Breaking and fixing HB+DB: A Short Tale of Provable vs. Experimental Security and Lightweight Designs

Ioana Boureanu[1] David Gerault[2] Pascal Lafourcade[2]
Cristina Onete [3]

[1]Univ. of Surrey, [2]Univ. Clermont Auvergne, [3]Univ. of Limoges

2 Feb. 2018
IRISA Rennes, SoSySec Seminar

(based on a WiSec2017 paper)

The Short Tale

The HB family of authentication protocols

- Lightweight (RFID)
- Based on the "learning parity with noise" (LPN) problem
- HB(2001), HB$^+$(2005) $\cdots$; 20+ protocols
- Difficult to get resistance against active MiM attacks

HB+DB (Pagnin *et al*, Wisec'15)

- New approach for active MiM security (in HB+)
- Basic idea: use distance bounding; test the hypothesis **experimentally**

- **We found attacks**
- We gave a **provable-security alternative**, in a fixed protocol called $\mathbb{BLOG}$

Outline

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

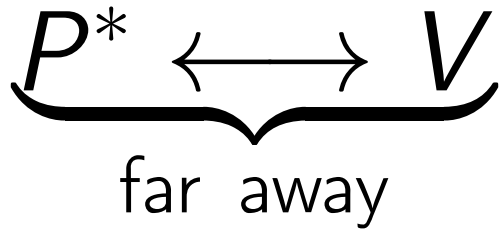
Lessons Learnt

Idea of DB

- a prover P (e.g., an NFC card) **authenticates** to a verifier V (e.g., an RFID reader),
  whilst proving that P is also **within a distance-bound** from V;

- the proximity proof is a challenge-response protocol in which the verifier V **measures the round trip times** between when V sends a challenge and when V gets the responses back from P

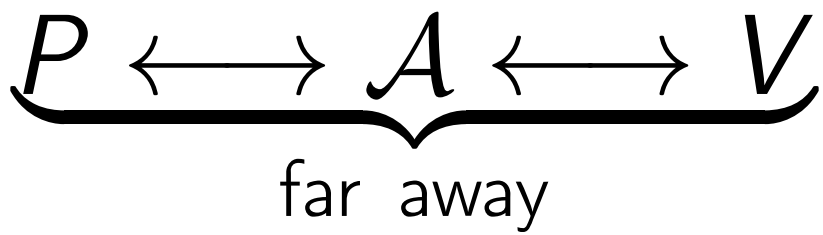
DB Threat1: Distance Fraud



a malicious prover P^* tries to prove that he is close to a verifier V

DB Threat2: Mafia Fraud

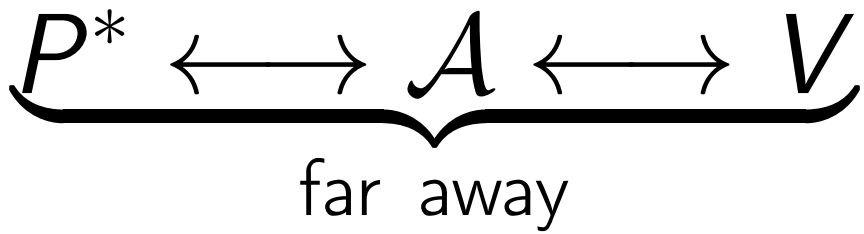
Major Security Problems with the "Unforgeable" (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them [Desmedt SECURICOM 1988]



an adversary $\mathcal{A}$ tries to prove that a prover P is close to a verifier V

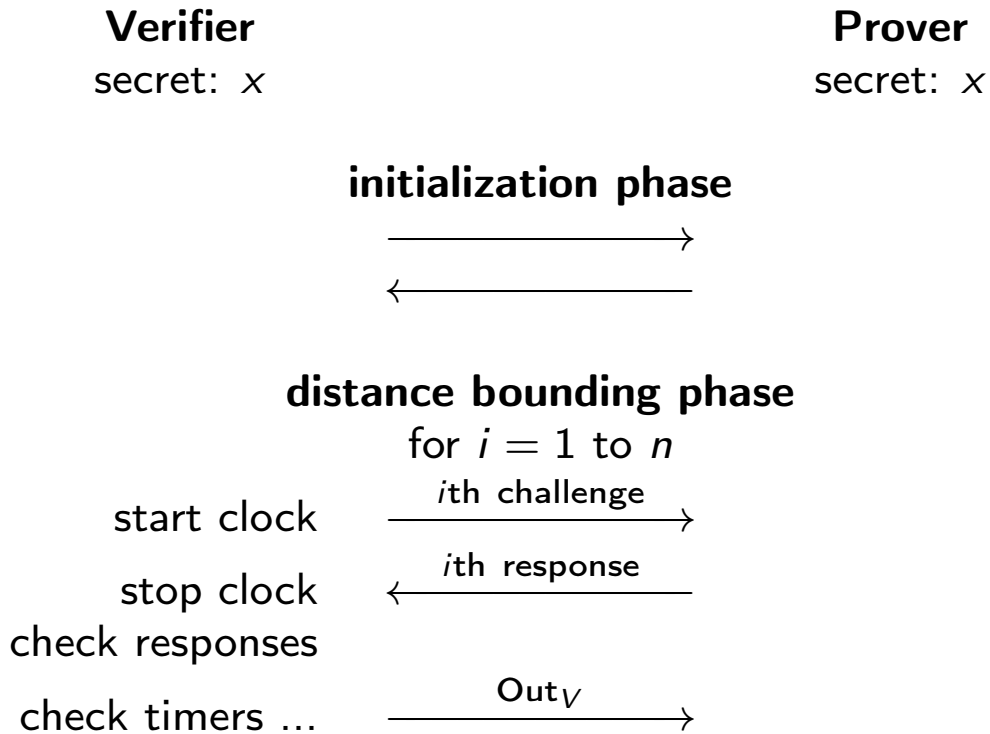
DB Threat3: Terrorist Fraud

Major Security Problems with the "Unforgeable" (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them [Desmedt SECURICOM 1988]



a malicious prover P^* helps an adversary $\mathcal{A}$ to prove that P^* is close to a verifier V without giving $\mathcal{A}$ another advantage

Recall DB: Main Technique



caveat: if the rapid bit-exchange is subject to noise, then the verifier needs to require that at least **a part** of the rounds are correct in order for him to accept

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

Lessons Learnt

The $\mathrm{LPN}_{\eta,x}$ problem

LPN Problem

Given samples of the form a and $a \bullet x \oplus \epsilon$, recovering x is difficult (i.e., the best 2016 algorithms were still sub-exponentials)

LPN Oracle
$\eta \in (0, \frac{1}{2}), x$

$a \xleftarrow{\$} \{0,1\}^{|x|}$

$\epsilon \leftarrow Ber_\eta$

$?$ ←

→ $a, (a \bullet x) \oplus \epsilon$

Where $a \bullet x = \bigoplus_{i=1}^{|x|} a_i \cdot x_i$

The HB+ protocol [Juels and Weis, CRYPTO 2005]

Verifier V

Prover P

Shared keys: x, y

Public parameter: η

$$\xleftarrow{\quad b \quad} \qquad b \xleftarrow{\$} \{0,1\}^{|x|}$$

$$a \xleftarrow{\$} \{0,1\}^{|x|} \qquad \xrightarrow{\quad a \quad}$$

$$\xleftarrow{\quad ((a \cdot x) \oplus (b \cdot y)) \oplus \epsilon \quad} \qquad \epsilon \leftarrow Ber_{\eta}$$

Repeat n times

If number of errors $\approx \eta \cdot n$, accept authentication

Obs.

- $\eta \in (0, \frac{1}{2})$; high η means hard key-recovery but high false acceptance for honest provers; typical: $\eta = \frac{1}{4}$ or $\eta = \frac{1}{8}$

Security

An *active* attacker, impersonating V, has to solve $\mathrm{LPN}_{\eta,y}$ to recover y.

"GRS" attacks on HB+ [Gilbert Robshaw Seurin, Eurocrypt 2005]

Key idea

A MiM flipping the i^{th} bit in the challenge, flips the response bit iff $x_i = 1$.

The attack

Flip a_i. If the authentication fails, then $x_i = 1$, otherwise $x_i = 0$.

b

$b \xleftarrow{\$} \{0,1\}^{|x|}$

$a \xleftarrow{\$} \{0,1\}^{|x|}$ $\xrightarrow{a}$ $a' = \text{flip}_i(a)$ $\xrightarrow{a'}$

$((a' \cdot x) \oplus (b \cdot y)) \oplus \epsilon$

$\epsilon \leftarrow Ber_\eta$

! Further active attacks

the attacker can impersonate P by sending $b = 0^{|x|}$, or the attacker can recover y in a similar way

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

Lessons Learnt

The HB+DB protocol (... a bit simplified)

– focused on thwarting the GRS attack on HB+DB

Basic Idea

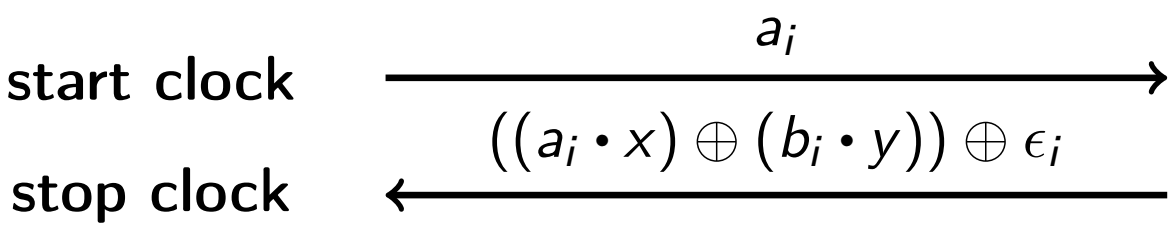
Flipping bits takes time! Close-by prover should respond faster!

Shared keys: x, y

Public parameter: η

Verifier V Prover P

$a_i \xleftarrow{\$} \{0,1\}^{|x|}$



MiM detection (in WiSec'15, by simulation in Matlab)

If a response takes too long to arrive, refuse authentication.

The HB+DB protocol (... a bit simplified)

– focused on thwarting the GRS attack on HB+DB

Basic Idea

Flipping bits takes time! Close-by prover should respond faster!

Shared keys: x, y, z

Public parameter: η

Verifier V

Prover P

$s \xleftarrow{\$} \{0,1\}^{|x|}$

s

For $i \in \{1, n\}$

$a_i \xleftarrow{\$} \{0,1\}^{|x|}$

$b_i \leftarrow f_z(s, i)$

$\epsilon_i \leftarrow Ber_\eta$

start clock

a_i

$((a_i \cdot x) \oplus (b_i \cdot y)) \oplus \epsilon_i$

stop clock

MiM detection (in WiSec'15, by simulation in Matlab)

If a response takes too long to arrive, refuse authentication.

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

Lessons Learnt

Why HB+DB's measures are not sufficient?

Why HB+DB's measures are not sufficient?

Problem No. 1

An active MiM adversary can overwrite one bit of the challenge, faster, without reading it!
Our MiM attack is faster & only slightly different to GRS: instead of $flip_i$, $write_i(1)$.

Why HB+DB's measures are not sufficient?

Problem No. 1

An active MiM adversary can overwrite one bit of the challenge, faster, without reading it!
Our MiM attack is faster & only slightly different to GRS: instead of $flip_i$, $write_i(1)$.

Problem No. 2

If one composes proximity-checking with authentication, one needs to treat the compound threats!
(1) In authentication, the prover is generally not dishonest! In DB, the prover can be dishonest!!

(2) The LPN-noise "inherited" for auth. security from HB+, hinders DB-security in HB+DB.
I.e., high LPN noises $\leftrightarrow$ large probability of distance-fraud and mafia-fraud (i.e., by random responses)!

Why HB+DB's measures are not sufficient?

Problem No. 1

An active MiM adversary can overwrite one bit of the challenge, faster, without reading it!
Our MiM attack is faster & only slightly different to GRS: instead of $flip_i$, $write_i(1)$.

Problem No. 2

If one composes proximity-checking with authentication, one needs to treat the compound threats!
(1) In authentication, the prover is generally not dishonest! In DB, the prover can be dishonest!!

(2) The LPN-noise "inherited" for auth. security from HB+, hinders DB-security in HB+DB.
I.e., high LPN noises $\leftrightarrow$ large probability of distance-fraud and mafia-fraud (i.e., by random responses)!

Problem No. 3

Compositions are not easy!
The security of HB+DB against passive attacks NO LONGER reduces to the LPN-problem!
I.e., all the LPN-noise, for what really?!

Why HB+DB's measures are not sufficient?

Problem No. 1

An active MiM adversary can overwrite one bit of the challenge, faster, without reading it!
Our MiM attack is faster & only slightly different to GRS: instead of $flip_i$, $write_i(1)$.

Problem No. 2

If one composes proximity-checking with authentication, one needs to treat the compound threats!
(1) In authentication, the prover is generally not dishonest! In DB, the prover can be dishonest!!

(2) The LPN-noise "inherited" for auth. security from HB+, hinders DB-security in HB+DB.
I.e., high LPN noises $\leftrightarrow$ large probability of distance-fraud and mafia-fraud (i.e., by random responses)!

Problem No. 3

Compositions are not easy!
The security of HB+DB against passive attacks NO LONGER reduces to the LPN-problem!
I.e., all the LPN-noise, for what really?!

Problem No. 4

HB+DB used 3 keys, but the security is only based on one!

Pb1:

Our Improved GRS

- in HB+DB, each bit is independently encoded into a high- or low-amplitude signal on the carrier link;
- in GRS on HB+DB, a bit-challenge is read then flipped

Pb1:

Our Improved GRS

- in HB+DB, each bit is independently encoded into a high- or low-amplitude signal on the carrier link;
- in GRS on HB+DB, a bit-challenge is read then flipped
- our setting:
  (i) during our attack, $\mathcal{A}$ can "speak louder" than the verifier, *i.e.*, send a signal with higher power, drowning out (part of) V's message,
  (ii) $\mathcal{A}$ knows the time interval between 2 challenges, and the bit period.

Pb1:

Our Improved GRS

- in HB+DB, each bit is independently encoded into a high- or low-amplitude signal on the carrier link;
- in GRS on HB+DB, a bit-challenge is read then flipped
- our setting:
  (i) during our attack, $\mathcal{A}$ can "speak louder" than the verifier, *i.e.*, send a signal with higher power, drowning out (part of) V's message,
  (ii) $\mathcal{A}$ knows the time interval between 2 challenges, and the bit period.
- plausible setting, if $\mathcal{A}$ located between a verifier and the prover
- instantaneous, hence un-detected by the experiments in HB+DB

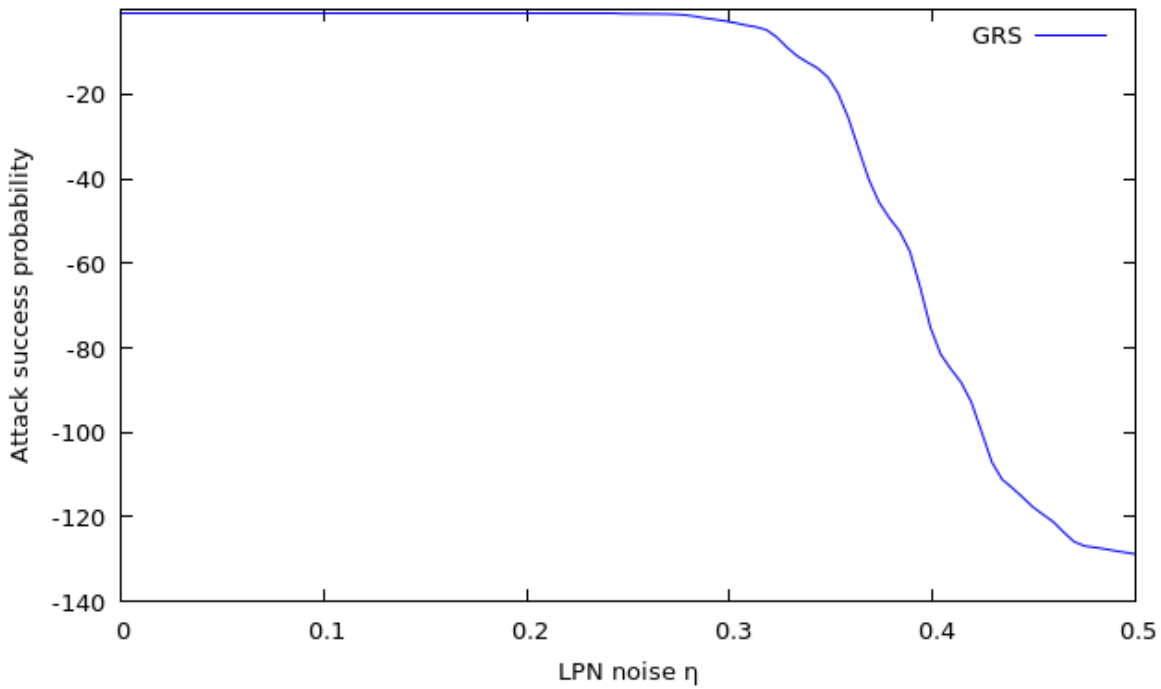
Limits of attacks: our MiM/GRS vs. random guessing

Our MiM

Recall: Our MiM attack $\equiv$ send a 1-bit in the challenge at a given position!

Intuition

In our MiM attack, if the challenge was 0, then it is flipped.



Behaviour

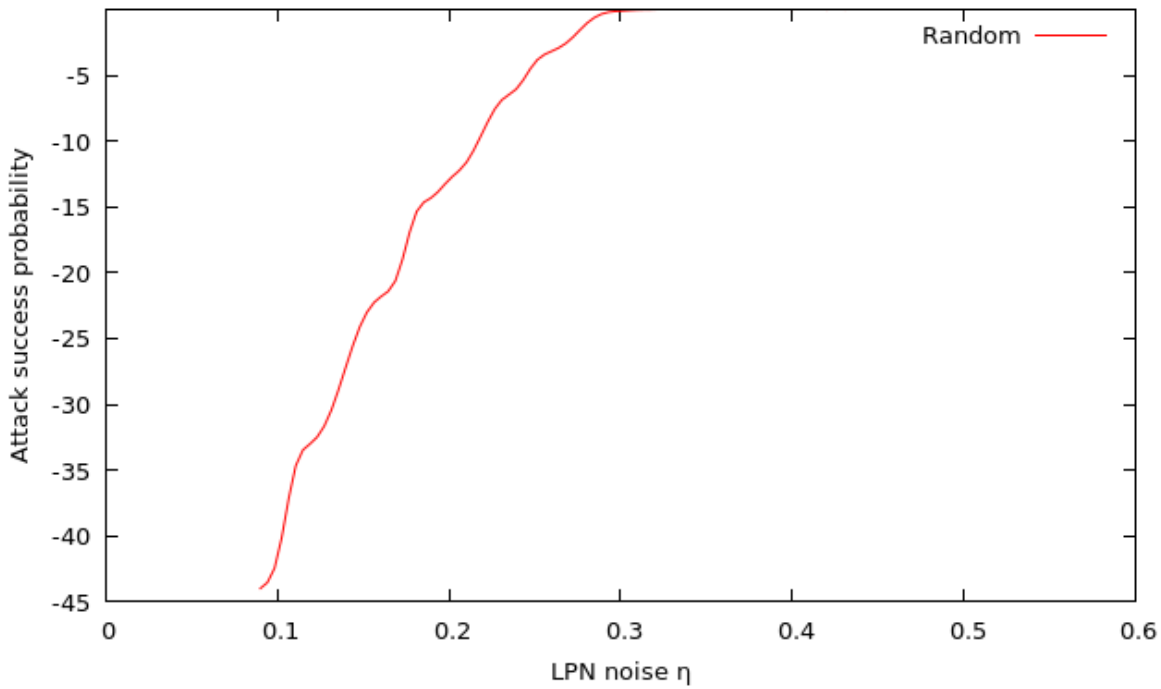
The success probability decreases as η increases.

Limits of our attacks: MiM vs. random guessing

Random guessing

Intuition

But ... as η increases $\Longrightarrow$ lots of errors need to be tolerated to not dismiss honest provers.



Behaviour

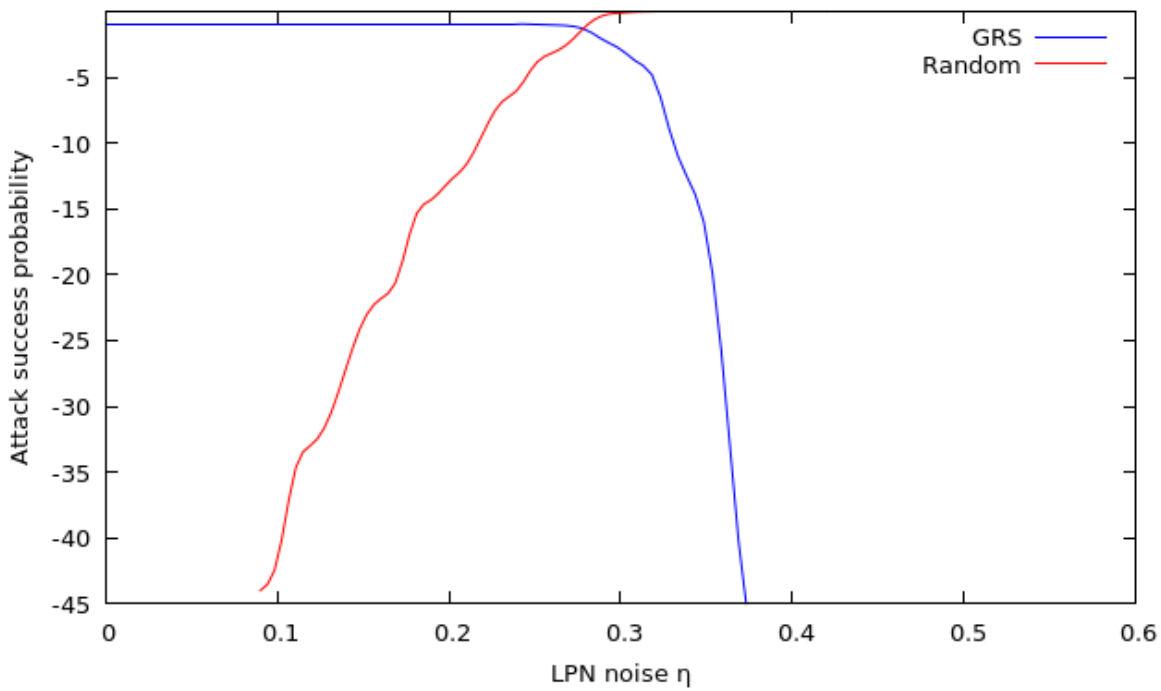
The higher η, the higher the probability to win by sending random responses.

Limits of our attacks: MiM vs. random guessing

Combining both attacks

Actual security

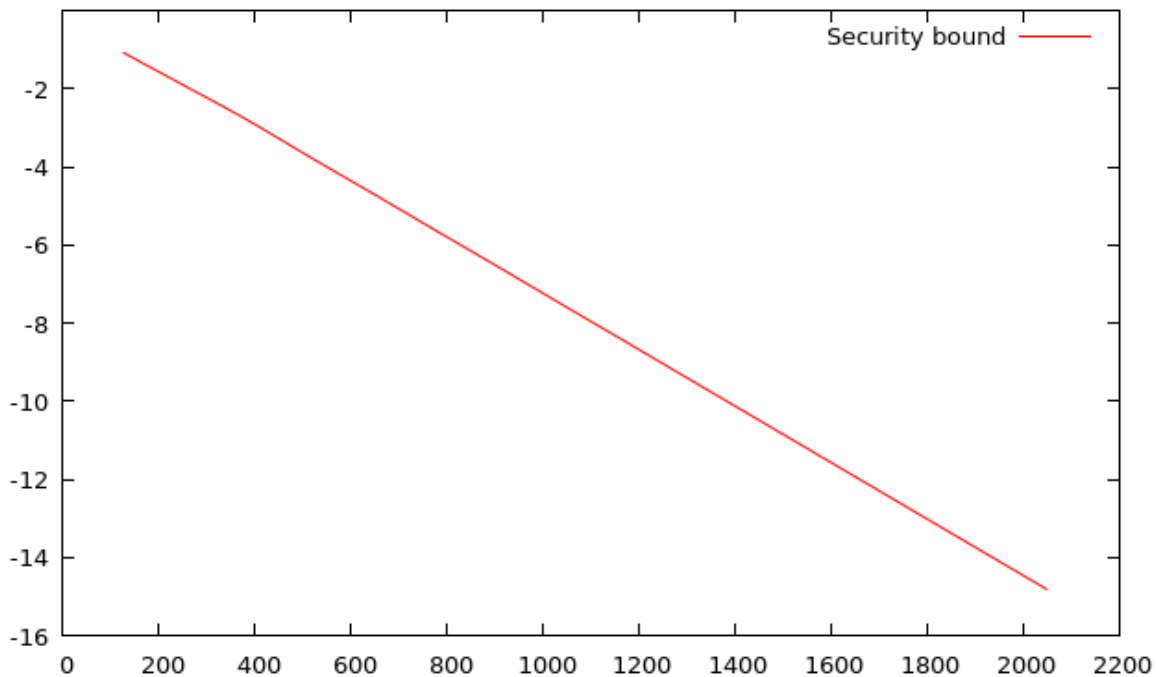
The intersection of the curves bounds the security for a given key size $n = |x|$



Pb2: Poor security level in HB+DB

Setting

For $|x| \in [128, 2048]$, $n = |x|$, $\min_{\eta=0}^{\frac{1}{2}}(max(p, q))$



e.g., for FRR=1%, for 128 rounds, the FAR (false acceptance rate) varies from 2^{-5} in the worst case ($\eta = 1/4$, $\nu = 0.1$) to 2^{-30} in the best-case scenario ($\eta = 1/8$, $\nu = 0.05$).

- $\mathcal{P}[Rand.\ guess = FAR] = p \equiv$ probability that tossing n fair coins results in $[n \cdot \mu - \tau, n \cdot \mu + \tau]$ successes

- $\mathcal{P}[Our\ MiM] = q = (\frac{1-p}{2} + \frac{1-FRR}{2})^{|x|}$

Pb3:

Passive-adv. security NOT based on LPN in HB+DB

Intuition:

- if HB+DB had no noise, then a passive adversary would be faced with a set of linear equations of the form $a_i \bullet x \oplus b_i \bullet y = r_i$, for publicly-known a_i and r_i values.

  So, HB+DB requires a non-zero LPN noise ϵ added into the responses r_i, to have passive adversaries be faced with an LPN instance $LPN_{x,\eta}$

Pb3:

Passive-adv. security NOT based on LPN in HB+DB

Intuition:

- if HB+DB had no noise, then a passive adversary would be faced with a set of linear equations of the form $a_i \bullet x \oplus b_i \bullet y = r_i$, for publicly-known a_i and r_i values.

  So, HB+DB requires a non-zero LPN noise ϵ added into the responses r_i, to have passive adversaries be faced with an LPN instance $LPN_{x,\eta}$

- in HB+DB, a passive adversary against is faced with a set of equations of the form $a_i \bullet x \oplus b_i \bullet y = r_i$, but in which only a_i and r_i are public, whilst the b_is remain known only to the honest parties.

  I.e., b_i randomises the padding to $a_i \bullet x$, which turns an honest execution of HB+DB into a computationally-hard problem for the observing adversary, without it being based on a hard instance of LPN.

Pb4:

Why 3 keys, if it is 1-key security?

Intuition:

- if $\mathcal{A}$ recovers x
- then, $\mathcal{A}$ starts a session sid with a far-away, honest P, and a separate session sid$'$ with V and wins in sid$'$, like so...

Verifier V Prover P

s

$b \xleftarrow{\$} \{0,1\}^{|x|}$

$a \xleftarrow{\$} \{0,1\}^{|x|}$

$a' = 0$

a

$(b \bullet y) \oplus \epsilon$

$\epsilon \leftarrow Ber_\eta$

$((a \bullet x) \oplus (b \bullet y)) \oplus \epsilon$

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

Lessons Learnt

Problems of the original protocol

- MiMs can active temper challenges
- (Unnecessarily) required LPN noise $\Longrightarrow$ high false-acceptance rate
- Many (not so useful) keys, many PRF calls

Proposition

We propose $\mathbb{BLOG}$ to fix some of these problems

BLOG

Verifier V

Shared keys: $x, \cancel{y, z}$
Public parameter: $\cancel{\eta}$

Prover P

$s \xleftarrow{\$} \{0,1\}^{|x|}$

s ←

$(xtemp||b) \leftarrow f_{\cancel{z}x}(s, \cancel{i})$

For $i \in \{1, n\}$

$a_i \xleftarrow{\$} \{0,1\}^{|x|}$

$\cancel{b_i \leftarrow f_z(s,i)}$
$\cancel{\leftarrow Ber_\eta}$

start clock a_i →

stop clock ← $a_i \bullet \cancel{x} xtemp \oplus b_i \bullet \cancel{y} \oplus \cancel{\epsilon_i}$

BLOG

Verifier V		Prover P
	$x \in \mathbb{Z}_2{}^k$ $f_x : \mathbb{Z}_2{}^k \to \mathbb{Z}_2{}^k$	
	Initialisation phase	
	$\xleftarrow{s}$	$s \leftarrow \mathbb{Z}_2{}^k$
$(x_{\text{temp}}\|\|b) \leftarrow f_x(s)$ V halts if $HW(x_{\text{temp}}) = 0$		$(x_{\text{temp}}\|\|b) \leftarrow f_x(s)$
	DB phase for $i = 1$ to n	
Start Clock $a_i \leftarrow \mathbb{Z}_2^k$	$\xrightarrow{a_i}$	
	$\xleftarrow{r_i}$	$r_i := (a_i \bullet x_{\text{temp}}) \oplus b_i$
calculate Δ_i as the a_i–r_i' RTT time **Stop Clock**		
	Verification phase	
V accepts if $\forall i \in \{1, \ldots, n\}$: $r_i = (a_i \bullet x_{\text{temp}}) \oplus b_i$ and $\Delta_i \leq t_{\max}$		

$\mathbb{BLOG}$'s Properties

- Provable security against
  - Active attacks
  - Distance bounding adversaries (MF, DF, TF)
- More lightweight(ish)

Protocol	Secure	Memory	Dot products	PRF
HB+DB	No	$3 \cdot \|x\| + (n+1) \cdot \|x\| + n$	$2 \cdot n$	n
$\mathbb{BLOG}$	Yes	$\|x\| + 3 \cdot \|x\|$	n	1

•seemingly great security for $\mathbb{BLOG}$, but .... this is with k-bit challenges!

Introduction to Distance-Bounding (DB.) Protocols

Lightweight, LPN-based Authentication $\equiv$ HB+

Lightweight LPN-based Auth. + DB $\equiv$ HB+DB

Our Attacks against HB+DB

A Provably Secure, Yet Partial Solution

Lessons Learnt

Conclusions...

- DB alone generally does not fix MiM attacks
- Composing security primitives is not trivial
- Designing a good DB protocol truly based on LPN-based may be desirable (as it could be lightweight) but it is challenging... There may be an interesting with achieving TF-resistance....

Thank you for your attention!
Questions?

The HB+DB protocol – entire

Verifier V | **Prover** P

$x \in \mathbb{Z_2}^k$
$y \in \mathbb{Z_2}^k$
$z \in \mathbb{Z_2}^k$
$\eta \in (0, \frac{1}{2})$
$f_z : \mathbb{Z_2}^k \times \{1, 2, \ldots, n\} \to \mathbb{Z_2}^k$

Init. phase

$\xleftarrow{s}$ $s \leftarrow \mathbb{Z_2}^k$

for $i \in \{1, \ldots, n\}$: $b_i = f_z(s, i)$

for $i \in \{1, \ldots, n\}$:
$b_i = f_z(s, i)$;
$\epsilon_i \leftarrow Ber_\eta$;
$c_i = (b_i \bullet y) \oplus \epsilon_i$

DB phase
for $i = 1$ to n

Start Clock

$a_i \leftarrow \mathbb{Z}_2^k$ $\xrightarrow{a_i}$

$\xleftarrow{r_i}$ $r_i := (a_i \bullet x) \oplus c_i$,

$r'_i := (a_i \bullet x) \oplus c_i \oplus \zeta_i$

where $\zeta_i \in \mathbb{Z_2}$ denotes the channel-noise

Stop Clock

Verification phase

V accepts if
$\Sigma_{i \in \{\mathbf{1}, \ldots n\}} (a_i \bullet x) \oplus (b_i \bullet y) \oplus r'_i \in [\mu - \tau, \mu + \tau]$
and $\Delta_i \leq t_{\mathbf{max}}$ for all $i \in \{1, \ldots, n\}$,
where μ is the expected mean of
$\Sigma_{i \in \mathbf{1}, \ldots, n} (\epsilon_i + \zeta_i)$,
and τ is a tolerance parameter.

Number of errors

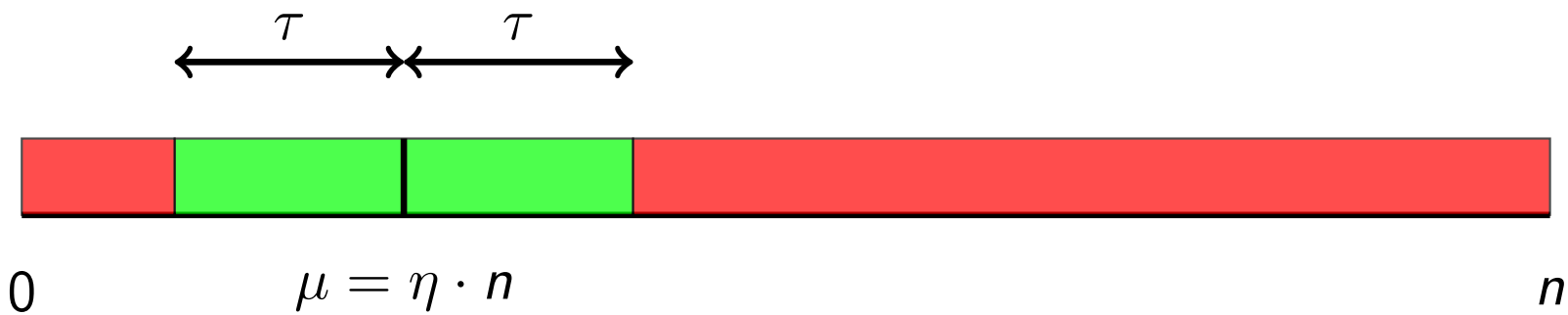
Verification

V counts the number of noisy (wrong) responses.

How many?

It should be roughly $n \cdot \mu$, but a security margin is necessary.

example: $\eta = 0.25$



Low tolerance τ leads to rejecting legitimate users, high τ increases the false acceptance rate.

Analysis

- $\mathcal{P}[Rand] = p =$ probability that tossing n fair coins results in $[n \cdot \mu - \tau, n \cdot \mu + \tau]$ successes.
- $\mathcal{P}[GRS] = q = (\frac{1-p}{2} + \frac{1-FRR}{2})^{|x|}$

Behaviour with false-rejection rate=0.01

- When $\eta = 0$, $p \approx 0$ and $q \approx (\frac{1}{2} + 0.495)^{|x|} \approx 0.995^{|x|} \approx 2^{-0.007 \cdot |x|}$
- When $\eta = \frac{1}{2}, p = 1$