THE YARROW-160 PRNG

John Kelsey, Bruce Schneier, and

Niels Ferguson

Counterpane Systems

101 E. Minnehaha Pkwy

Minneapolis, MN 55419

{*kelsey,schneier,niels*}*@counterpane.com*

http://www.counterpane.com

OVERVIEW OF TALK

1. Introduction and Context

2. Design Philosophy and General PRNG Structure

3. Design of Yarrow-160 and Resistance to Attacks

4. Open Issues

INTRODUCTION

- Cryptography requires RANDOM numbers.
  - Keys, IVs, Nonces, etc.
- Deterministic computers designed to be predictable, not random.
  - But they can often **observe** nondeterministic behavior in their own hardware.
- Possible solutions:
  1. Add random number generator (RNG) hardware to computers.
  2. Use Pseudorandom Number Generators (PRNGs)

PRNGs and PSEUDORANDOM NUMBERS

- A PRNG is an algorithm that:

  1. Collects UNPREDICTABLE VALUES from operations of computer.

  2. Uses them to derive or update an unguessable KEY.

  3. Uses key to generate PSEUDORANDOM NUMBERS.

     – Deterministic; outputs are function of KEY.

     – Computationally infeasible to distinguish outputs from random without knowledge of KEY.

     – Usually based on other cryptographic mechanism, e.g. block cipher.

TWO COMPETING DESIGN PHILOSOPHIES

1. Entropy Queues: PGP, /dev/random, Cryptlib

   - Assume sufficient entropy for all outputs.
   - Task of PRNG is to distill out entropy and queue it up for use.
   - When don't see sufficient entropy, shut down or generate pseudorandom outputs.

2. Pseudorandom Generators: ANSI X9.17 Key Generator, RSAREF PRNG, Yarrow

   - Collect and distill sufficient entropy for a KEY.
   - Generate pseudorandom outputs from KEY.

YARROW DESIGN PHILOSOPHY

- Once we have unguessable KEY, it's easy to generate random-looking bits.

- Hard problems are:

  1. Getting sufficient entropy for initial KEY.

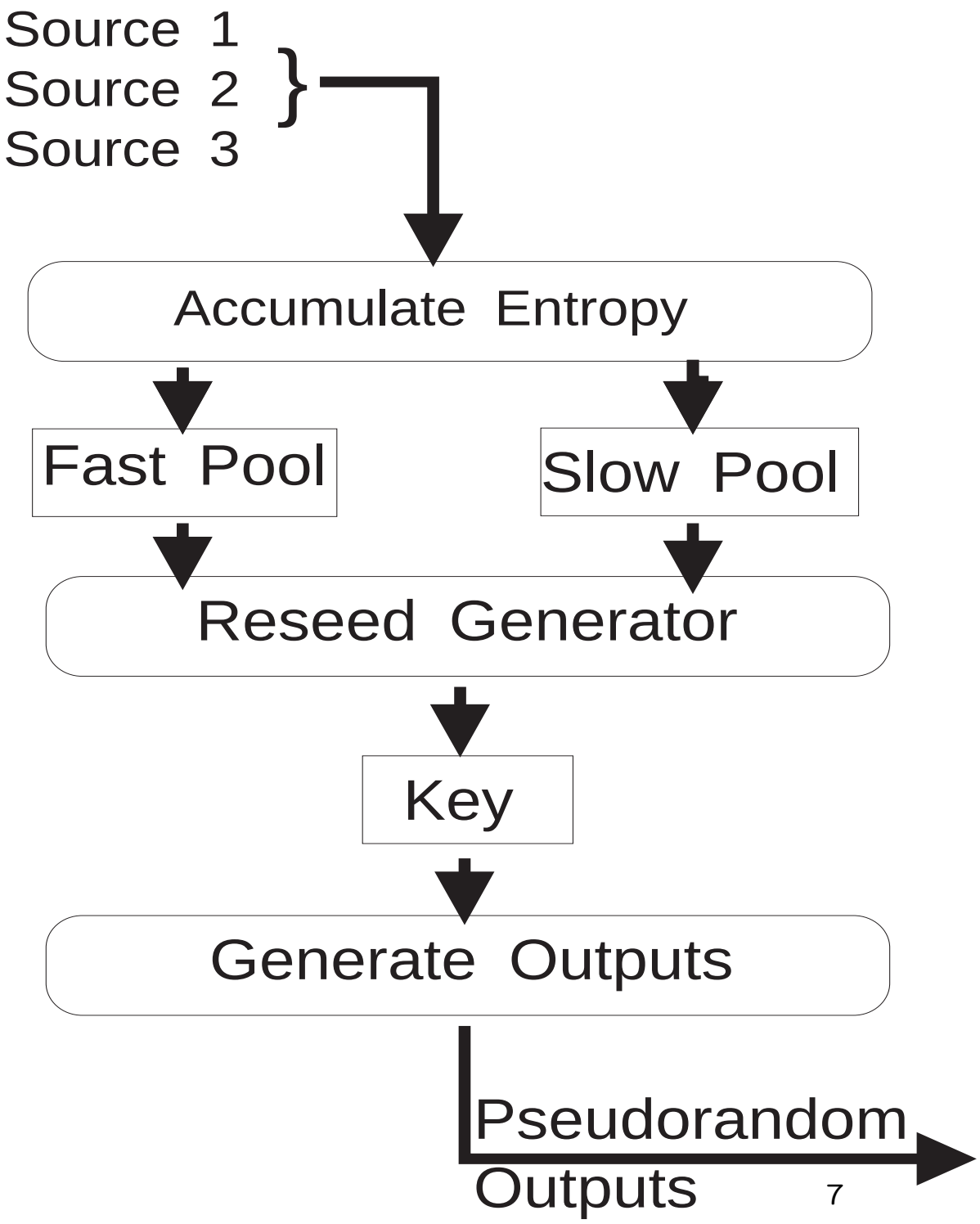
  2. Measuring entropy to know when it's sufficient.

  3. Reseeding to recover from compromise.

  4. Surviving attacker-control over some entropy sources.

- Attack-Oriented Design—base specific design on known attacks.

The Yarrow PRNG: Overview



THE GENERALIZED YARROW DESIGN: COMPONENTS

- SOURCES provide entropy to the PRNG.
- Entropy is accumulated into the POOLS.
- Each pool keeps a running ENTROPY ESTIMATE from each source.
- When a pool estimates it has enough entropy, it RESEEDS.
- RESEEDING updates the KEY from one or both POOLS.
- The KEY is used to GENERATE PSEUDORANDOM OUTPUTS.

Cryptanalytic Attacks on Yarrow-Type PRNGs

1. **Guessed Entropy**—Guess inputs more easily than expected.

2. **Direct Cryptanalysis**—Cryptanalyze generator mechanism.

3. **Input-Based**—Attack based on knowledge/contro over inputs.

4. **Compromise Extension**—Compromise PRNG state, then extend effects as far as possible.

   (a) Iterative Guessing

   (b) Backtracking

   (c) Too-Slow Reseeding

ENTROPY AND SOURCES

- Treat different sources of entropy separately, e.g.:

  1. Keyboard timings.

  2. Packet arrival timings.

  3. Microphone inputs.

- Estimate entropy from each source based on that source's properties.

- Sources are implementation-dependent.

- Good selection and entropy estimation of sources necessary to resist entropy-guessing attacks.

ACCUMULATING ENTROPY IN POOLS

- Each pool keeps running hash of all inputs since last reseed.
- Each pool keeps estimate of entropy from each source.
- Fast pool reseeds often, to quickly recover from compromise.
- Slow pool reseeds rarely, to almost certainly recover from compromise.
- Hash used in pools must resist chosen-input attacks.

RESEEDING

- New key is function of both key and pool.
- Reset entropy estimates in pool after reseeding.
- Reseed can be made computationally expensive to resist entropy-guessing attacks.
- Reseeding must wait until we can reseed with unguessable seed.
  - Otherwise, vulnerable to iterative guessing attack.

OUTPUT GENERATION

- Generate pseudorandom outputs from key.
- Must resist backtracking after compromise.
- Must resist direct cryptanalysis of outputs.

YARROW-160 COMPONENTS

- Specific sources amd entropy estimates implementation dependent.
- Entropy Accumulation done with SHA1.
- Pools are SHA1 hashing contexts.
- Reseed using SHA1 and triple-DES.
- Key is a three-key triple-DES key.
- Generate pseudorandom outputs using triple-DES in counter-mode.
- **Design security is 160 bits.**

YARROW-160 RESEEDING WITH SHA1

- Reseed may be made computationally expensive.

- Reseed works as follows:

  1. Generate 20 bytes of output and hash into pool.

  2. Let X_0 =hash of pool.

  3. For $i = 1$ to n, let $X_i = SHA1(X_{i-1})$.

  4. Extend X_n to 168 bits and use as new key.

  5. Reset counter C to zero.

  6. Reset all entropy estimates in pool(s) used to zero.

GUESSING PAST RESEED LIKE A DICTIONARY ATTACK

- If insufficient entropy in pool (like too-short password), can guess.

- Include timestamp at reseed as form of "salt."

- Include key+pool in reseed; both needed to learn new key.

- Make reseed expensive to make guessing entropy more expensive.

- Reseed slow pool using fast pool contents as well. (Slow pool reseed is last chance to recover from compromise.)

YARROW-160 ACCUMULATES ENTROPY WITH SHA1

- Indistinguishable from full entropy.
  - If we ever see pair of distinct input sequences resulting in same pool, we have hash collision.
- Resistant to chosen-input attacks.
  - Hashes designed with user chosen-input attacks in mind.
- Efficient.

YARROW-160 FAST POOL

- Purpose: reseed quickly in event of compromise. If enough entropy, should resume secure PRNG operations as quickly as possible. **Resist attacks based on too-slow reseeding.**

- Rule: Reseed when any source estimate reaches 100 bits.

- Typically use computationally cheap reseed.

- Reset all estimates in fast pool to zero after reseed.

YARROW-160 SLOW POOL

- Purpose: eventually reseed securely in event of compromise. Even if estimates are optimstic, should still reseed securely. **Resist iterative guessing attacks.**

- Rule: Reseed when any two sources reach 160 bits.

- Reseed includes data in fast pool.

- Typically use computationally expensive reseed.

- Reset all estimates in both pools to zero after reseed.

RESISTANCE TO ITERATIVE GUESSING ATTACKS

- If estimates about best source accurate, fast pool reseeds are enough.

- If not, slow pool reseeds should eventually reseed securely.

- If all estimates far too optimistic, nothing can save PRNG from attack.

YARROW-160 OUTPUT GENERATION

- Generate outputs 64 bits at a time.
- $output \leftarrow E_{KEY}(C)$
- $C \leftarrow C + 1$
- Strength equivalent to that of three-key 3DES.
- Every few outputs, let $KEY \leftarrow$ next 168 bits of output.
  - Prevents backtracking.
  - Prevents block size birthday paradox problems.

ATTACKS: SUMMARY

- Reseed mechanism design adds difficulty to guessed-entropy attack.
- Cryptanalysis attack resisted by triple-DES.
- Input attacks resisted by SHA1.
- Iterative guessing attacks resisted by two pools.
- Backtracking attacks resisted by rekeying 3DES every few output blocks.

OPEN ISSUES

- Characterizing sources and reliably measuring entropy.

- Porting to new cryptographic primitives, e.g., AES.

- Integrating in hardware noise source without loss of security.