Randomness in Cryptography January 10, 2013

Lecture 1: One-Time MACs, (XOR)Universal hashing, Weak Keys

Lecturer: Yevgeniy Dodis *Scribe: Eric Miles*

In today's lecture we study one-time message authentication codes (MACs) which are secure in an information-theoretic sense. We will see that, compared to information-theoretically secure encryption, significantly better parameters can be achieved. We will also study such MACs in the setting of imperfect randomness, i.e. when the secret key is not drawn from the uniform distribution but rather is only guaranteed to have some min-entropy.

1 Class Organization

Before starting the technical material, we make two remarks on the class itself.

First, all registered students will be expected to scribe roughly two lectures, and all visitors are encouraged to scribe one lecture.

Second, throughout the lectures there will be a number of problems to be solved outside of class time, with varying levels of difficulty. *Exercises* are simple problems which will usually just involve a routine calculation. *Questions* will require slightly more work, but will still be on the easier end of the spectrum. At the other end of the spectrum we have *projects*, which will be more open-ended and for which the solution may not be known. Finally *quesjects* will be somewhere in between the latter two, still requiring work but with a somewhat clearer path to a solution.

2 One-time MACs

We start by defining a (one-time) *message authentication code* (MAC). The setting is the following: we have two parties, A(lice) and B(ob), who share a secret key $r \in \{0,1\}^m$. A wants to send a message $x \in \{0,1\}^n$ to B along with a tag $t \in \{0,1\}^\lambda$ that allows B to verify that the message came from A. To do so, they use a function $\mathsf{Tag} : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^\lambda$; specifically, A sends x and $t := \mathsf{Tag}(r, x)$, and B receives (x', t') and verifies that $t' = \mathsf{Tag}(x', r)$. Throughout, we will use Tag_r to denote the function $\mathsf{Tag}(r, \cdot)$.

In general one can (and does) consider randomized MACs, but today we will only consider deterministic MACs. As a result, the correctness property, namely that B will always accept a message with a valid tag, is immediate.

To define the security of a MAC, consider the following game G_r parameterized by $r \in \{0,1\}^m$. There are two players: a challenger C who receives r as input, and an adversary E(ve) who receives no input. G_r has the following three steps.

1. E chooses $x \in \{0,1\}^n$ and sends x to C.

2. C computes and sends $t := \mathsf{Tag}_r(x)$ to E.

3. E outputs $(x', t') \in \{0,1\}^n \times \{0,1\}^\lambda$.

We say that E *wins* G_r if $x' \neq x$ and $\mathsf{Tag}_r(x') = t'$, and write $\mathsf{Adv}_E(r) := \Pr[E \text{ wins } G_r]$ to denote E's *advantage*. In general we write $\mathsf{Adv}_E^{G_r}(r)$ if we need to specify the game.

Our goal in this lecture is to obtain an efficient function Tag such that, for every computationally unbounded adversary E, $\mathsf{Adv}_E(r)$ is negligible in λ. Clearly if r is fixed, this is impossible as we can consider an E that has r hardwired. Thus, the following security definition considers secret keys r that are chosen probabilistically.

Here and throughout the lecture notes, U_m denotes the uniform distribution on $\{0,1\}^m$. We will use capital letters to denote random variables and/or the distributions from which they are sampled, and lower-case letters to denote specific values.

Definition 1 Let R be a distribution on $\{0,1\}^m$ and $\delta > 0$. A function $\mathsf{Tag} : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^\lambda$ is a (R,δ)*-secure one-time MAC* if for every E,

$$\mathbb{E}_{r \leftarrow R}[\mathsf{Adv}_E(r)] \leq \delta,$$

When $R \equiv U_m$, we simply say δ*-secure.* ♢

This definition captures what we intuitively want from a (one-time) MAC, because any eavesdropper E who overhears one message from A to B does not gain enough information to then forge any message to B from A.

In constructing MACs, there are two general goals. The first is to minimize the tag length λ and the error δ for given key and message lengths m, n. The second, more common goal is to minimize the tag and key lengths λ, m for a given message length n and error δ.

We will construct MACs from a certain type of hash functions, defined next.

Definition 2 Let $n, \lambda, p \in \mathbb{N}$ and $\delta > 0$. A family of functions $H = \{h_a : \{0,1\}^n \to \{0,1\}^\lambda \mid a \in \{0,1\}^p\}$ is δ*-almost XOR-universal (δ-AXU)* if $\forall x \neq x' \in \{0,1\}^m, y \in \{0,1\}^\lambda$:

$$\Pr_{A \leftarrow U_p}[h_A(x) \oplus h_A(x') = y] \leq \delta. \tag{1}$$

If $\delta = 2^{-\lambda}$ (which is optimal when $\lambda < n$), we say H is *XOR-universal (XU)*. If (1) holds only for $y = 0^n$, namely

$$\Pr_{A \leftarrow U_p}[h_A(x) = h_A(x')] \leq \delta. \tag{2}$$

we say δ*-almost universal (δ-AU)* (or *universal* when $\delta = 2^{-\lambda}$), respectively. ♢

2.1 Constructing MACs from δ-AXU functions

Before constructing δ-AXU functions, we show how to construct MACs from them.

Theorem 1 *Let $H = \{h_a : \{0,1\}^n \to \{0,1\}^\lambda \,|\, a \in \{0,1\}^p\}$ be a δ-AXU function family. Then, parsing $r = (a,b) \in \{0,1\}^p \times \{0,1\}^\lambda$, the function*

$$\mathsf{Tag}_r(x) := h_a(x) \oplus b$$

is a δ-secure one-time MAC with key length $m = p + \lambda$.

Exercise 1 *Find a counterexample to Theorem 1 if instead* $\mathsf{Tag}_r(x) := h_a(x)$ *(i.e. if* $\oplus b$ *is omitted).*

Recall the game G_r that defines the security of a given MAC Tag_r. We now prove Theorem 1 by defining another game G'_r with the following properties: first, any adversary with advantage ε in G_r implies the existence of an adversary with advantage ε in G'_r; second, every adversary has advantage bounded by δ in G'_r when H is δ-AXU. (In fact, as a syntactic convienience we will define two games G'_r, G''_r with these properties.)

Proof: We first restate the game G_R when $R \leftarrow U_m$. Throughout the proof, we assume wlog that the adversary E is deterministic and computationally unbounded, and also that E never outputs $X' = X$ (as then she loses the game for sure).

$G_R :=$

1. E chooses and sends $X \in \{0,1\}^n$ to C.
2. C samples $(A, B) \leftarrow U_p \times U_\lambda$ and sends $T = \mathsf{Tag}_{(A,B)}(X) = h_A(X) \oplus B$ to E.
3. E computes and outputs (X', T'), and wins if $X \neq X'$ and $\mathsf{Tag}_{(A,B)}(X') = T'$.

We define the game G'_R to be the same as G_R with the following change to step 3: E computes X', T' but instead outputs $(X', Y = T' \oplus T)$, and wins if $h_A(X) \oplus h_A(X') = Y$. Notice, this is only a syntactic change, since

$$h_A(X') \oplus B = T' \text{ iff } (h_A(X') \oplus B) \oplus (h_A(X) \oplus B) = T' \oplus T \text{ iff } h_A(X) \oplus h_A(X') = Y$$

Hence, clearly we have

$$\max_E \left(\mathsf{Adv}_E^{G_R}(R)\right) = \max_E \left(\mathsf{Adv}_E^{G'_R}(R)\right)$$

when $R \leftarrow U_m$.

We now define a third game G''_R, which is only different from G'_R in the way the tag T is computed.

$G''_R :=$

1. E chooses and sends $X \in \{0,1\}^n$ to C.
2. C samples $T \leftarrow U_\lambda$ and sends it to E.
3. E computes and outputs (X', Y).
4. C samples $A \leftarrow U_p$, and E wins if $h_A(X) \oplus h_A(X') = Y$.

Notice, because B is sampled uniformly in game G'_R, we have that T is distributed uniformly in both G'_R and G''_R. Moreover, T is independent from A, which justifies the "delayed" sampling of A in step 4 of game G''_R. Thus the changes from G'_R to G''_R preserve the distribution of each random variable, and we have

$$\max_E \left(\mathsf{Adv}_E^{G'_R}(R)\right) = \max_E \left(\mathsf{Adv}_E^{G''_R}(R)\right).$$

Furthermore, because A is sampled at random *after* $X \neq X'$ and Y are defined, the fact that H is δ-AXU implies that

$$\max_E \left(\mathsf{Adv}_E^{G''_R}(R)\right) \leq \delta$$

which implies the theorem. □

Note the importance of sampling B uniformly at random in the proof of this theorem.

2.2 Constructing δ-AXU functions

We now turn to constructing δ-AXU function families $H = \{h_a : \{0,1\}^n \to \{0,1\}^\lambda \,|\, a \in \{0,1\}^p\}$. But first, we note the following lower bounds on the key size p.

if H is...	then...
XU	$p \geq n$
δ-AXU	$p \geq \log(1/\delta) + \log(n/\lambda)$
universal	$p \geq n - \lambda$
δ-AU	$p \geq \log(1/\delta) + \log((n-\lambda)/\lambda)$

The first construction we consider is trivially XU, but has very poor key length $p = n\lambda$.

Construction 1 *Let the key $a \in \{0,1\}^{\lambda \times n}$ be a matrix, and define $h_a(x) := a \cdot x$.*

We now observe that by instead letting a come from the set of so-called Hankel matrices, we can save on the key length.

Definition 3 A matrix $a \in \{0,1\}^{\lambda \times n}$ is a *Hankel matrix* if each reverse diagonal is constant. That is, for each $2 \leq i \leq \lambda$ and each $1 \leq j \leq n-1$, $a_{i,j} = a_{i-1,j+1}$. ♢

Construction 2 *Let the key $a \in \{0,1\}^{\lambda \times n}$ be a Hankel matrix, and define $h_a(x) := a \cdot x$.*

Question 1 *Prove that Construction 2 is XU.*

Note that a Hankel matrix is specified by giving a single bit for each of the $n + \lambda - 1$ reverse diagonals. Thus we have $p = n + \lambda - 1$, which is $< 2n$ when $\lambda \leq n$ and thus within a constant factor of the XU lower bound.

The next construction uses finite fields and achieves $p = n$, matching the XU lower bound. We assume some implicit bijection between $\{0,1\}^n$ and the finite field $GF(2^n)$ defined by an irreducible $GF(2)$-polynomial of degree n.

Construction 3 *Let the key $a \in GF(2^n)$, and define $h_a(x)$ to be the lower-order λ bits of $a \cdot x$ (where multiplication is in $GF(2^n)$).*

Question 2 *Prove that Construction 3 is XU.*

Our final XU construction achieves the same key length, but uses inner products over the finite field $GF(2^\lambda)$ and will be more convenient to modify later.

Construction 4 *Assume that $n = b\lambda$ for some $b \in \mathbb{N}$. Let the key $a = (a_1, \ldots, a_b) \in GF(2^\lambda)^b$. Then parse x as $(x_1, \ldots, x_b) \in GF(2^\lambda)^b$, and define $h_a(x) := \langle a, x\rangle = \sum_i a_i x_i$.*

Lemma 1 *Construction 4 is XU.*

Proof: Fix $x \neq x' \in GF(2^\lambda)^b$ and $y \in GF(2^\lambda)$. Define $z = x - x' \neq 0^b$. Then we have

$$\Pr_a[h_a(x) \oplus h_a(x') = y] = \Pr_a[\langle a, x\rangle \oplus \langle a, x'\rangle = y] = \Pr_a[\langle a, z\rangle = y]$$

because addition and subtraction in $GF(2^\lambda)$ both correspond to bit-wise $\oplus$. We claim that the latter probability equals $2^{-\lambda}$, which implies the lemma. To see this this, assume wlog that $z_1 \neq 0$, and note that for any setting of $a_2, \ldots, a_b$ we have

$$\Pr_{a_1}[\langle a, z\rangle = y] = \Pr_{a_1}[a_1 = c] = 2^{-\lambda}$$

where $c := (y - \sum_{i \geq 2} a_i z_i) \cdot z_1^{-1} \in GF(2^\lambda)$. □

To achieve only universality as opposed to XOR-universality, we can save λ bits in the key (and thus match the lower bound) by fixing $a_1 = 1 \in GF(2^\lambda)$.

We now modify Construction 4 to obtain a δ-AXU family for $\delta < n \cdot 2^{-\lambda}$ while reducing the key length to λ. This is done by replacing $(a_1, \ldots, a_b)$ with $(a, a^2, \ldots, a^b)$ for a single $a \in GF(2^\lambda)$.

Construction 5 *Assume that $n = b\lambda$ for some $b \in \mathbb{N}$. Let the key $a \in GF(2^\lambda)$. Then parse x as $(x_1, \ldots, x_b) \in GF(2^\lambda)^b$, and define $h_a(x) := \sum_i a^i \cdot x_i$.*

Lemma 2 *Construction 5 is $(2^{-\lambda} \cdot n/\lambda)$-AXU.*

Proof: Fix $x \neq x'$ and y as before, and let $z = x - x' \neq 0^b$. Then, if we define $z_0 := y$, we have

$$\Pr_a[h_a(x) \oplus h_a(x') = y] = \Pr_a\left[\sum_{i=0}^{b} a^i \cdot z_i = 0\right].$$

Thus $h_a(x) \oplus h_a(x') = y$ only for those a that are roots of the polynomial $\varphi(s) := \sum_{i \leq b} z_i \cdot s^i$. Since φ is of degree $\leq b$ and thus has $\leq b = n/\lambda$ roots, this implies the lemma. □

Letting $\delta = 2^{-\lambda} \cdot n/\lambda$, we see that Construction 5 achieves key length $p = \lambda < \log n + \log(1/\delta)$. (In general one can have constructions that decouple p from λ, but we will not consider those here.) The following corollary is immediate.

Corollary 3 *For every n and δ, there is a δ-AXU family with $p = \lambda = \log(n/\delta)$.*

2.3 Putting it together

Combining the results of the preceding subsections, the following main theorem is proved. Recall that for a MAC, n denotes the message length, m denotes the key length, λ denotes the tag length, and δ denotes the maximum advantage of any adversary.

Theorem 2 *There exist δ-secure one-time MACs in each of the following parameter regimes.*

1. *For any n and λ, $m = 2\lambda$ and $\delta = n \cdot 2^{-\lambda} = n \cdot 2^{-m/2}$.*
2. *For any n and m, $\lambda = m/2$ and $\delta = n \cdot 2^{-m/2}$.*
3. *For any n and δ, $m = 2\log(n/\delta)$ and $\lambda = \log(n/\delta)$.*

It is interesting to note that if one only cares about message authentication rather than encryption, Shannon's well-known lower bound in the setting of one-time statistical security, namely that key length $\geq$ message length, does not hold.

We remark on the optimality of this MAC construction. First, there is a lower bound by Alon (unpublished) which shows that any MAC must satisfy

$$m \geq \log n + 2\log(1/\delta) - \log\log(1/\delta). \tag{3}$$

The construction in Theorem 2 essentially achieves this bound up to the constant factor 2 on $\log n$. Second, a paper by Gemmell and Naor [2] notes that the existence of a MAC with $m = \log n + 2\log(1/\delta)$ can be proved non-constructively, which again improves on Theorem 2 only by the constant factor 2 on $\log n$.

Quesject 1 *Prove either or both of the above bounds, namely the lower bound (3) and the (non-constructive) MAC that achieves $m = \log n + 2\log(1/\delta)$.*

Before moving on, we note the following two simple lower bounds. First, the tag length λ must be at least $\log(1/\delta)$; this is because an adversary can correctly guess a tag with probability $2^{-\lambda}$. Second, the key length $m \geq 2\log(1/\delta)$ (even when $n = 1$). We will not prove it now (see next lecture), but the intuition is that when $R \leftarrow U_{2\log(1/\delta)}$, $\mathsf{Tag}_R(x)$ has $\log(1/\delta)$ bits of entropy, and for any $x' \neq x$ the value $\mathsf{Tag}_R(x')$ has $\log(1/\delta)$ bits of entropy even conditioned on $\mathsf{Tag}_R(x)$. Note that when the message length $n = 1$, this can be achieved by parsing $r = (r_0, r_1) \in \{0,1\}^{2\log(1/\delta)}$ and defining $\mathsf{Tag}_r(x) = r_x$ where $x \in \{0,1\}$.

3 MACs with imperfect randomness

We now begin to study a question which we will continue in the next lectures, namely: is it possible to build a MAC from an imperfect source of randomness?

To make sense of this question we must formalize what is meant by "imperfect". This is done by defining the notion of the *entropy* of a given distribution R. There are multiple types of entropy that one can define; the most common form is *Shannon entropy*, denoted $\mathbf{H}_{sh}(R)$, which we will not define here. Shannon entropy is typically not the "right" notion of entropy for cryptography, because it is possible to define pathological distributions that have high Shannon entropy but are useless to cryptographic algorithms. The main type of entropy that we will consider is *min-entropy*, denoted $\mathbf{H}_\infty(R)$ and defined next. Later we will also consider *collision entropy*, which is denoted $\mathbf{H}_2(R)$. For any distribution R, these three types of entropy satisfy $\mathbf{H}_\infty(R) \leq \mathbf{H}_2(R) \leq \mathbf{H}_{sh}(R)$.

DEFINITION 4 Let R be a distribution. The *predictability* of R is defined by $\mathsf{Pred}(R) := \max_r(\Pr[R = r])$, and the *min-entropy* of R is defined by $\mathbf{H}_\infty(R) := \log(1/\mathsf{Pred}(R))$. When $\mathbf{H}_\infty(R) \geq k$ we say that R is a *k-source.* ♢

Note that R is a k-source if and only if $\Pr[R = r] \leq 2^{-k}$ for every r in the support of R. Also, the value $\mathsf{Pred}(R)$ is equal to the maximum, over all computationally unbounded adversaries E, of $\Pr_R[E \text{ guesses } R]$.

With this definition in hand, we now define MACs with imperfect randomness and prove a general transformation from perfect randomness to imperfect randomness.

DEFINITION 5 A function Tag is a *(k, δ)-secure one-time MAC* if it is an (R, δ)-secure one-time MAC for all k-sources R. ♢

Theorem 3 *If* Tag *is a δ-secure MAC with key length m, then for every $k \leq m$ it is also a $(k, 2^{m-k} \cdot \delta)$-secure MAC.*

Informally speaking, a theorem such as this one holds for any cryptographic task which deals with "unpredictability" (as opposed to the stronger notion of "indistinguishability"). Theorem 3 follows immediately from the next lemma, where $f(r) = \mathsf{Adv}_E(r)$ is indeed non-negative.

Lemma 4 *For every function $f : \{0,1\}^m \to \mathbb{R}^{\geq 0}$ and every k-source R on $\{0,1\}^m$,*

$$\mathbb{E}[f(R)] \leq 2^{m-k} \cdot \mathbb{E}[f(U_m)].$$

Proof: Because $\Pr[R = r] \leq \mathsf{Pred}(R)$ for all r by definition, we have

$$\begin{aligned} \mathbb{E}[f(R)] &= \sum_r \Pr[R = r] \cdot f(r) \\ &\leq \mathsf{Pred}(R) \cdot 2^m \cdot \sum_r \frac{1}{2^m} \cdot f(r) \\ &= 2^{m - \mathbf{H}_\infty(R)} \cdot \mathbb{E}[f(U_m)]. \end{aligned}$$

Notice, the inequality crucially used the fact that $f \geq 0$. Indeed, the result is wrong for general f, as we will see later. □

Combining Theorems 1, 2, and 3, we obtain the following.

Theorem 4 *For any k such that $m/2 + \log n < k \leq m$, the function* Tag *defined in Theorem 1 is a $(k, n \cdot 2^{m/2-k})$-secure MAC with tag length $\lambda = m/2$.*

In other words, for every n and δ, every $m \geq 2\log(n/\delta)$, and every $(m \geq)$ $k \geq m/2 + \log(n/\delta)$, there exists a (k, δ)-secure MAC with tag length $\lambda = m/2$.

3.1 Conditional Min-Entropy and Direct Proof of Theorem 4

We conclude today's lecture by giving a more direct proof of Theorem 4 that in particular does not use the general transformation of Theorem 3. To do so we need some simple facts about min-entropy, as well as the following notion of conditional min-entropy which comes from [1].

DEFINITION 6 Let A, B be two jointly-distributed random variables, and define

$$\mathsf{Pred}(A \mid B) := \mathbb{E}_{b \leftarrow B}[\mathsf{Pred}(A \mid B = b)] = \max_{E} \left(\Pr_{A,B}[E(B) = A] \right).$$

Then, the *conditional min-entropy* of A given B is $\mathbf{H}_\infty(A \mid B) := \log(1/\mathsf{Pred}(A \mid B))$. ♢

Note that $\mathbf{H}_\infty(A \mid B)$ is *not* equivalent to $\mathbb{E}_{b \leftarrow B}[\log(1/\mathsf{Pred}(A \mid B = b)]$ (which has the $\mathbb{E}$ and log switched). This latter definition turns out not to be very useful for cryptography, since $2^{-\mathbf{H}_\infty(A|B)} = \mathsf{Pred}(A|B) = \max_E(\Pr(E(B) = A)$ measures the best probability E can guess A given B.

Lemma 5 *For every distribution Z and every deterministic function g: $\mathbf{H}_\infty(Z) \geq \mathbf{H}_\infty(g(Z))$.*

Proof: This is equivalent to $\mathsf{Pred}(Z) \leq \mathsf{Pred}(g(Z))$, which holds because applying g to the output of any predictor for Z gives a predictor for $g(Z)$. □

Lemma 6 *For all distributions A, B with $|Support(B)| \leq L$, the following two (equivalent) statements hold.*

1. $\mathbf{H}_\infty(A) \geq \mathbf{H}_\infty(A \mid B) \geq \mathbf{H}_\infty(A, B) - \log L \geq \mathbf{H}_\infty(A) - \log L.$
2. $\mathsf{Pred}(A) \leq \mathsf{Pred}(A \mid B) \leq L \cdot \mathsf{Pred}(A, B) \leq L \cdot \mathsf{Pred}(A).$

Proof: The statements are equivalent by definition. The only non-trivial inequality is $\mathsf{Pred}(A \mid B) \leq \mathsf{Pred}(A, B) \cdot L$, which we now prove following [1, Lem. 2.2].

$$\begin{aligned}
\mathsf{Pred}(A \mid B) &= \mathbb{E}_{b \leftarrow B}[\mathsf{Pred}(A \mid B = b)] \\
&= \sum_b \max_a (\Pr[A = a \mid B = b]) \cdot \Pr[B = b] \\
&= \sum_b \max_a (\Pr[A = a \wedge B = b]) \\
&\leq \sum_b \max_{a,b'} \left(\Pr[A = a \wedge B = b']\right) \\
&= L \cdot \max_{a,b'} \left(\Pr[A = a \wedge B = b']\right) = L \cdot \mathsf{Pred}(A, B).
\end{aligned}$$

A more "algorithmic" way to prove this result is to turn any predictor $E(b)$ for A given $b \leftarrow B$ into a predictor E' for (A, B) as follows. E' samples uniformly random $b \leftarrow$ Support(B), and then runs $a \leftarrow E(b)$, and outputs (a, b). Intuitively, irrespective of the actual distribution B, the random sample of b from Support(B) is "correct" (call this event Cor) with probability at least $1/L$. Moreover, it is easy to see (exercise) that conditioning on Cor does not affect the marginal distribution of "real" (A, B). Thus, $\mathsf{Adv}_{E'}(A, B) \geq \frac{1}{L} \cdot \mathsf{Adv}_E(A|B)$. □

We now turn to the direct proof of Theorem 4. Recall that the MAC we are considering is defined by

$$\mathsf{Tag}_{(a,b)}(x) := b + \sum_{i=1}^{d} x_i \cdot a^i$$

where $a, b \in GF(2^\lambda)$ and $x \in \{0,1\}^n$ is parsed as $(x_1, \ldots, x_d) \in GF(2^\lambda)^d$ for $d := n/\lambda$.

Observe that for any $x \neq x' \in GF(2^\lambda)^d$ and any $t, t' \in GF(2^\lambda)$, the system of equations

$$b + \sum_{i=1}^{d} x_i \cdot a^i = t$$

$$b + \sum_{i=1}^{d} x'_i \cdot a^i = t'$$

has $\leq d$ solutions for a, and these solutions are the same for each $b \in GF(2^\lambda)$. We define the *position* of a given (a, b), denoted $\mathsf{Pos}_{(a,b)}(x, x') \in \{1, \ldots, d\}$, as follows. Let $t = \mathsf{Tag}_{(a,b)}(x)$ and $t' = \mathsf{Tag}_{(a,b)}(x')$, and let $a_1, a_2, \ldots$ be the lexicographically-ordered set of solutions to the above system with this t, t'. Then define $\mathsf{Pos}_{(a,b)}(x, x') := i$ where $a = a_i$.

Proof: Let E be a computationally unbounded adversary, and assume wlog that E is deterministic. Recall the following game between E and the challenger C that defines Tag's security; here we split the final step into two parts as a technical convenience.

$G_{(A,B)} :=$

1. E chooses and sends $x \in (GF(2^\lambda))^d$ to C.
2. C samples $(A, B) \leftarrow GF(2^\lambda) \times GF(2^\lambda)$ and sends $T = \mathsf{Tag}_{(A,B)}(x)$ to E.
3. E computes X' (as a function of T) and sends it to C.
4. E computes T' (as a function of T) and sends it to C.

E wins $G_{(A,B)}$ if $X' \neq x$ and $\mathsf{Tag}_{(A,B)}(X') = T'$.

For the X' that E outputs, let $T^* := \mathsf{Tag}_{(A,B)}(X')$ denote its real tag. Because X' is a function of T only, it is clear that the strategy that maximizes $\mathsf{Adv}_E(A, B)$ is to try to compute in step 4 this value T^*, given only T. Thus denoting $\delta := \max_E(\mathsf{Adv}_E(A, B))$, we must have $\log(1/\delta) \geq \mathbf{H}_\infty(T^* \,|\, T)$, and so it suffices to prove $\mathbf{H}_\infty(T^* \,|\, T) \geq k - \lambda - \log d$ as follows.

$$\begin{aligned}
\mathbf{H}_\infty(T^* \,|\, T) &\geq \mathbf{H}_\infty(T^*, T) - \lambda \\
&\geq \mathbf{H}_\infty(T^*, T, \mathsf{Pos}_{(A,B)}(x, X')) - \lambda - \log d \\
&\geq \mathbf{H}_\infty(A, B) - \lambda - \log d \\
&= k - \lambda - \log d.
\end{aligned}$$

The first two inequalities hold by Lemma 6 since the support size of T and $\mathsf{Pos}_{(A,B)}(x, X')$ is 2^λ and d, respectively. The third inequality holds by Lemma 5, because there is a deterministic g such that $g(T^*, T, \mathsf{Pos}_{(A,B)}(x, X')) = (A, B)$; note that g can compute x and X' because the former is fixed and the latter depends only on T. □

References

[1] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In *EUROCRYPT 2004*.

[2] Peter Gemmell and Moni Naor. Codes for interactive authentication. In *CRYPTO 1993*.