ThumbPod Puts Security Under Your Thumb

UCLA students develop a prototype fingerprint authentication device incorporating a Virtex-II FPGA.

by Patrick Schaumont
Ph.D. Candidate, Embedded Security,
EE Department, UCLA
schaum@ee.ucla.edu

Ingrid Verbauwhede
Associate Professor, Embedded Security,
EE Department, UCLA
ingrid@ee.ucla.edu

We live in a networked world, where your digital alter ego is almost as important as your physical self. Your digital identity, usually comprising a series of electronic records, determines whether you can buy a car, get a loan, or open a bank account.

Yet the link between your physical and digital selves is surprisingly weak and insecure. Identity theft and credit card fraud are ubiquitous. Indeed, we have grown used to verification via trivial secrets such as your mother's maiden name. But trivial secrets require only trivial guesswork to be revealed.

A more unique identifier is possible through the use of biometrics – unique physical features such as DNA, fingerprint data, iris composition, or facial structure. These are distinctive keys that cannot be forgotten or lost and, with appropriate precautions, are difficult to counterfeit.

The ability to authenticate yourself through biometrics opens up enormous possibilities in embedded and portable contexts, from key chains to credit cards. The extraction of unique features from biometrics data, however, is a complex signal-processing problem that requires a significant amount of computational power.

One solution may be ThumbPod, an FPGA-based prototype (Figure 1) of an embedded fingerprint authentication system created by seven graduate students and their advisor at the University of California, Los Angeles. ThumbPod (*www.thumbpod.com*) includes a fingerprint sensor, an embedded processor with memory, dedicated processing units, and a serial communication channel.

According to design specifications, the unit must deliver 1,000 fingerprint-matching operations, each comprising an estimated 250 million operations, on a single battery. Given the complexity of the embedded fingerprint recognition, we use an FPGA prototype to quickly and easily evaluate critical design decisions.

In its final form, ThumbPod could be integrated into an embedded device not larger than a key chain fob (Figure 2).

The Security Pyramid

A security application is only as safe as its weakest link. ThumbPod is no exception. Security must be considered at all design stages and design abstraction levels.

We employ a "security pyramid" to identify four different design abstraction levels in ThumbPod that affect safe operation:

- The *protocol* level expresses how ThumbPod connects to an application; that is, how it will be used to open an electronic lock. A critical aspect in the design of this protocol is that fingerprints are sensitive personal data, and thus should be processed with adequate privacy. Unlike many other biometrics systems, our ThumbPod processes all fingerprint data locally. Even the fingerprint sensor is local and private for each person, and cannot be shared.

- The *algorithm* level collects the functions and algorithms used as building blocks in the ThumbPod security protocol. This includes fingerprint signal processing to extract a unique template from a fingerprint, consisting of minutiae. It also includes encryption using the Advanced Encryption Standard.

- The *architecture* level defines the target onto which the algorithms are mapped. We combine a soft-core, 32-bit,

Figure 1 – ThumbPod prototype

LEON2 SPARC processor with hardware acceleration co-processors for signal processing and encryption. This enables the general-purpose processing required to integrate the application, as well as the intensive data processing necessary for fingerprint matching, to operate in an embedded context.

- The *circuit* level deserves special attention, because it is a potential backdoor for security hackers. With careful technology mapping, however, we can make ThumbPod resistant to power and timing attacks.

Figure 2 – ThumbPod conceptual drawing

System Architecture

ThumbPod's high design complexity makes it impossible to capture everything in one design layer. Instead, we approached the design as a stack of three machines, each one bootstrapping the next (Figure 3).

At the bottom layer is a Virtex™-II XC2V1000 FPGA. It has a rich set of on-chip resources providing interconnection, on-chip storage, and logic. The prototype is implemented on an Insight Electronics development board that also offers 32 MB of DDR RAM for background storage.

On the FPGA, we configured a soft-core processor with two hardware co-processors. The processor is a SPARC-compliant, 32-bit LEON2 from Gaisler Research (*www.gaisler.com*). An encryption processor and a discrete Fourier transform (DFT) signal processor provide acceleration for the critical processing parts of the design.

On top of the LEON2, we ran an embedded Java kilobyte virtual machine (KVM). The KVM offers our application developers a smooth transition from application development to the embedded design.

In addition, all platform-specific features, such as the fingerprint sensor or hardware co-processors, can be integrated into a single programming environment using native interfaces. (The ThumbPod website [*www.thumbpod.com*] collects additional publications that elaborate on the performance and test issues occurring through the use of these interfaces.)

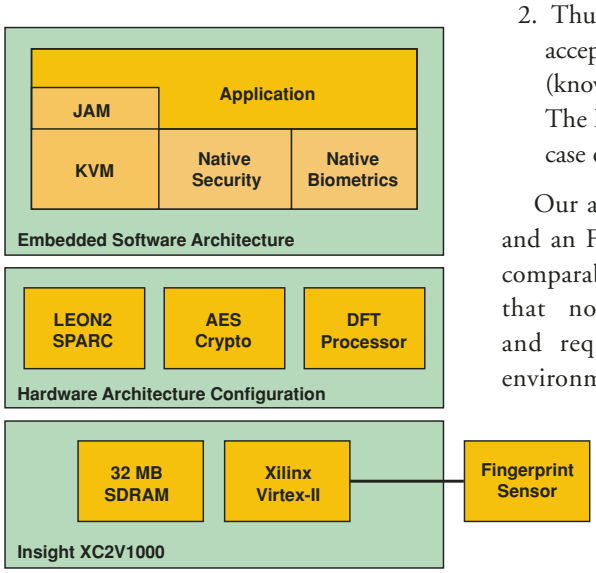

Figure 3 – ThumbPod system architecture

Fingerprint Matching

When you put your finger on the fingerprint sensor, it takes an image. From this raw image, we extract a set of unique features, known as minutiae (Figure 4) through a sequence of image processing steps (Figure 5). We obtain the minutiae beginning with the raw grayscale image of the fingerprint and proceeding through a sequence of image processing steps.

The set of minutiae forms a secret key that is stored securely in ThumbPod as a template. It cannot be changed afterwards, nor can it ever be extracted. During actual use, this template is matched to a newly captured fingerprint. This matching process returns a score between 0 and 100, which is used to decide acceptance or rejection of the ThumbPod user.

The bulk of the processing power in ThumbPod is devoted to the extraction of minutiae. We started with a reference software implementation from NIST (National Institute of Standards and Technology). Subsequently, we adapted this for embedded processing. This includes conversion from floating-point to fixed-point operation, reduction of the memory requirements, and the introduction of a DFT hardware accelerator.

The operation of ThumbPod must be as reliable as possible. We verified the quality of the fingerprint detection and matching algorithms by evaluating their probability for error.

Two types of mistakes could occur:

1. ThumbPod could reject a fingerprint corresponding to the true template (known as a "false reject," or FFR).
2. ThumbPod, however, might also accept an incorrect fingerprint (known as a "false accept," or FAR). The latter case is, of course, the worse case of the two.

Our algorithms gave an FRR of 0.5% and an FAR of 0.01%. These figures are comparable to commercial-grade systems that normally run on workstations and require power-hungry processing environments.

Design Flow Using GEZEL

ThumbPod is programmed in a combination of Java, C, and VHDL. Our design flow ensures that the appropriate verifications are complete before the design is brought to the FPGA (Figure 6).

A key element of the design flow is the GEZEL design environment (*www.ee.ucla.edu/~schaum/gezel*) that supports the development and prototyping of our coprocessors. GEZEL provides instruction-set co-simulation as well as VHDL code generation. In combination with the FPGA, GEZEL technology provides a high-level exploration environment for complex system-on-chip architectures such as ThumbPod.

Our design flow consists of three distinct steps, each with an increasing amount of implementation detail.

Figure 4a – Raw fingerprint data from sensor

Figure 4b – Minutiae location with superimposed cleaned-up fingerprint image

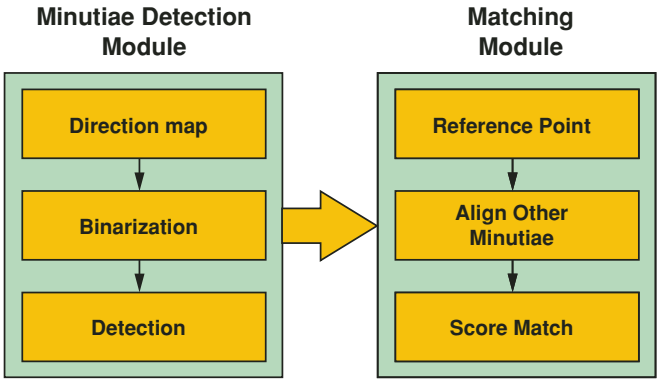

Figure 5 – Fingerprint minutiae detection and matching

We developed the top-level security protocol in Java. Using simulation on a workstation, we verified how the protocol behaves under security exceptions such as replay attacks or the use of false fingerprints. We also checked functional aspects, such as protocol communication timeouts and transmission errors.

Next, we integrated the C code of the fingerprint detection and matching algorithms into the Java code. We used the native method capabilities of Java. We developed this C code as a separate software IP core. We tested it extensively in overnight simulations before integrating it. These overnight simulations were required to check the quality loss resulting from fixed-point refinement of the fingerprint algorithms.

We then ported the ensemble of Java and C code to the embedded Java KVM. The KVM is executed on top of the LEON2 instruction-set simulator. Using this setup, we could simulate how the ThumbPod application downloaded as an applet to the prototype platform – and how the protocol worked in combination with a server application.

Once the simulation ran on top of an instruction-set simulator, we began including models of the actual target architecture, particularly the hardware coprocessors. The GEZEL environment captured cycle-true descriptions of the coprocessor architectures using a specialized programming language. The models of the AES and DFT coprocessors were co-simulated with the LEON2 instruction set simulator.

Through VHDL code generation, we obtained seamless connection to the FPGA platform. We followed a standard FPGA design flow based on Synplicity® synthesis software and Xilinx ISE software.

Conclusion

We demonstrated our ThumbPod prototype in a University Booth at the 2003 Design Automation Conference in Los Angeles. ThumbPod is a first-of-its-kind device that demonstrates portable and embedded biometrics processing as an FPGA prototype.

Embedded biometrics is a field in full expansion. As information technology further penetrates our everyday life, the need for privacy and authentication will continue to grow. Passwords and secret questions will become impractical compared to the convenience of biometrics.

But the complex signal processing that comes with embedded biometrics will require innovative architectures. These architectures combine sequential processing with the power efficiency of parallel hardware.

Using platform FPGAs and reconfigurable technology, we can build and test embedded biometrics solutions like ThumbPod in a way that outperforms the design time and cost of other technologies, such as COTS solutions or ASICs.

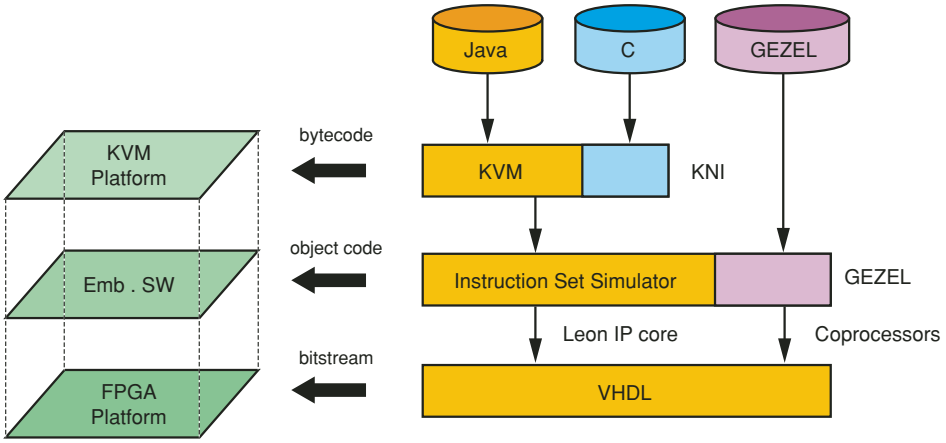

Figure 6 – ThumbPod design flow

Constructing the ThumbPod system required a significant amount of dedication and time. ThumbPod is the result of the efforts of seven graduate students at the University of California, Los Angeles, over a period of eight months. They are Yi Fan, Alireza Hodjat, David Hwang, Bo-Cheng Lai, Kazua Sakiyama, Patrick Schaumont, and Shenglin Yang.

The project enjoyed the support of many people, including the students' advisor, Professor Ingrid Verbauwhede, Jiri Gaisler from Gaisler Research, and Daryl Specter from Insight Electronics.

We would also like to thank the Xilinx University Program team, including Jeff Weintraub and Anna Acevedo, for their support and generous donation.