A Platform Solution for Secure Supply-Chain and Chip Life-Cycle Management

Joseph P. Skudlarek, Tom Katsioulas, and Michael Chen, Mentor Graphics

Fragmentation of the system-on-chip supply chain has introduced many vulnerabilities in electronic devices. A proposed platform solution lets supply-chain participants authenticate, track, provision, and analyze their products during the entire chip life cycle through a single root of trust in the form of unique chip IDs.

Globalization and outsourcing have introduced numerous vulnerabilities in electronic devices, including counterfeit integrated circuits (ICs), by fragmenting the system-on-chip (SoC) supply chain throughout the life cycle of design, manufacture, distribution, and field use. The number of vulnerabilities is expected to dramatically increase as billions of devices with SoCs, many provided by untrusted foundries, connect to the Internet of Things.

Silicon supply-chain security is an industry-wide issue that requires a holistic approach and collaboration among IC suppliers, foundries, assembly and test houses, contract manufacturers, and electronic device distributors. We propose a platform solution that combines hardware, software, and protocols to let supply-chain participants authenticate, track, provision, and analyze chips during the entire life cycle. By providing end-to-end security with a reliable root of trust, it could significantly mitigate supply-chain vulnerabilities.

PLATFORM OVERVIEW

Our proposed platform enables connection of SoCs to a *secure server*, tracks them at each step in the supply chain, and securely provisions them in the field. This lets IC suppliers minimize counterfeit components as well as offer new value-added services during the SoC life cycle that were not possible before. It also provides

0018-9162/16/$33.00 © 2016 IEEE

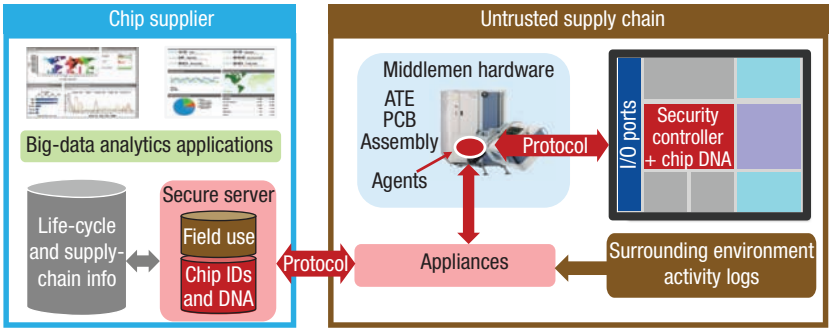

FIGURE 1. Platform overview. The security controller embedded on each chip connects to a secure server and supports the protocols that enable enrollment, authentication, tracking, and in-field provisioning. Each chip has a unique ID and DNA—helper data and a unique key to protect select data in transit—making every server connection with a chip distinct from that of every other chip. In addition to the on-chip hardware, the platform includes appliances and agents: at-server validation and configuration mechanisms and associated middleware that run on the supply chain's untrusted sites. ATE: automated test equipment; PCB: printed circuit board.

better visibility of the SoC life cycle from design to birth to proliferation to decommissioning.

Figure 1 shows a simplified view of the platform in which there is a single *security controller* embedded on a chip. The controller supports the protocols that enable chip enrollment, authentication, tracking, and in-field provisioning. *Enrollment* registers the chip with the secure server. *Authentication* verifies that the chip is known to the system—that is, the chip has been enrolled and has a sound history. *Tracking* establishes the chip's provenance; it certifies that the authentic chip has a detailed chain of custody. *Provisioning* enables and disables individual chip features including intellectual property (IP) blocks and I/O or debug ports. One variation of provisioning is "metering," whereby portions of the design or the full SoC can be provisioned to expire either based on chip status in the supply chain or how often the chip checks in with the server.

A key feature of the platform is use of a physically unclonable function (PUF) to provide each chip with its own unique ID and a unique key to protect select data in transit to the chip. Every server connection with a chip is distinct from that of every other chip, so compromising one chip's unique key does not compromise other chips.

The secure protocols not only utilize the hardware and software provided, but also specify the participation of the chip author and the chip manager that controls the secure server. The chip manager incorporates its own business logic and models to authorize the enrollment, authentication, tracking, and provisioning of individual chips.

In addition to the on-chip hardware, the platform includes the corresponding at-server validation and configuration mechanisms and associated middleware that runs on the untrusted sites of the supply chain—noted in Figure 1 as *appliances* and *agents*, respectively. In other words, it comes with the two self-validating ends of the protocol and a software development toolkit (SDK) to help make the connection between them. The SDK includes middleware for the agents (both at-fab and in-field) as well as for the secure server.

HARDWARE OPERATIONS

The platform uniquely identifies and reliably authenticates each chip. It then tracks the chip and creates an audit trail that establishes its provenance. In addition, selected chip functions, including debug modes and I/O ports, can be enabled or disabled in the field after manufacturing in a secure way based on the chip's unique ID.

A one-time operation gathers enrollment data from the manufactured chip and stores it in a secure server, usually soon after wafer testing. This data consists of the chip's ID and DNA—protected helper data and the chip key—and is subsequently used to reliably identify the chip and to enable authentication and provisioning.

Authentication reliably demonstrates to the secure server that it is indeed the chip it claims to be. This operation relies on a conventional challenge–response protocol that uses the protected chip key to encrypt the challenge and response, and a cipher-based message authentication code (MAC) to ensure the integrity of the data exchanged. The multicycle authentication protocol ensures that the chip is at the other end of the connection.

The secure server uses the chip's unique key to transmit the chip's current configuration of enabled and disabled features, which are based on the chip's unique ID. This configuration update, or provisioning, can be carried out in the field.

At important stages in the supply chain the chip connects to the secure server. The server authenticates the chip at each stage and records that event, establishing a reliable audit trail for the chip's progress and providing proof of provenance.

SECURITY CONTROLLER AND OTHER ON-CHIP RESOURCES

As Figure 2 shows, the security controller interfaces to the outside world via Joint Test Action Group (JTAG) or other I/O ports and executes our protocols utilizing low-level cryptographic primitives. These primitives are used

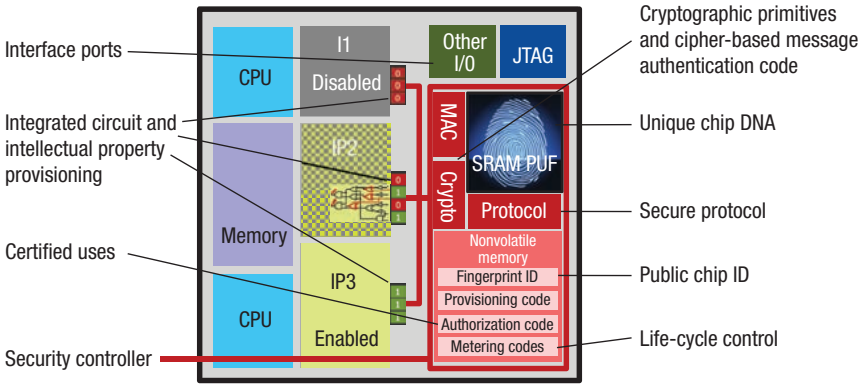

FIGURE 2. Embedded security controller functionality and features. A key feature is a physically unclonable function (PUF), based on static RAM (SRAM) cells, that provides each chip with its own unique ID and DNA. JTAG: Joint Test Action Group; MAC: Message Authentication Code.

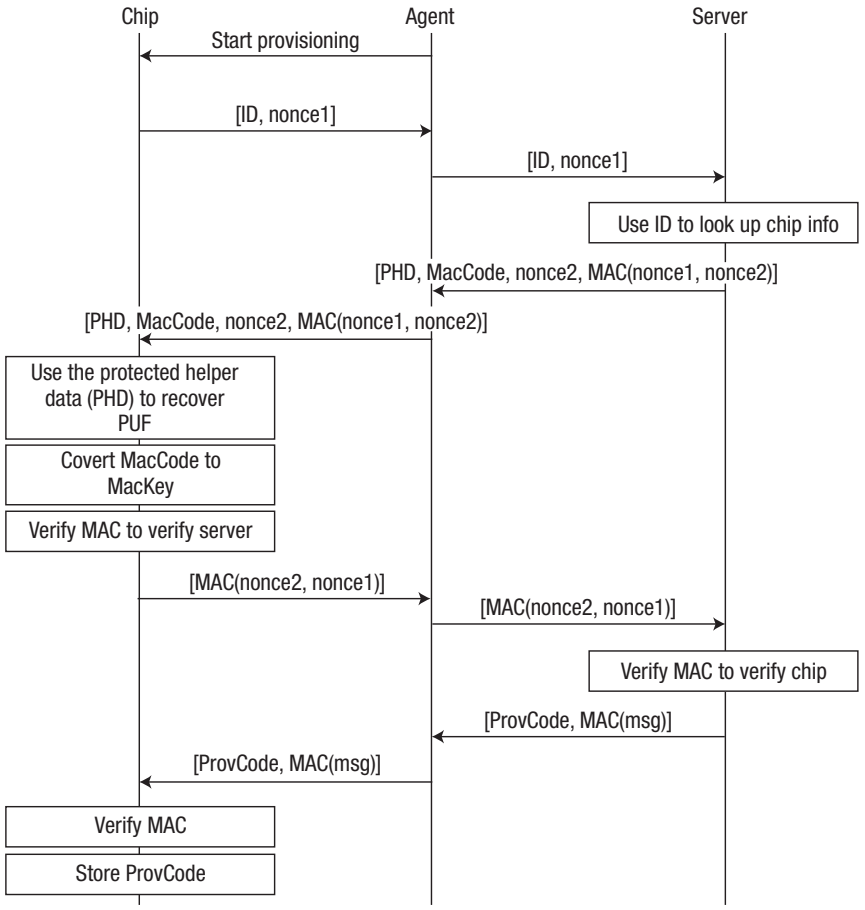

FIGURE 3. Simplified provisioning protocol used to enable or disable chip features. A provisioning bit sequence is loaded into the logical configuration register, and each third-party IP reads its portion of the register to control its provisioned features.

to generate pseudorandom values and MACs and to decrypt protected values. The PUF, based on static RAM (SRAM) cells, provides part of the secret key that is used to manage secure communications with the server.

A logical configuration register stores the enable/disable state of provisioned subsystems and ensures that a given chip configuration controls any third-party IP. As Figure 3 shows, a provisioning bit sequence is loaded into the configuration register, and each IP reads its portion of the register to control its provisioned features.

Design goals

In developing the hardware, we had several design goals.

To maximize detection of possible counterfeit chips, it is necessary to minimize the trust required of various supply-chain contributors. Our solution limits required trust to the design house, which adds our hardware to its chip, and the chip manager, which provides a secure server that connects to the chip with our protocols. In general, we do not rely on secure channels or reliable intermediaries for security because we authenticate the packets that are transmitted between the chip and server and encrypt sensitive information.

To provide strong protection at modest cost, we avoid the more expensive public-key cryptosystems like Rivest–Shamir–Adleman (RSA) or elliptic curve cryptography (ECC); instead, we use industry-standard symmetric encryption like Rijndael, the superset of the Advanced Encryption Standard (AES), between agent and chip, coupled with PUF-provided secrets. We also sought to keep costs low by not requiring a secure channel. Moreover, we leverage existing chip infrastructure and design-flow methodologies; for example, our initial hardware communications mechanism uses a JTAG interconnect and the existing design for test (DFT) and automated pattern generation (ATPG) methodologies

to support chip enrollment at initial power-up during wafer testing.

To encourage wider adoption, the hardware is designed to be independent of foundry and process node. We provide SystemVerilog register-transfer level files and a gate-level netlist, and rely on the designer to supply the necessary SRAM and nonvolatile memory (NVM). We also provide verification IP for system validation of the chip connected to a virtual server, allowing the designer to validate the major hardware operations and possibly to test for various attacks.

To lower deployment costs, we minimize wafer testing time at the manufacturing site using a feed-forward solution: the tester gathers enrollment data from the chip without interacting with the server and then releases the chip. The enrollment data is later sent by an appliance to the secure server and can be batched for more efficient operation.

To reduce the attack surface, all sensitive information should be dynamically computed or stored on the chip encoded in NVM. To achieve these goals, we use a PUF with low-level cryptographic circuitry. The PUF provides a hidden secret available only after power-up, and the cryptographic circuitry processes protected information. We also control key hardware resources like NVM to prevent attackers from corrupting the chip state, which could disable provisioned portions of the chip. No secrets are stored in NVM; we leave the chip in a disabled state at power-off and provide the provisioning ability to re-enable it at the next stage of the supply chain.

Discarded alternatives

We considered but rejected other hardware design alternatives. One was to require a trusted personalizer early in the life cycle to inject a secret, which might be stored in one-time programmable memory, into the chip. However, this solution requires including trusted transfer of the wafers or dies to a trusted personalizer, and leaves secrets at rest on the chip. We also rejected on-chip public-key cryptography. While great advances have been made in ECC to reduce gate count and processing time, these costs are still significant.

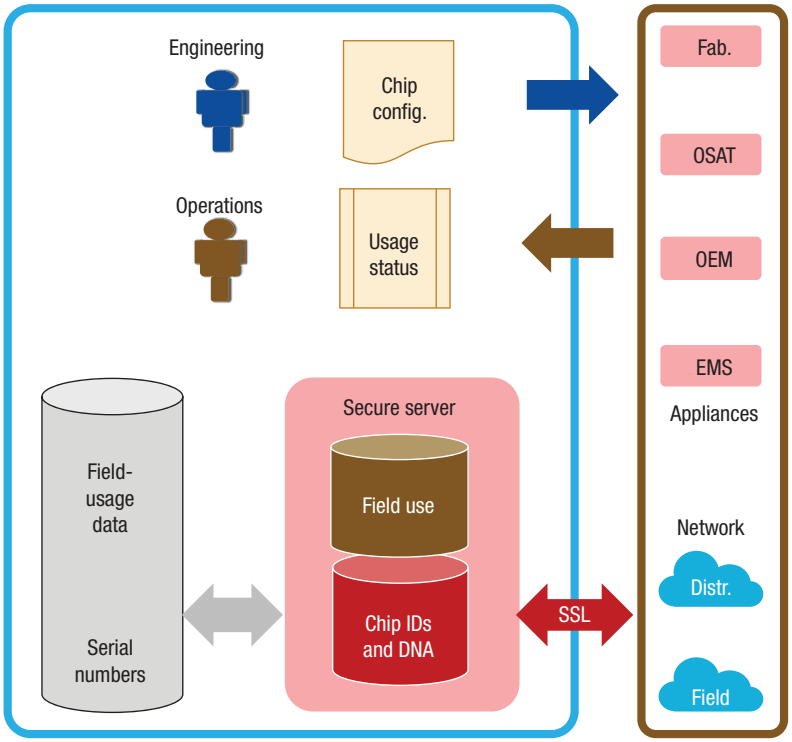

FIGURE 4. Server connectivity with supply-chain participants via Secure Sockets Layer (SSL). The server stores the unique ID and DNA of all chips enrolled during wafer testing at chip fabricators and ties this to additional field-use data gathered from appliances at various vendor sites including assembly and test (OSAT) companies, original equipment manufacturers (OEMs), and electronic manufacturing suppliers (EMSs).

SECURE SERVER

The secure server is an enterprise-grade system that exchanges information with chips containing a security controller through appliances and agents inside middlemen equipment that physically connect with a chip via its I/O ports. As Figure 4 shows, it stores the unique ID and DNA of all chips enrolled during wafer testing and ties this to additional data gathered from appliances at various vendor sites, such as log files from outsourced assembly and test (OSAT) companies, printed circuit board (PCB) debug information from original equipment manufacturers (OEMs), or PCB bill of materials and quality data from electronic manufacturing suppliers (EMSs).

The secure server provides greater visibility into the chip's field-use status and its internal state, and remote controllability by injecting various codes to provision and personalize the chip based on its unique ID. It supports users, roles, and permissions with respect to who can view the chip's status and has privileges to provision it. Authorized operators can observe the chip's status, receive notifications and alerts about state changes in the chip, and obtain real-time and historical reports of the chip's progress in the supply chain.

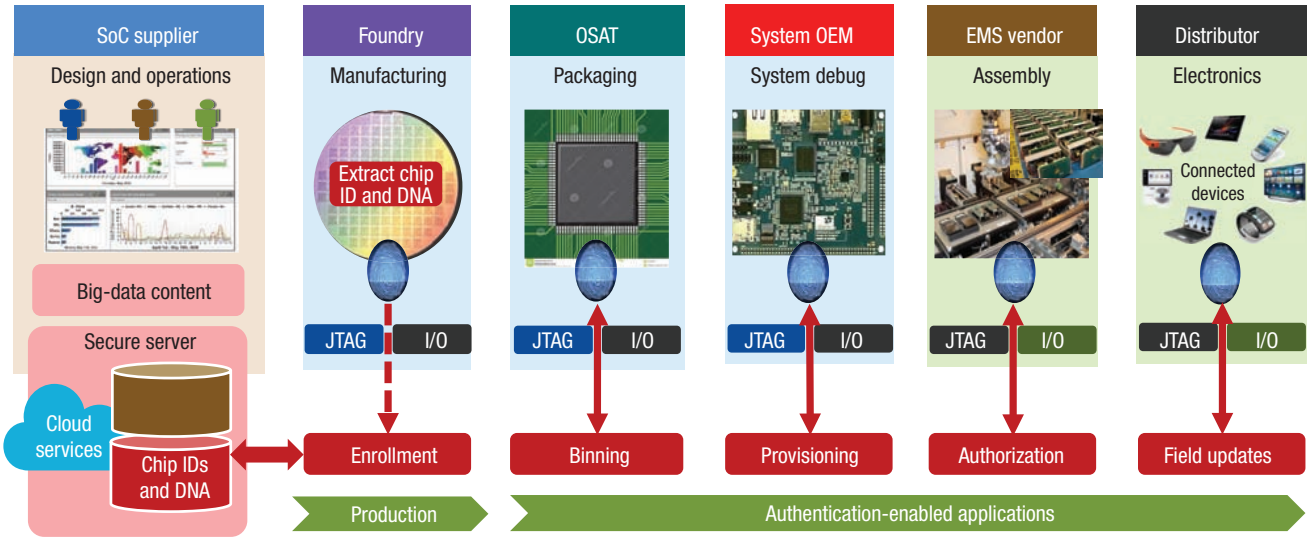

FIGURE 5. Secure supply-chain process. Unique chip IDs let all supply-chain participants track their products through a single root of trust. Real-time monitoring coupled with big-data analytics simplify chip life-cycle management. SoC: system on chip.

During a chip's life cycle in the supply chain, the appliances gather field-use information tied to the chip's unique ID and send it to the secure server, which stores it in a database for next-generation big-data analytics applications. Such information might include wafer lot, wafer map, die IDs from the foundry, package labels, serial numbers, stock-keeping units (SKUs) from OSAT firms, field failures at the PCB or device level, and chip status.

From the secure server, engineering and operations managers can provision various chip features including functional modes, debug ports, application codes, device certificates, life-cycle meters, and field updates.

APPLIANCES AND AGENTS

Appliances' primary role is to collect information from various agents and activity logs from the middlemen that connect to the chip's I/O ports and send this data via Secure Sockets Layer (SSL) to the secure server. One appliance can serve several agents running on multiple devices. The secure server places the data in its database to maintain a complete historical record of activities during chip manufacturing that can be queried later.

These agents have five primary responsibilities:

- connect to the chip's security controller,
- connect to the secure server via the appliance or a connected device,
- accept client requests,
- perform client authorization, and
- execute authorized requests.

In the current platform, agents connect to the security controller via JTAG, but future designs could include other interconnects, including the ARM Advanced Microcontroller Bus Architecture (AMBA). Agents connect to the secure server via the appliance SSL. Additionally, the server validates agents and applies access controls based on their identity. We use a secure connection to support authentication and to reduce the attack surface.

Some agents, like enrolling and tracking agents, are fixed functions. Others, like in-field agents, can process requests on demand for operations including authentication and provisioning. To authenticate a chip, for example, a client's agent connects to the chip and the server to run the authentication protocol and reports the result to the client. After a request has been approved, the agent acts as a middleman in the protocols needed to process the request.

Certain operations, like provisioning, are restricted to authorized users. The agent acts as the gatekeeper and, in cooperation with the appliance and secure server, implements the business logic necessary to authorize the client and the request. For example, a business might require that the client pay a fee and accept a license agreement before a chip can be provisioned in the field. The agent ensures that those requirements are met.

SECURE SUPPLY-CHAIN PROCESS

Our proposed platform can drive a secure supply-chain process, whereby chip suppliers and system OEMs can track their products through a single root of trust in the form of unique chip IDs. The ability to provision hardware at each step in the process, or configure it to self-deactivate if it does not regularly check in with the secure server, makes it possible to manage the chip life cycle, especially in mission-critical applications.

As Figure 5 shows, the platform accomplishes this using a noninvasive methodology that builds on current supply-chain infrastructure:

- *Design*—embed a security controller into each chip and tie its configuration register to third-party IP.
- *Enrollment*—upload the chip's unique ID and DNA—its "birth certificate"—into the secure server upon initial power-up during wafer testing.
- *Binning*—obtain activity logs during package testing and configure SKUs.
- *Provisioning*—configure the chip, IP, functional modes, and JTAG or I/O ports during system debug.
- *Authorization*—obtain logs on authorized use from devices or distributors during assembly.
- *Field updates*—provision the chip for the application and metering during device updates.

The platform's benefits during product development include the ability to leverage existing design flows, high reliability and tamper resistance, and increased configurability. The benefits during volume production include continuous real-time monitoring coupled with big-data analytics, which will reduce field failures and material returns, improve yield, simplify lifecycle management, provide more personalized chips for specific uses and applications, and lead to creation of a security knowledge base, which in turn will enable the implementation of best practices to cope with the attack surface and hackers' growing sophistication.

SECURITY ANALYSIS

Our platform solution adheres to Kerckhoff's principle: a cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

FURTHER READING

- R.J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed., Wiley, 2008.
- M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*, NIST special publication 800-38B, Nat'l Inst. of Standards and Technology, 2005; http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.
- N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design Principles and Practical Applications*, Wiley, 2010.
- R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*, Springer, 2013.
- A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
- S.M. Plaza and I.L. Markov, "Solving the Third-Shift Problem in IC Piracy with Test-Aware Logic Locking," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 4, no. 6, 2015, pp. 961–971.
- M. Rostami, F. Koushanfar, and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," *Proc. IEEE*, vol. 102, no. 8, 2014, pp. 1283–1295.
- U. Ruhrmair and D.E. Holcomb, "PUFs at a Glance," *Proc. Conf. Design, Automation & Test in Europe* (DATE 14), 2014, pp. 1530–1591.
- M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*, Springer, 2015.

Attack defenses

To protect against replay attacks, our protocols incorporate at least one nonce (a one-time and nonrepeated value) that makes each transaction unique. Thus, the same transaction will never occur twice, precluding a replay.

We use two approaches to defend against man-in-the-middle attacks. First, during enrollment, the use of a MAC prevents an attacker from changing data without being detected. Second, during authentication, the secure server sends protected data that only the chip can decode, preventing an attacker from impersonating the chip.

A focused-ion beam (FIB) attack physically alters a chip by adding or removing connections. We reduce the risk of FIB attacks by making it more difficult to discover what locations would need to be changed and requiring many to be changed to bypass security.

Security is not absolute—a sufficiently powerful and patient adversary can often comprise any system if the economics justify it. We do not provide complete protection against reverse-engineering and re-masking attacks, but we make them more difficult by obfuscating the circuit representation and minimizing secrets at rest.

To reduce the risk of birthday attacks, which are used to find collisions in cryptographic hash functions, we use 128-bit values or greater, which implies that about 2^{64} (about 1.8e19) random samples would be needed to get a single collision.

ABOUT THE AUTHORS

JOSEPH P. SKUDLAREK is a staff engineer in the System Level Engineering Division at Mentor Graphics, where he focuses on silicon security projects with an emphasis on physically unclonable functions. He received an MS in computer science from Stanford University. Skudlarek is a Senior Member of ACM. Contact him at joseph_skudlarek@mentor.com.

TOM KATSIOULAS heads the market development ecosystem strategy for the Design for Security program in the System Level Engineering Division at Mentor Graphics. He received an MS in electrical engineering and computer science from the University of Massachusetts Amherst. Contact him at tom_katsioulas@mentor.com.

MICHAEL CHEN manages leading-edge technology efforts for the System Design Management and Design for Security initiatives in the System Level Engineering Division at Mentor Graphics. He received a BS in electrical engineering and in information and computer science from the University of California, Irvine. Chen is a licensed professional engineer and member of IEEE. Contact him at michael_chen@mentor.com.

Trust model

Business requirements—for example, to permit provisioning—dictate the trust that must be established between an agent and a server. Thus, we isolate built-in trust to the chip and server. The agents act as conduits for chip–server communication, and the appliances serve as communication concentrators. But neither appliances nor agents are trusted for chip communication—the protocols ensure the messages' integrity and authenticate the chip and server.

Key management

An integral part of our platform's security is the PUF-generated secret value, which is used as part of the key for cryptographic exchanges. As the PUF value is unique to a chip, communication is only valid for that chip. During enrollment, the chip generates

- an ID, which is a large random number and expected to be unique;
- protected helper data, which can be stored on the server and is used by the chip to reestablish a stable PUF value; and
- protected chip data, which provides the chip-specific information needed by the secure server to generate unique-to-that-chip encrypted data.

We use a fail-hard fail-fast strategy—if the protected helper data is not valid for the chip, or the encoded values are not valid for that chip, the chip goes into a hard error state. This thwarts attacks and reduces the attack surface.

The key management needed to allow agents to talk with servers, and the permission management controlling which agents can perform which protocols, are business-logic specific, and beyond the scope of this article.

We continue working to improve the platform. One primary goal is enhanced protection for disabled functionality on a chip, or circuit locking. The basic idea is that the circuit incorporates locking gates, and will not function properly without the appropriate multibit key. This is to prevent a single-location point of attack—for example, if a single wire carried the decision to activate the chip functionality, then by changing just that bit (through, say, a FIB attack), the circuit would be easily enabled. Another aim is logic obfuscation—hiding the function of select logic on a chip to make reverse engineering very difficult. Both of these capabilities will require new tools and are expected to drive next-generation design-flow methodologies for security compliance.