

The Sources of Randomness in Smartphones with Symbian OS

Jan Krhovják, Petr Švenda, Vašek Matyáš,
Luděk Smolík

Faculty of Informatics
Masaryk University, Brno

- Basics on random number generation
 - True- & pseudo- random number generators
 - Specifics of mobile devices
- Analysis of selected sources on Nokia N73
 - Entropy estimation
 - Microphone input
 - Camera input
- Practical pseudorandom number generator
 - Performance comparison



Basics on random number generation

- Random data in cryptography
 - Cryptographic keys, padding values, nonces, etc.
 - Quality and unpredictability is critical
- Generating truly random numbers
 - Based on nondeterministic physical phenomena
 - Radioactive decay, thermal 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
- Quality assurance – statistical testing

- 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)
 - Better categorization
 - External & internal environment (+ mutual interactions)

- 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)
 - How many random bits is extractable per one time unit?
- Attacker can force source to produce most probable values => those values contains minimum entropy
 - 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 then or equal then Shannon entropy

Microphone input

- Selected device: smartphone Nokia N73
 - Symbian OS, JavaME, good camera, etc.

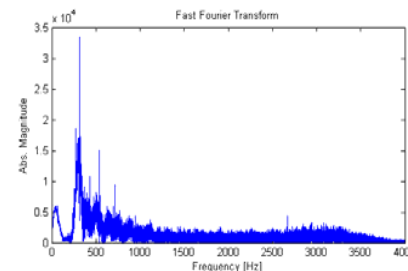
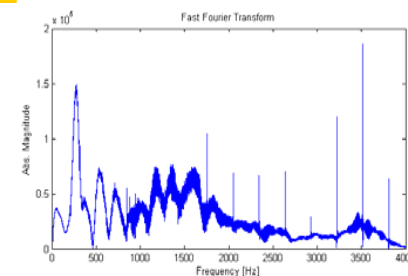
- Nokia N73 voice input

- Embedded or hands-free microphone
- Modulation method, sampling frequency => ~ 16 kB/s
 - 16-bit pulse coded modulation (a signed PCM)
 - Sampling a sound wave at frequency 8000 Hz

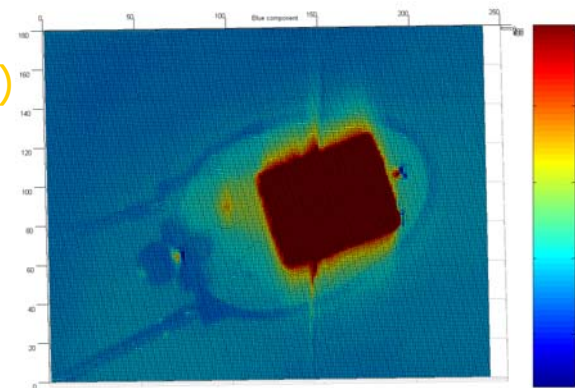


- Entropy in input sound signal

- Focused on noise originated in microphone
- Basic analysis (embedded/hands-free)
 - Fast/discrete Fourier transform => quality
 - Histogram analysis => upper bound

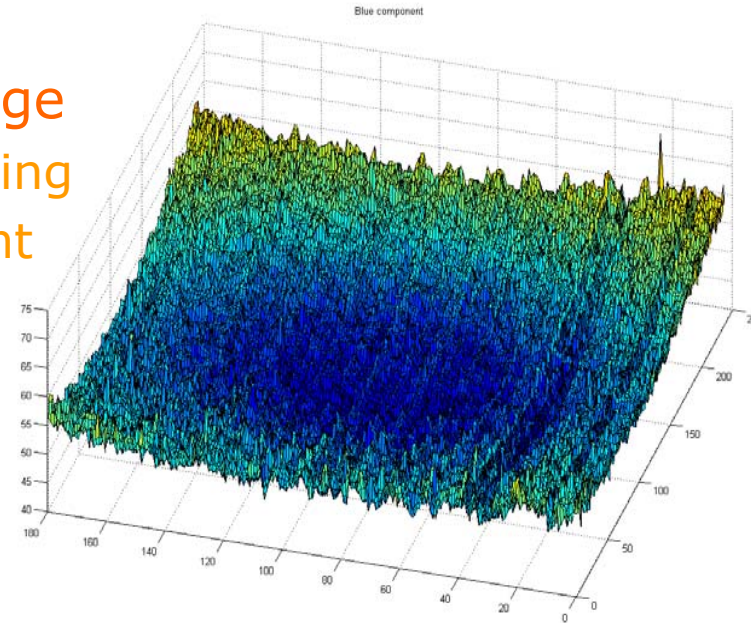


- Digital optical input devices
 - Array of semiconductor photo-sensors
 - Several chip designs
 - CCD, CMOS, EMCCD, ICCD, etc.
 - Different sensitivity, noise level, exposure time
 - More than 6 sources of noise
 - Mostly thermal noise => sensitivity to temperature
 - Higher temp. == higher noise
- 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 °C to 45 °C



Camera input entropy estimation

- 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



- Independency of pixels in image (& between images)
 - Matlab *corrcoef* cross-correlation function [OK]
 - Neighboring pixels & pixels in the same row
 - Matlab auto-correlation and FFT/DFT [OK]
 - Vector of values taken in time from single pixel (12 fps)
 - NIST test battery [green component always passed]
 - Bit-streams generated from R/G/B pixel values

- Pseudorandom number generator
 - Often based on cryptographic primitives (AES, SHA-xxx)
 - Serve as fast entropy extractors
 - No mathematical guarantee of security
 - Amount of raw data from sources limited by the performance of mobile device
- Performance comparison (tested on SHA-1)
 - Nokia N73 (Symbian v9.1) ~ 2200.00 kB/s
 - Nokia N73 (JavaME) ~ 426.00 kB/s
 - Sony-Ericsson k750i (JavaME) ~ 84.00 kB/s
 - Nokia 6230 (JavaME) ~ 67.00 kB/s
 - Nokia 6021 (JavaME) ~ 4.65 kB/s

- Mobile device contains several randomness sources
 - 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)
 - Other sources seems to be suitable
- Analysis of selected sources on Nokia N73
 - Microphone & camera input have great potential
 - Big throughput and inherently presented internal noise
 - Min-entropy (upper bound) is 2/4 bits per audio sample/subpixel
 - Our analysis found several defects in camera input
 - Due to sensor technology & post-processing
 - Statistical tests of random data from camera noise promising
 - Symbian OS performance significantly higher than JavaME
 - Possibility to extract entropy from high throughput sources