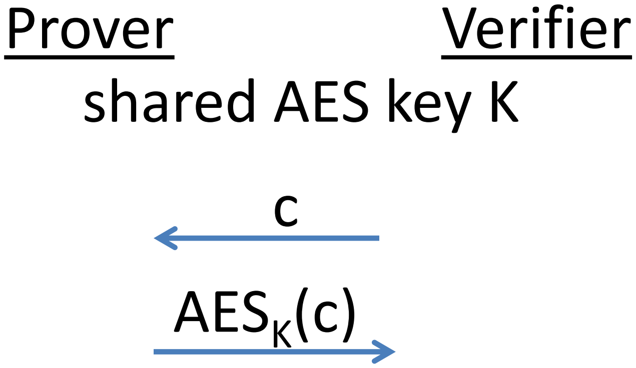
Lapin

(an efficient authentication protocol based on Ring-LPN)

Stefan Heyse, Eike Kiltz,
Vadim Lyubashevsky,
Christof Paar, Krzysztof Pietrzak

Authentication Protocols



Prover Verifier

shared AES key K

c

$\mathrm{AES}_K(c)$

HB-style authentication
protocols based on LPN

suitable for *light-weight* authentication

Lightweight Authentication - Motivation

Lightweight authentication has many applications

- "We need security with less than 2000 gates for RFID tags"
  Sanjay Sarma (MIT AUTO-ID Labs) @ CHES 2002

- $3 trillion damage annually due to product piracy*

  → replacement parts and devices need authentication

*Source: www.bascap.com

- Remote keyless entry systems for buildings, cars…

Lightweight Authentication - Motivation

- Many embedded applications are very cost-sensitive
  → we need **lightweight** authentication

- Since ≈ 2006 a lot of research on lightweight ciphers (PRESENT and many other proposals)

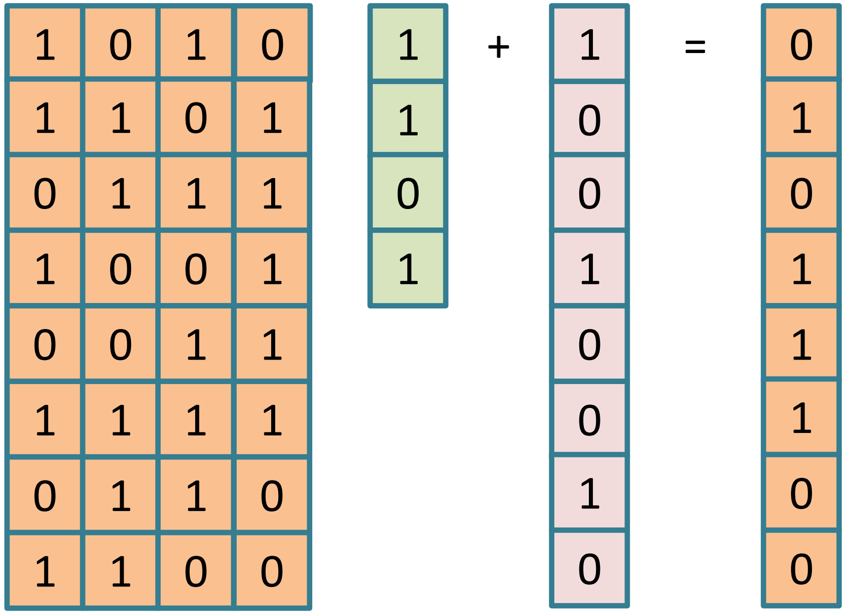
- All previous lightweight ciphers…
  - are optimized for hardware complexity (gate count), even though the vast majority of embedded applications run in software / firmware
    → very small code attractive for many applications
  - are not based on hardness assumptions

Learning Parity with Noise (LPN)

We have access to an oracle who has a secret **s** in $\mathbf{Z}_2^n$

On every query, the oracle:

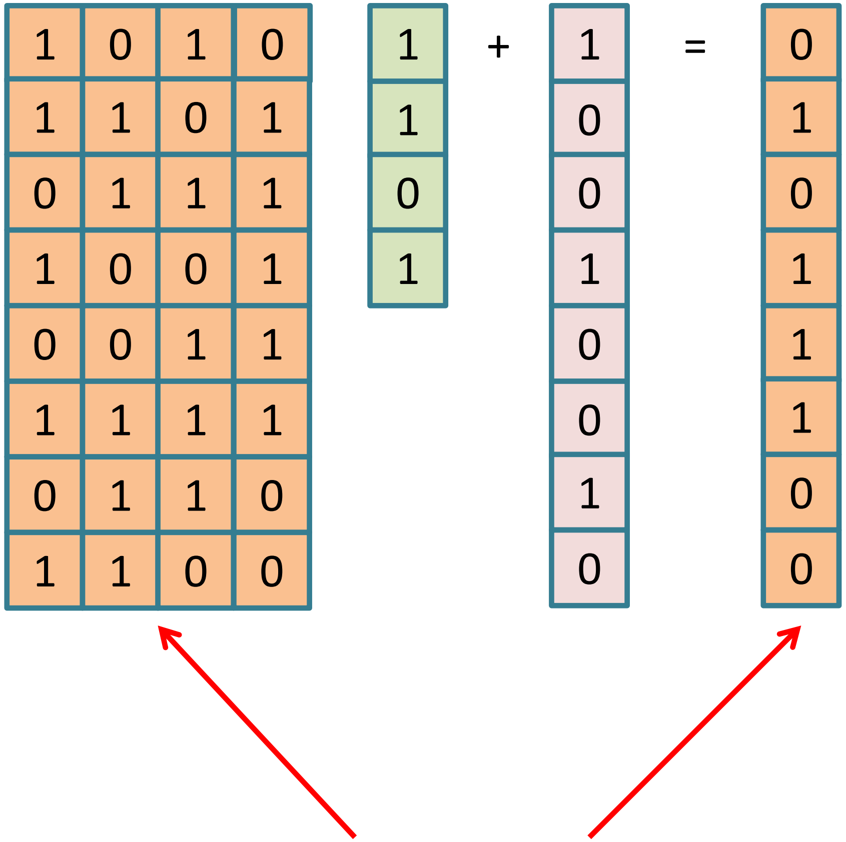
1. Picks **r** $\leftarrow \mathbf{Z}_2^n$
2. Picks a \`noise’ e $\leftarrow \beta_{¼}$ (i.e. e= 0 w.p. ¾ and 1 w.p ¼)
3. Outputs (**r**, t=<**r**,**s**> + e)



$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

The goal: Find **s**

Decision LPN



$$\begin{bmatrix}1&0&1&0\\1&1&0&1\\0&1&1&1\\1&0&0&1\\0&0&1&1\\1&1&1&1\\0&1&1&0\\1&1&0&0\end{bmatrix}\begin{bmatrix}1\\1\\0\\1\end{bmatrix}+\begin{bmatrix}1\\0\\0\\1\\0\\0\\1\\0\end{bmatrix}=\begin{bmatrix}0\\1\\0\\1\\1\\1\\0\\0\end{bmatrix}$$

can’t distinguish from uniform

Thm [BFKL ‘93]: Decision-LPN is as hard as LPN

HB Protocol [HB ‘01]

Prover | **Verifier**

common secret $\mathbf{s}$ in $\mathbf{Z}_2^n$

$\xleftarrow{\mathbf{r}_1, \dots, \mathbf{r}_k}$

Pick $\mathbf{r}_1, \dots, \mathbf{r}_k \leftarrow \mathbf{Z}_2^n$

For $1 \le j \le k$

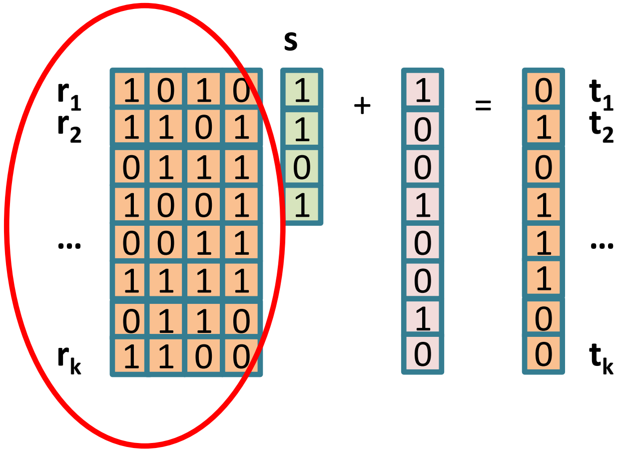
generate $e_j \leftarrow \beta_{1/4}$

set $t_j = \langle \mathbf{r}_j, \mathbf{s} \rangle + e_j$

$\xrightarrow{t_1, \dots, t_k}$

Accept iff for more than 60% of j, $t_j = \langle \mathbf{r}_j, \mathbf{s} \rangle$

As secure as LPN against a passive adversary



$$\begin{array}{c} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \\ \\ \dots \\ \\ \\ \mathbf{r}_k \end{array} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \overset{\mathbf{s}}{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{array}{c} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \\ \\ \dots \\ \\ \\ \mathbf{t}_k \end{array}$$

$kn \approx 2^{18}$ bits!!

HB Protocol [HB '01]

Prover **Verifier**

common secrets $\mathbf{s}_1,\dots,\mathbf{s}_k$ in $\mathbf{Z}_2^n$

$\xleftarrow{\mathbf{r}}$ Pick $\mathbf{r} \leftarrow \mathbf{Z}_2^n$

For $1 \le j \le k$

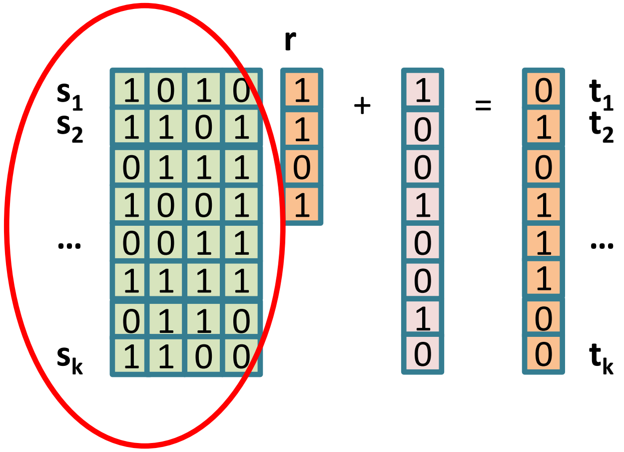
generate $e_j \leftarrow \beta_{1/4}$

set $t_j = <\mathbf{r},\mathbf{s}_j> + e_j$

$\xrightarrow{t_1, \dots, t_k}$

Accept iff for more than 60% of j, $t_j = <\mathbf{r},\mathbf{s}_j>$

As secure as LPN against a passive adversary



$$\begin{array}{l} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \\ \\ \dots \\ \\ \\ \mathbf{s}_k \end{array} \begin{bmatrix} 1&0&1&0 \\ 1&1&0&1 \\ 0&1&1&1 \\ 1&0&0&1 \\ 0&0&1&1 \\ 1&1&1&1 \\ 0&1&1&0 \\ 1&1&0&0 \end{bmatrix} \overset{\mathbf{r}}{\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \\ \\ \dots \\ \\ \\ \mathbf{t}_k \end{array}$$

$kn \approx 2^{18}$ bits!!

HB Protocol + Toeplitz Matrix [GRS ‘08]

Prover **Verifier**

common secrets $\mathbf{s}_1,\ldots,\mathbf{s}_k$ in $\mathbf{Z}_2^n$

$\mathbf{r}$ ←

Pick $\mathbf{r} \leftarrow \mathbf{Z}_2^n$

For $1 \le j \le k$

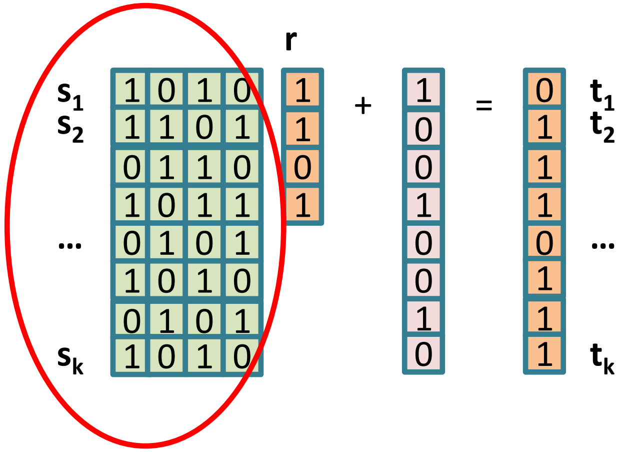
generate $e_j \leftarrow \beta_{1/4}$

set $t_j = \langle \mathbf{r},\mathbf{s}_j \rangle + e_j$

$t_1, \ldots, t_k$ →

Accept iff for more than 60% of j, $t_j = \langle \mathbf{r},\mathbf{s}_j \rangle$

As secure as “Toeplitz-LPN” against a passive adversary



$$\begin{array}{lccccccccccc}
 & & & & & \mathbf{r} & & & & & \\
\mathbf{s}_1 & 1 & 0 & 1 & 0 & 1 & + & 1 & = & 0 & \mathbf{t}_1 \\
\mathbf{s}_2 & 1 & 1 & 0 & 1 & 1 & & 0 & & 1 & \mathbf{t}_2 \\
 & 0 & 1 & 1 & 0 & 0 & & 0 & & 1 & \\
 & 1 & 0 & 1 & 1 & 1 & & 1 & & 1 & \\
\ldots & 0 & 1 & 0 & 1 & & & 0 & & 0 & \ldots \\
 & 1 & 0 & 1 & 0 & & & 0 & & 1 & \\
 & 0 & 1 & 0 & 1 & & & 1 & & 1 & \\
\mathbf{s}_k & 1 & 0 & 1 & 0 & & & 0 & & 1 & \mathbf{t}_k
\end{array}$$

$k+n-1 \approx 2^{10}$ bits

HB Protocol + Ring (field) $\mathbf{Z}_2[x]/<\mathbf{f}(x)>$

Prover **Verifier**

common secrets $\mathbf{s}_1,\dots,\mathbf{s}_k$ in $\mathbf{Z}_2^n$

Pick $\mathbf{r} \leftarrow \mathbf{Z}_2^n$

$\xleftarrow{\mathbf{r}}$

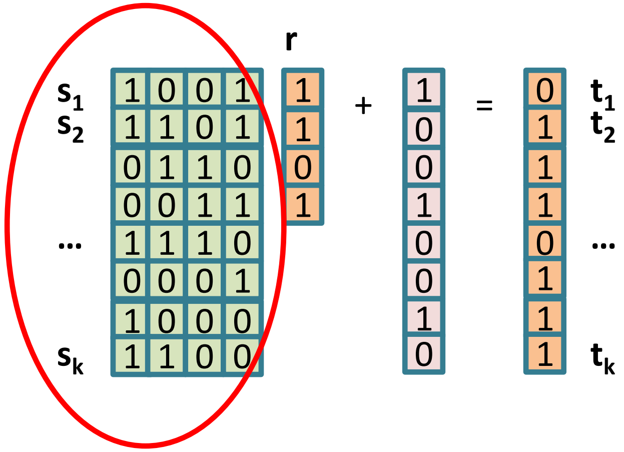
For $1 \le j \le k$
generate $e_j \leftarrow \beta_{1/4}$
set $t_j = <\mathbf{r},\mathbf{s}_j> + e_j$

$\xrightarrow{t_1, \dots, t_k}$

Accept iff for more than 60% of j, $t_j = <\mathbf{r},\mathbf{s}_j>$

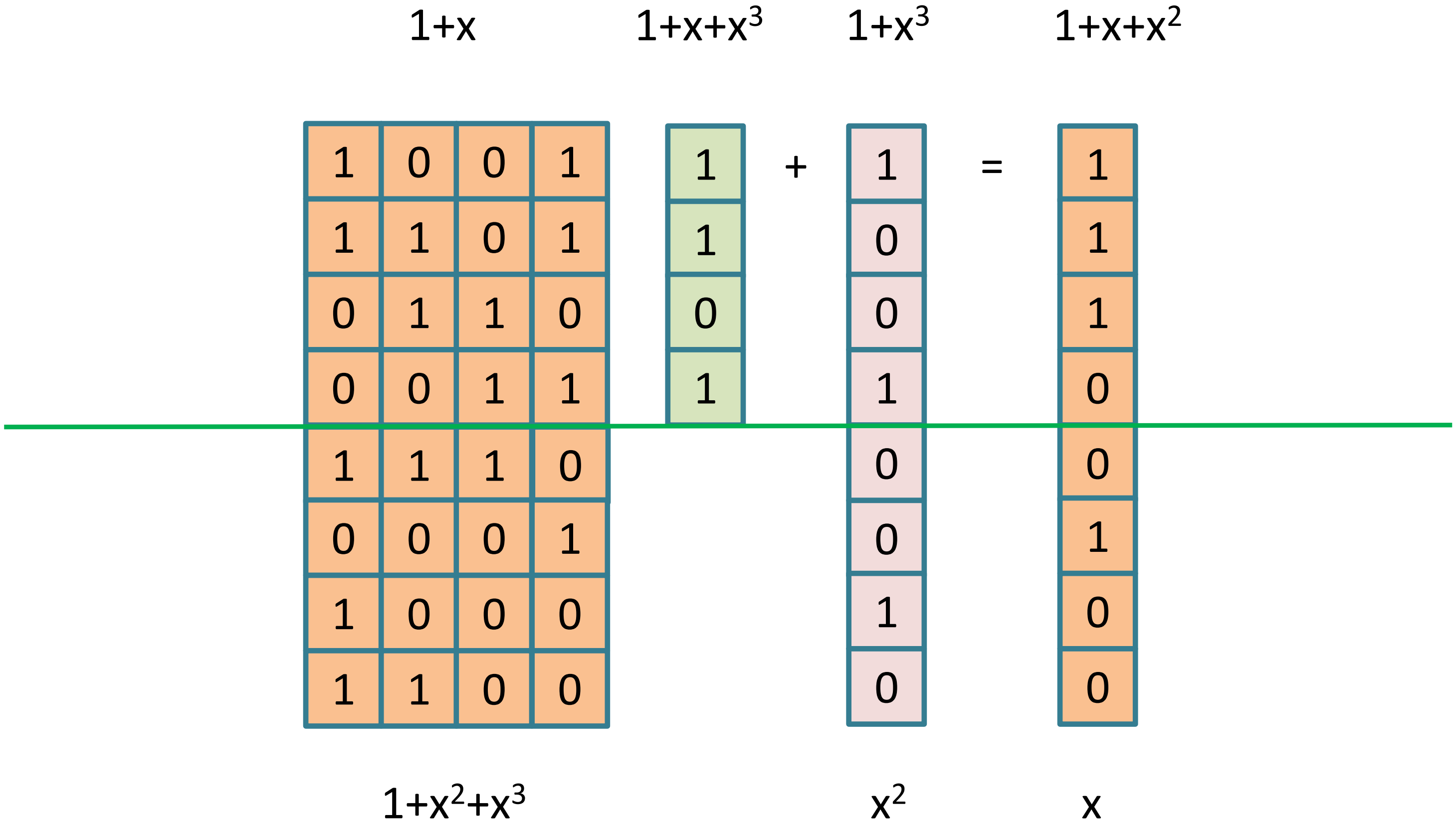
As secure as “Ring-LPN” against a passive adversary

≈ 2^9 bits



$$\begin{array}{ccccccccccc} & & & & & \mathbf{r} & & & & & \\ \mathbf{s}_1 & 1 & 0 & 0 & 1 & 1 & + & 1 & = & 0 & \mathbf{t}_1 \\ \mathbf{s}_2 & 1 & 1 & 0 & 1 & 1 & & 0 & & 1 & \mathbf{t}_2 \\ & 0 & 1 & 1 & 0 & 0 & & 0 & & 1 & \\ & 0 & 0 & 1 & 1 & 1 & & 1 & & 1 & \\ \dots & 1 & 1 & 1 & 0 & & & 0 & & 0 & \dots \\ & 0 & 0 & 0 & 1 & & & 0 & & 1 & \\ & 1 & 0 & 0 & 0 & & & 1 & & 1 & \\ \mathbf{s}_k & 1 & 1 & 0 & 0 & & & 0 & & 1 & \mathbf{t}_k \end{array}$$

HB Protocol + Field $\mathbf{Z}_2[x]/<x^4+x+1>$



$$\left[\begin{array}{cccc}1&0&0&1\\1&1&0&1\\0&1&1&0\\0&0&1&1\\\hline 1&1&1&0\\0&0&0&1\\1&0&0&0\\1&1&0&0\end{array}\right]\left[\begin{array}{c}1\\1\\0\\1\end{array}\right]+\left[\begin{array}{c}1\\0\\0\\1\\\hline 0\\0\\1\\0\end{array}\right]=\left[\begin{array}{c}1\\1\\1\\0\\\hline 0\\1\\0\\0\end{array}\right]$$

Ring-LPN Problem

$\mathbf{f}(x)$ = polynomial of degree n

$\mathbf{R}=\mathbf{Z}_2[x]/<\mathbf{f}(x)>$

(Decision) Ring-LPN problem

$\mathbf{s} \leftarrow \mathbf{R}$

$\mathbf{r} \leftarrow \mathbf{R}$

$\mathbf{e} \leftarrow \beta^{n}_{1/8}$

$\mathbf{t}=\mathbf{rs}+\mathbf{e}$

Output (**r**,**t**)

$\mathbf{r} \leftarrow \mathbf{R}$

$\mathbf{t} \leftarrow \mathbf{R}$

Output (**r**,**t**)

Distinguish between the two distributions

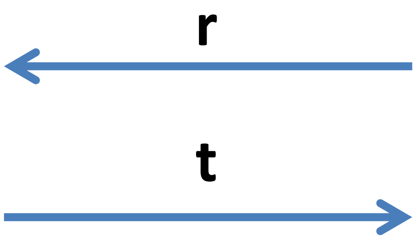
Hardness of Ring-LPN

- Very little known
- For irreducible **f**(x), seems as hard as general LPN
- For reducible **f**(x) … one needs to be careful
  - **f**(x) = $x^n + 1$ (where n is a power of 2), there is a $2^{\sqrt{n}}$ algorithm
- No known connection between decision and search versions

HB Protocol + Ring (field) $\mathbf{Z}_2[x]/\langle \mathbf{f}(x)\rangle$

<u>Prover</u> <u>Verifier</u>

common secret **s** in $\mathbf{Z}_2[x]/(f(x))$



r

Pick $\mathbf{r} \leftarrow \mathbf{Z}_2[x]/(f(x))$

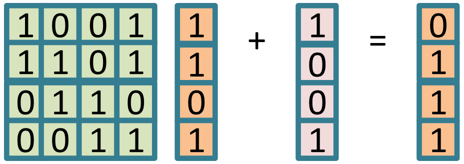
generate $\mathbf{e} \leftarrow \beta^{n}_{1/8}$

set **t=rs+e**

t

Accept iff **t+rs** is 0 for more than 60% of the coefficients

As secure as “Ring-LPN” against a passive adversary



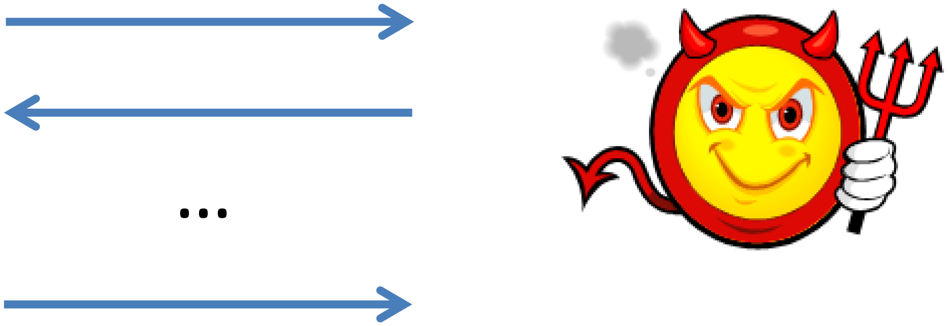
$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

What about active attacks?

Active Attack Model

Prover

Adversary Phase 1



Active Attack Model

Adversary Phase 2

Verifier

Accept!

Adversary wins

HB Protocol with Active Security

[JW ‘05, KS ’06, GRS ’08, …]

Prover

Verifier

secret size doubled

3 Rounds

security proof uses rewinding (not tight):

adversary succeeding with probability δ lets us break LPN with probability δ^2

Our Result

- 2 round *efficient* protocol based on Ring-LPN

- Uses ideas from [KPCJV ‘10]
  - [KPCJV ‘10] is a 2-round LPN-based protocol
  - It suffers from the same efficiency drawback as HB
  - Don’t know if it can be instantiated with a Toeplitz matrix

New Authentication Protocol

Prover

Verifier

common secrets **s**, **s'** in $\mathbf{R}=\mathbf{Z}_2[x]/\langle \mathbf{f}(x)\rangle$

$\mathbf{R}^*$ is the set of all invertible elements in **R**

D is a subset of **R** such that for all $\mathbf{c} \neq \mathbf{c'}$ in **D**, **c**+**c'** is in $\mathbf{R}^*$

Pick $\mathbf{c} \leftarrow \mathbf{D}$

$\xleftarrow{\mathbf{c}}$

generate $\mathbf{r} \leftarrow \mathbf{R}^*$

generate $\mathbf{e} \leftarrow \beta^{n}_{1/8}$

set $\mathbf{z} = \mathbf{r}(\mathbf{sc}+\mathbf{s'})+\mathbf{e}$

$\xrightarrow{(\mathbf{r},\mathbf{z})}$

Accept iff **r** is in $\mathbf{R}^*$ and more than ¾ of the entries of $\mathbf{z} + \mathbf{r}(\mathbf{sc}+\mathbf{s'})$ are 0

Security Proof

$c^* \leftarrow D,\ a \leftarrow R,\ s' = c^*s+a$

Phase 1

$\xleftarrow{\quad c \quad}$

$\xrightarrow{(r', t=r's+e)}$

$r = r'(c+c^*)^{-1}$

$z = t+ra$

$= r(sc+s')+e$

$\xrightarrow{\quad (r,z) \quad}$

Phase 2

$\xrightarrow{\quad c^* \quad}$

$\xleftarrow{\quad (r,z) \quad}$

$\xleftarrow{t=r's+e}$

if **r** is in $\mathbf{R}^*$ and more than ¾ of the entries of $\mathbf{z} + \mathbf{r}(\mathbf{s}\mathbf{c}^*+\mathbf{s}')$ are 0.
else

$\xleftarrow{(r',t) \text{ is random}}$

Performance Comparisons

8-bit AVR ATmega163 smartcard implementations

Protocol	Online Time (cycles)	Offline Time (cycles)	Code Size (bytes)
f(x)=x^{621}+… (reducible)	30,000	82,500	1356
f(x)=x^{532}+x+1 (irreducible)	21,000	174,000	459
AES-Based	10,121	0	4644

Open Problems

- Man-in-the-middle security?
  - There is a $2^{k/2}$ time MIM attack against our protocol (requires $2^{k/2}$ observations)
  - Can we design a *practical* protocol *provably secure* against man-in-the-middle attacks?
    - Big step taken in [DKPW '12]
    - Is Lapin already secure against MIM attacks?
- How hard is the Ring-LPN problem?
  - Is there a search-decision reduction?
- A 2-round protocol with Toeplitz matrices?

Thank You!