# CWE Detail – CWE-1259

## Description

The System-On-A-Chip (SoC) implements a Security Token mechanism to differentiate what actions are allowed or disallowed when a transaction originates from an entity. However, the Security Tokens are improperly protected.

## Extended Description

Systems-On-A-Chip (Integrated circuits and hardware engines) implement Security Tokens to differentiate and identify which actions originated from which agent. These actions may be one of the directives: 'read', 'write', 'program', 'reset', 'fetch', 'compute', etc. Security Tokens are assigned to every agent in the System that is capable of generating an action or receiving an action from another agent. Multiple Security Tokens may be assigned to an agent and may be unique based on the agent's trust level or allowed privileges. Since the Security Tokens are integral for the maintenance of security in an SoC, they need to be protected properly. A common weakness afflicting Security Tokens is improperly restricting the assignment to trusted components. Consequently, an improperly protected Security Token may be able to be programmed by a malicious agent (i.e., the Security Token is mutable) to spoof the action as if it originated from a trusted agent.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Related Attack Patterns (CAPEC)

* CAPEC-121
* CAPEC-681

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Files or Directories, Execute Unauthorized Code or Commands, Bypass Protection Mechanism, Gain Privileges or Assume Identity, Modify Memory, Modify Memory, DoS: Crash, Exit, or Restart — Notes:

## Potential Mitigations

**•** Architecture and Design: Security Token assignment review checks for design inconsistency and common weaknesses. Security-Token definition and programming flow is tested in both pre-silicon and post-silicon testing. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** Let's assume the system has two agents: a Main-controller and an Aux-controller. The respective Security Tokens are "1" and "2". Register Description Default AES\_ENC\_DEC\_KEY\_0 AES key [0:31] for encryption or decryption 0x00000000 AES\_ENC\_DEC\_KEY\_1 AES key [32:63] for encryption or decryption 0x00000000 AES\_ENC\_DEC\_KEY\_2 AES key [64:95] for encryption or decryption 0x00000000 AES\_ENC\_DEC\_KEY\_3 AES key [96:127] for encryption or decryption 0x00000000 AES\_KEY\_ACCESS\_POLICY AES key access register [31:0] 0x00000002

## Notes

**•** Maintenance: This entry is still under development and will continue to see updates and content improvements. Currently it is expressed as a general absence of a protection mechanism as opposed to a specific mistake, and the entry's name and description could be interpreted as applying to software.