CS 7880 Graduate Cryptography September 17, 2015

Lecture 2: MACs, Statistical Distance, Statistical Security

Lecturer: Daniel Wichs *Scribe: Giorgos Zirdelis*

1 Topics Covered

- Better MAC construction
- Secret sharing
- Statistical Distance
- Statistical Security of Encryption

2 Better MAC construction

The MAC construction that we saw in the previous lecture was not very practical in the sense that we can improve the construction by using the same key and tag length as before but for a bigger message space. That is, for some prime p we have:

- $\mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$
- $\mathcal{M} = \mathbb{Z}_p^d$ for some $d \geq 1$
- $\mathcal{T} = \mathbb{Z}_p$

(We can replace $\mathbb{Z}_p$ by any finite field $\mathbb{F}$. In particular, it's useful to use a binary extension field $\mathbb{F}_{2^\ell}$ since in that case keys and messages are just bit string of length ℓ.).

For a key $\mathbf{k} = (x, y)$ and message $\mathbf{m} = (m_1, \ldots, m_d)$ we define

$$\text{MAC}(\mathbf{k}, \mathbf{m}) = \sum_{i=1}^{d} m_i x^i + y$$

where mutliplication and addition are performed over the field $\mathbb{Z}_p$.

Theorem 1 *The above* MAC *has 1-time security with* $\varepsilon = \frac{d}{p}$.

Proof: Let X, Y be two independent random variables denoting uniform samples from $\mathbb{Z}_p$.

Set $\mathbf{K} = (X, Y)$. Then for any $\mathbf{m}$ and t we have that:

$$\begin{aligned} &\Pr[\text{MAC}(\mathbf{K}, \mathbf{m}) = t] = \\ &\Pr\left[\sum_{i=1}^{d} m_i X^i + Y = t\right] = \\ &\Pr\left[Y = t - \sum_{i=1}^{d} m_i X^i\right] = \frac{1}{p}. \end{aligned} \tag{1}$$

The last line follows by considering the probability only over Y, even for a worst-case choice of X.

For any $\mathbf{m}' \neq \mathbf{m}$ and any t', t consider the following probability,

$$\begin{aligned}&\Pr[\mathrm{MAC}(\mathbf{K},\mathbf{m}') = t', \mathrm{MAC}(\mathbf{K},\mathbf{m}) = t] = \\ &\Pr\left[\sum_{i=1}^{d} m'_i X^i + Y = t', \sum_{i=1}^{d} m_i X^i + Y = t\right].\end{aligned} \tag{2}$$

For both events of equation 2 to hold, we obtain the following non-linear system of two polynomial equations on X and Y,

$$\begin{cases}\sum_{i=1}^{d} m'_i X^i + Y = t' \\ \sum_{i=1}^{d} m_i X^i + Y = t\end{cases}.$$

Furthermore, substracting the second equation from the first this is equivalent to the system of equations:

$$\begin{cases}\sum_{i=1}^{d} (m'_i - m_i) X^i + (t' - t) = 0 \\ Y = t - (\sum_{i=1}^{d} m_i X^i)\end{cases}$$

The first equation is a non-zero polynomial of degree d in X over $\mathbb{Z}_p$, hence the probability of X being a root is $\leq \frac{d}{p}$. Conditioned on any fixed value of X, the probability of the second equation is then $1/p$ over the choice of Y. therefore

$$\Pr[\mathrm{MAC}(\mathbf{K},\mathbf{m}') = t', \mathrm{MAC}(\mathbf{K},\mathbf{m}) = t] \leq \frac{d}{p^2}. \tag{3}$$

Combining equations 1 and 3, the probability of Eve succeeding is

$$\begin{aligned}&\Pr[\mathrm{MAC}(\mathbf{K},\mathbf{m}') = t' \mid \mathrm{MAC}(\mathbf{K},\mathbf{m}) = t] = \\ &\frac{\Pr[\mathrm{MAC}(\mathbf{K},\mathbf{m}') = t', \mathrm{MAC}(\mathbf{K},\mathbf{m}) = t]}{\Pr[\mathrm{MAC}(\mathbf{K},\mathbf{m}) = t]} \leq \frac{d}{p^2} p \leq \frac{d}{p}.\end{aligned}$$

Hence the MAC scheme has 1-time security of $\varepsilon = \frac{d}{p}$. □

Example 1 *Consider a message of size 2^{33} bits (4GB). We need two field elements for the key and $d = 2^{26}$ blocks of size $p \in (2^{128}, 2^{129})$ to represent the message and at most 129 bits to describe p. With this input, we get security of $\varepsilon \leq 2^{-102}$ and key size of $129 + 129 = 256$ bits.*

From Example 1 we see that we can authenticate big messages with a very small key compared to message size, but we still need one key per message. As the next theorem states, we cannot do any better if we want statistical security.

Theorem 2 *To authenticate q messages with security $\varepsilon = 2^{-r}$ we need a key of size $(q+1)r$.*

Proof: ommited. □

Intuitively, think of $q+1$ messages where, for each message the adversary gets a tag of the corresponding message. Since each tag should be hard to guess with probability 2^{-r} even conditioned on all the other tags, the tags in total need to have $(q+1)r$ bits of entropy, which can only come from the key.

We can also combine encryption and authentication by sending a tag of the encrypted message, i.e. by sending $c = \mathsf{Enc}(k_1, m)$ and $t = \mathrm{MAC}(k_2, c)$.

3 Secret Sharing

Lets consider the scenario where do not want to keep a secret message as a whole, but we wish to split it across number of n computers such that every computer gets a "share" of the secret. We want the adversary to not be able to recover the message even if he manages to get $n-1$ shares of the message. However, if we have all n shares than the message can be recovered.

Definition 1 For message space $\mathcal{M}$, share space $\mathcal{S}$ and a number of parties n we define a secret sharing scheme to consist of a randomized sharing procedure $Share : \mathcal{M} \to \mathcal{S}^n$ and a recovery procedure $Rec : \mathcal{S}^n \to \mathcal{M}$. Additionally, the following two properties must be satisfied:

- **Perfect Correctness**: For every message $m \in \mathcal{M}$, $\Pr[Rec(Share(m)) = m] = 1$.
- **Perfect Security** Let M be some random variable over $\mathcal{M}$. Let $(S_1, \ldots, S_n) = Share(M)$ be a random variable for the sharing of a message M (the randomness is over both the choice of the message and the randomness of the sharing algorithm). Let $A \subseteq \{1, \ldots, n\}$ be some set of corrupted parties of size $|A| = n-1$ and let S_A denote the set of shares seen by the adversary $S_i : i \in A$. We require that S_A and M are independent random variables, which means that the adversary learns nothing about the message. This property is the analogous of perfect secrecy. (Note that $|\mathcal{M}| \leq |\mathcal{S}|$ because the n-th share has to reveal the message.)

$\diamondsuit$

First, let's assume that $n = 2$. Then we can create a secret sharing scheme based on any perfectly secret encryption scheme: one party gets the secret key and another party gets the ciphertext. Individually, neither party learns anything about the message but together they can recover it completely. In the case of one-time pad, one party gets K and the other gets $M \oplus K$.

Next, we generalize this to any number of parties n. Let $(\mathbb{G}, +)$ be an additive group (e.g., bit-string of length ℓ under the XOR operation). Let $\mathcal{M} = \mathcal{S} = \mathbb{G}$.

- $Share(m)$: Choose $s_1, \ldots, s_{n-1}$ uniformly at random from $\mathbb{G}$.
  Set $s_n = m - (s_1 + s_2 + \cdots s_{n-1})$ and output $(s_1, \ldots, s_n)$.
- $Rec(s_1, \ldots, s_n)$: Output $m = s_1 + \cdots + s_n$.

For any r.v. M and any set $A \subseteq \{1, \ldots, n\}$ of size $|A| = n-1$ and any $m \in \mathcal{M}, \vec{s} \in \mathcal{S}^{n-1}$ we have:

$$\Pr[S_A = \vec{s} | M = m] = 1/|\mathbb{G}|^{n-1}.$$

(since conditioned on $M = m$, the event $S_A = \vec{s}$ occurs for a unique choice of $s_1, \ldots, s_{n-1}$). Therefore this probability is the same for all m and hence S_A, M are independent as desired.

3.1 Threshold Secret Sharing

With a t-out-of-n threshold secret sharing we have the same scenario as before in which we have distributed a message among n parties. But now we also set a threshold t such that any $t+1$ parties can recover the message (correctness) but strictly no less than $t+1$ (security). Essentially, in the previous case we had $t = n-1$.

One solution to this problem is to simply enumerate all subsets of parties of size $t+1$ and use the scheme described above to secret share the message among the $t+1$ parties so that all of them are needed to recover. However, since we would need to do this for every subset of size $t+1$, the efficiency of the sharing procedure and the size of the share would grow with $\binom{t+1}{n}$ which can be exponential in n.

A much better construction for threshold secret sharing is the one by A. Shamir [Sha79] which we describe next. Let n be the number of parties and $t < n$ be a threshold. Also let $\mathcal{M} = \mathbb{Z}_q$ be the message space, and $\mathcal{S} = \mathbb{Z}_q$ be the share space where q is a prime number with $q > n$. (We can replace $\mathbb{Z}_q$ with any finite field $\mathbb{F}$ such that $|\mathbb{F}| > n$.) To share a message $m \in \mathcal{M}$ we follow the next steps:

Share(m)

- Choose t uniformly random coefficients $c_1, \ldots, c_t \leftarrow \mathbb{Z}_q$ and set $c_0 = m$.
- Define polynomial $p(x) = \sum_{j=0}^{t} c_j x^j$
- Distribute $s_i = p(i)$ for $i = 1, \ldots, n$.

To recover a message $m \in \mathcal{M}$ we follow the next steps:

Recover($\{(i, s_i)\}_{i \in Z}$). Given $t+1$ shares s_i of users $i \in Z$ for some $Z \subseteq \{1, \ldots, n\}$, $|Z| = t+1$, we can use Lagrange interpolation to recover the polynomial $p(x)$. This is possible since we are given $t+1$ evaluations of a polynomial of degree t. In particular:

- Let $Z = \{z_0, \ldots, z_t\}$ and define $y_i = s_{z_i}$ so that $y_i = p(z_i)$.
- Then we can recover $p(x)$ as follows:

$$p(x) = \sum_{i=0}^{t} \left(\prod_{j \neq i} \frac{x - z_j}{z_i - z_j} \right) y_i$$

There are two equivalent ways to represent a polynomial, either by specifying its coefficients or by knowing its evaluations. In the this scheme we go back and forth between those two representations. Going from coefficients to evaluations is easy because we just evaluate the polynomial, and for going from evaluations to coefficients we use interpolation.

Theorem 3 *Shamir's secret sharing scheme has perfect secrecy.*

Proof: For any message m and any t distinct points $z_1, \ldots, z_t \subseteq \mathbb{Z}_q \backslash \{0\}$, the probability of $p(z_1) = s_1, \ldots, p(z_t) = s_t$ is $1/q^t$ since all the coefficients, except the constant one, where chosen uniformly at random. Therefore, no matter what message m we choose, an adversary learns nothing about the message m, i.e.

$$\Pr[p(z_1) = s_1, \ldots, p(z_t) = s_t \mid M = m] = \frac{1}{q^t}$$

where M is a random variable on $\mathcal{M}$. So the probability of the adversary guessing the correct message from this scheme is $1/q$, which is equivalent to guessing the message uniform at random without any information from the scheme. □

4 Statistical Distance

As we saw with perfect secrecy and statistical security there are some drawbacks regarding practicality, i.e. we can use a key only once at the length of the key is bigger than the message sometimes.

What if we want to relax the notion perfect secrecy? For example, say for a random key K and for all m_0, m_1 we have $c_0 = \mathsf{Enc}(K, m_0)$ and $c_1 = \mathsf{Enc}(K, m_1)$. Can we know if c_0 and c_1 are similar?

In order to do that, we first must be able to tell how much random variables differ one from another. For that purpose we use the statistical distance of two random variables.

Definition 2 Let X, Y be two random variables that takes values in $\mathcal{V}$ (i.e., $\mathcal{V}$ is the union of supports of X and Y). The statistical distance between X and Y is defined as follows in three equivalent ways:

$$\mathsf{SD}(X,Y) = \max_{f:\mathcal{V}\to\{0,1\}} |\Pr[f(X) = 1] - \Pr[f(Y) = 1]| \tag{1}$$

$$\mathsf{SD}(X,Y) = \max_{\mathcal{W}\subseteq\mathcal{V}} |\Pr[X \in \mathcal{W}] - \Pr[Y \in \mathcal{W}]| \tag{2}$$

$$\mathsf{SD}(X,Y) = \frac{1}{2}\sum_{v\in\mathcal{V}} |\Pr[X = v] - \Pr[Y = v]| \tag{3}$$

♢

Theorem 4 *The above definitions for statistical distance are equivalent.*

Proof:

- $1 \leftrightarrow 2$: We can convert back and forth between a set $\mathcal{W} \subseteq \mathcal{V}$ and an indicator function $f : \mathcal{V} \to \{0,1\}$ by defining $f(v) = 1 \Leftrightarrow v \in \mathcal{W}$
- $3 \leftrightarrow 2$. Let $\mathcal{T} \subseteq \mathcal{V}$ be the set

$$\mathcal{T} = \{v \in \mathcal{V} \ : \ \Pr[X = v] - \Pr[Y = v] \geq 0\}.$$

It's clear that one of $\mathcal{W} = \mathcal{T}$ or $\mathcal{W} = \mathcal{V} \setminus \mathcal{T}$ must maximize $\max_{\mathcal{W} \subseteq \mathcal{V}} |\Pr[X \in \mathcal{W}] - \Pr[Y \in \mathcal{W}]|$ depending on whether the difference is positive or negative. Actually, it turns out that this value is the same in both cases. This is because

$$\Pr[X \in \mathcal{T}] + \Pr[X \in \mathcal{V} \setminus \mathcal{T}] = 1 = \Pr[Y \in \mathcal{T}] + \Pr[Y \in \mathcal{V} \setminus \mathcal{T}]$$

and hence

$$\Pr[X \in \mathcal{T}] - \Pr[Y \in \mathcal{T}] = \Pr[Y \in \mathcal{V} \setminus \mathcal{T}] - \Pr[X \in \mathcal{V} \setminus \mathcal{T}].$$

Therefore $\max_{\mathcal{W} \subseteq \mathcal{V}} |\Pr[X \in \mathcal{W}] - \Pr[Y \in \mathcal{W}]| = \Pr[X \in \mathcal{T}] - \Pr[Y \in \mathcal{T}]$.

This shows:

$$\begin{aligned}
\mathsf{SD}(X,Y) &= \frac{1}{2}\sum_{v \in \mathcal{V}} |\Pr[X = v] - \Pr[Y = v]| \\
&= \frac{1}{2}\left(\sum_{v \in \mathcal{T}} (\Pr[X = v] - \Pr[Y = v]) + \sum_{v \in \mathcal{V} \setminus \mathcal{T}} -(\Pr[X = v] - \Pr[Y = v])\right) \\
&= \frac{1}{2}\left(\Pr[X \in \mathcal{T}] - \Pr[Y \in \mathcal{T}] + \Pr[Y \in \mathcal{V} \setminus \mathcal{T}] - \Pr[X \in \mathcal{V} \setminus \mathcal{T}]\right) \\
&= \frac{1}{2} 2(\Pr[X \in \mathcal{T}] - \Pr[Y \in \mathcal{T}]) \\
&= \max_{\mathcal{W} \subseteq \mathcal{V}} |\Pr[X \in \mathcal{W}] - \Pr[Y \in \mathcal{W}]|
\end{aligned}$$

□

Example 2 *Let X be a uniform random variable over $\{1, \ldots, 2^{100}\}$ and Y be a uniform random variable over $\{1, \ldots, 2^{100} - 1\}$. The support of those two random variables differ in exactly one element, 2^{100}. Their statistical distance is $\frac{1}{2^{100}}$ as the probability of X taking that value is $\frac{1}{2^{100}}$ and for Y is 0.*

Definition 3 An encryption scheme Enc has ε-statistical secrecy if for any $m_0, m_1 \in M$ and key K, $\mathsf{SD}(\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m_1)) \leq \varepsilon$. ♢

Theorem 5 *If an encryption scheme has ε-statistical secrecy then $\varepsilon \geq 1 - \frac{|\mathcal{K}|}{|\mathcal{M}|}$, where $|\mathcal{K}|$ and $|\mathcal{M}|$ are the key and message space, respectively.*

Bibliography

[Sha79] Adi Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, November 1979.