Computational Fuzzy Extractors

Benjamin Fuller

Joint work with Xianrui Meng and Leonid Reyzin

August 17, 2013

How Should People Authenticate?

Passwords?

Lots of evidence that passwords don’t have enough entropy for crypto

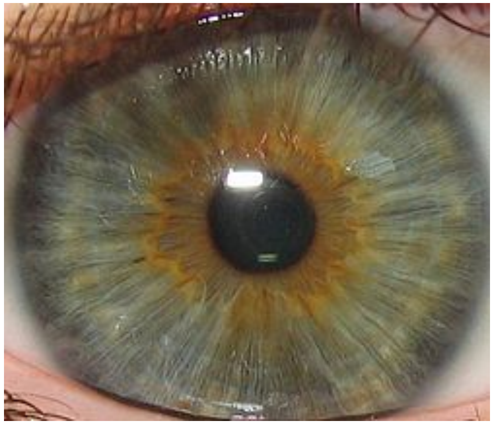
iPhone PIN	Frequency	iPhone PIN	Frequency
1234	10.713%	9999	0.451%
1111	6.016%	3333	0.419%
0000	1.881%	5555	0.395%
1212	1.197%	6666	0.391%
7777	0.745%	1122	0.366%
1004	0.616%	1313	0.304%
2000	0.613%	8888	0.303%
4444	0.526%	4321	0.293%
2222	0.516%	2001	0.290%
6969	0.512%	1010	0.285%

datagenetics.com

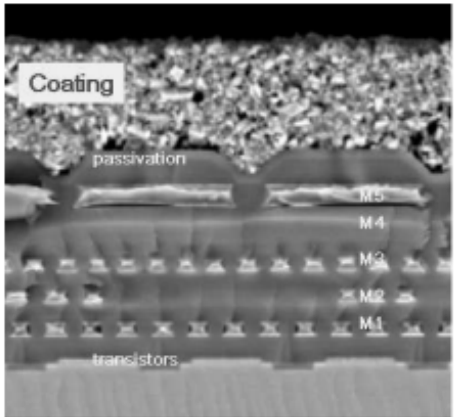
Biometrics/Physical Unclonable Functions?

High entropy, but suffer from noise

Current techniques for removing noise impose large entropy losses and prevent use in authentication



wikipedia.org



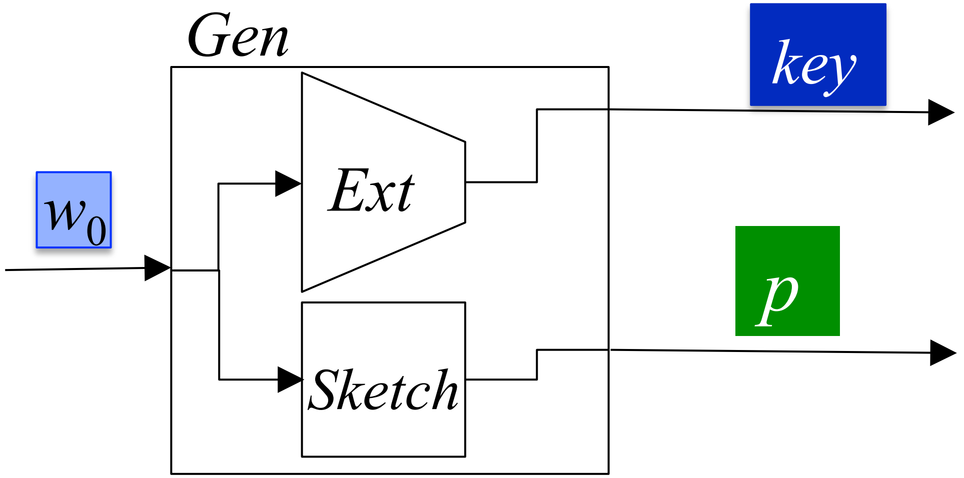
Tuyls et al. 2006

Fuzzy Extractors

Fuzzy Extractors derive reliable keys from noisy data

[DodisOstrovskyReyzinSmith08]

- Correctness: Gen, Rep give same key if w_0 and w_1 are "close"
- Traditional Construction
  - Derive key with *extractor*
  - *Error-correct* with *Secure Sketch*

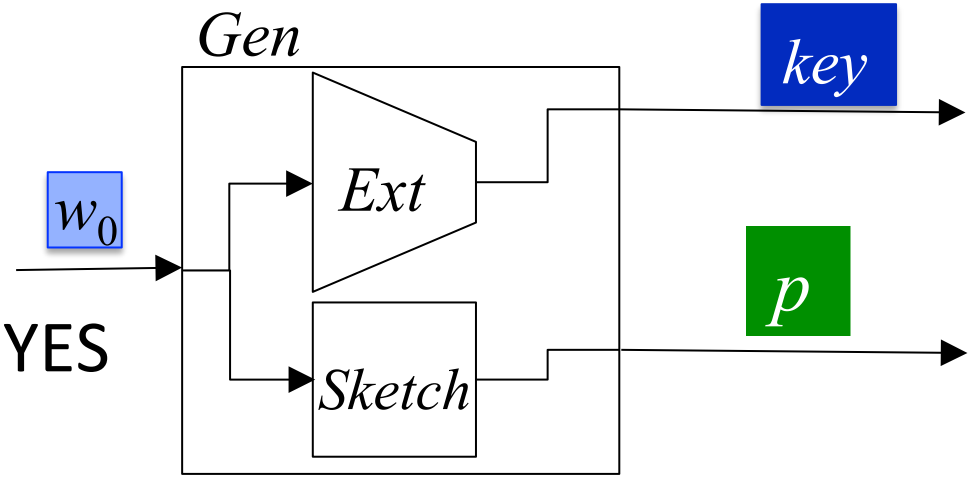


- Security info-theoretic: key close to uniform conditioned on p
- Entropy losses prevent adoption (for irises there is 0 entropy after using a secure sketch)

Can we do better in computational setting?

Can we do better in computational setting?

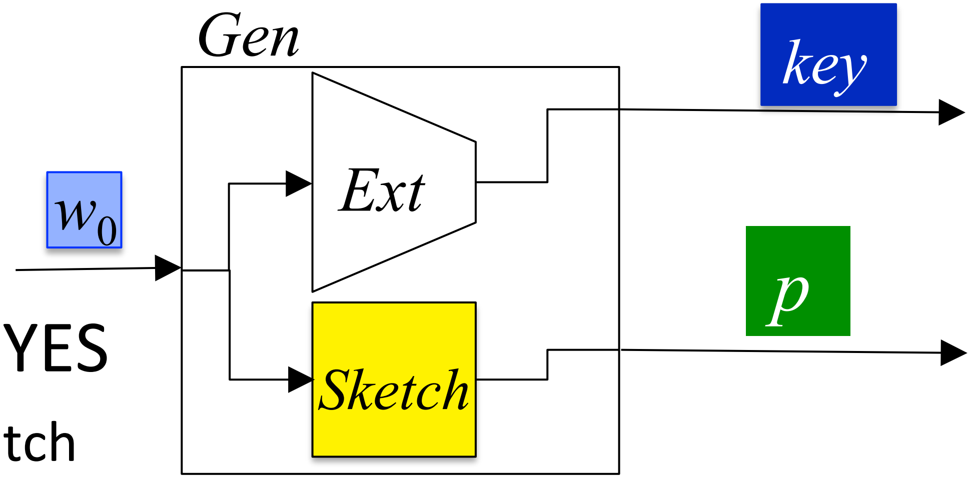
- Using sketch-and-extract: NO
  - <u>Thm:</u> Defining secure sketches using computational entropy is unlikely to help



- Using a new construction: YES

Can we do better in computational setting?

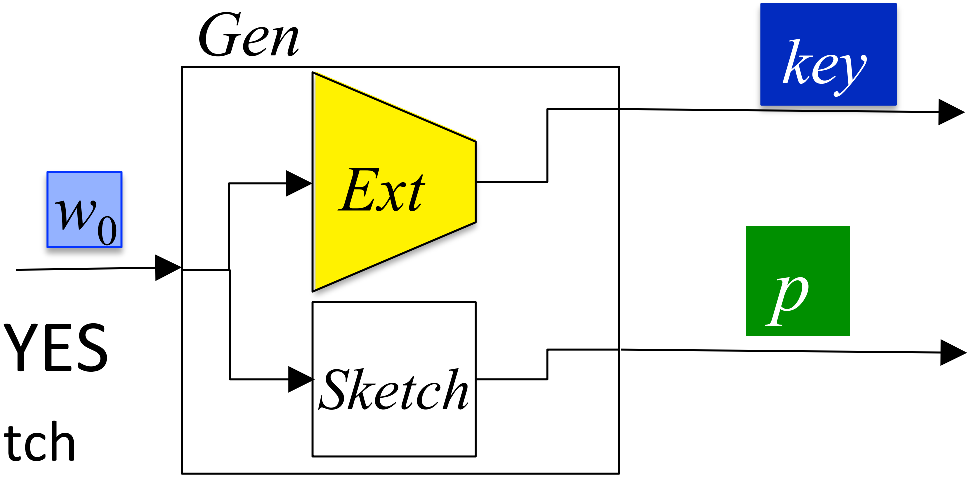
- Using sketch-and-extract: NO
  - Thm: Defining secure sketches using computational entropy is unlikely to help



- Using a new construction: YES
  - Know we can't change the sketch

Can we do better in computational setting?

- Using sketch-and-extract: NO
  - Thm: Defining secure sketches using computational entropy is unlikely to help

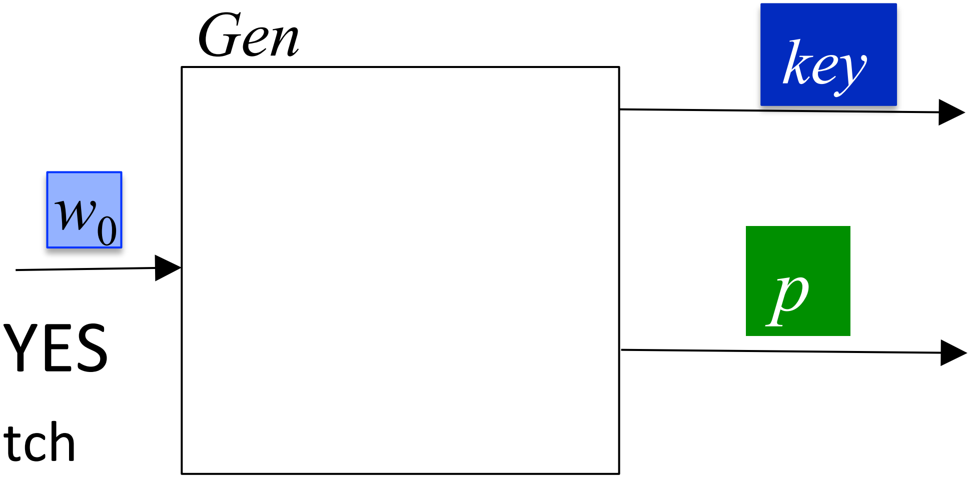


- Using a new construction: YES
  - Know we can't change the sketch
  - Could use computational extractor

    (Must have enough entropy remaining after the sketch)

Can we do better in computational setting?

- Using sketch-and-extract: NO
  - Thm: Defining secure sketches using computational entropy is unlikely to help

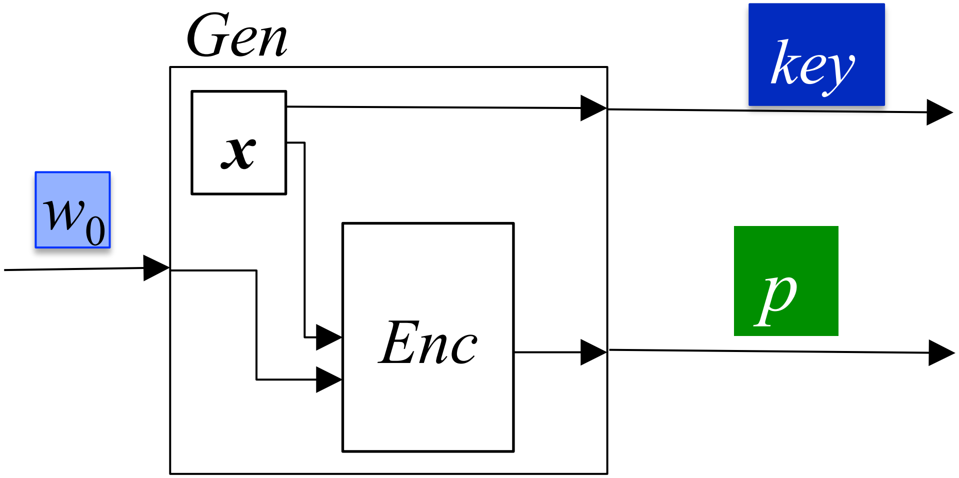


- Using a new construction: YES
  - Know we can't change the sketch
  - Could use computational extractor

    (Must have enough entropy remaining after the sketch)
  - We make the whole process computational

Computational Fuzzy Extractor

- Key idea: instead of trying to hide w_0, we use private randomness x as our key
- Encrypt x using w_0
- Need encryption algorithm that allows decryption from close w_1
- Our encryption algorithm is the "code-offset" secure sketch instantiated with random linear code (security from LWE)
- First fuzzy extractor where $|key|$ independent of error tolerance



Open Problems

- Show security for arbitrary high entropy sources
- Support higher error rates

Thanks!

To appear at Asiacrypt ‘13
Available:

http://eprint.iacr.org/2013/416