

Random Numbers and Mobile Devices

Jan Krhovjak, Petr Svenda, Vashek Matyas

Faculty of Informatics
Masaryk University, Brno
Czech Republic

- Basics on random number generation
 - True- & pseudo- random number generators
 - Specifics of mobile devices
- Analysis of selected sources
 - Entropy estimation (theory & practice)
 - Fundamentals
 - Our methodology
 - Results of experiments (Nokia, E-TEN)
 - Microphone input
 - Camera input
- Distributed generating
 - Basic ideas
 - Chicken-and-Egg problem



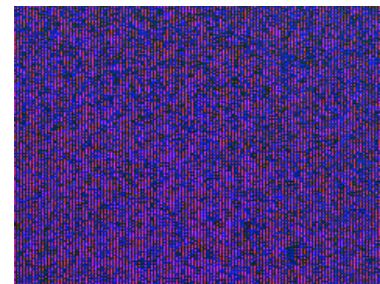
Basics on random number generation

- Random data in cryptography
 - Cryptographic keys, padding values, nonces, etc.
 - Quality (statistical testing) and unpredictability is critical
- Generating truly random numbers
 - Based on nondeterministic physical phenomena
 - Radioactive decay, thermal (white) noise, etc.
 - In deterministic environments hard and slow
- Generating pseudorandom numbers
 - Based on deterministic algorithm
 - Short input (seed) – truly random data
 - Output – pseudorandom data, computationally indistinguishable from truly random data
 - Hybrid semi-deterministic approach
 - Truly random data periodically improves inner state or pool



Specifics of mobile devices

- True random number generator
 - Quality strongly dependent on source of randomness
 - Possibility of influencing by attacker
 - General purpose computer systems
 - Many sources exist (hardware/software based, user inputs)
 - Mobile devices
 - Typically located only inside the chip (SIM card)
- Mobile device-dependent sources of randomness
 - Based on specific HW components of device
 - Microphone, digital camera, touchable LCD, battery level
 - Based on mobile nature of device
 - Information about current location, strength of transmitted signal (or other signal characteristics)



Entropy estimation – fundamentals

- Basic measure for randomness is called *uncertainty* or *entropy* (average-case)
 - $H_1(X) = -\sum_{x \in X} P_X(x) \log P_X(x)$
 - Sample x is drawn from random distribution X with probability $P_X(x)$
 - Logarithm base corresponds to units (2 => bits)
- Better measure is *min-entropy* (worst-case)
 - $H_\infty(X) = \min_{x \in X} (-\log P_X(x)) = -\log(\max_{x \in X} P_X(x))$
 - Always “less than” or “equal to” Shannon entropy
- How many random bits are extractable per time unit?

Entropy estimation – our methodology

- Obvious problem
 - Sampled data can be (and typically are) non-uniform
 - This typically implies unrealistic entropy estimation
- Ideal solution
 - Remove all stat. defects & dependences
 - Entropy estimation and extraction (or vice versa)
- Our *attacker-aware* methodology
 - Data captured with as little noise as possible
 - Computing of histograms, probabilities, entropy estimation
 - Statistical testing (NIST, auto- and cross- correlation)
 - Success: no impact to entropy estimation
 - Failure: simple “entropy extraction” and/or new lower estimation

Selected mobile devices & initial thoughts

- Programmable mobile devices
 - Smartphone Nokia N73 (Symbian OS, JavaME)
 - PDA phones E-TEN M700 & X500 (WM5/WM6)



- Some low-level sources have no sufficient precision (API restrictions) or have a slow refresh frequency
 - Battery level and signal strength (only ten values scale)
 - GPS position (only one measurement per second)

Sources of randomness – mobile devices

- Sound input

- Embedded or hands-free (only N73) microphone
 - Different sensitivity (dependent on solidity of membrane)
 - Different modulation method, sampling frequency



- Optical input

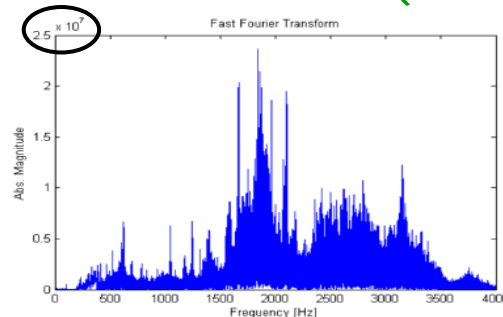
- Array of semiconductor photo-sensors
 - Several chip designs (CCD, CMOS, EMCCD, ICCD, etc.)
 - Different resolution, sensitivity, noise level, exposure time
- More than 6 sources of noise
 - Thermal noise, readout noise, amplification noise, quantization/discretization noise...



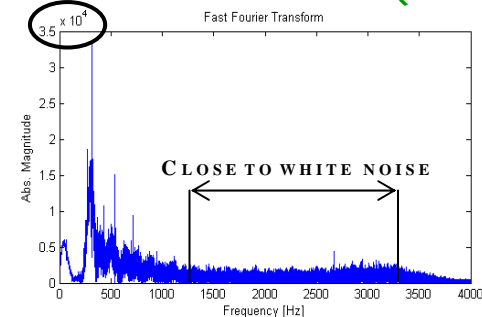
Microphone input (Nokia N73)

- Nokia N73 voice input (16-bit PCM, 8000 Hz, ~16 kB/s)
 - Focused on noise originated in microphone (in quiet room)
 - Sharp peak on the beginning of each sample removed
 - Histograms of noise recordings yields limit values
 - 12/13 (embedded mic.) & –9/8 (hands-free mic.)
 - Min-entropy (hf): **0.5** bits; Shannon entropy (hf): **2.9** bits
 - Upper bound of entropy (hf): $\log_2 18 = \mathbf{4.2}$ bits
 - Autocorrelation shows some dependencies
 - Taking only every 2nd, 3rd, 4th value decreased correlation
 - Lower estimated entropy – at least 5×
- Additional spectral analysis (FFT/DFT)

Embedded mic. (music)



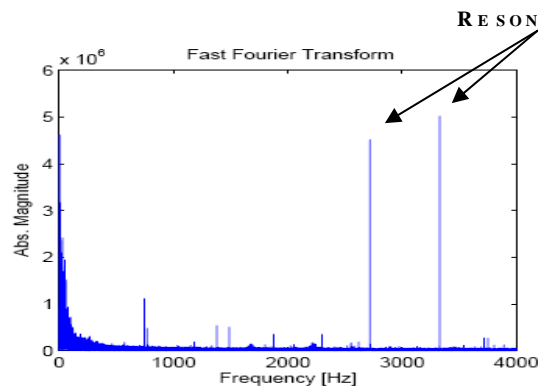
Hands-free mic. (noise)



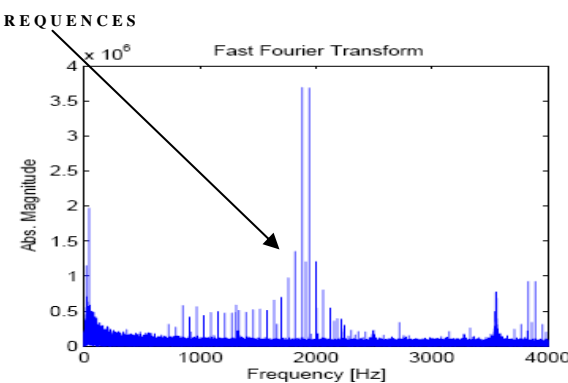
Microphone input (E-TEN M700/X500)

- E-TEN M700/X500 voice input (16-bit PCM, 8000 Hz)
 - Extremely sensitive microphones (higher variance)
 - No sharp peak on the beginning of each sample
 - Histograms of noise recordings yields limit values
 - Min-entropy: **0.016** bits & **0.023** bits
 - Shannon entropy: **10.1** bits & **9.4** bits
 - Upper bound of entropy: **11.8** bits & **10.6** bits
 - Autocorrelation (similar results)
- Additional spectral analysis of noise (FFT/DFT)

E-TEN M700

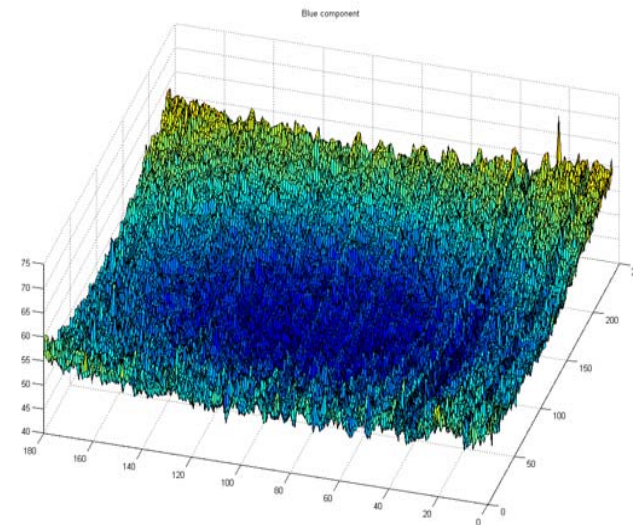
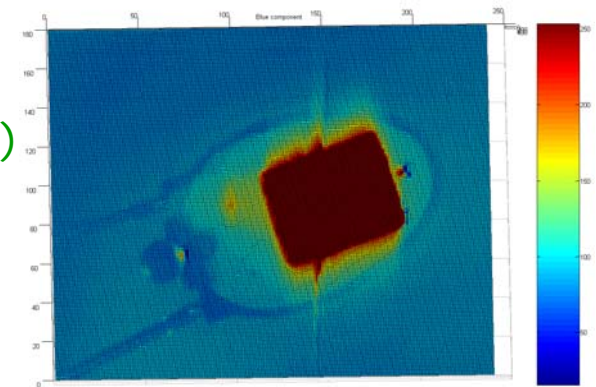


E-TEN X500



Camera input (Nokia N73)

- Nokia N73 uses CMOS based 3.2 Mpix camera
 - View finding instead of high-resolution picture
 - No post-processing
 - noise reduction, compression
 - Fast data acquisition (12 fps, ~1600 kB)
 - 1 frame, 240×180 pixels, ~130 kB
 - Closed camera cover
 - Defense against overexposure =>
 - Temperatures 5 to 45 °C
- Systematic defects in camera image
 - Sensor technology & post-processing
 - Avg. value of (blue color component)
 - Hot pixels around borders
 - Significant rips in the rows
 - Centered circle rips
 - Different intensity towards centre



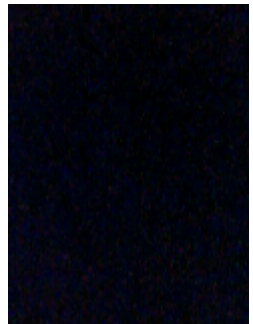
Camera input (Nokia N73) entropy estimation

- Independency of pixels in image (& between images)
 - Matlab *corrcoef* cross-correlation function
 - Pixels in the same row and column
 - No significant correlation found
 - Matlab auto-correlation and FFT/DFT
 - Vector of values taken in time from single pixel (12 fps)
 - No significant correlation & almost uniform spectrum
 - NIST test suite
 - Bit-streams generated from R/G/B pixel values
 - Green component passed all tests
- Entropy estimation for single pixel (5 °C)

Color	Shannon entropy [bit]	Minimal entropy [bit]
Red	3.9	3.2
Green	4.0	3.3
Blue	4.8	3.9

Camera input (E-TEN M700/X500)

- E-TEN M700/X500 devices use 2.0 Mpix camera
 - Special video driver for capturing screen via USB
 - Faster retrieving and storing of images (resolution 240×320)
 - Precision reduced from 8 to 5/6 bites per pixel
 - No camera cover; covered lens => turning camera off
 - Different camera sensors
 - Visually different noise (confirmed by stat. analysis)
- Effects of temperature (from 8 °C to 50 °C)
 - 35 °C and more => significant influence
 - Noise prevents even software camera switch off
 - It allows capturing of closed camera cover images
 - Automatic ISO level correction
 - Expected: decreasing temp. => decreasing noise level
 - Observed: decreasing temp. => increasing noise level
 - Only to certain level, then decreasing as expected



Camera input (E-TEN M700/X500) entropy estimation

- Data captured at 8 °C
- No significant correlations found
- E-TEN M700

Color	Shannon entropy [bit]	Minimal entropy [bit]
Red	4.5	3.0
Green	4.3	2.9
Blue	5.4	3.7

- E-TEN X500

Color	Shannon entropy [bit]	Minimal entropy [bit]
Red	3.1	1.2
Green	3.1	2.1
Blue	2.5	0.7

Conclusion to randomness sources

- Microphone & camera input have great potential
 - Big throughput and inherently presented internal noise
 - Estimated entropy for **microphone** input [bits per second]
 - Based on min-entropy & reduction by factor 5
 - Nokia N73: **800**; E-TEN M700: **25.6**; E-TEN X500 is **36.8**
 - Estimated entropy for **camera** input (huge numbers)
 - Three different colors, min-entropy between 0.7 and 3.9, resolutions 240×180 or 240×320 => more than 100 000s bits **in one** picture
- Subject of our next research
 - **Quantitative analysis** (and impact to entropy)
 - Microphone: spectral analysis
 - Camera: defects due to sensor, pre- & post-processing
 - **Entropy extraction techniques**
 - Towards to provable security against active adversaries

- Local generation
 - Generating truly random (and pseudorandom) numbers
 - TRNG: sampling of nondeterministic physical phenomena
 - PRNG: deterministic algorithm (the input == seed)
 - Security relies on quality of used sources of randomness
 - Mobile phones – e.g., noise in audio/video input
 - Standalone mobile device can be under attack
- Distributed generation
 - Involving several remote (mobile) devices
 - At the beginning – always a consuming device
 - At the end – always a generating device or computer
 - Trusted and semi-trusted (mobile) devices
 - Problem is assessing quality of received random data
 - Reputation-based entropy estimation

Gathering random data in distributed environments

- The more generating parties \Rightarrow the lower probability to influence all of them
 - Different devices and their environments
 - Different communication paths
- Two basic approaches
 - Obtaining random data per explicit requests
 - Random data as product of ongoing communication
- Consuming device can obtain not only post-processed but also a raw random data
 - Inherent for ongoing audio/video conferences
 - Post-processing performed on the consuming device
 - Cryptographic PRNGs, entropy extractors...

The main problem

- The defense for all attacker types always degrade to securing communication links/paths (M-P protocols)
 - Link security (owner of infrastructure)
 - End-to-end security (user applications)
- Chicken-and-Egg problem
 - Transfer of random numbers must be always secured
 - Classical crypto. mechanisms require random numbers
- There is no way, how to break this circle!
 - All encrypted data can be viewed as a pseudorandom
 - With the seed == encryption key
 - The “entropy” in obtained data is limited
 - We can use these data only for some applications

Issues regarding the entropy...

- Group entropy & pooling
 - Sharing cryptographic keys among more participants
 - Distinct pools for each communication group
- Remote entropy sources
 - Secured audio/video streams (known by all parties)
 - Secure transformation to different streams
 - PRNG or cipher with different keys? Entropy is ???
- Can we “cheat” or somehow “avoid” the circle?
 - Without need additional random number...
 - Without securing the path by the encryption...

- Post authentication by audio-visual means
 - Ability to recognize user face/voice and other behavioral characteristic
 - Visual checking of D-H values
- Steganography
 - Hiding data with additional entropy into the stream
 - Only legitimate recipient can extract such random data
- Anonymity
 - Hiding the recipient of random data
 - Attacker does not know who is the victim