Auto entropy Collection Provider– ACP 1.0

Revision 1.0, October 20, 2024

John G. Underhill – john.underhill@protonmail.com

This document is an engineering level description of the ACP 1.0 entropy provider.

1. Introduction

The Auto Entropy Collection Provider (ACP) is designed to gather entropy from multiple system-level sources; system statistics, operating system random providers, and platform random providers (e.g., RDRAND). These entropy sources are used to seed a cSHAKE XOF generator, which produces a cryptographically secure pseudo-random byte stream. ACP provides pseudo-random output suitable for cryptographic key generation, generator seeds, initialization vectors, and other security-related applications.

2. Protocol Description

The ACP protocol uses several functions to gather entropy and generate random numbers.

2.1 Entropy Collection (acp_collect_statistics)

- **Purpose**: Gathers entropy from various system-level sources and compresses the collected statistics using SHA3.
- **Function Logic**:
  1. Collects timestamps, process handles, user, drive, and memory statistics and hashes them using SHA3-512.
  2. Collects pseudo-random from the operating system random provider.
  3. Collects random from the system (RDRAND) entropy provider.
  4. Combines these seeds to initialize an instance of cSHAKE-512.

2.2 Random Byte Generation (qsc_acp_generate)

- **Purpose**: Generates an array of random bytes using multiple entropy sources.
- **Inputs**:

  output: A pointer to the byte array that will receive the random bytes.

  length: The number of random bytes requested.

  Returns: True for success.

  - **Function Logic**:

1. Collects and hashes system statistics, generates an operating system random provider seed, and a platform random provider seed.
2. Seeds the cSHAKE XOF function to produce the pseudo-random output.

2.3 Integer Generation (qsc_acp_uint16, qsc_acp_uint32, qsc_acp_uint64)

- **Purpose**: Generates random unsigned integers of different sizes (16, 32, or 64 bits).
- **Function Logic**:

  Uses qsc_acp_generate to gather random bytes and converts them into unsigned integers.

  Returns: the random integer.

3. Mathematical Description

The ACP protocol is based on the cSHAKE-512 function from the Keccak SHA-3 family, which operates using a sponge construction. The primary entropy sources are combined and hashed into a seed that is then expanded into pseudo-random output using cSHAKE-512.

Entropy Collection: The function collects entropy from various operating system sources and compresses them using a using the hash function SHA3-512 into a 512-bit seed ***S***:

$$S = \text{SHA3-512}(system\_statistics)$$

The system statistics are sets of data collected from the operating system:

- logged-in user statistics
- system timestamp
- computer name
- current process id
- logged-in user name
- system uptime timer
- hard drive statistics (total, used, remaining bytes)
- memory statistics (main memory total, used, and remaining bytes)

Random Byte Generation: The seed ***S***, along with system random bytes generated with the operating system random provider ***R***, and custom seed supplied by the system hardware random provider ***P***, is used to initialize the cSHAKE-512 XOF function.

where ***Z*** is the random output:

$$Z = \text{cSHAKE}(R, S, P)$$

4. References

NIST FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

Intel RDRAND Instruction: Documentation

https://software.intel.com/en-us/articles/intel-digital-random-number-generator-drng-software-implementation-guide

SP 800-90C: Recommendation for Random Bit Generator (RBG) Constructions

https://csrc.nist.gov/publications/detail/sp/800-90c/final

SP 800-22 Rev. 1a: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications

https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final

Microsoft Cryptographic Service Providers Documentation: A comprehensive guide to understanding and using Microsoft's Cryptographic Service Providers (CSPs), detailing their architecture, supported cryptographic algorithms, and implementation in Windows applications.

https://learn.microsoft.com/en-us/windows/win32/seccrypto/microsoft-cryptographic-service-providers

Random Number Generation in Linux: Understanding /dev/random, /dev/urandom, and Cryptographic APIs.

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/LinuxRNG/LinuxRNG_EN_V5_0.pdf?__blob=publicationFile&v=3

Conclusion

ACP is a robust and flexible entropy gathering system and pseudo-random generator designed to provide cryptographically secure random bytes. By leveraging multiple operating system entropy sources and combining them with the operating system entropy provider, the platform entropy provider, and the cSHAKE-512 function from SHA-3, ACP ensures that the pseudo-random bytes it generates is both secure and unpredictable. ACP is particularly well-suited for use in cryptographic key generation, initialization vectors, and other security-critical operations that require high-quality randomness.

Key Strengths:

1. **Diverse Entropy Sources**: ACP gathers entropy from a wide range of system sources, including timers, process handles, and memory statistics. Platform-specific random number generators like RDRAND, as well as the operating system cryptographic random generator. This diversity ensures a high degree of entropy is injected into the generator, making it difficult for attackers to predict the internal state.

2. **Post-Quantum Security**: By utilizing SHA3-512 and cSHAKE-512, ACP offers strong post-quantum security, ensuring that even in the face of future quantum computing advancements, the generated random bytes will remain secure.
3. **High Entropy Output**: The use of SHA3-512 compression on system entropy sources and the cryptographic strength of cSHAKE-512 ensures that the random output remains of high entropy and cryptographically secure.

Recommendations:

ACP is recommended for any cryptographic system that requires secure and reliable random number generation. Its flexibility, high security, and post-quantum resilience make it suitable for applications such as key generation, cryptographic protocols, and secure communication systems. ACP's ability to combine diverse entropy sources and maintain a high degree of randomness even in constrained environments makes it a dependable solution for a variety of cryptographic needs.

In conclusion, ACP provides a comprehensive and secure mechanism for collecting entropy and generating random bytes. Its design ensures long-term security and adaptability to evolving cryptographic standards and requirements, making it an ideal choice for secure, high-performance cryptographic operations.