

Using HB Family of Protocols for Privacy-Preserving Authentication of RFID Tags in a Population

Tzipora Halevi¹, Nitesh Saxena¹, Shai Halevi²

¹ Polytechnic Institute of New York University,

² IBM T. J. Watson Research Center

Radio Frequency Identification (RFID)



- Increasingly used in daily life
 - Military, commercial and medical domains
- Authentication needed to prevent tag forgery and counterfeiting
- Privacy concerns require tag anonymity
 - Protect tag from unauthorized readers
 - Tracking is a main concern
 - RFID denounced as 'big brother' surveillance tool

RFID Limitations

- Tags are low-cost devices
 - Have limited computation and storage capabilities
 - Traditional authentication protocols may not be applicable

Privacy Concerns: Examples

- RFID's used in Passports
 - Can be used to track the person's location, eavesdrop on personal information
- Libraries books with RFID chips
 - Can be used to track books on 'hotlists'
 - Tracked by law enforcement agencies
 - Indicators of terrorist or illegal activities
- RFID used for commercial every-day products
 - Unlike bar codes, each tag ID is different
 - RFID's can be read from a distance, through bags, clothes, etc.

Authentication and Privacy

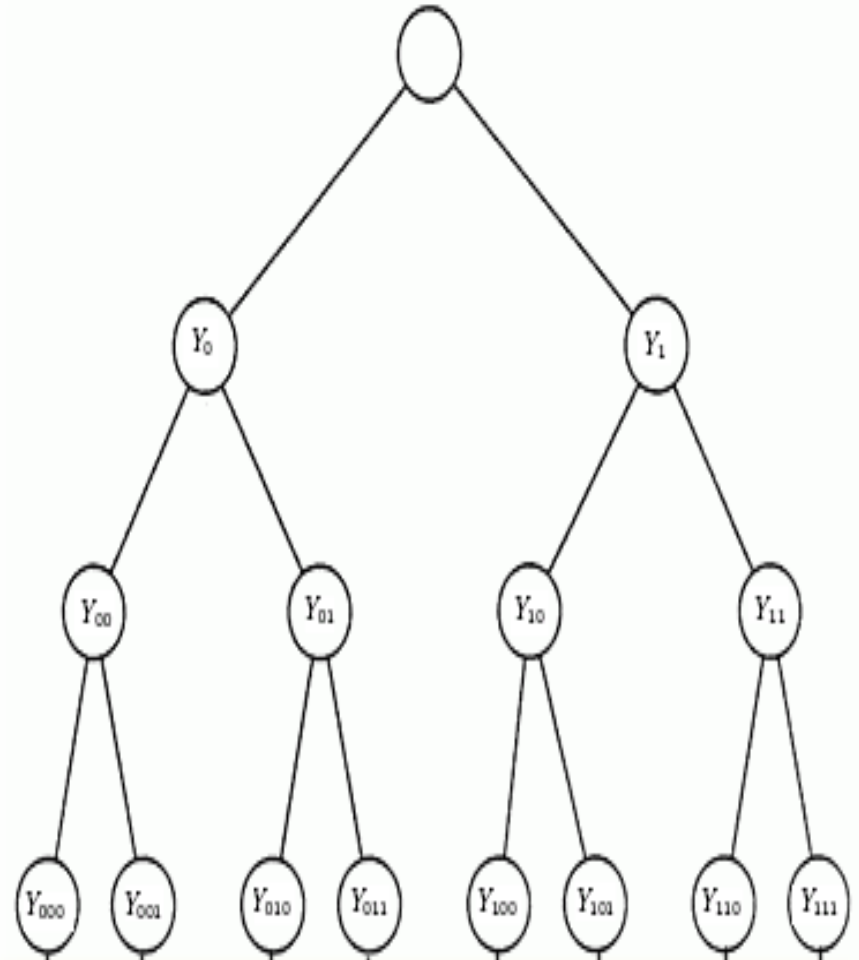
- Reader must use keys to authenticate tag
- Multiple Tags (millions or billions) registered with a single reader
- How to find the right keys to use?
- Tag can send ID
 - Violate tag privacy, eavesdropper learns tag identity
 - Attacker can use it to track tags, holders
 - Passports, books, etc
- How to authenticate without sending IDs?

Privacy-Preserving Authentication

- Exhaustive Search – Trying to match the tag results to all the possible tag secrets
 - Does not scale – $O(n)$ calculations
 - Reader computation is also limited
 - For some authentication protocol, such as HB+, False Accept Rate (FAR) also grows $x n$
 - may become too large for a non-valid tag
- Tree-based authentication schemes: Molnar-Wagner [CCS04]

Tree-Based Authentication

- Keys in the nodes
- Each tag associated with a leaf, gets all the keys on root-leaf path
- To authenticate
 - Start at the root
 - At each level, use “light exhaustive search” to decide on the next child to go to



Tree-Based Authentication

- Reduces computation to $O(\log_b N)$
 - Adds some memory and communication overhead

Previous Work

- Privacy-preserving authentication used PRF'S
- Tree-based scheme: Molnar-Wagner [CCS04]
 - 2D Mesh scheme, pseudonyms :Cheon et al. [ePrint09], Burmester et al. [AsiaCCS08]
- However, low-cost RFID may not be able to calculate PRF's
 - Have limited computational resources
- HB series of protocols designed for use by low-cost RFIDs

Background: HB+ Protocol

- "Human Protocols": Hopper-Blum [Asiacrypt 01]
- Use HB+ for RFIDs: Juels-Weis [Crypto 05]
- Parallel HB+: Katz-Shin [ePrint 05]
- Other variants since: Gilbert et al. [Eurocrypt08]

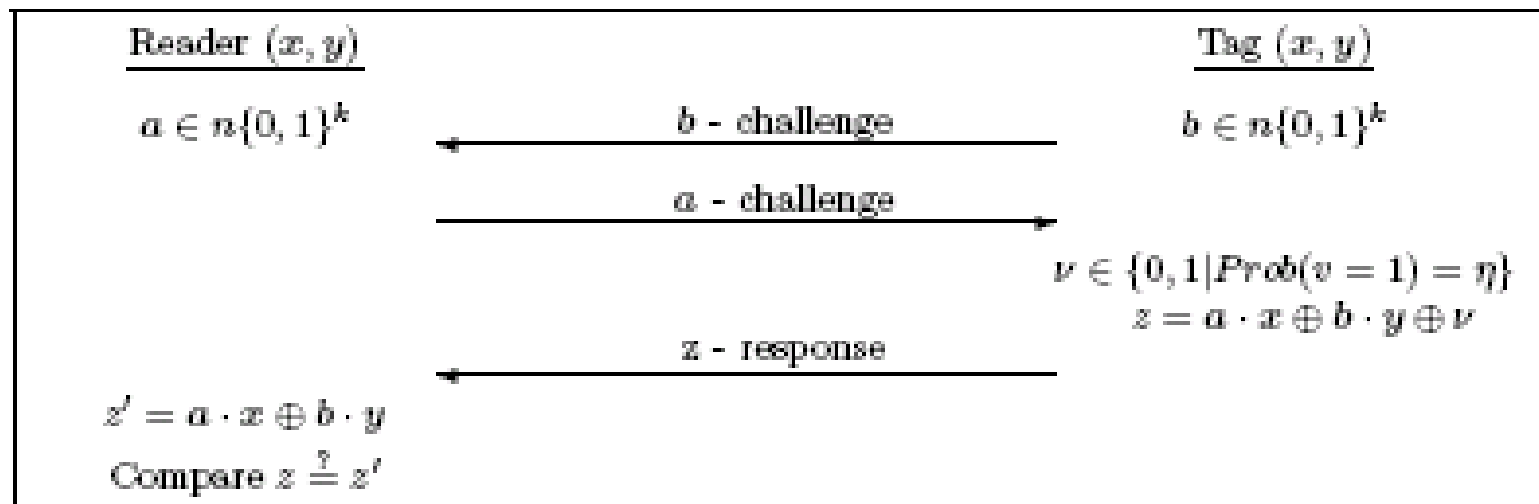


Fig. 1. One round of the HB+ Protocol

HB+ Protocol parameters

- For security with memory up to 2^{65} bytes
 - Key sizes needed : $K_x = 80$ and $K_y = 330$
(Levieil-Fouque [SCN06])
- Error rates for different parameters of HB+:
 - Two noise levels, 0.125 or 0.25

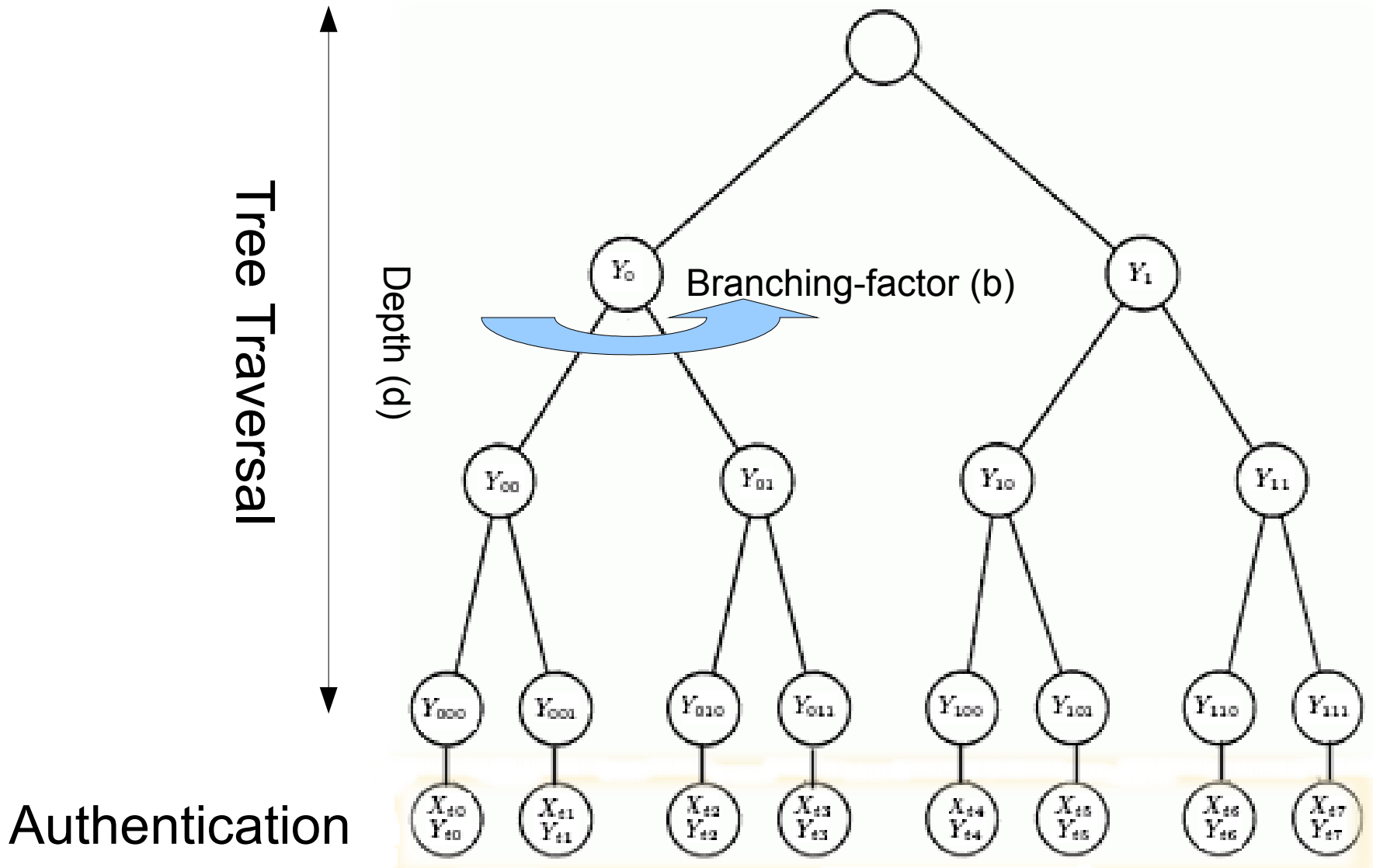
r	η	FRR	FAR
80	0.25	0.44	$4 \cdot 10^{-6}$
50	0.125	0.44	$2 \cdot 10^{-8}$

Our Contribution:

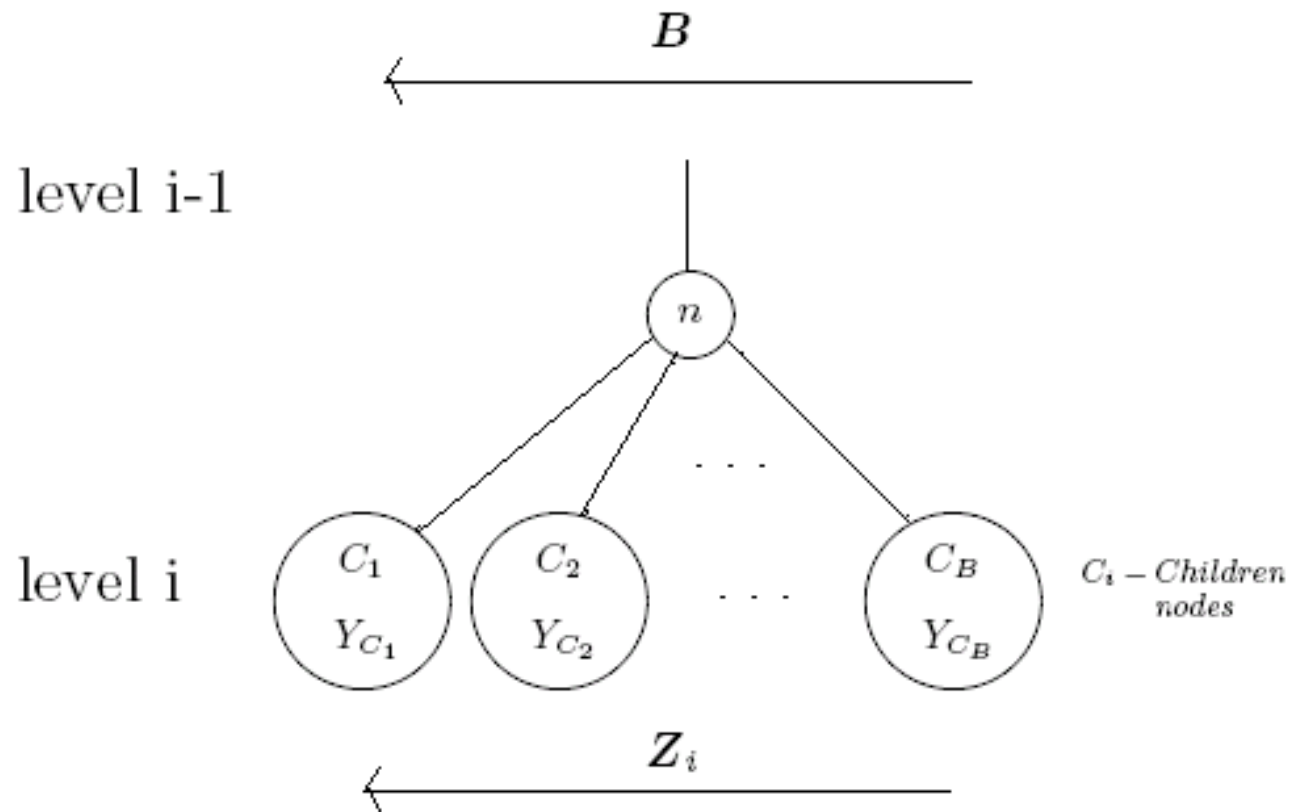
Tree-based HB like algorithm

- Tree-based protocol utilizing HB-like protocols
 - Reduces computations for both reader and tags
- Two logical steps:
 - Tree-Traversal Stage
 - Reader identifies most likely tag to authenticate
 - Authentication Stage
 - The identity of the “most likely tag” is verified
- FAR does not grow with number of tags
 - The same as the underlying authentication protocol
- FRR grows very slowly

Tree-HB+



Tree-Traversal Stage



Choose C_j such that Z_i closest to $B \cdot Y_{C_j}$

Tree-Traversal Stage

- Tag sends the challenge matrix and all the vectors Z_i
 - $Z_i = \mathbf{B}y_i \oplus v_i$
 - Noise vector v_i generated for each stage
- Reader chooses most likely node for each stage
 - Goes down the tree until arrives to one of the leaves

Optimizations (Tree-traversal)

- Observation: $F_y(\mathbf{B}) = \mathbf{B}y \oplus \text{noise}$ behaves like a pseudo-random function; Applebaum [Crypto09]
 - Replaces traditional PRF's for tree-traversal stage
 - \mathbf{B} random matrix
- Use only tag-generated matrix in tree-traversal
- Same matrix used for all levels in the tree
- Response (z_i) can be much shorter in tree-traversal stage
- Result almost as efficient as the underlying HB+

Authentication Stage

- The most likely leaf is authenticated
 - Using the HB+ protocol
 - Reader sends a challenge matrix $\mathbf{A}^{r \times k}$
 - Tag chooses a noise vector $v_t^{r \times 1}$
 - Tag sends vector $Z_i = \mathbf{A}x_i \oplus \mathbf{B}y_i \oplus v_t$
 - Reader computes $Z_i' = \mathbf{A}x_i \oplus \mathbf{B}y_i$
 - If Hamming distance between Z_i and Z_i' less than threshold, tag authenticated
 - HB# can be used instead of HB+

Protocol Comparison

- Protocol comparison for branching $b=1000$
 - Other parameters as in original HB+
 - Response length=80, noise level=0.25
 - Reader computation for Tree-HB+ reduced significantly compared to exhaustive search
 - Security (FAR) significantly improved

Table 2. Protocol comparison for a population of $N = 10^6$ tags.

Method	Reader Computation	Communication	Tag Memory	FAR	FRR
ES HB+	$10^6 \cdot C_{HB+}$	26960	336	0.98	0.44
Tree HB+	$2000 \cdot C_{HB+}$	27120	848	$4 \cdot 10^{-6}$	0.6
Tree PRF	$2000 \cdot C_{PRF}$	1024	256	0	0

Improving Performance

- Iterating the protocol if tag rejected reduces FRR drastically, only increase FAR slightly
 - $FRR_{\text{new}} = FRR^2$
 - $FAR_{\text{new}} = 2 \times FAR$
- Complexity only grows by a factor $(1+FRR)$
 - $C_{\text{new}} = C(1+FRR)$
 - With probability $(1-FRR)$, protocol only runs once
- Can transform protocols with large FRR (such as original HB+) to receive lower FRR

Example: Choice of Parameters

- System Parameters Required:
 - Tag population 10^6 tags
 - $FRR=10^{(-4)}$ and $FAR=10^{(-8)}$
 - Security with memory up to 2^{65} bytes
- Repeat algorithm 4 times
 - For one run, we need $FAR\sim 10^{-9}$ and $FRR\sim 0.08$
 - Dictates choice of response-length r

Example

- $N=10^6$ tags, 4 iterations
- Tree-depth 2 or 3 (branching 100 or 1000)
- Two noise levels, 0.125 or 0.25
- Security against attacks with upto 2^{65} space

ε	d	β	k_x	k_y	r	r_{tr}	C_{rdr}	C_{tag}	comm	mem	FRR	FAR
0.25	2	1000	80	330	212	102	$7.49E + 7$	$1.71E + 5$	96804	740	$6.0E - 5$	$6.5E - 9$
0.25	3	100	80	330	212	83	$9.23E + 6$	$1.88E + 5$	96854	1400	$6.0E - 5$	$6.5E - 9$
0.125	2	1000	80	440	86	40	$3.92E + 7$	$8.88E + 4$	49778	1400	$3.9E - 5$	$6.4E - 9$
0.125	3	100	80	400	86	32	$4.74E + 6$	$9.66E + 4$	49796	1840	$4.1E - 5$	$6.4E - 9$

Legend: ε - error-rate, d - depth, β -branching factor, k_x, k_y - key lengths, r - response length in auth. stage, r_{tr} - response length in tree stage, C_{rdr} - expected reader computation, C_{tag} - expected tag computation, comm - expected total communication, mem - tag memory requirements.

Conclusions

- Developed HB-Type protocol for privacy preserving authentication
 - For low-cost tags incapable of PRF calculations
 - Reduces reader load compared to PRF-based protocols
- Significant improvements over naive use of HB+ in tree-based scheme
 - Reduce significantly communication
 - Almost as efficient as underlying HB+ protocol

Simulations

Method	Comp. Time (seconds) mean \pm stdev	FAR		FRR	
		expected / observed	expected / observed	expected / observed	expected / observed
ES (HB+)	110.180 \pm 0.9656	0.76 / 0.75	0.365 / 0.450		
Tree HB+ ($d = 2$)	0.131 \pm 0.0234	1.4E-6 / 0	0.432 / 0.422		
Tree HB+ ($d = 3$)	0.019 \pm 6.75E-4	1.4E-6 / 2.14E-6	0.382 / 0.382		
Tree PRF(AES)	7.046 \pm 0.0561	—	—		

- Used $r = 96$, provides tree-HB+ FRR value similar to original HB+ protocol FRR
- Large run-time improvement for Tree-HB+ relatively to compared protocols
 - Computation time faster than PRF by order of 368 ($d=3$) and 53 ($d=2$)
- Run on IBM Thinkpad, Intel CPU, T2400, 1.83 GHz and 1.99 GB RAM, Simulations done on Matlab and Java libraries for PRF

Security Analysis

- Attack model:
 - Attacker can eavesdrop on communication
 - Attacker can communicate directly with reader or tag
 - Can NOT modify messages sent between them
- Authentication stage as resilient against impersonation as HB+
- Privacy security relies on hardness of LPN
 - Based on $F_x(a) = \mathbf{B}y \oplus \textit{noise}$ behaving like a pseudorandom function

Security Analysis

- There is no security against man-in-the-middle attacks: Gilbert et al. [ePrint 05], Quafi et al. [Asiacrypt 08]
- Like all tree-based protocols, somewhat vulnerable to tag compromise