Authenticating Pervasive Devices with Human Protocols

Presented by Xiaokun Mu

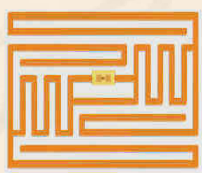
Paper Authors:

Ari Juels
RSA Laboratories

Stephen A. Weis
Massachusetts Institute of Technology

Authentication Problems

- It seems inevitable that many applications will come to rely on basic RFID tags or other low-cost devices as authenticators.
- (RFID = Radio Frequency Identification)



- **An RFID tag used by Wal-Mart**

Why we use RFID tag?

- Combat counterfeiting and theft (4 examples)

FDA proposed attaching RFID tags to prescription drug containers in an attempt to combat counterfeiting and theft.

Supermarket Products

Library Books

Smartrip Metro Card

Skimming Attack of RFID Tag

- Most RFID devices today promiscuously broadcast a static identifier with no explicit authentication procedure. This allows an attacker to surreptitiously scan identifying data in what is called a skimming attack. Besides the implicit threat to privacy, skimmed data may be used to produce cloned tags, exposing several lines of attack.
- For example, in a swapping attack, a thief skims valid RFID tags attached to products inside a sealed container. The thief then manufactures cloned tags, seals them inside a decoy container, and swaps the decoy container with the original.
- Clone creates Denial-of-Service

Example specification for a 5-10 cents low-cost RFID tag

- **Storage:** 128-512 bits of read-only storage.
- **Memory:** 32-128 bits of volatile read-write memory.
- **Gate Count:** 1000-10000 gates.
- **Security Gate Count Budget:** 200-2000 gates.
- **Operating Frequency:** 868-956 MHz (UHF).
- **Scanning Range:** 3 meters.
- **Performance:** 100 read operations per second.
- **Clock Cycles per Read:** 10,000 clock cycles.
- **Tag Power Source:** Passively powered by Reader via RF signal.
- **Power Consumption:** 10 microwatts.
- **Features:** Anti-Collision Protocol Support Random Number Generator

Humans vs. RFID Tags

- Like people, tags can neither remember long passwords nor keep long calculations in their working memory.
- Tags are better at performing logical operations.
- Tags are also better at picking random values.
- Tag secrets can be completely revealed through physical attacks.
- Physically attacking people tends to yield unreliable results.

How to utilize the similarities?

- Adopting human authentication protocols in low-cost pervasive computing devices.
- Allowing a person to log onto an un-trusted terminal while someone spies over his/her shoulder, without the use of any scratch paper or computational devices.
- A simple password would be immediately revealed to an eavesdropper.

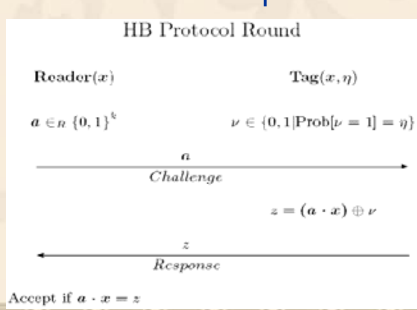
The HB Protocol

- This paper focuses primarily on the human authentication protocols of Hopper and Blum.
- Hopper and Blum's secure human authentication protocol is only secure against passive eavesdroppers.
- Authors augment the HB protocol against active adversaries that may initiate their own tag queries.

How does HB work?

- Suppose Alice and a computing device *C* share an *k*-bit secret ***x***, and Alice would like to authenticate herself to *C*. *C* selects a random challenge ***a*** ∈ {0, 1}*k* and sends it to Alice. Alice computes the binary inner-product ***a · x***, then sends the result back to *C*. *C* computes ***a · x***, and accepts if Alice's parity bit is correct.
- In a single round, someone imitating Alice who does not know the secret ***x*** will guess the correct value ***a · x*** half the time. By repeating for *r* rounds, Alice can lower the probability of naively guessing the correct parity bits for all *r* rounds to 2^{-r}.

A single round of the HB authentication protocol



A single round of the HB authentication protocol

- the tag plays the role of the Alice and the reader of the authenticating device *C*. Each authentication consists of *r* rounds, where *r* is a security parameter.
- Of course, an eavesdropper capturing *O*(*k*) valid challenge-response pairs between Alice and *C* can quickly calculate the value of ***x*** through Gaussian elimination.
- To prevent revealing ***x*** to passive eavesdroppers, Alice can inject noise into her response. Alice intentionally sends the wrong response with constant probability *η* ∈ (0, 1/2). *C* then authenticates Alice's identity if fewer than *ηr* of her responses are incorrect.

Implementation of HB protocol

- Calculations are very simple to implement in hardware. (AND, OR, XOR operations)
- Noise bit *v* can be cheaply generated. (thermal noise, shot noise, diode breakdown noise)

Remarks of HB

- The HB protocol can be also deployed as a *privacy-preserving* identification scheme.
- A reader may initiate queries to a tag without actually knowing whom that tag belongs to.
- Based on the responses, a reader can check its database of known tag values and see if there are any likely matches.
- This preserves the privacy of a tag's identity, since an eavesdropper only captures an instance of the LPN problem.

Learning Parity in the Presence of Noise

- Suppose that an eavesdropper, i.e., a passive adversary, captures q rounds of the HB protocol over several authentications and wishes to impersonate Alice. Consider each challenge a as a row in a matrix A; similarly, let us view Alice's set of responses as a vector z. Given the challenge set A sent to Alice, a natural attack for the adversary is to try to find a vector x1 that is functionally close to Alice's secret x. In other words, the adversary might try to compute a x1 which, given challenge set A in the HB protocol, yields a set of responses that is close to z. (Ideally, the adversary would like to figure out x itself.)

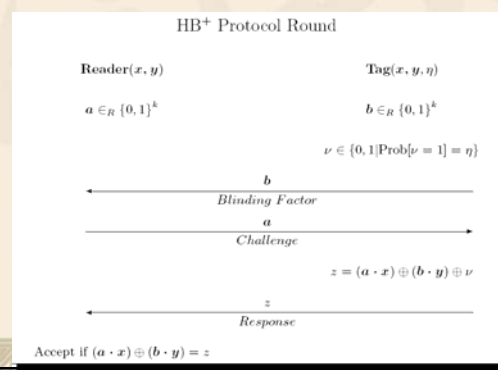
The LPN Problem

- may also be formulated and referred to as the Minimum Disagreement Problem.
- also known as the syndrome decoding problem.
- to be NP-Hard, and is hard even within an approximation ratio of two.
- is not efficiently solvable in the statistical query model
- is both pseudo-random and log-uniform. (HB)

HB and HB+

- HB protocol is only secure against passive eavesdroppers.
- HB+ protocol is effective to active eavesdroppers.
- HB+ has more parameter than HB.

A single round of the HB+ protocol



Defending Against Active Attacks

- adaptive (non-random) challenges.
- additional k-bit random secret $\boldsymbol{y}$.
- the tag in the HB+ protocol first generates random k-bit “blinding” vector $\boldsymbol{b}$ and sends it to the reader.
- Tag computes $z = (\boldsymbol{a} \cdot \boldsymbol{x}) \oplus (\boldsymbol{b} \cdot \boldsymbol{y}) \oplus v$, and sends the response z to the reader.

Defending Against Active Attacks

- HB+ requires the tag (playing the role of the human), to generate a random k-bit string $\boldsymbol{b}$ on each query. If the tag (or human) does not generate uniformly distributed $\boldsymbol{b}$ values, it may be possible to extract information on $\boldsymbol{x}$ or $\boldsymbol{y}$.

Security Intuition

- In the augmented protocol HB+, an adversary can still, of course, select $\boldsymbol{a}$ challenges to mount an active attack.
- The tag effectively prevents an adversary from actively extracting $\boldsymbol{x}$ or $\boldsymbol{y}$ with non-random $\boldsymbol{a}$ challenges.
- $(\boldsymbol{b} \cdot \boldsymbol{y}) \oplus v \oplus (\boldsymbol{a} \cdot \boldsymbol{x})$ prevents an adversary from extracting information through non-random $\boldsymbol{a}$ challenges.

Security Intuition

- The value $(\boldsymbol{b} \cdot \boldsymbol{y}) \oplus v$ effectively “blinds” the value $\boldsymbol{a} \cdot \boldsymbol{x}$ from both passive and active adversaries.
- An adversary able to efficiently learn $\boldsymbol{y}$ can efficiently solve the LPN problem.
- The blinding therefore protects against leaking the secret $\boldsymbol{x}$ in the face of active attacks.
- Without knowledge of $\boldsymbol{x}$ or $\boldsymbol{y}$, an adversary cannot create a fake tag that will respond correctly to a challenge $\boldsymbol{a}$.

Security Proofs

- Notation and Definitions: define a tag-authentication system in terms of a pair of probabilistic functions (R, T), namely a reader function R and a tag function T.
- T is defined in terms of a noise parameter η, a k-bit secret $\boldsymbol{x}$, and a set of q random k-bit vectors $\{\boldsymbol{a}(i)\}_q(i=1)$
- Let q be the maximum number of protocol invocations on T in this experiment.

Security Proofs

- For protocol HB, we denote the fully parameterized tag function by $T_{x,\boldsymbol{A},\eta}$.
- On the i-th invocation of this protocol, T is presumed to output $(\boldsymbol{a}(i), (\boldsymbol{a}(i) \cdot \boldsymbol{x}) \oplus v)$.
- Here v is a bit of noise parameterized by η.
- This models a passive eavesdropper observing a round of the HB protocol.

Slide 23

m k1 m k, 4/9/2008

Security Proofs

- For HB+, we denote a fully parameterized tag function as $T_{x,y,\eta}$.
- On the *i*-th invocation of T for this protocol, the tag outputs some random $\mathbf{b}(i)$.
- outputs $z = (\mathbf{a}(i) \cdot \mathbf{x}) \oplus (\mathbf{b}(i) \cdot \mathbf{y}) \oplus v$.
- the reader $R_{x,y}$ takes as input a triple ($\mathbf{a}$, $\mathbf{b}$, z) and outputs either "accept" or "reject".

Security Proofs

- For both protocols HB and HB+, we consider a two-phase attack model involving an adversary comprising a pair of functions $A = (A_{query}, A_{clone})$, a reader R, and a tag T.
- In the first, "query" phase, the adversarial function A_{query} has oracle access to T and outputs some state σ.
- The second, "cloning" phase involves the adversarial function A_{clone}.

Security Proofs

- A_{clone} takes the full experimental state as input.
- Presume that a protocol invocation takes some fixed amount of time.
- Characterize the total protocol time by three parameters:
  1. *the number of queries to a T oracle, q;*
  2. *the computational runtime t1 of A_{query};*
  3. *the computational runtime t2 of A_{clone}.*

Security Proofs

- Let D be some distribution of $q \times k$ matrices.
- let $R\leftarrow$ denote uniform random assignment.

Experiment $\mathbf{Exp}_{\mathcal{A},D}^{HB-attack}[k,\eta,q]$

$$x \xleftarrow{R} \{0,1\}^k;$$
$$\mathbf{A} \xleftarrow{R} D$$
$$\sigma \leftarrow \mathcal{A}_{query}^{\mathcal{T}_{x,\mathbf{A},\eta}};$$
$$\mathbf{a}' \xleftarrow{R} \{0,1\}^k;$$
$$z' \leftarrow \mathcal{A}_{clone}(\sigma, \mathbf{a}', \text{"guess"});$$
$$\text{Output } \mathcal{R}_x(\mathbf{a}', z').$$

Experiment $\mathbf{Exp}_{\mathcal{A}}^{HB^+-attack}[k,\eta,q]$

$$x, y \xleftarrow{R} \{0,1\}^k;$$
$$\sigma \leftarrow \mathcal{A}_{query}^{\mathcal{T}_{x,y,\eta}};$$
$$\mathbf{b}' \leftarrow \mathcal{A}_{clone}(\sigma, \text{"initiate"});$$
$$\mathbf{a}' \xleftarrow{R} \{0,1\}^k;$$
$$z' \leftarrow \mathcal{A}_{clone}(\sigma, \mathbf{a}', \mathbf{b}', \text{"guess"});$$
$$\text{Output } \mathcal{R}_{x,y}(\mathbf{a}', \mathbf{b}', z').$$

Security Proofs

- Consider A's advantage for key-length k, noise parameter η, over q rounds. In the case of the HB-attack experiment, this advantage will be over matrices $\mathbf{A}$ drawn from the distribution D:
- Let $Time(t1, t2)$ represent the set of all adversaries A with runtimes $t1$ and $t2$, respectively. Denote the maximum advantage over $Time(t1, t2)$:

$$\mathsf{Adv}_{\mathcal{A},D}^{HB-attack}(k,\eta,q) = \left| Pr\left[\mathbf{Exp}_{\mathcal{A},D}^{HB-attack}[k,\eta,q] = \text{"accept"}\right] - \frac{1}{2} \right|$$

$$\mathsf{Adv}_{D}^{HB-attack}(k,\eta,q,t_1,t_2) = \max_{\mathcal{A}\in Time(t_1,t_2)} \{\mathsf{Adv}_{\mathcal{A},D}^{HB-attack}(k,\eta,q)\}$$

Reduction from LPN to HB-Attack

- ***A*** may actually be negligible over modified (***A***, ***z***) values, i.e., over the distribution R_{A_i}.
- Matrices are not independent over this distribution.
- Any two sample matrices are identical in all but one column.
- it is possible in principle that ***A*** loses its advantage over this distribution of matrices and that the reduction fails to work.

Reduction from LPN to HB-Attack

Lemma 1. *Let* $\mathsf{Adv}_U^{HB-Attack}(k, \eta, q, t_1, t_2) = \epsilon$, *where* U *is a uniform distribution over binary matrices* $\mathbb{Z}_2^{q \times k}$, *and let* $\mathcal{A}$ *be an adversary that achieves this* ϵ*-advantage. Then there is an algorithm* $\mathcal{A}'$ *with running time* $t_1' \leq kLt_1$ *and* $t_2' \leq kLt_2$, *where* $L = \frac{8(\ln k - \ln \ln k)}{(1-2\eta)^2 \epsilon^2}$, *that makes* $q' \leq kLq + 1$ *queries that can correctly extract all* k *bits of* x *with probability* $\epsilon' \geq \frac{1}{k}$.

- It might even be possible to devise a rigorous reduction that uses a single matrix ***A*** for all columns. We leave these as open questions.
- It is entirely possible that the adversary's advantage is preserved when, for each column *j*, samples are drawn from the ***RA****ji* subspace for a matrix ***A****j* .

Reduction from HB to HB+ Attack

- Lemma 3 is the main technical core of the paper, but its proof must be omitted here due to lack of space.

Lemma 3. *If* $\mathsf{Adv}_U^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$, *then*

$$\mathsf{Adv}_U^{HB-Attack}(k, \eta, q', t_1', t_2') \geq \frac{\zeta^3}{4} - \frac{\zeta^3 + 1}{2k},$$

where $q' \leq q(2 + \log_2 q)$, $t_1' \leq kq't_1$, $t_2' \leq 2kt_2$, *and* $k \geq 9$.

Two main technical challenges in the proof.

- Finding the right embedding of *w* in a secret bit of the simulated HB+-oracle.
- Comes in the rewinding and extraction. There is the possibility of a non-uniformity in the responses of *A*+. An important technical lemma is necessary to bound this non-uniformity.

Reduction of LPN to HB+-Attack

- By combining Lemmas 1 and 3, we obtain a concrete reduction of the LPN problem to the HB+-attack experiment.
- The theorem follows directly from Lemmas 1 and 3.

Theorem 1. *Let* $\mathsf{Adv}^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$, *where* U *is a uniform distribution over binary matrices* $\mathbb{Z}_2^{q \times k}$, *and let* $\mathcal{A}$ *be an adversary that achieves this* ζ*-advantage. Then there is an algorithm that can solve a random* $q' \times k$ *instance of the LPN problem in time* (t_1', t_2') *with probability* $\frac{1}{k}$, *where* $t_1' \leq k^2 Lq(2+\log_2 q)t_1$, $t_2' \leq 2k^2 Lt_2$, $q' \leq kLq(2 + \log_2 q)$, *and* $L = \frac{128k^4(\ln k - \ln \ln k)}{(1-2\eta)^2(\zeta^3(k-2)+2)^2}$.

To put this in asymptotic terms, the LPN problem may be solved by an adversary where $\mathsf{Adv}^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$ in time $O(\frac{(k^5 \log k)(q \log q)\ t}{(1-2\eta^2)\zeta^6})$, where $t = t_1 + t_2$.

Conclusion and Open Questions

- Presents a new authentication protocol named HB+ that is appropriate for low-cost pervasive computing devices.
- The HB+ protocol is secure in the presence of both passive and active adversaries.
- The HB+ should be implemented within the tight resource constraints of today's EPC-type RFID tags.
- The security of the HB+ protocol is based on the LPN problem, whose hardness over random instances remains an open question.

Open Questions

- Open question 1: whether the two-round variant of HB+ is secure.
- Open question 2: the hardness of the "Sum of *k* Mins" has not been studied as much as the LPN problem, nor is it clear whether this protocol can efficiently be adapted for low-cost devices.

THANK YOU !!!