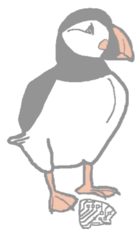
The spammed
Code Offset Method

Boris Škorić
Niels de Vreede

PUFFIN workshop
3 Nov. 2013



eprint.iacr.org/2013/527

Outline

- Helper data schemes
  - privacy-preserving biometric databases
  - Physically Obfuscated Keys
- The Code Offset Method
- Adding fake enrollment data
  - while retaining efficient reconstruction
  - LDPC codes, syndromes, ...
- Analysis
  - security
  - storage/work

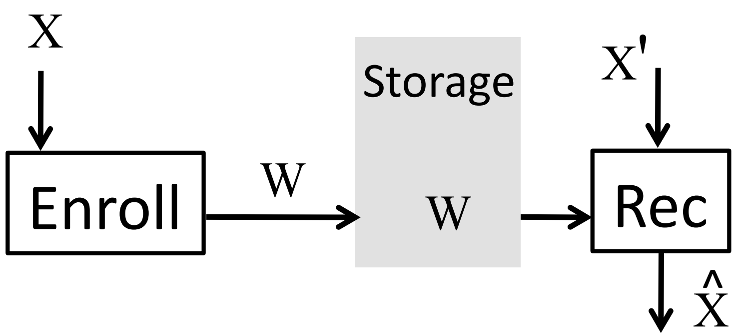
trade-off

Scenario 1: privacy-preserving biometrics DB

Aim: store only the hash of a user's fingerprint/iris/...

Problem: **noise**

Solution: helper data scheme (Secure Sketch)



Desired properties:

- High prob. of correct reconstruction.
- W does not leak much about X.

Database entry:

W	**salt**	**Hash(W,salt,X)**

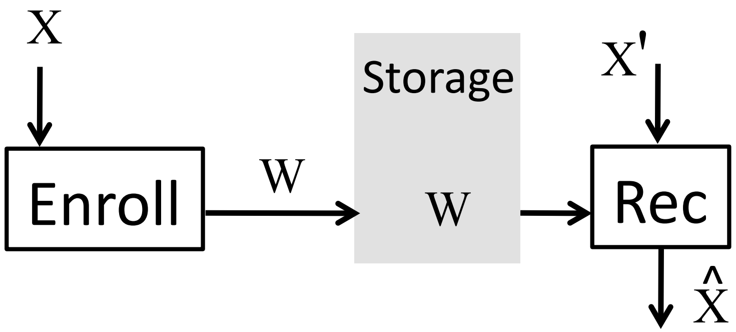
Figure of merit: $H(X|W)$

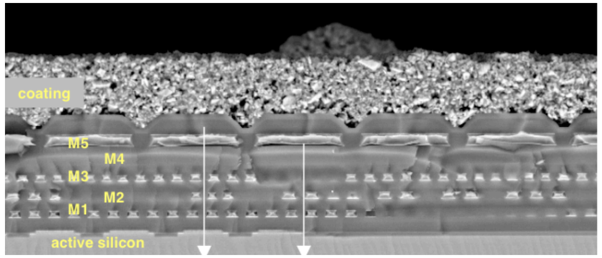
Scenario 2: Physically Obfuscated Key

Aim: Alternative technology for read-proof key storage.
Obtain key from measurement on complex physical system ("PUF").

Problem: **noise**

Solution: helper data scheme





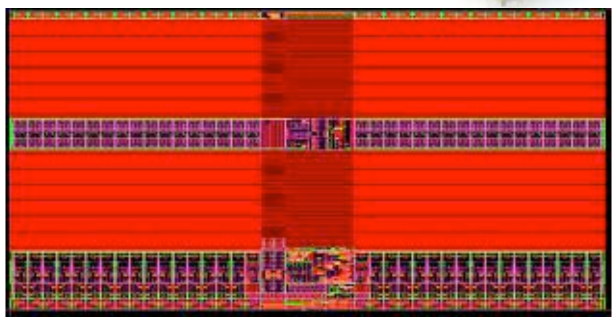


Figure of merit: $H_2(X|W)$

Intermezzo: Error-correcting codes

k-bit message μ.

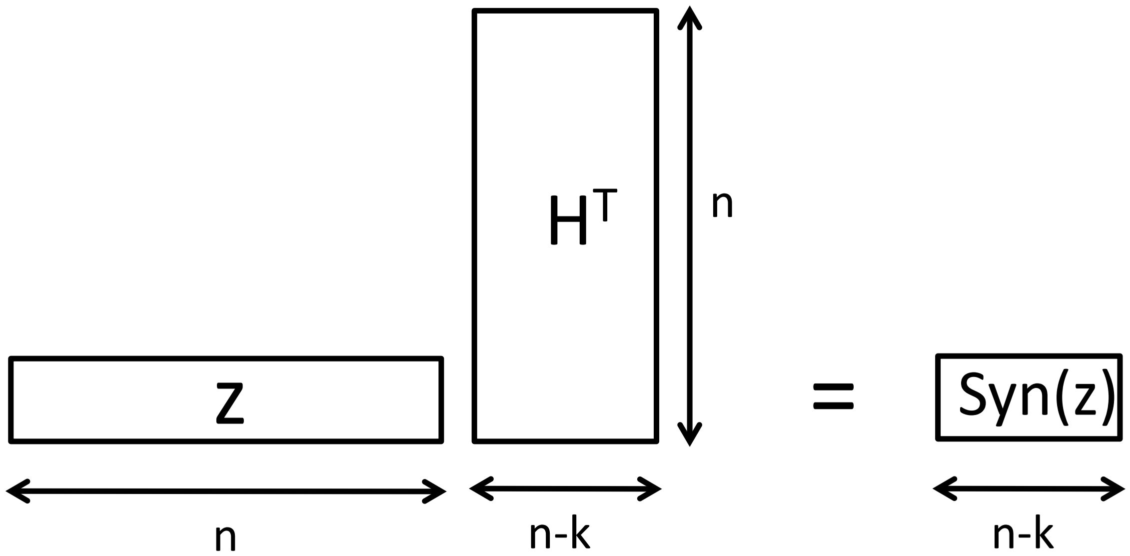
n-bit codeword C_μ.

n-bit noise pattern e.

$z = C_\mu + e$

Syndrome $Syn(z) = Syn(C_\mu) + Syn(e) = Syn(e)$

$$\boxed{C_\mu H^T = \underline{0}}$$



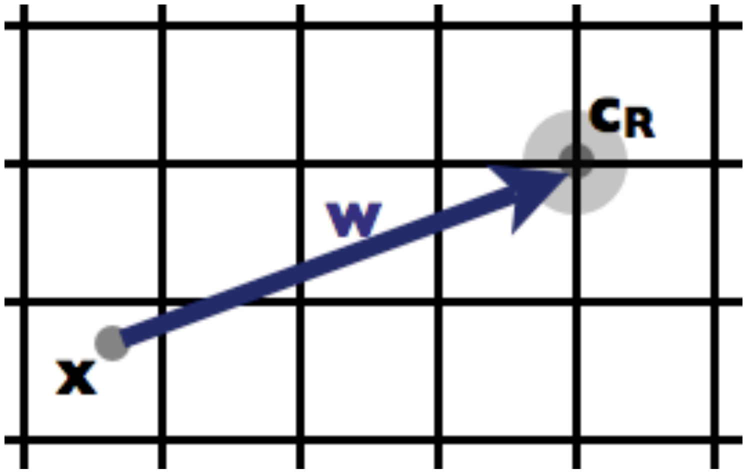
"<u>L</u>ow-<u>D</u>ensity <u>P</u>arity <u>C</u>heck" matrix [Gallager 1960]

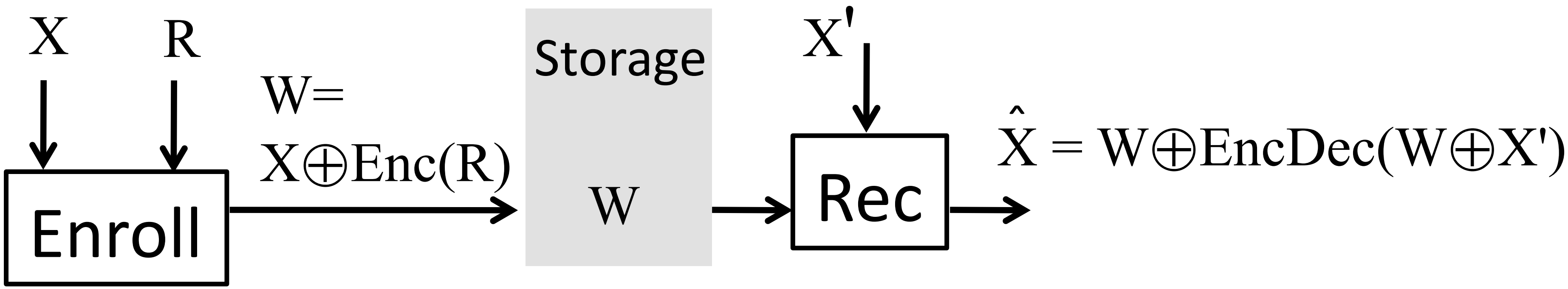
Code Offset Method

[Juels & Wattenberg 1999]

"The mother of all Secure Sketches"

- Source $X \in \{0,1\}^n$.
- Uniformly random $R \in \{0,1\}^k$.
- Binary linear error correcting code (Enc, Dec).
  Message size k; codeword size n.



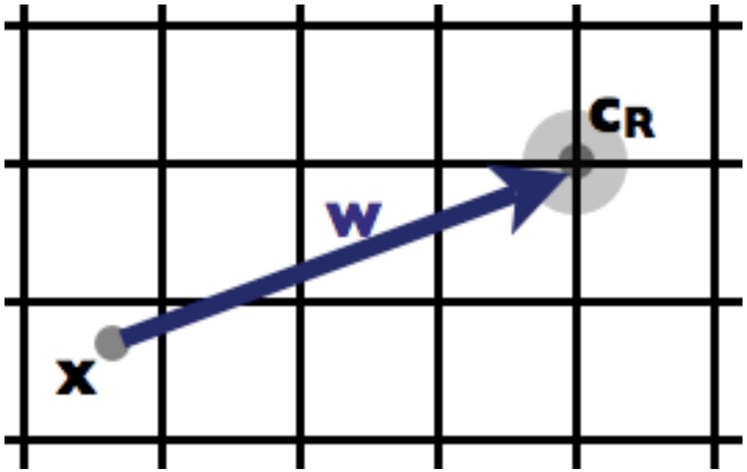


Code Offset Method: analysis

How good is this?

If X is uniform:

- H(R|W)=H(R); **no leakage about R!**
- H(X|W) = H(R) = k
  W leaks n–k bits about X



If X is *not* uniform

- W leaks about R
- W still reveals Syn(X)

Can we do better?

Fake helper data

Idea: hide W in lots of fake helper data (with same distribution)

Biometrics database, entry for one user:

<u>Legitimate party</u>:

- Has X'
- Reconstruction by brute force: Try all entries

<u>Attacker</u>:

- Does not have X'
- Brute force attack
- effort multiplied by m/2 $\Longrightarrow$ log(m/2) bits of security gained

More efficient scheme

- Use LDPC code
  - parity check matrix is sparse
  - $X' \approx X$ implies $\mathrm{Syn}(X') \approx \mathrm{Syn}(X)$
- Store $\mathrm{Syn}(X)=\mathrm{Syn}(W)$ and all $\mathrm{Syn}(W^{fake})$
  - can be computed from W and W^{fake}
  - reveals nothing new
  - Code Offset Method possible with only syndrome

$\mathrm{Syn}\ W_1^{fake}$	•••	$\mathrm{Syn}\ X$	•••	$\mathrm{Syn}\ W_{m-1}^{fake}$	Hash(table,X)

Fast reconstruction:
- Compute $\mathrm{Syn}(X')$
- Prioritize entries with $\mathrm{Syn}(W_i) \approx \mathrm{Syn}(X)$.

Security analysis

Without spam: H(X|W) = H(Syn X)

With spam: $\mathsf{H}(X|\mathbf{\Omega}) \geq \mathsf{H}(X|W) + \log m - \frac{m-1}{\ln 2}\mathbb{E}_x q_{\mathtt{Syn}(x)}$

$$\mathsf{H}(X|\mathbf{\Omega}) \geq \mathsf{H}(X) - \frac{1}{m} \cdot \frac{2^{n-k}-1}{\ln 2}$$

Ω: the helper data list

q_a = Prob[Syn X=a]

Typically, $\frac{m-1}{\ln 2}\mathbb{E}_x q_{\mathtt{Syn}(x)}$ is of order $\frac{m}{2^{n-k}}$

m → 2^{n-k}: Leakage gets close to zero.

The size of the table (assuming LDPC)

biometrics (1 user) *phys. obfuscated key*

		k=64			k=128	
#err	n	log m	**Mem**	n	log m	**Mem**
1	72	4	**16 B**	138	5	**40 B**
		8	**0.3 KB**		10	**1.2 KB**
2	78	7	**0.2 KB**	146	9	**1.1 KB**
		14	**29 KB**		18	**0.6 MB**
3	85	10.5	**3.8 KB**	154	13	**26 KB**
		21	**5.5 MB**		26	**0.2 GB**

- *n values are approximate*
- *Listed values for log m: (n-k)/2 and n-k*
- *Choose m that fits in memory* $\Longrightarrow$ **sec. gain log(m)-1 bits**

Summary

We added a new "knob" to the Code Offset Method

- better use of source entropy
- price: size of enrollment data

trade-off

- security analysis: Shannon entropy
- Rényi entropy [not shown]
- interesting for low source entropy

Work in progress:

- explicitly choose LDPC codes
- *generate* the table (with PRNG)
  - security ↔ memory tradeoff becomes security ↔ work tradeoff

Acknowledgements

- Ruud Pellikaan
- Ludo Tolhuizen