# Computing Systems and Security

## Literature Review

**SUGGESTED CITATION:** Computer Science Teachers Association. (2026). *Computing systems and security: Literature review.* https://csteachers.org/k12standards/research



 

We conducted a literature review aimed at synthesizing research on learning progressions for computing systems and for networks and the Internet. Using terms related to the subconcepts of the existing Networks & the Internet and Computing Systems topics (e.g., *computing devices*, *cybersecurity*, *networks*, *hardware*, *physical computing*) and *K–12* as keywords, we identified 143 potential articles to include. After examining the resulting titles and abstracts, we narrowed the list to 47 papers that seemed most pertinent to this review. Many articles reported on specific interventions, so we concentrated on seven papers that provided information on common student conceptions and misconceptions; student ideas, segmented by age or grade level; and differences in strategies used by more and less successful learners.

## Hardware and Software (Including Troubleshooting)

### What topics should be covered?

Przybylla and Grillenberger (2021) conducted a qualitative content analysis of textbooks, Computer Science education research articles, and course materials to identify the key concepts for teaching embedded systems and hardware and software. Their work was motivated by (a) K–12 students' interest in physical computing, (b) the importance placed on these topics in learning standards and frameworks across countries, and (c) many teachers' lack of specialized knowledge in this domain.

They synthesized their findings into a model that identifies a set of core technologies (i.e., applications of hardware and software design), practices (i.e., commonly used methods in hardware and software design), design principles (i.e., goals of the hardware and software design process), and mechanics (i.e., technical features of embedded systems) (see Figure 1 and Table 1).

Notes about the model:

- This model is an initial step in defining competencies for hardware and software topics. It needs to be validated by experts.
- The list of core technologies may change over time as new technologies are developed and current technologies become outdated.

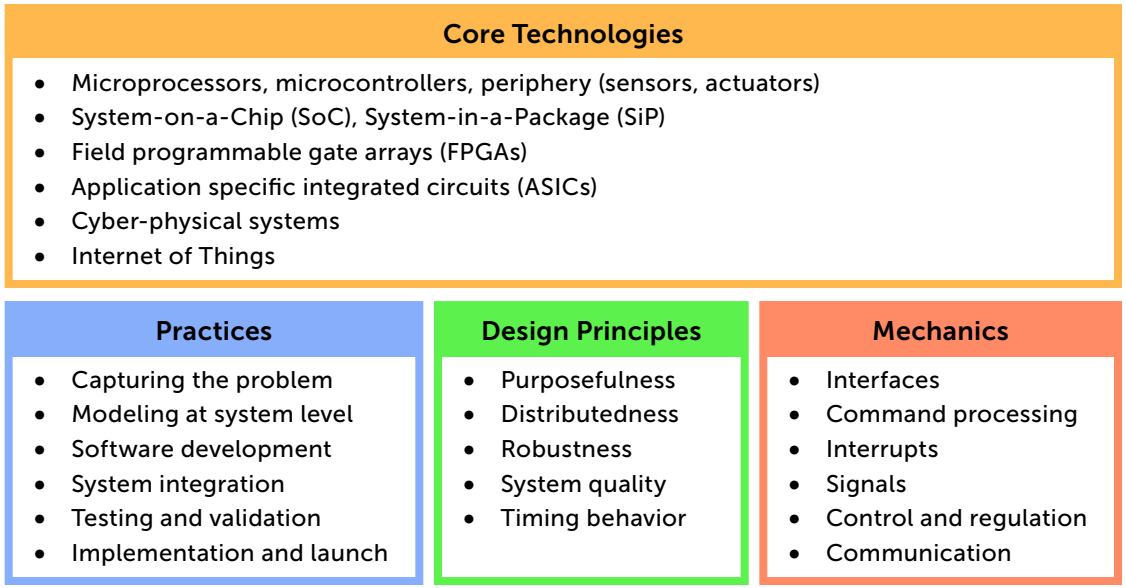


**Figure 1.** *Przybylla and Grillenberger's (2021) Model of Key Concepts of Embedded Systems and Hardware/Software Co-Design*

 

**Table 1.** *Definitions of Concepts in Przybylla and Grillenberger's (2021) Model*

| Model Component | Key Concepts |
|---|---|
| **Practices** | **Capturing the problem:** Identification and description of a problem, determination of goals<br>**Modeling at system level:** Includes the selection of suitable hardware components, interfaces, software to control the whole system<br>**Software development:** Programming sketches and functions for controlling individual system components<br>**System integration:** Assembling the hardware and software components, designing the housing if necessary<br>**Testing and validation:** Checking compliance with the requirements, simulation, debugging<br>**Implementation and launch:** Integration of the system into its runtime environment, networks, etc. |
| **Design Principles** | **Purposefulness:** Hardware/software systems are designed and optimized for a specific application<br>**Distributedness:** Demonstrated by modularity, parallelization, multitasking, interrupts<br>**Robustness:** Model errors have only minimal impact on stability and performance; systems are tamper resistant and not prone to failure due to disturbance rejection<br>**System quality:** Performance, reliability, accuracy, energy efficiency, resource consumption<br>**Timing behavior:** Real-time capability, timeliness |
| **Mechanics** | **Interfaces:** For realizing inputs and outputs with sensors and actuators<br>**Command processing:** Refers to commands, pipelining, and scheduling<br>**Interrupts:** A means of dealing with concurrency in distributed systems<br>**Signals:** Signals that can be transmitted, described (e.g., fall/rise time, analog/digital), and processed (e.g., discrete/event-driven/timed, sampling, pulse-width-modulation [PWM], filter)<br>**Control and regulation:** To meet system requirements and fulfill tasks<br>**Communication:** Communication via buses and networks is related to signal transmissions and timing behavior |

### Students' understanding of troubleshooting systems that involve hardware and software

The debugging process typically involves four phases: fault isolation, hypothesis generation, solution generation, and testing and verification. Strategies employed during debugging may include forward or backward reasoning to isolate problems, creating hypotheses, generating solutions, and verifying or testing these hypotheses or solutions. It has been observed that pairs of students using both forward reasoning (e.g., searching for bugs while making sense of the code) and backward reasoning (e.g., searching for bugs after observing incorrect behavior) were more effective in identifying issues, providing them with more opportunities for verification and testing.

Research, mostly at the secondary level, has identified several challenges learners face when debugging computing systems:

- Forming a comprehensive view of the systems they work with
- Generating initial theories and conceiving of multiple explanations for the root of an issue
- Coordinating the various representations of physical computing systems (e.g., the artifact itself, paper-based representations, code)
- "Tinkering," or making unsystematic changes that lead to even more bugs
- Pinpointing the wrong locations for bugs (for instance, pinpointing bugs solely in the hardware)
- Focusing solely on finding issues in one part of a system (e.g., circuitry) while overlooking mistakes in another part (e.g., the code)
- Erroneously concluding that the bugs have been fixed
- Ignoring logical issues within the code and assuming code without syntax errors is free of bugs

Over time, students improved in defining coding bugs, transitioning from undefined to semi-defined to well-defined issues. More successful debugging involved a variety of strategies; less successful debugging concentrated solely on isolating a problem or hypothesizing a cause.

**[Students' debugging strategies] may include forward or backward reasoning to isolate problems, creating hypotheses, generating solutions, and verifying or testing these hypotheses or solutions.**

## Networks

Brom et al. (2023) conducted a systematic review of 27 studies from the past 20 years to identify and organize the ideas that children ages 3 to 15 hold about Internet-enabled technologies. They identified 60 themes (see Figure 2) related to (a) general perceptions of the Internet; (b) Internet infrastructure; (c) data flow, or how data moves across the Internet and is stored; and (d) contradictions in a child's understanding. While the review focused specifically on the Internet, the identified themes have applicability to the broader topic of networks.

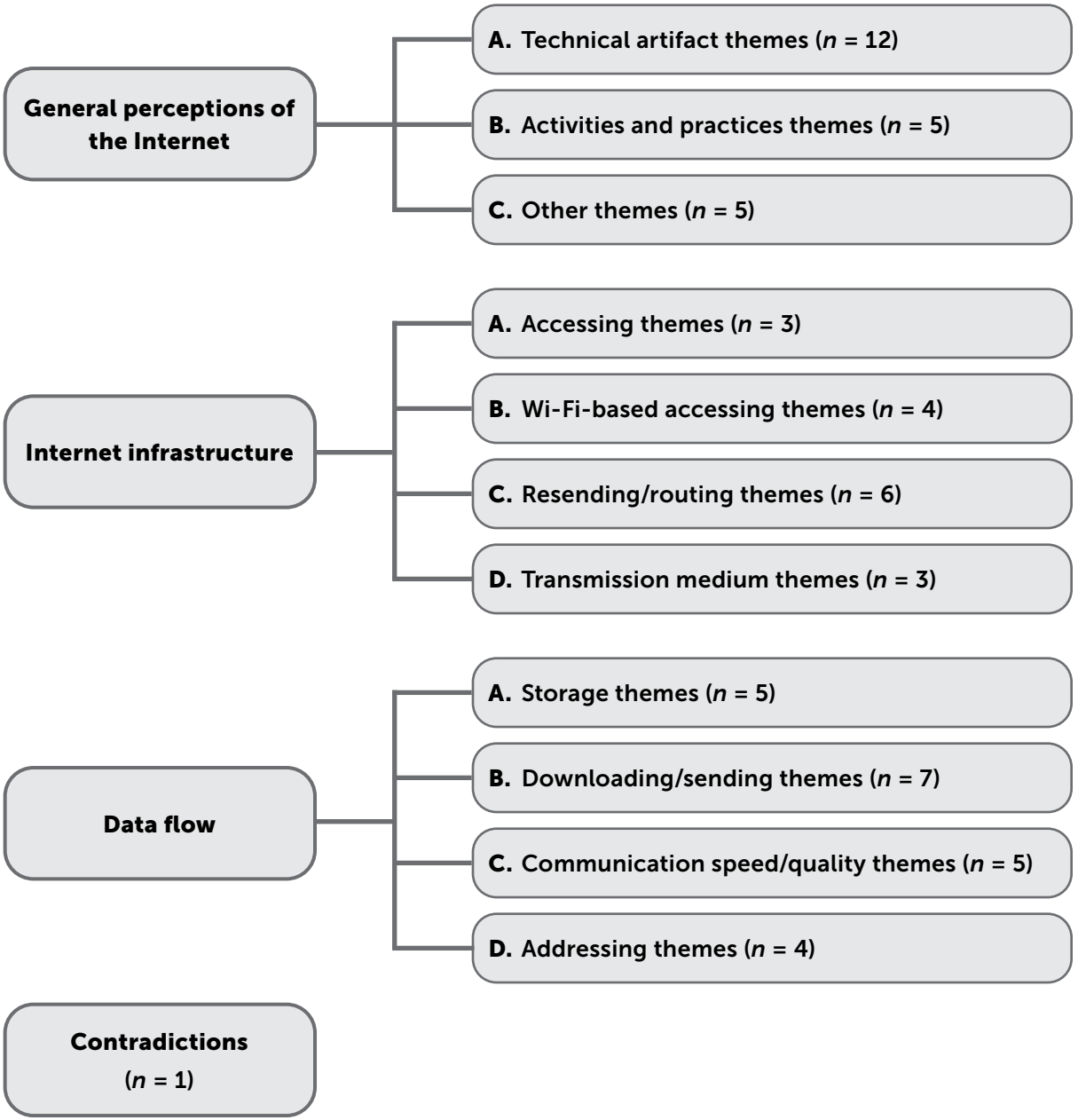


**Figure 2.** *Brom et al.'s (2023) Categorization of Student Ideas about the Internet*

 

### Students' general perceptions of the Internet

Children's understanding of the Internet varies with age. Younger children (ages 3–9) often conflate the Internet with the devices that access it. This confusion becomes less common, but does not entirely vanish, as children get older. Some children, particularly in younger groups, (a) believe the Internet lives within devices and (b) have difficulty distinguishing online and offline activities (e.g., they think that viewing videos through streaming services is an offline activity). However, some 5-to-7-year-olds are aware that devices need to connect to something external to enable Internet functionalities.

Some children conceive of the Internet as a specific application (e.g., a search app). Some children possess a proto-network understanding of the Internet, viewing the Internet as an entity linking devices through a central point or without a clear internal structure. Older children's (10–15 years) ideas are more complex. They perceive the Internet as a large, global network with a complex, but not fully understood, architecture. A correct and detailed understanding of the Internet as a network of networks with specific servers and connections is rare and typically only seen in children over the age of 10 or 11.

***The Internet's functions.*** Children ages 3–12 often see the Internet not as a technical artifact, but rather as Internet-related activities (like gaming or shopping), a source of information or content (such as videos), or a communication medium. Some children under 6 or 7 years of age are familiar with Internet-related activities but do not know the term *Internet*, or know the word but can't connect it to the activities it enables.

**A correct and detailed understanding of the Internet as a network of networks with specific servers and connections is rare and typically only seen in children over the age of 10 or 11.**

## Students' ideas about Internet infrastructure

Young children have some awareness of different Internet access methods, such as cables, mobile hotspots, and cellular data. However, this knowledge is relatively uncommon and is likely influenced by personal experience and the Internet infrastructure children are exposed to. This understanding improves slightly as children get older, but comprehensive knowledge in this area remains inconsistent. Many children around the age of 5 know that Wi-Fi is related to Internet access. By the time they are approximately 7 years old, they are often aware of the physical presence of a Wi-Fi router in their homes.

***Internet and connectivity.*** Despite some knowledge of hardware, confusion about the Internet and its connectivity persists across all age groups. For example, by grades 5 and 9, only a few students differentiate between Wi-Fi and the Internet. Even fewer understand that devices can connect to each other via Wi-Fi without an Internet connection.

***Data transmission.*** Some children above grade 5 hold a simplistic view of direct transmission of data over the Internet. They believe that end-user devices communicate directly with each other. The concept of a modular network with various networking devices (e.g., routers, switches, base transceiver stations) that function as data crossroads is nearly absent in children's understanding. Instead, more common, but inaccurate, single-hop conceptions exist, where data is thought to travel to a satellite, a central computer, or a tower before reaching the target device.

***Medium of transmission.*** Older children, about 9 years and above, often believe that signals serve as the main medium of transmission. However, students mention cables, the more typical medium in modern times, less frequently. There is little data on whether adolescents are aware of the existence of undersea cables, which are a key part of global Internet infrastructure.

## Students' ideas about data flow

The concept of distributed data storage on numerous servers is understood by adolescence. This knowledge emerges as early as grade 5 but is typically seen as more advanced. Common misconceptions about data storage include the belief that it is centralized in one main computer and, less commonly, the belief that it is held in satellites or only within user devices. More outdated or rare misconceptions include storage within a modem or in clouds in the sky. Studies focusing on very young children's understanding are rare.

***Downloading and sending data.*** There is strong evidence that many children ages 3–9 have a basic understanding of downloading apps to a phone, termed *pre-downloading*. Little research has examined older children's concept of downloading, but researchers have noted that many 12-to-16-year-olds know how to save a photo found online. Misconceptions about streaming include thinking that videos are returned to the server after watching or that they are played directly from the server. Also, there is a split belief that data is either sent in packets or in one complete part.

***Communication speed and quality.*** Some 5-year-olds understand that proximity to an access point affects Internet connection. Children above grade 4 or 5 have various correct or partially correct theories about factors influencing connection speed, such as cable thickness and data size affecting transmission speed and the idea that longer distances result in longer transmission times. However, there are also incorrect beliefs, such as a central computer causing a bottleneck.

***Addressing.*** About half of the children in grade 5 and older vaguely understand that computers have some form of an address, which could relate to an IP address concept. The idea of addressing through a phone number is strong among children in grade 5 and older, potentially influenced by the use of smartphones and messaging apps. Few children attribute additional correct features to these addresses, such as uniqueness.

 

## Cybersecurity

### Current approaches to teaching cybersecurity

Cybersecurity is incorporated to varying extents in K–12 CS education. The K–12 Computer Science Framework emphasizes privacy and security, with cybersecurity falling under the Networks & the Internet concept area. Curricula should cover risks to information security, particularly in networked systems, and protection methods and their implications. In the UK, privacy protection is introduced from early education, with more advanced stages covering a wider range of online identity and privacy protection strategies as well as future technologies. The German association for computer science encourages a focus on secure data privacy, authenticity, and integrity, and assessing networked systems for privacy and safety. At higher levels, students create specific security measures. These CS education frameworks aim for students to understand cybersecurity and not simply to focus on using security technologies. Cybersecurity is sometimes introduced in mathematics education, but with more focus on basic encryption methods and less focus on practical applications.

### Students' understanding of cybersecurity topics

Many students have gaps in their understanding of networked communication (see the Networks section), which is vital for forming accurate mental models about cybersecurity. Some students lack comprehensive understanding of key concepts: encryption mechanisms, data transport over the Internet, and potential security threats. Lindmeier and Mühling (2020) identified correct and incorrect ideas that high school students hold about cybersecurity (see Table 2). The authors noted a tendency for cybersecurity lessons to focus on outdated symmetric encryption methods (e.g., Caesar cipher). They recommended students be taught about technologies like HTTPS and the TLS protocol, which underpin data confidentiality, integrity protection, authentication, and trust online.

**Table 2.** *Students' Ideas About Cybersecurity (Lindmeier & Mühling, 2020)*

| Incorrect Ideas About Cybersecurity | Correct Ideas About Cybersecurity |
|---|---|
| • **More is more:** Security of communication increases with the number of cryptographic strategies applied.<br>• **Network structure enhances security:** Features or characteristics of the network used for data transport (e.g., number of intermediate hops, server locations, transfer speed) are needed for or positively affect security.<br>• **Server forbidden:** Servers (or servers in certain countries or from certain companies) should not be allowed on the transport path in order to ensure secure messaging. | • **Awareness of multi-hop transmissions:** A message is actually not sent directly from end device to end device, but instead will be handled by other systems along its path.<br>• **End-to-end encryption:** Students demonstrate a method of secure communication that prevents third parties from accessing data. |

## References and Works Consulted

Brom, C., Yaghobova, A., Drobna, A., & Urban, M. (2023). "The internet is in the satellites!": A systematic review of 3–15-year-olds' conceptions about the internet. *Education and Information Technologies, 28*(11), 14639–14668. https://doi.org/10.1007/s10639-023-11775-9

Hennessy Elliott, C., Nixon, J., Chakarov, A. G., Bush, J. B., Schneider, M. J., & Recker, M. (2024). Characterizing teacher support of debugging with physical computing: Debugging pedagogies in practice. *ACM Transactions on Computing Education, 24*(4), 1–28. https://doi.org/10.1145/3677612

Jayathirtha, G., Fields, D. A., & Kafai, Y. B. (2020). Pair debugging of electronic textiles projects: Analyzing think-aloud protocols for high school students' strategies and practices while problem solving. In M. Gresalfi & I. S. Horn (Eds.), *The interdisciplinarity of the learning sciences, 14th International Conference of the Learning Sciences (ICLS) 2020* (Vol. 3, pp. 1047–1054). International Society of the Learning Sciences. https://www.yasminkafai.com/s/Jayathirtha_2020-ISLS-DEBUGGING-ETEXTILES.pdf

Lindmeier, A., & Mühling, A. (2020). Keeping secrets: K–12 students' understanding of cryptography. In T. Brinda & M. Armoni (Eds.), *Proceedings of the 15th Workshop in Primary and Secondary Computing Education (WiPSCE) 2020*. Association for Computing Machinery. https://doi.org/10.1145/3421590.3421630

Morales-Navarro, L., Fields, D. A., Barapatre, D., & Kafai, Y. B. (2024). Failure artifact scenarios to understand high school students' growth in troubleshooting physical computing projects. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE) 2024* (Vol. 1, pp. 874–880). Association for Computing Machinery. https://doi.org/10.1145/3626252.3630855

Percival, N., Narain, S., & Lee, C. S. (2023). Modern cryptography education of middle school students: A review of current works. In *Proceedings of the 24th Annual Conference on Information Technology Education (SIGITE) 2023* (pp. 33–38). Association for Computing Machinery. https://doi.org/10.1145/3585059.3611428

Przybylla, M., & Grillenberger, A. (2021). Fundamentals of physical computing: Determining key concepts in embedded systems and hardware/software co-design. In M. Berges, A. Mühling, & M. Armoni (Eds.), *Proceedings of the 16th Workshop in Primary and Secondary Computing Education (WiPSCE) 2021*. Association for Computing Machinery. https://doi.org/10.1145/3481312.3481352

 