GE vs GM: Efficient side-channel security evaluations on full cryptographic keys

CHES 2022

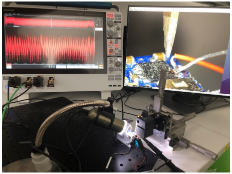
Anca Rădulescu, PG Popescu and Marios Choudary

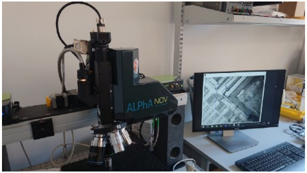
Leuven, 21 September 2022

Thanks Christ, the UPB team and Virgil Gligor from CMU


(The UPB campus – left: our Church; right: the rector offices)

Side-channel attack security evaluations





Images from https://medium.com/@charles.guillemet/ledger-donjon-3e04e0ce49a9

SCA evaluations necessary:

- During product manufacturing to assess security of products
- For governments, to establish some required standards
- For security industry (e.g. automotive, banking) to ensure that third-party products (e.g. smartcards) have a sufficient level of security
- To obtain a uniform level of security certification (e.g. Common Criteria EAL4+)

SCA security evaluation tools for short data (e.g. key byte)

- Commonly used security level estimation metrics: Success Rate (SR), Guessing Entropy (GE) aka Rank
- Less common (yet...): Massey's Guessing Entropy (GM)
- A mess of guessing entropy measures and notations
  - 1994: James Massey proposes $E[G]$
  - 1997: Christian Cachin terms it 'Guessing Entropy' $E[G(X)]$ and present conditional version $E[G(X|y)]$
  - 2007: Köpf and Basin use the conditional guessing entropy in the context of side-channel attacks
  - 2009: FX Standaert et al. present (empirical) Guessing Entropy in framework for SCA evaluations
- Bigger problem: GE and GM both run in $O(N \log N)$
  - Do not directly scale for large keys (impractical for $N > 2^{16}$)
  - We need special methods for full-key security evaluations

SCA security evaluation tools for full keys (e.g. 128-bit AES key, 4096-bit RSA key)

Two main approaches for full-key security evaluations:

- Key enumeration for large keys ([Charvillon et al. 2012, Poussier et al. 2016])
- **Security level estimation for large keys**:
  - Empirical Guessing Entropy (Rank) estimation ([Charvillon et al. 2013, Glowacz et al. 2015, Zhang et al. 2020])
  - Massey's Guessing Entropy (GM) bounds ([Choudary and Popescu 2017])

SCA security evaluation tools for full keys (e.g. 128-bit AES key, 4096-bit RSA key)

Our main goal – comparing full-key SCA evaluation tools:

- FSE'15 rank estimation [Glowacz et al. 2015]
  - One of the fastest GE estimation methods to date
  - Works well up to 256 key bytes, with good precision
- GM bounds [Choudary and Popescu 2017]
  - Mathematical, rigurous bounds for GM
  - Fastest and most scalable full-key evaluation method to date
  - Works with 1024-byte keys and beyond
- GEEA rank estimation [Zhang et al. 2020]
  - One of the newest methods for GE estimation on large keys
  - Lower STD than FSE'15

GM vs GE computation

$$\text{(Massey's)GM} = \frac{1}{N}\sum_{q=1}^{N}\sum_{i=1}^{|\mathcal{S}|} i \cdot P(k_i|X = \mathbf{X}_q)$$

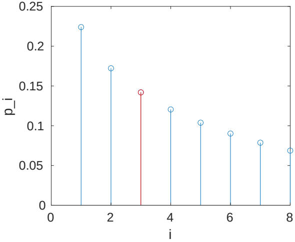
$$\text{(Empirical)GE} = \frac{1}{N}\sum_{q=1}^{N}\{\text{rank of } k\star \text{ in experiment } q\}$$

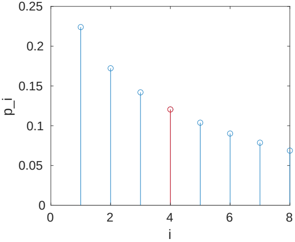
$$(P(k_1|\mathbf{X}_q) \geq \ldots \geq P(k_i|\mathbf{X}_q) = P(k\star|\mathbf{X}_q) \geq \ldots \geq P(k_{|\mathcal{S}|}|\mathbf{X}_q))$$

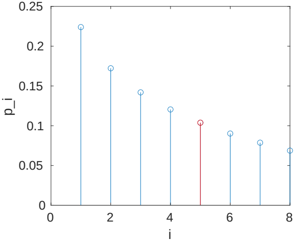
Observations:

- Same complexity (need to sort all the list of probabilities)
- Both dependent on acquired datasets ($\mathbf{X}_q$)
- Different use of probabilities
- GE requires knowledge of correct key, GM does not

GM vs GE simple example

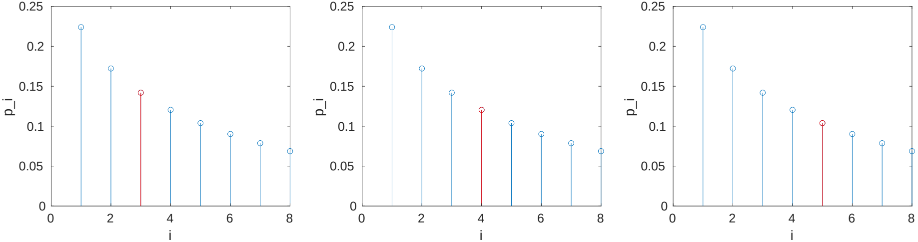






GM	3.63	3.63	3.63
GE	3	4	5

GM vs GE simple example



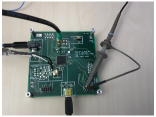
GM	3.63	3.63	3.63
GE	3	4	5

→ GE provides actual (empirical) estimation of rank

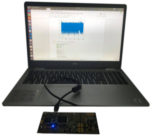
→ GM is generally a lower bound for GE [KB'07]

Experimental datasets

- We used three different datasets:
  - *Simulated* dataset (Hamming weight of AES S-box output mixed with Gaussian noise): $\mathbf{x}_i = \mathrm{HW}(\mathrm{S\text{-}box}(k \oplus p_i)) + r_i$
  - *XMEGA* dataset (AVR XMEGA AES engine)

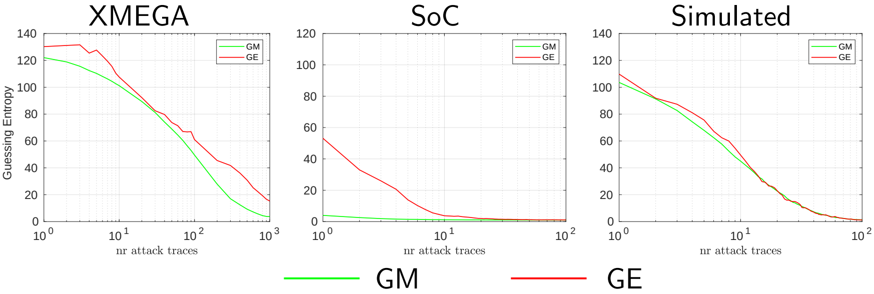


  - *SoC* dataset (ChipWhisperer-Lite with STM32F303 32-bit ARM)



- We used Template Attacks to obtain lists of probabilities for each AES key byte $(p_1, p_2, \ldots, p_{256})$

On the utility of GM



Observation 1: GM is generally a lower bound for GE
$\rightarrow$ Can be used to confirm security is above a certain treshold

Observation 2: we may combine both measures to determine the quality of a leakage model

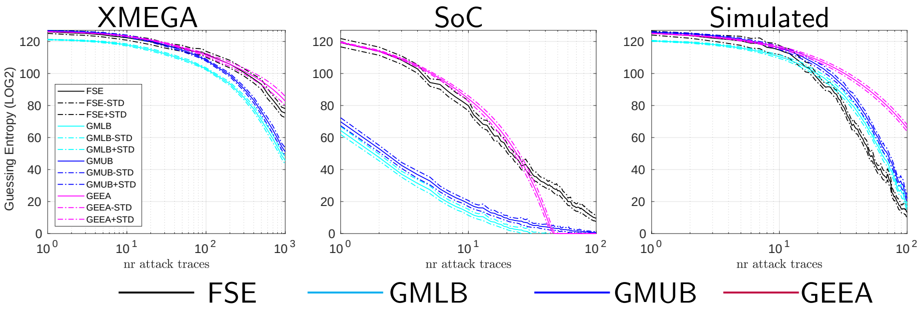
GM close to GE $\rightarrow$ good model (e.g. in *Simulated* dataset)

GM departs from GE $\rightarrow$ bad model (e.g. in *SoC* dataset)

Analysis of full-key evaluation tools

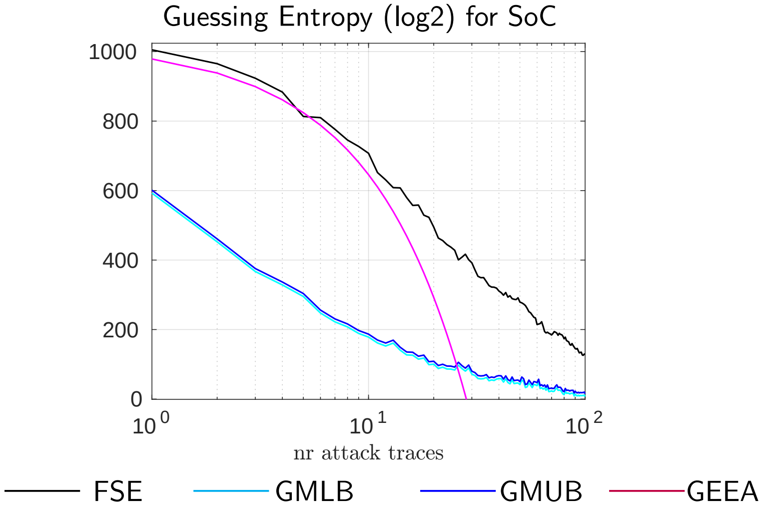
- We focus on the three representative methods
  - FSE'15 (Glowacz et al. 2015)
  - GM Bounds (Choudary and Popescu 2017)
  - GEEA (Zhang et al. 2020).

Precision analysis on 128-bit data (16-byte results)



	Median STD		
FSE'15	1.84	2.67	2.89
GM Bounds	0.74	1.34	2.22
GEEA	0.56	0.81	1.77

Scalability and usability analysis on larger data (128 bytes)



Scalability and computation analysis on large data (16/128/1024-byte results)

Computation time (s) for XMEGA/SoC/simulated

	16 bytes	128 bytes	1024 bytes
FSE'15	29/60/172	1027/5336/4689	Not practical
GM Bounds	1/1/1	2/6/6	40
GEEA ($M = 10^4, 10^6$)	17/18/26	432/415/473	Not practical

Overall analysis and usability guidelines

- FSE'15:
  - Good approximation of GE
  - Works well for up to 256 key bytes
  - Slow computation for large keys

Overall analysis and usability guidelines

- FSE'15:
  - Good approximation of GE
  - Works well for up to 256 key bytes
  - Slow computation for large keys
- GM Bounds:
  - Guaranteed, tight bounds for GM
  - (Typically) Lower bound for GE/FSE
  - Can be used with very large keys

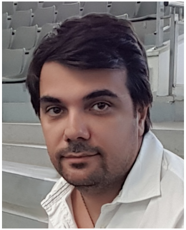
Overall analysis and usability guidelines

- FSE'15:
  - Good approximation of GE
  - Works well for up to 256 key bytes
  - Slow computation for large keys
- GM Bounds:
  - Guaranteed, tight bounds for GM
  - (Typically) Lower bound for GE/FSE
  - Can be used with very large keys
- GEEA:
  - High accuracy (low STD)
  - Deviates from GE/FSE within similar computation time
  - Needs more analysis to provide some guarantees

Overall analysis and usability guidelines

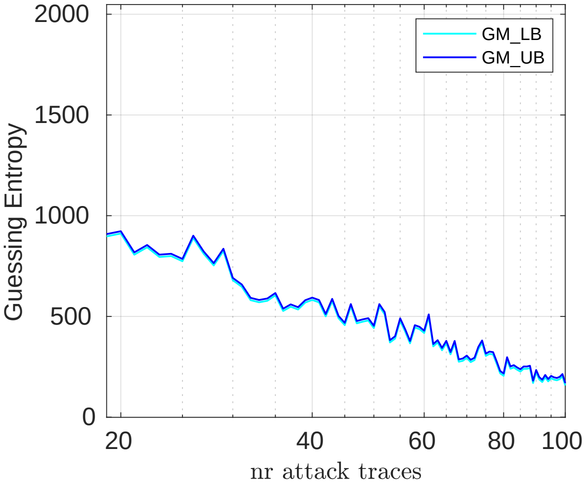
- Conclusions:
  - Use GM Bounds for a very fast security evaluation (lower bound) – works with very large keys https://gitlab.cs.pub.ro/marios.choudary/gmbounds
  - Use FSE'15 or other GE estimation algorithm for accurate estimate of key rank
  - (Optionally) Use a key enumeration algorithm to output list of keys in decresing probability

Greetings from the UPB (GM Bounds) Team

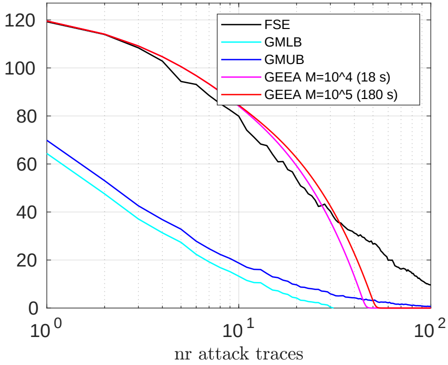




Appendix

GM Bounds (log2) on 1024-byte key (SoC data)



GEEA with varying amount of data (SoC, 16 bytes)



- GEEA computation on large keys uses random selection of subkey computations (comparison vectors)
- Needs very large M (large computation) to approach GE/FSE
- May not be able to follow GE within given computing power