Offline HW/SW Authentication for Reconfigurable Platforms

Eric Simpson, Patrick Schaumont
Overview

- Security considerations for reconfigurable platforms
- Why today’s security mechanisms are insufficient
- New approach to securing reconfigurable designs
Reconfigurable Designs

- **Reconfigurable**: chip whose logic function is programmed by customer *after* the IC has been fabricated
  - Design represented by bitstream, not a physical chip
- Security considerations for bitstream?

![Diagram of System Design Process]

1. **Development**
2. **Bitstream**
3. **Program Chip**
Bitstream Piracy Example

- Does physical chip need to be stolen for your system to be pirated?
Current Bitstream Security

- Bitstream is stored encrypted offchip
  - Decrypted upon entering FPGA
  - Then used to configure chip
How to deal with larger designs?

- Increasing density of FPGAs
  - 185 thirty-two bit RISC processors on a single chip

"Using IP library elements in a ‘cut-and-paste’ design style is the only way to reach the necessary design productivity"

- Muscular Methods for Mammoth Designs
FPGA IP Market

- Practically non-existent IP market for reconfigurable targets

  “Commercial model for IP cores involves large up-front fees reminiscent of ASIC NRE charges”

  - T. Kean, Algotronix

- No security assurances between system developers and IP providers
Bitstream Encryption is Insufficient

- Secure interaction between multiple parties involves three components:
  (1) Privacy
  (2) Authenticity
  (3) Integrity

- Bitstream encryption only provides privacy
HW/SW Mutual Authentication

- Allows secure, authenticated distribution and integration between multiple parties
Identity

- In order to authenticate something, its identity needs to be established

- HW Identity
  - Characterized by the physical silicon of the chip

- IP Identity
  - Sequence of processor executed opcodes
  - Bitstream that represents a custom logic function
HW Identity

- Standard security and authentication module manufactured in each chip
- Contains a Physically Unclonable Function (PUF)
  - Uniquely identify a chip by utilizing the inherent variation in the underlying silicon
PUF Challenge/Response

- At a high-level, PUF is characterized by its challenge, response pairs

Challenge = 0x1234

- FPGA (1)
  - PUF
  - Response = 0x00AB

- FPGA (2)
  - PUF
  - Response = 0xF01A
IP Identity

- Nothing physical to characterize about IP
- Represented by sequence of ones, zeros and name we give it
Authentication Protocol

- Can assign identities to:
  1. Hardware
  2. IP

- Authentication protocol is divided into two phases:
  1. Enrollment
  2. Request and Distribution

Identity

Securely Authenticate HW and IP?
Enrollment Protocol

- Used to establish repository of HW/IP identities
Enrollment Protocol

- HW Identity Transmitted by Chip Manufacturer
  - $HW\#$: Hardware ID (e.g. manufacturer serial number)
  - $<CRP>$: List of challenge, response pairs

- IP Identity Transmitted by IP Provider
  - $IP\#$: IP ID (e.g. Name and release version)
  - $Hash(IP\ Data, \ IP\#)$
Secure IP Request and Distribution

- Enrollment continues in background
- Design Example:
  - Prototype portable TIVO player with HDTV capability
  - TIVO isn’t focused on designing HDTV decoders
  - TIVO utilizes third-party HDTV core in their system
Secure IP Request and Distribution

Sys Developer → TTP → IP Provider

1. IP#, HW#, Nonce

2. IP#, HW#, C_{tp}
   \{IP#, Hash(IP, IP#), C_{ip}, Nonce\}R_{tp}

3. IP#, HW#, Nonce, R_{ip}

4. IP#, HW#, 
   \{length, Nonce, IP\}R_{ip}
System Block Diagram

- Xilinx Virtex-II FPGA
Security Module
Loading external IP

- Secure IP is stored off-chip in the following three-part format:

  1. *Opcode*[load]
  2. *Ctcp, {IP#,Hash (IP, IP#) ,Cip,Nonce}Rttp*
  3. *{Length,Nonce, IP}Rip*

- Not limited to single IP module
- At runtime can swap modules in and out
Generating CRPs

- Generating a CRP list requires the following message:

  1. Opcode[CRP] , Seed, # of pairs to generate

- Seed = 64-bit random number
- # of pairs to generate = 64-bit integer
CRP Generation Algorithm

\[ C_0 = \text{PUF} (\text{PUF}(\text{seed})) \]
\[ R_0 = \text{PUF}(C_0) \]

For \( i = 1 \) to \# \ of \ pairs \ to \ generate
\[ C_i = \text{PUF}(R_{i-1}) \ ^{i} \ ^{R_{i-1}} \]
\[ R_i = \text{PUF}(C_i) \]
Conclusion

- Bitstream encryption alone is insufficient to cope with multiple IP originators
- Our mutual HW/SW authentication scheme is able to cope with systems integrating multiple sources of IP
- More lightweight than other trusted-computing ideas
- System can deployed in an offline context
- Backward compatible with existing approaches to downloading FPGA bitstreams