob is present. When this happens, the information that the correct key fob is within range is leaked to all observers: the driver, who obviously knows that the car is his, and any observers in the same SD. If a car is known to be locked and someone enters it without physically unlocking the door, that person is likely to have the correct key fob. If not, the door would remain locked. Thus, the secure information "I have the key" has been leaked. Information has flowed from the secure domain of the car to the unsecured domain of the outside world. In contrast, if the door is not known to be locked, then an observer cannot deduce that the person entering the car has the key because the car might have already been unlocked.

Security domains

The SDs in purely physical systems are controllable. The castle wall defines an unsecured outside from the more secure safe area and the most secure keep. If the walls are breached, the innermost keep could be defended to the last man to keep the lord of the castle and his family safe. To a large extent, the same is true of modern physical assets. Many military installations are walled or fenced in to keep casual observers outside. Even more sensitive areas can be further guarded or secured with strong locks and doors. This model carries over to modern computer systems as well, with different access levels serving as walls and guards. In both cases, SD boundaries are easily defined.

In a CPS, however, SDs can overlap and intrude into each other, creating complex domains, as Figure 2 shows. These patterns do not remotely fit the castle analog. Any physical asset in an SD might be observable, which means that the outside threat area encroaches on the SD. Moreover, system assets might not be contiguous, as in an electrical smart grid or a water-delivery system, or parts of the CPS's control mechanisms might be miles apart but still electronically connected, as in a commuter train or smart traffic-light system. In these cases, connections are purely electronic but might still leak information.

A GENERAL SECURITY METHOD

Because a CPS divides nicely into physical and cyber realms and information flow, securing it can, for the most part, be done in three major steps. The steps, which are really loops, can be expressed fairly simply.

Analyze physical security

The first step in securing any CPS must be physical security. Without physical security there is no point in attempting cybersecurity. As in configuring a firewall (network or physical), it is best to start by eliminating all access so that it is impossible to miss an access path. Once the assets are physically secured, the next step is to analyze each user to determine what physical access they require to use the CPS. Usually only engineering and maintenance staff will require physical access, but there will always be people who think they need to touch the equipment. A rule of thumb is to view anyone with physical access as a physical threat, including security personnel.

Analyze cybersecurity

Many well-understood methods exist to secure cyber assets. For example, analysis might start with an access-control model such as Harrison–Ruzzo–Ullman,[3] SDs such as Bell–LaPadula[4] and Lipner,[5] and encryption. Such models are inherently hierarchial; Bell–LaPadula, in particular, formalizes the view of security partitioned as high and low. However, a strict interpretation of Bell–LaPadula is problematic. For two entities within a system to communicate, they must be at the same security level; if they are not, then one must lower its security level to the other's. Trust models are then required to control whether a high-security entity divulges information to a low-security entity. In a distributed system, essentially all entities must be at the same security level or they cannot communicate with each other. In a CPS, maintaining the same security level is more complex because the system has both physical and cyber parts.

Secure information flow

When a cybersystem controls a physical system, the challenge is to secure the flow of information between the physical and cybersystems' various components. Any control system must be at the same security level as the process it is controlling so that it can communicate with that process. Similarly, the controlled process's security level must be higher than that of the entity that owns it; otherwise, if attacked, the owning entity can corrupt the physical process. For example, an electric utility attacked by a virus could disrupt the electricity flow to customers. Thus, what becomes a key concern is not access but information flow—both its prevention, as in confidentiality and privacy, and assurance, as in integrity.

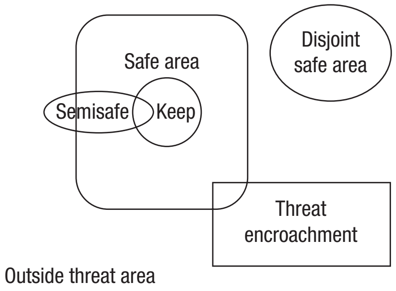


FIGURE 2. Complex security domains (SDs) common in cyber-physical systems (CPSs). The security domains (SDs) correspond to the castle image in Figure 1. In a CPS, SDs can overlap, creating more complex security issues.

Information flow can be understood in terms of models such as noninterference,[6] noninference,[7] nondeducibility (ND),[8] and MSDND.[2] Interestingly, because MSDND is a more general model, if a system is MSDND secure, the other information flow models are no longer a concern.

NONDEDUCIBILITY AND SECURITY

Many critical CPSs are vulnerable to attacks that do not steal or modify information but simply interrupt the flow of control information within the CPS. For example, if messages about the location of airliners are blocked, the air-traffic controller might accidentally direct an airplane into the path of another airplane. If control messages are modified within the

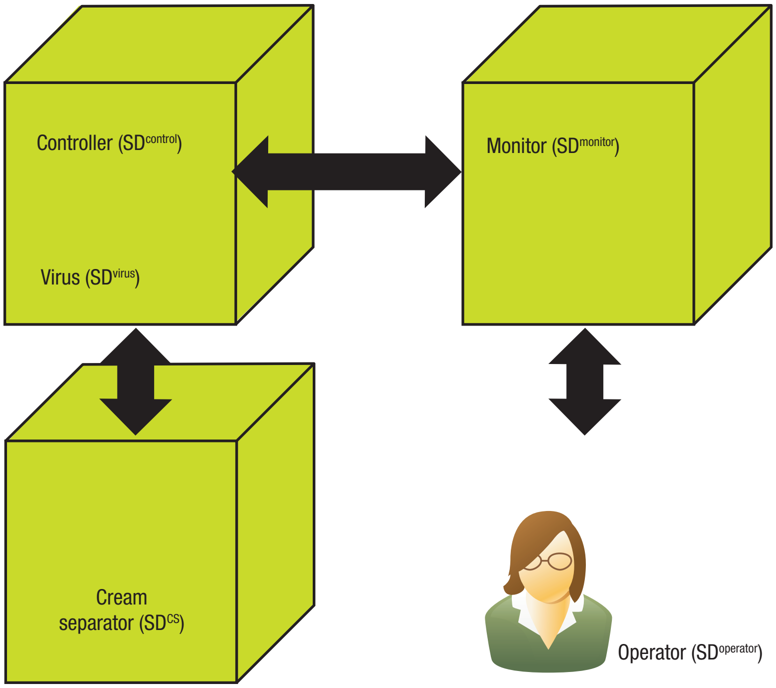


FIGURE 3. Model of the five SDs in an ice cream company showing an attack on its controller in SD$^{\text{control}}$. The company can use Multiple Security Domain Nondeducibility (MSDND) analysis to find where the virus might hide while it is disrupting the information flow among domains. The company can then modify the information flows to make it impossible for the virus to turn MSDND security against the system.

proposed electrical smart grid, power could be misrouted. In other words, a CPS can be destroyed or rendered useless simply by disrupting critical information flow.

The risk of disrupting information flow is high when SDs overlap. We developed MSDND mainly to address scenarios in which a CPS's information flows are difficult to describe with existing models. An example is when states cannot be correctly deduced. If two states are mutually exclusive—that is, only one state can be true so the other must be false—and an agent cannot evaluate either state, the resulting states are MSDND secure. Simply put, if you cannot correctly deduce the state of a true or false variable, then that variable is MSDND secure.

A formal definition of MSDND is defined over a set of CPS state changes as

$$\text{MSDND} = \forall w \in W : w \vdash \Box[s_x \ \mathbf{xor}\ s_y] \wedge [w \vDash (\nexists V_x^i(w) \wedge \nexists V_y^i(w))] \quad (1).$$

In the special case where s_x is φ = T and s_y is ¬φ = T, MSDND reduces to

$$\text{MSDND} = \forall w \in W : [w \vDash (\nexists V_\varphi^i(w))] \quad (2).$$

In the set of state variables $x_i \in X$, each set of states x_i can be viewed as defining a world, $w_i \in W$. The transitions between these worlds are given by functions *wRw′*, where *w*,*w′* ∈ *W* and R is the transition relation between the worlds *w*→*w′*. To know the state of any $x_i \in X$, the model *must* have some valuation function that returns the variable's truth value. These valuations are denoted by $V_i^j(w)$, which returns the truth value of x_i as seen by entity *j*. It is entirely possible that one entity can evaluate x_i and another cannot. For example, a SysAdmin might be able to see that file FRED exists while user Sam might not. Given the logical expression $\varphi = x_i$ is true, the expression must either be true or false in every world. There is no other choice.

Therefore, φ is MSDND secure in world *w* if and only if equation 1 or 2 is true. In other words, if for every possible combination of events an entity cannot evaluate a logical expression, that expression is MSDND secure.

Schrödinger's cat

To better undestand MSDND, consider Schrödinger's cat, a classic thought experiment. A cat is placed in a box with a silent apparatus that can randomly kill the cat or not. The question is posed to two observers, Ike and Tina: Is the cat alive or dead? If the apparatus has gone off, the cat is dead. If the apparatus has not gone off, the cat is alive. No one is allowed to check on the cat, so no one can know its status.

Thus, while the box remains closed, neither observer has any information about the cat's status and neither can correctly deduce if the cat is alive or dead. If φ = the cat is alive, then for both observers φ is obviously MSDND and the cat's status is MSDND secure.

But suppose Tina has secretly attached a heart monitor to the cat and can hear the cat's heartbeat. In this case, the cat's status is MSDND secure for Ike because he cannot evaluate the statement φ = the cat is alive. However, Tina can hear the cat's heartbeat and definitely knows the cat is alive. Therefore, φ is MSDND secure for Ike but not for Tina.

PRESERVING INFORMATION-FLOW INTEGRITY

Two examples illustrate how our MSDND model differs from other methods in preserving a CPS's integrity.

Cream separator

Consider a cyber-physical process within a high-quality ice cream

TABLE 1. Security domains (SDs) in a cream separator's process control system.

System component	Domain	Valuation
Cream separator	SD^{CS}	V^{CS}
Virus*	SD^{virus}	V^{virus}
Controller	$SD^{control}$	$V^{control}$
Monitor	$SD^{monitor}$	$V^{monitor}$
Human operator	$SD^{operator}$	$V^{operator}$

*Although it is not a system component per se, the virus is hiding in a component: the I/O buffer between the cream separator and controller.

company that has discovered the exact butter fat content to make perfect ice cream. Too much butter fat and the ice cream does not have enough taste; too little and the ice cream does not taste smooth and rich. The process is rigidly controlled by a centrifuge connected to a programmable logic controller (PLC).

Unknown to the ice cream company, a rival dairy has inserted a virus into the controller for the cream separator, as shown in Figure 3. The monitor will send an alert if the cream separator is not functioning properly, but the virus intercepts all messages to and from the controller. Regardless of the separator's actual speed, the virus reports that all is well. Because it never sends information to the rival dairy, the virus is particularly hard to detect.

Traditional methods will not spot the virus, but an MSDND analysis will. At the actual separator, the speed is *not* MSDND secure because the sensors on the separator correctly read the speed. However, when the separator reports the speed reading to the controller, the virus intercepts the reading and reports to the controller that the separator is operating at the correct speed regardless of the actual speed. Likewise, the virus intercepts any messages from the controller to speed up or slow down the separator. Eventually the separator will spin at the wrong speed and the ice cream produced will be of poor quality. In essence, the virus is hiding behind MSDND.

The infected system can be divided into five separate SDs, as defined in Table 1. In this model, not only are there multiple SDs, but the notion of high and low security is not relevant. Setting φ equal to "cream is being separated properly" and given normal conditions, the controller and monitor system will adjust the separator to ensure that φ is true.

If the system is operating correctly (no virus), all operations by the controller are successfully carried out and reported back to the monitor, deducibly. In other words, every action in a domain is uniquely identifiable in another domain; the cream separation process is reported correctly to the operator, operator comments are carried out by the controller to the cream separator, and so on.

The key to demonstrating the integrity of such a system is to show that the desired information flow cannot be disrupted. An analysis showing that the system is not MSDND secure proves that each observation or command can be uniquely attributed to its corresponding command and observation. The first step is to establish that the cream separator correctly reports its status and dutifully follows commands sent to it from within the controller. Thus, an MSDND analysis will show that the cream's status is not MSDND secure at the cream separator; it will seek to determine *why* the physical cream separator correctly reports the cream's status.

This determination involves two propositions.

Proposition 1. The first proposition is, "The cream status is not MSDND secure at the cream separator." It is obvious that (φ **xor** ¬φ) = true, so the first condition for MSDND is met by definition. However, the separator directly measures the cream and therefore both $V^{CS}_{\varphi}(w)$ and $V^{CS}_{\neg\varphi}(w)$ are correctly evaluated for any *w*, and the conditions for MSDND are not met. That is,

$$\text{notMSDND} = \forall w \in W : w \vdash [\varphi \mathbf{xor} \neg\varphi] \wedge [w \vDash \left(\exists V^{CS}_{\varphi}(w) \wedge \exists V^{CS}_{\neg\varphi}(w)\right)].$$

To maximize its disruption, the virus can completely block the information flow from the cream separator to the controller. However, this case is easily detected by examing the timeouts on the readings from the separator. The more insidious case is where the virus fabricates readings to create false information flow. These fabricated readings cause an observation at the monitor to be consistent with multiple possibilities within the physical system, essentially making the system nondeducible from a human operator's perspective.

Proposition 2. The second proposition is, **"**If the system is MSDND secure for $SD^{control}$, then any entity *i* within $SD^{control}$, $SD^{monitor}$, and $SD^{operator}$ will believe all is well." Obviously (φ **xor** ¬φ) = true, so the first condition for MSDND is met. If φ cannot be correctly evaluated in $SD^{control}$, then both conditions are met. The virus always returns to $SD^{control}_{\varphi}$ = true, so regardless of the cream separator's status, the infected system reports to

TABLE 2. SDs in a drive-by-wire car.

System component	Domain	Valuation
Car's electronic control system	SD^{car}	V^{car}
Corporation network	SD^{corp}	V^{corp}
Driver	SD^{driver}	V^{driver}
Traction control (TC) system	SD^{TC}	V^{TC}

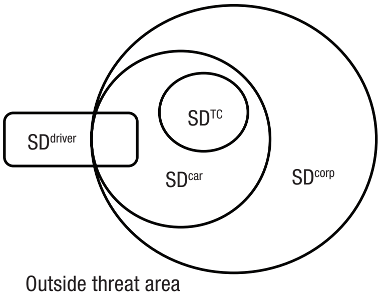


FIGURE 4. SDs in a drive-by-wire car. When a network controls the car, there must be some way to know how the car will respond in various modes. An MSDND analysis will reveal when the traction control (TC) system and not an attacker is in control.

$SD^{control}$ that all is well. Any entity in $SD^{control}$ will report the same message and so on up to $SD^{operator}$. No matter what, the trusting human operator will suspect nothing, the separator will spin at the wrong speed, and the cream will be ruined.

But what if the operator is not so trusting and elects to periodically check the separator's speed gauge? Doing so will instantly reveal that something is wrong, and the MSDND security of the virus attack would be broken. So analyzing the information flow to find MSDND-secure flows would save the ice cream company, as long as measures are taken to ensure that all information flows are not MSDND secure. The odd result is that breaking the attack's security effectively spoils that attack. In other words, information-flow analysis has made it impossible for the attacker to use security against the company.

Obviously, the cream separator example is a thinly veiled description of how the Stuxnet virus operated.[9]

Drive-by-wire car

Complex security domains are evident in a drive-by-wire car equipped with remote assistance such as OnStar or Toyota Connect. The model consists of a corporation that provides remote assistance to drivers such as navigation, remote unlock, and remote shutdown, and an automobile with on-board drive-by-wire functionality and an advanced form of traction control (TC). We assume the car has three operational modes: normal, hazardous road conditions, and corporate remote. The challenge for security is to understand which commands the car is responding to when it is in a particular mode.

Normal operations. The driver can operate the vehicle as normally expected. The driver knows that he controls the car.

Hazardous road conditions. Most current automobiles are equipped with varying degrees of TC systems to automatically correct for a loss of traction. TC systems can be viewed as a superset of the automatic braking system (ABS). While a TC system operates, the car will attempt to correct a skid and counter anything the driver does that would make the skid worse.

Corporate remote operations. If the car is equipped with a service similar to OnStar, the corporation can issue commands to the car, and the driver must trust the corporation to act in his best interests.[2,10] Many recent television commercials have touted the benefits of accessing your car through the Internet or corporate connections, but is this really an advantage? Admittedly, it is great fun to lock and unlock your car from a cellphone,[11] but what happens when your cellphone is hacked or lost? What if the corporation or network is hacked and the hacker decides to simply power off as many cars as possible?[12]

Obviously, the car will only respond to one set of commands from either the driver, TC system, or corporate network. Depending on the car's current mode, the driver might not be able to distinguish who or what is actually in control. Table 2 and Figure 4 show the SDs in a drive-by-wire car. Remote operation by the corporation exists in one SD, and the driver operation exists in another. What the driver can and cannot ascertain is governed by the information flow that exists among both cyber and physical domains. Classical models of information flow and deducibility break down in this cyber-physical environment.

For example, if the TC system takes control of the car, the driver notices a complete lack of response to driver commands. Athough this control wresting might initally seem disconcerting, the TC system has more timely knowledge than the driver about road conditions and any lack of traction by the tires. Because of this, the TC system can react more accurately than the driver.

But what if someone uses the network to take control of the car? The car will now respond only to the network commands.

Two questions arise: Who is in control of the car, and can the driver correctly deduce who is in control? Obviously, the car will respond to commands from only one source at a time. The highest command level is the corporation network, the TC module, or the driver. If the commands come from the driver, all is well and the driver can correctly deduce who is in control. However, the driver sees the same loss of control and unexpected actions when either the TC system or the corporation is in control, but cannot determine which one actually is.

This situation is exactly what is required to show that the car's control is MSDND secure from the driver. In this case, MSDND has been turned against the driver to hide a possible attack. Not only that, but because a TC system failure would behave in the same way, the attack's source is also MSDND secure by the same reasoning. In a CPS, it is highly probable that security tools and methods can be turned against the system. When attacking a CPS, disrupting the normal flow of commands is often enough to damage or destroy that system.[2,10]

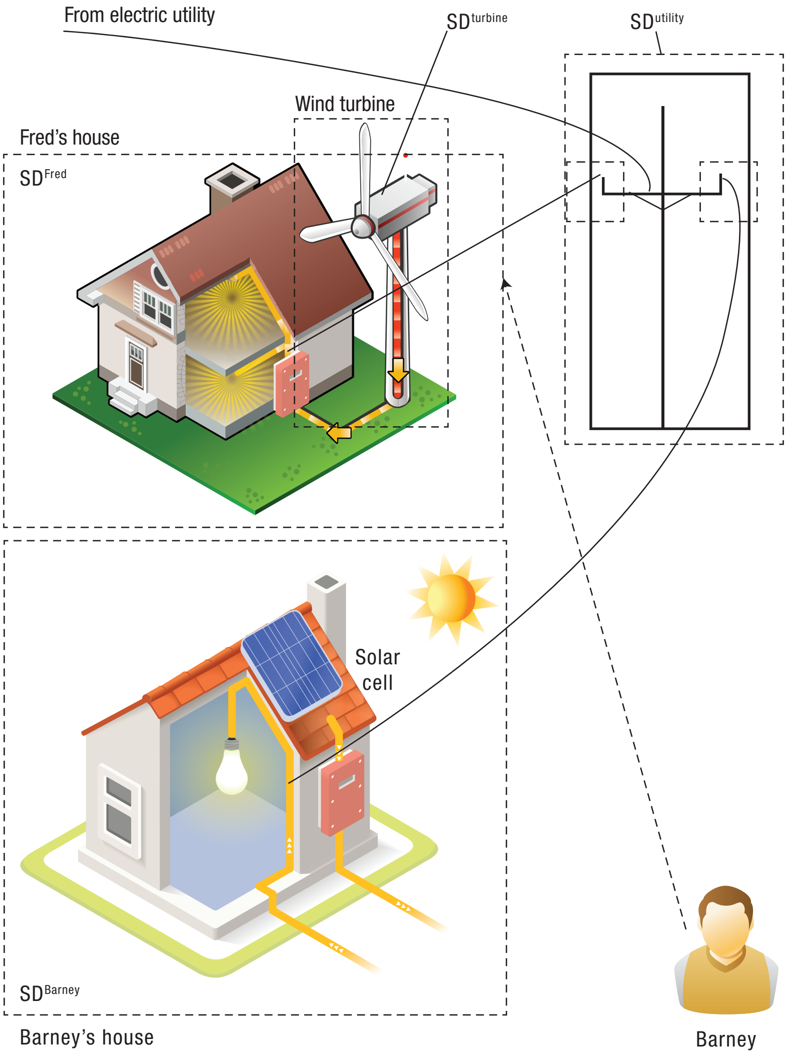


FIGURE 5. Multiple SDs in a simplified smart grid. Two neighbors, Fred and Barney, agree to purchase a renewable resource each and then share the profit from selling their power output to the utility. But eventually, Fred breaks the agreement and begins secretly siphoning power from his wind turbine into a battery inside his house, which Barney cannot see. To discover Fred's deception, Barney can use an MSDND analysis.

ASSESSING CONFIDENTIALITY AND PRIVACY EFFECTS

In addition to preserving integrity, our MSDND model can be applied when privacy and confidentiality issues prohibit the evaluation of certain information such as electrical use and meter readings in the future smart electrical grid. Consider the case shown in Figure 5, in which two neighbors, Fred and Barney, agree to each purchase a renewable resource and then share their power output.

Fred purchases a wind turbine and Barney purchases a solar panel. When the sun shines, both use Barney's power, and when the wind blows, they use Fred's power. They agree to sell back any excess power to the utility company and share the profit. Their simple smart grid operates well, and for a while both are satisfied and enjoy reduced energy costs.

Eventually, however, Fred no longer wants to share the profits, so he buys a battery, installs it in his house, and changes his system's operation. When the wind blows, he sends the excess to his battery. Later, when it is economically profitable, he sells

TABLE 3. SDs in a simplified smart grid.

System component	Domain	Valuation
Fred's house	SD^{Fred}	V^{Fred}
Fred's meter	SD^{Fmeter}	V^{Fmeter}
Fred's wind turbine	$SD^{turbine}$	$V^{turbine}$
Barney's house	SD^{Barney}	V^{Barney}
Barney's meter	SD^{Bmeter}	V^{Bmeter}
Utility company's meter	$SD^{utility}$	$V^{utility}$

his stored energy back to the utility. Because Barney cannot monitor the power use inside Fred's house and doesn't know about the battery, Fred's dishonesty can be completely hidden as long as he carefully times when he sells power back to the utility. If Fred is smart and sells back only when the turbine is spinning and Barney does not need extra power, Fred can hide his activities because Fred's power use is MSDND secure. Fred has turned cyber-physical security against Barney.

So what are the implications of Fred's handiwork? Obviously this situation is complex, but it can be simplified by combining some of the nested SDs shown in Figure 5, which are defined in Table 3. Because of privacy concerns, the load in each house cannot be directly examined, so domains inside the house are not relevant, including that for Fred's battery because no one other than Fred knows it exists.

An MSDND analysis involves three propositions—all of which hinge on the idea that CPSs typically have components that leak information simply because they cannot be hidden. Observations of the physical system are key in many attacks, so hiding as much physical activity as possible is critical to CPS security.

Proposition 1. The first proposition is, "When Fred is storing turbine power, the existence of the battery is MSDND secure from Barney." As a proof, let ψ = "The turbine is turning," φ = "Fred is using all the power from the turbine," and $\neg\varphi$ = "Fred is *not* using all the power from the turbine but is storing power."

Barney can see the turbine spinning, so ψ = true. The power is going somewhere inside of Fred's house so $\varphi\mathbf{xor}\neg\varphi$, which meets the first condition for MSDND. But because of privacy issues, Barney cannot observe the load in Fred's house to know if Fred is using all the turbine's power, so $\nexists V_{\varphi}^{\text{Barney}}$ and φ is MSDND secure from Barney. That is, Barney cannot correctly deduce the existence or nonexistence of Fred's battery.

However, Barney becomes suspicious enough to sneak over and monitor Fred's power transfer to the utility over Fred's power line into his house. Barney also monitors his own power line and observes that he is drawing power from the utility and that Fred is not providing power but the wind turbine is spinning. Because he does not know about the battery, Barney believes Fred's power transfer is consistent with Fred's using all the power he generates. Barney cannot correctly determine whether φ is true or false. Thus, φ is MSDND secure and Fred's actions are completely hidden from Barney.

Proposition 2. The second proposition is, "Whether Fred is using all the power from the turbine or storing power is MSDND secure from Barney when Fred is not sending power to the utility." The proof of this proposition follows directly from the proof for proposition 1.

However, Fred monitors Barney's power, and when Barney is not drawing power from the grid, Fred discharges his battery to the electric utility, making a profit for himself. Barney's power use is not MSDND secure from Fred because Barney is not cheating. Suppose, however, that Barney observes his own power from the utility, Fred's power to the utility, and the spinning of the wind turbine. If the wind turbine is not spinning, information flows from Fred to Barney, and Barney can deduce $\neg\varphi$ and know that his neighbor is being dishonest.

Proposition 3. The third proposition is, "If the turbine is not spinning and Fred is selling power to the utility, then the existence of a battery in Fred's house is *not* MSDND secure from Barney." As a proof, let $\neg\psi$ = "The turbine is not turning," φ = "Fred is using all the power from the turbine," and $\neg\varphi$ = "Fred is not using all the power from the turbine but is storing power."

If the turbine is not turning and Fred is selling power, then ¬ φ = true because Fred is selling power from somewhere. So by observing the turbine, Fred's meter, and the utilities meter, Barney can determine that Fred has electrical storage in his house and is cheating Barney out of his share of profits. The existence of the battery is *not* MSDND secure under these conditions.

WHY PERFORM A NONDEDUCIBILITY ANALYSIS?

We have shown that MSDND has uses in shielding activities from outside observers as in Schrödinger's cat and Fred and Barney's houses. This very powerful tool preserves the confidentiality aspects of security and should be used whenever confidentiality is appropriate in a CPS. The drive-by-wire car and cream separator examples showed how an attacker can hide behind ND. In both cases, CPS security can be ensured and enhanced by breaking all known ND-secure information flows—not a trivial undertaking.

Early threat detection

If all information flows are deducible, then an attack will be detected sooner. In many cases the attack could even be foiled, as in the cream separator. The company could have the controller handle any small speed perturbations but also equip the cream separator with a physical alarm (such as a loud horn) that would trigger when the separator's speed was critically out of normal range. If the system becomes infected by a virus or is simply faulty and the perturbations reached the point of causing the cream to be ruined, the alarm would sound and the operator could shut down operations until the problem was corrected. At worst, the damage would be minimized.

Tracing the threat to its source

Although we did not directly address this idea, if all information flows are deducible—*not* nondeducibly secure—the attack or failure can easily be traced to its source. Studies show that an attack can be detectable but its source can be nondeducible.[13,14] This scenario is better than not detecting the attack, but correcting the attack without knowing the exact source is problematic. Besides, no one wants to let the attacker get away to try again.

We have described some of the serious challenges that CPSs present to security professionals. If the CPS is viewed only as a set of physical things controlled by secure electronics, an adversary could easily observe the system's physical actions and deduce the secure actions that must be hidden to protect the CPS. The complex interrelations of the CPS's two sides (cyber and physical) produce ample opportunities for information flows that can be used as an attack vector. Because of the collateral damage when a CPS is not properly controlled, any attack can lead to property damage or loss of life.

Physical and traditional cybersecurity must be in place and must be the best possible. Without physical security there is no point in cybersecurity, and without cybersecurity the CPS is not secure. However, because the two sides of every CPS are inextricably intertwined, physical and cybersecurity are not enough. The system can be observed, which leaks information that flows between the system's cyber and physical assets. To secure them, all information flows must be examined and made deducible to the organization.

ABOUT THE AUTHORS

GERRY HOWSER is an assistant professor of computer science at Kalamazoo College. His research interests include cyber-physical security, cyber-physical systems (CPSs), and the application of modal logic to computing. Howser received a PhD in computer science from the Missouri University of Science and Technology. He is a member of the IEEE Computer Society and ACM. Contact him at ghowser@kzoo.edu or research@gerryhowser.com.

BRUCE MCMILLIN is a professor of computer science and associate dean of engineering and computing at the Missouri University of Science and Technology. His research interests include cyber-physical security, CPSs, and smart living. McMillin received a PhD in computer science from Michigan State University. He is a member of the IEEE Computer Society's Board of Governors. Contact him at ff@mst.edu.

MSDND is a powerful tool to model the subtle information disruption that prevents CPS information from reaching its target intact and can easily be applied to find attack vectors that are MSDND secure. The CPS can then be modified to reduce or eliminate those attacks—an essential step in the race to secure the system before it is attacked.

REFERENCES

1. E.A. Lee, “Cyber-Physical Systems—Are Computing Foundations Adequate?” *NSF Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, vol. 2, 2006; ptolemy.eecs.berkeley.edu/publications/papers/06/CPSPositionPaper.
2. G. Howser and B.M. McMillin, “A Multiple Security Domain Model of a Drive-by-Wire System,” *Proc. 37th IEEE Computer Software and Applications Conf.* (COMPSAC 13), 2013, pp. 369–374.
3. M.A. Harrison, W.L. Ruzzo, and J.D. Ullman, “Protection in Operating Systems,” *Comm. ACM*, vol. 19, no. 8, 1976, pp. 461–471.
4. D. Bell and L.J. LaPadula, *Computer Security Model: Unified Exposition and MULTICS Interpretation*, tech. report ESD-TR-75-306, NIST, 1976; csrc.nist.gov/publications/history/bell76.pdf.
5. S.B. Lipner, “Non-Discretionery Controls for Commercial Applications,” *Proc. IEEE Symp. Security and Privacy (SP 82)*, 1982, pp. 184–194.
6. J.A. Goguen and J. Meseguer, “Security Policies and Security Models,” *Proc. IEEE Symp. Security and Privacy (SP 82)*, 1982, pp. 11–20.
7. J. McLean, “Security Models and Information Flow,” *Proc. IEEE CS Symp. Research in Security and Privacy*, 1990, pp. 180–187.
8. D. Sutherland,”A Model of Information,” *Proc. 9th Nat’l Computer Security Conf.* (NCSC 86), 1986, pp. 175–183.
9. G. Howser and B.M. McMillin, “A Modal Model of Stuxnet Attacks on Cyberphysical Systems: A Matter of Trust,” *Proc. 8th Int’l Conf. Software Security and Reliability* (SERE 14), 2014, pp. 225–234.
10. G. Howser and B. McMillin, “Modeling and Reasoning about the Security of Drive-by-Wire Automobile Systems,” *Int’l J. Critical Infrastructure Protection*, vol. 5, no. 3, 2012, pp. 127–134.
11. C. Woodyard, “Start, Unlock, or Honk Horn of Your GM Car from a Cellphone,” *USA Today*, 22 Jul. 2010; content.usatoday.com/communities/driveon/post/2010/07/start-unlock-or-honk-horn-of-your-gm-car-from-a-smart-phone/1#.WBEhvC0rK70.
12. K. Poulsen, “Hacker Disables More than 100 Cars Remotely,” *Wired*, 17 Mar. 2010; www.wired.com/threatlevel/2010/03/hacker-bricks-cars.
13. T. Roth and B.M. McMillin, “Breaking Nondeducible Attacks on the Smart Grid,” *Proc. 7th Int’l Workshop Critical Information Infrastructures Security* (CRITIS 12), 2012, pp. 80–91.
14. T. Roth and B.M. McMillin, “Physical Attestation of Cyber Processes in the Smart Grid,” *Critical Information Infrastructures Security*, vol. 8328, 2013, pp. 96–107.