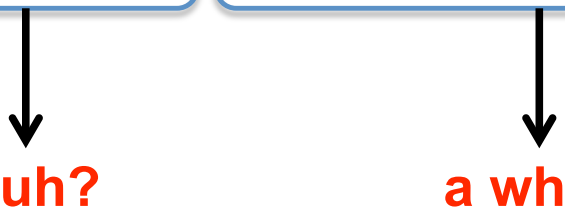
An efficient fuzzy extractor for limited noise

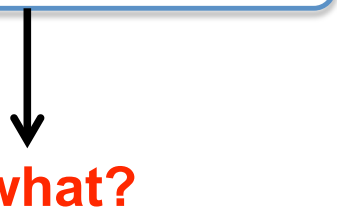
Boris Škorić and Pim Tuyls

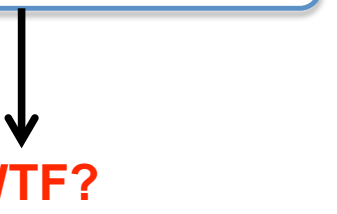
WIC Symposium
28-29 May 2009

Outline

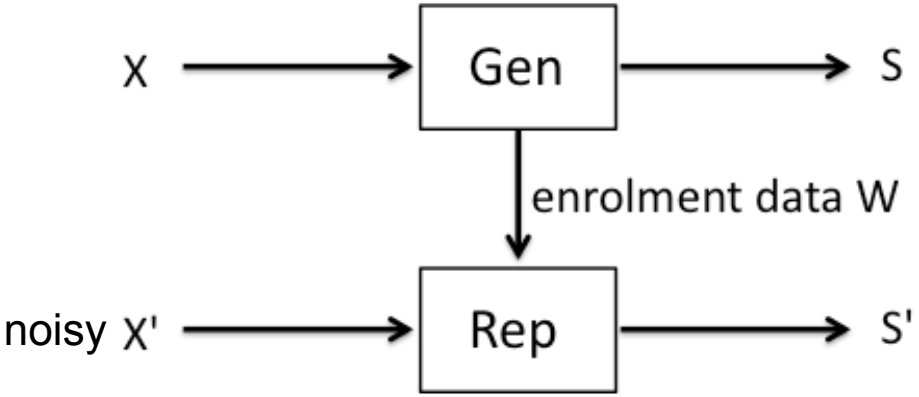
An efficient fuzzy extractor for limited noise







Fuzzy Extractor



Dodis et al. 2003
Juels+Wattenberg 1999
Linnartz+Tuyls 2003

Properties

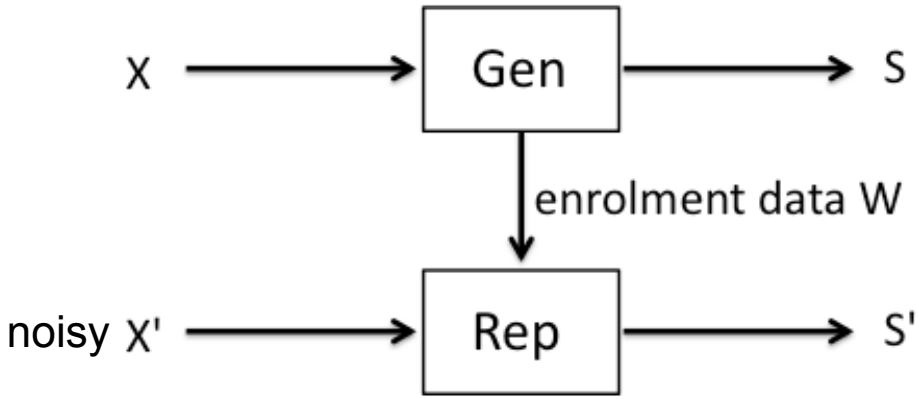
- Secrecy and uniformity: **Δ(WS; WU) ≤ ε**.
  "S given W is almost uniform"
- Error correction: **If X' sufficiently close to X, then S'=S**.
- Robustness [Boyen et al. 2005]:
  Detection of active attack against W

Applications

- privacy preserving biometrics
- anti-counterfeiting ("object biometrics")
- PUF-based key storage



Fuzzy Extractor: Efficiency



What's so special?

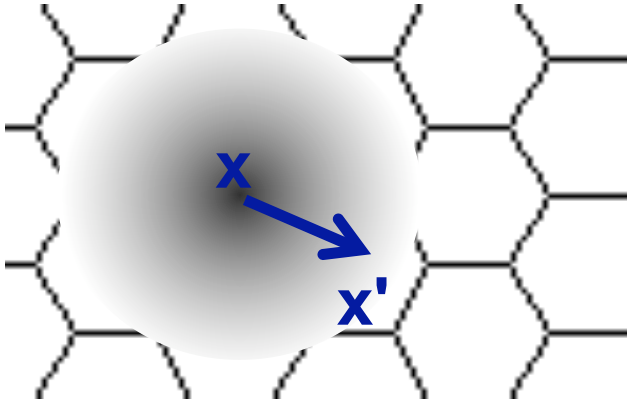
- Redundancy data (in W) must not leak info about secret S.
- Make near-uniform S from non-uniform X.
- How to authenticate W when there is no trusted authority?

"Efficiency"

- Extract as many reproducible bits from X as possible.
- Low storage requirements.
- Small computational load.

Limited noise

Example



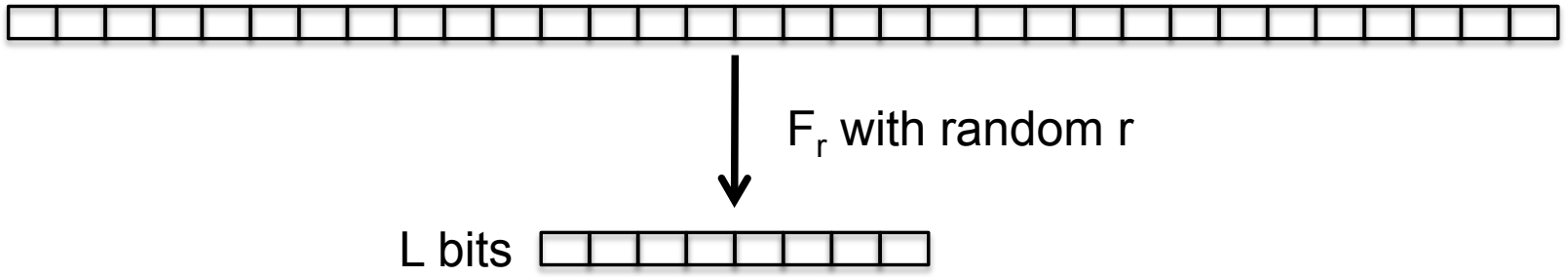
Common class of noise

- Considerable prob. that x' ≠ x.
- Small number of likely x'.

Problematic for error correcting codes

- Codes work best with low error rate
- Cannot exploit non-uniform error patterns (low entropy of errors)
- Entropy loss.

Universal hash functions

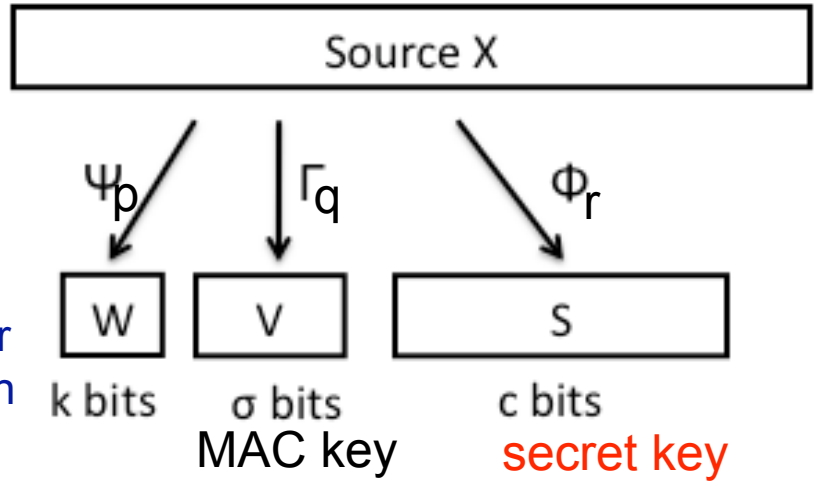


- Def: δ-**almost universal** hash functions F_r.
  For fixed x and x':

$$\text{Prob}[F_R(x) = F_R(x')] \leq 2^{-L}(1+\delta)$$

- Not a cryptographic hash
- Main purpose: uniformity
- Light-weight implementation in hardware and software.
- Information-theoretic properties.
- Does not rely on unproven security assumptions

Fuzzy Extractor based on universal hash functions



Publicly stored enrolment data:
p,q,r,w, m:=MAC(v; pqrw)

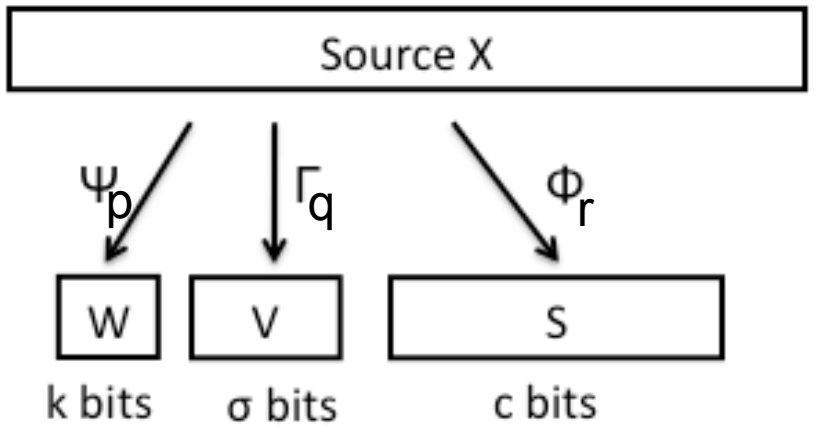
attack

p', q', r', w', m'

Key reconstruction procedure

- Measure x'. Read p', q', r', w', m'.
- Make list L of likely candidates.
  - Must be manageable!
- Find x in L such that $\Psi_{p'}(x)=w'$.
  - Sort of Slepian-Wolf
- Compute $v'=\Gamma_{q'}(x)$.
- Check if MAC(v'; p'q'r'w')=m'.
- If okay, reconstruct secret $s=\Phi_{r'}(x)$.

Fuzzy Extractor based on universal hash functions



- All three security functionalities achieved by universal hashes!
  - ➢ error correction
  - ➢ uniform key
  - ➢ manipulation detection key
- This scheme exploits low entropy of error patterns.

Theorem: If $c \leq \max_{\rho} \left[\mathsf{H}_2^{\rho}(X) + 2 - \log \frac{1}{\varepsilon(\varepsilon - \rho) - \delta/4} \right] - k - \sigma$

then Δ(PQRWM **S**; PQRWM **U**) ≤ ε .

Conclusions

Fuzzy extractor is necessary security primitive for:

- privacy preserving biometrics
- anti-counterfeiting ("object biometrics")
- PUF-based key storage

Construction based on (almost-)universal hash functions

- Slepian-Wolf coding
- Only works for "limited" noise
- Less entropy loss than error-correcting code
- Efficient to implement