Entropy Sources – Industry Realities and Evaluation Challenges

Sonu Shankar and Ashit Vora

September 12th 2013

Agenda

- Introduction and Background
- Typical CSPRNG Implementation
- NIST SP 800-90B
- Annex-D NDPP v1.1
- Entropy Sources - Design Examples
- Industry Status Quo
- Conclusions
- Looking Ahead...

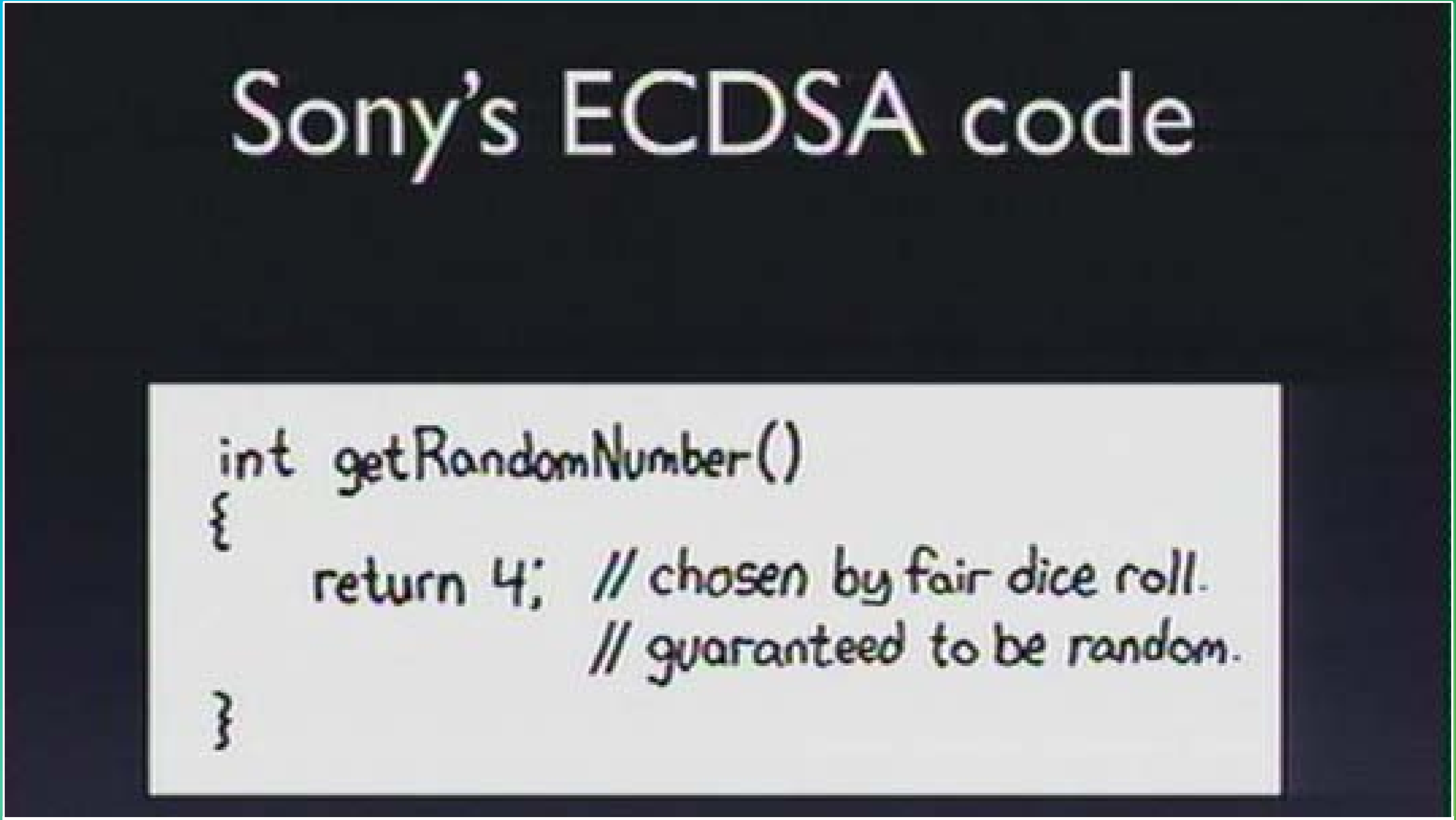
Introduction and Background

Random numbers and security

- Cryptographic key generation (MACSec, IPSec, SSH, TLS ...)
- Nonces and initialization vectors (802.11i, EAP, MACSec...)
- Padding schemes, signatures (DSA, OTPs...)
- Using poor random numbers (random != unique) can have catastrophic consequences. And cause severe embarrassment!
- Repeating primes in public keys (Research by Nadia Heninger et al 2012)

  59,000 duplicate keys were repeated owing to use of poor random numbers (1% of all certificates, or 2.6% of self-signed certificates)

  ~25,000 SSL keys were factorable by computing the GCD of all pairs of RSA moduli
- Recovery of ECDSA private key (fail0verflow hack CCC 2010)



Ouch!

(fail0verflow ECDSA Sony PS3 hack – CCC 2010)

TRNGs, PRNGs, CSPRNGs

- True Random Number Generator (TRNG)
  - Generates unpredictable, statistically independent, irreproducible bits (wow! Ideal)
  - Sampling/digitization of naturally occurring physical phenomena (a.k.a slow :-/)
- Pseudorandom Number Generator (PRNG)
  - Deterministic algorithm that generates output closely resembling a TRNG
  - Requires a *seed* to initialize underlying deterministic model (State Space >> Seed Space)
  - Designed for simplicity/performance. Not secure(!!)
- Cryptographically Secure PRNG (CSPRNG)
  - Computationally complex PRNG (May use cryptographic hashes, ciphers)
  - Non-trivial, computationally infeasible to accurately infer internal state (backward/forward secrecy)
  - But wait... still requires a seed (periodically) that possesses high "Entropy"

Entropy

- Uncertainty associated with a random variable; The expected value of information contained in a message (Claude E. Shannon 1948)
- Entropy is how we quantify unpredictability. (What space must an adversary search to determine a key?)
- For a random variable X with n outcomes,

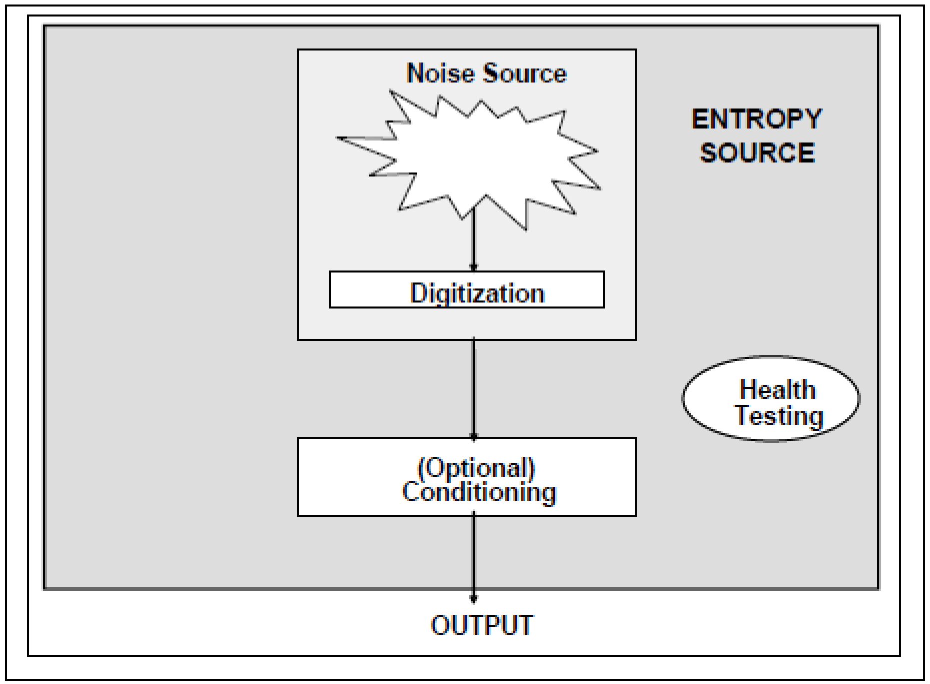
$$\{x_i : i = 1, 2, \ldots, n\}, \mathrm{H(X)} = -\sum_{i=0}^{n} P(x_i) log_2 P(x_i)$$

- More relevant parameter in practice - "Minimum Entropy" $-H_{min} = \; log_2(P_{max})$
- Most likely output has probability 1/4 = 2 bits of min-entropy

Typical CSPRNG Implementation

Typical CSPRNG Implementation

- Entropy source
  - Noise source - Hardware-based, dependent on electronic noise (thermal/Johnson, shot) - Ring Oscillator jitter or electronic metastability
  - Conditioner (to reduce bias) - SHA-1 engine, Linear Feedback Shift Register, Yuval Peres (von-Neumann) corrector
  - Health tests - Continuous tests on noise source, repetition count, adaptive proportion
- Software CSPRNG implementation
  - SP 800-90A DRBG (For example, a CTR-DRBG based on AES-256)
  - Seeded (and reseeded periodically) using the entropy source - 256 bits at least every 2^48 requests
  - Health tests - Continuous random number generator test



NIST SP 800-90B Draft August 2012

NIST SP 800-90B

NIST SP 800-90B

- First formal recommendation that describes required properties, design and testing for entropy sources
- Entropy Source = Noise Source + (Optional) Conditioner + Health Tests
- Statistical tests to assess min-entropy for both i.i.d and non-i.i.d sources
- Raw noise sampling necessary to perform tests
- Health tests mandatory on raw noise samples (continuously)

Annex-D NDPP v1.1

Annex-D NDPP v1.1

- Design Description

  Documentation that describes the design of the entropy source as a whole, interaction of all components including post-processing.

  Design review is critical to ensure the source is robust enough to perform with a certain min-entropy estimate when deployed in large numbers (> ~1 million Cisco Aironet 3600 APs deployed. IP phones?)

- Entropy Justification

  Technical argument describing source of unpredictability and justification of probabilistic behavior.

  Specify expected entropy rate and process of seeding underlying CSPRNG

- Operating Conditions

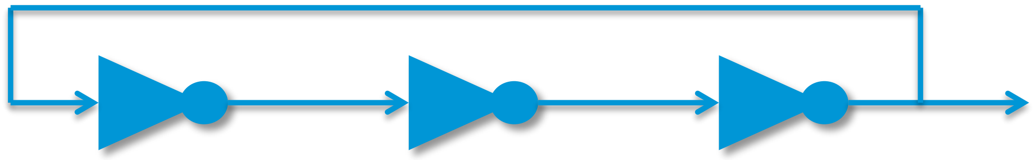
  Information on operational ranges (temperature, operational voltage) within which normal operation can be expected

- Health Tests

  Describe entropy source health tests, rate and conditions under which performed, results expected and justification for use

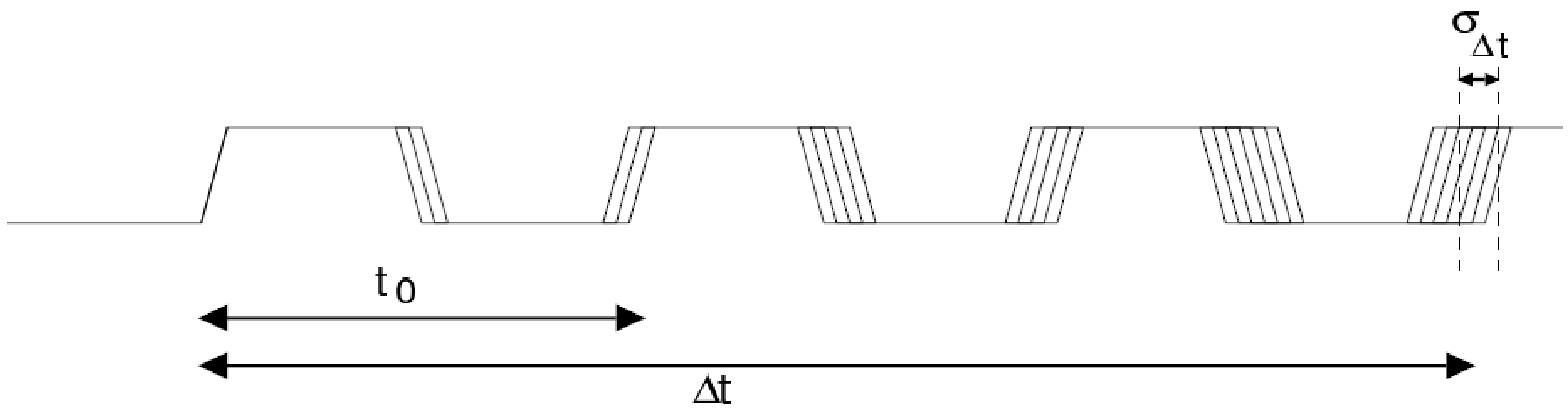
Entropy Sources - Design Examples

Ring Oscillators



- Odd number of NOT gates connected in series with a wire inversion
- Output oscillates between two voltage levels
- Oscillations begin spontaneously above a threshold voltage

Ring Oscillators (continued)



- Property exploited for entropy – Ring oscillator jitter (fluctuations in oscillator period due to electronic noise)
- Jitter causes increasing uncertainty in signal transition times

Ring Oscillator Jitter

Boltzmann constant

Temperature

Supply voltage

$$\sigma_{\Delta t} \approx \sqrt{\frac{8}{3\eta}}\sqrt{\frac{kT}{P}\frac{V_{\mathrm{DD}}}{V_{\mathrm{char}}}}\sqrt{\Delta t}$$

Proportionality constant

Time after oscillation start

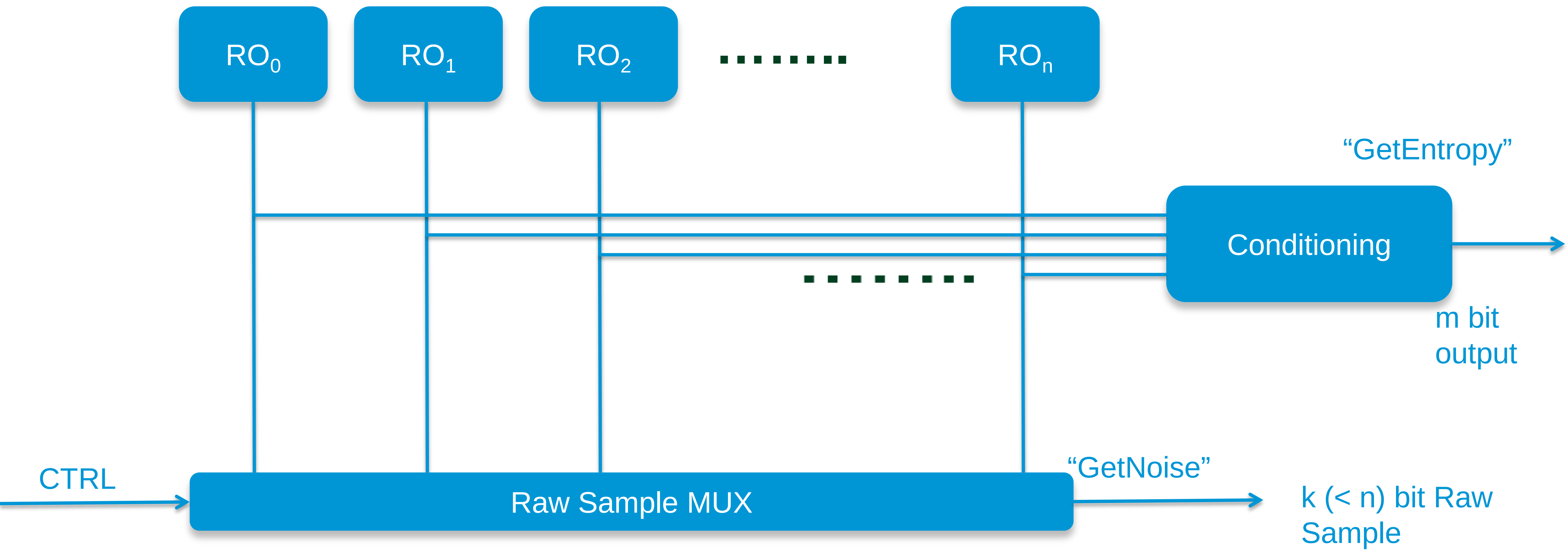
Characteristic voltage

Jitter standard deviation

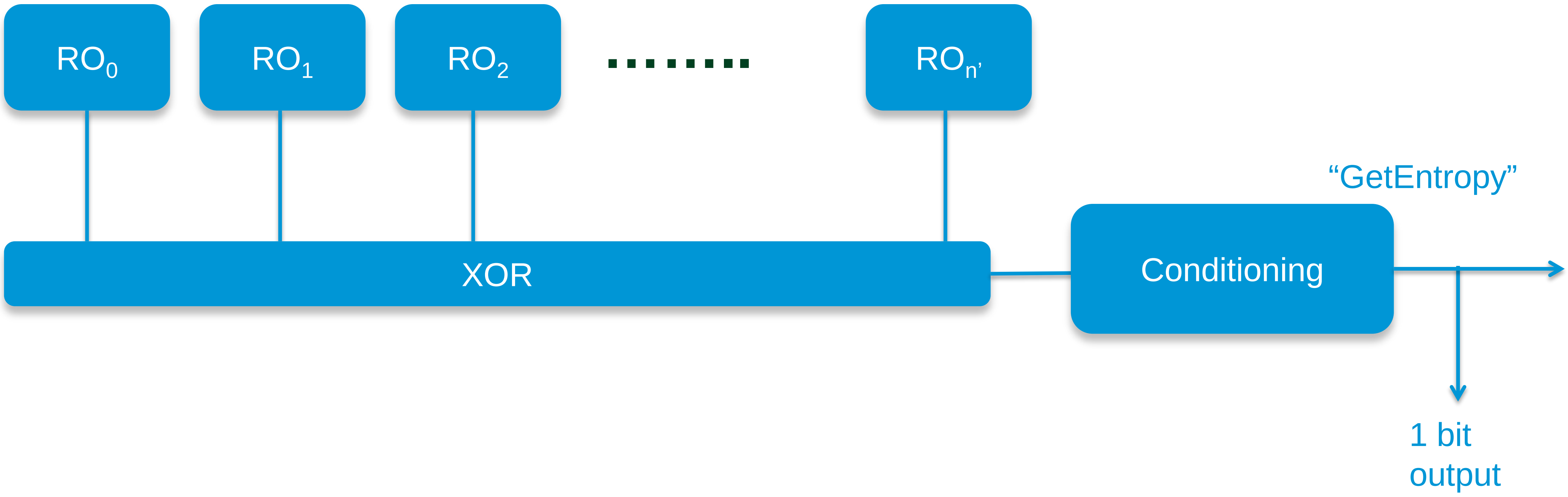
Power consumption

- Above is for a single-ended CMOS RO derived by Hajimiri et al. – “Jitter and Phase Noise in Ring Oscillators”, *IEEE J. Solid-State Circuits 34(6) (1999) 790-804.* (Reference for equation and figure on prev slide)

Generic RO-based Design 'A'



Generic RO-based Design 'B'



Design 'C'

Design 'C'

?

Industry Status Quo

Industry Status Quo

- Most designs are based on Ring Oscillator jitter (Some on electronic metastability)
- Dedicated entropy sources are rare. Always part of another multi-purpose chip.
- Access to raw unconditioned noise is not always available (800-90B? Errrr...)
- When available, consecutive raw sampling of entire conditioner input is non-trivial
- Noise source health tests ("equivalent" to 800-90B recommendations) are rare in hardware
- Design/architecture details are not always made available

Conclusions

Conclusions

- Requirements applicable across a wide range of applications, deployment scenarios and computational capabilities
- Parts (and vendors), designs, quality, ability to assess vary across devices
- Certain robust, well-designed entropy sources currently don't have access to raw noise (No min-entropy assessment for you!)
- Certain vendors are uncomfortable sharing any detailed description of noise source design (some do not provide any details whatsoever). Multi-party NDAs not practical

Looking Ahead...

Looking Ahead…

- Ideally, BOTH min-entropy estimation and design review required for a thorough evaluation
- However, a deep design review as part of the evaluation will not be scalable (Min-entropy estimation + noise source health test requirements should suffice)
- The burden of performing detailed design review should lie with the network device vendor to ensure robust operation in specific applications and deployments
- Need for a dialog between Industry and Government about requirements that provide security assurance while considering real world issues such as IPR, device capabilities, deployment scenarios etc

Looking Ahead… (continued)

- Industry (design and policy changes)
  - Access to raw noise is essential for min-entropy statistical analysis as well as noise source health tests (New silicon = Adoption delays ~2-4 years)
  - To facilitate sound entropy analysis, vendors will need to be open to sharing (at least) high-level design descriptions enough to justify a robust, reliable, probabilistic source of noise
- Government (policy changes)
  - Several designs exist in the industry (Noise - Ring Oscillator jitter, RS latch metastability, Post-processing - Cryptographic hash, Von Neumann corrector, linear codes)
  - Concerns around level of IP required currently to pass design review
  - A single min-entropy estimation tool should be made publicly available

Thank you.