Randomness in Cryptography April 25, 2013

Lecture 15: Privacy Amplification against Active Attackers

Lecturer: Yevgeniy Dodis *Scribe: Travis Mayberry*

1 Last Time

Previously we showed that we could construct a *robust extractor* which allowed us to extract using an authenticated seed. That is, we could guarantee that a seed had not been changed using only the randomness in our imperfect source. Specifically, our robust extractor could extract m bits of uniform randomness from an n-bit source, X, with min-entropy k, while ensuring that the seed S had not been tampered with since its original "authentication". We defined P to be the "helper information", which included the seed and the authentication of the seed. We were able to achieve robust extraction for:

(a) $k > \frac{n}{2}$

(b) $m << k$ $(m = \Theta(k - \frac{n}{2}))$

We were also able to show matching lower bounds for robust extractors. Particularly, we showed that the $k \geq \frac{n}{2}$ requirement is inherent. This is unfortunate, especially considering other settings in which we can extract from even very low entropy sources.

2 Beating 1/2 entropy rate

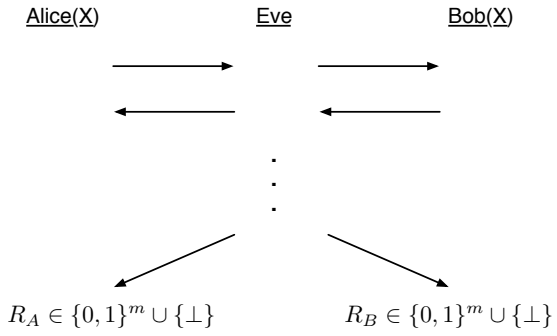
Option 1: Computation Assumptions

In the random oracle model, it is simple to construct robust extractors for any k, using $P = (S, \mathsf{Hash}(X, S))$. The extractor then verifies the hash and extracts using S.

There have also recently been standard model constructions [2][1], but they are suboptimal.

Option 2: Interactive Setting

We will concentrate on the interactive setting, which will allow us to construct more efficient robust extracts as well as some other interesting primitives like secure key agreement. Our interactive protocols will look like this:



Alice and Bob will communication for one or more rounds, through a channel controlled by the adversary Eve. In the end, they should each obtain a uniform secret R, or $\perp$ if Eve has actively modified the messages. We call this protocol *privacy amplification with active attacker*.

DEFINITION 1

An interactive protocol (P_B, P_B), executed by Alice and Bob on a communication channel fully controlled by an active adversary EVE, is a (k, m, ε)-*privacy amplification protocol* if it satisfies the following properties whenever $H_{\inf}(X) \geq k$:

1. **Correctness**: If Eve is passive, then $Pr[R_A = R_B \wedge R_A \neq \perp \wedge R_B \neq \perp] = 1$ (Alice and Bob will obtain the same key).

2. **Pre-application Robustness**: Even f Eve is active, $\Pr[R_A \neq R_B \wedge R_A \neq \perp \wedge R_B \neq \perp] \leq \varepsilon$ (Eve cannot cause Alice and Bob to have different keys without them aborting the protocol).

3. **Post-application Robustness**: If Eve is additionally given the key R_A the moment she completes the left execution (P_A, P_E), and the key R_B the moment she completes the right execution (P_E, P_B), the protocol is still robust. That is, even with oracle access to Alice and Bob, the protocol is still secure. For example, if Eve completed the left execution before the right execution, she may try to use R_A to force Bob to output a different key $R_B \notin \{R_A, \perp\}$, and vice versa.

4. **Extraction**: Given a string $r \in \{0,1\}^m \cup \{\perp\}$, let $\mathsf{purify}(r)$ be $\perp$ if $r = \perp$, and and a fresh m-bit random string otherwise. Letting E' denote Eve's view of the protocol, we require that

   $$\delta((R_A, E'), (\mathsf{purify}(R_A), E')) \leq \varepsilon \text{ and } \delta((R_B, E'), (\mathsf{purify}(R_B), E')) \leq \mathit{epsilon}$$

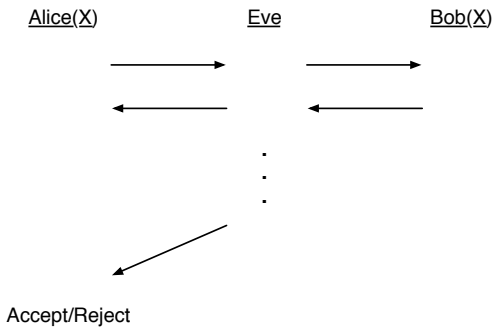
   Namely, whenever a party does not reject, its key looks like a fresh random string to Eve.

$\diamondsuit$

Our hope is to get $m \approx k$ in a small number of rounds. We know that the number of rounds has to be at least two when $k < \frac{n}{2}$, but we have yet to see if we can actually do it in two. By contrast, if Eve is passive we can do it in one round simply with a good extractor $(A \xleftarrow{S} B, R = \mathsf{Ext}(X; S))$.

2.1 Liveness test

First, we will consider a simplified version of this protocol where Bob simply proves to Alice that she is actually communicating with him. We will call this a *liveness test*.



A liveness test protocol is *correct* if $PR[A(X) \leftrightarrow B(X,S) \text{ accepts}] = 1$. That is, if Bob is actually communicating with Alice then Alice should output accept every time.

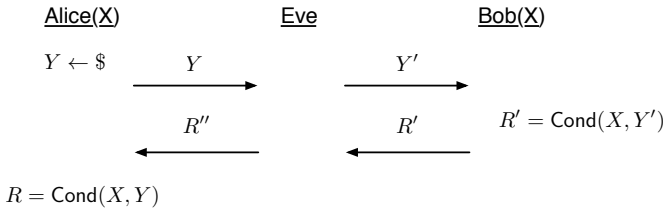
A liveness test is (k,ε)-secure if $\forall \varepsilon, Pr[A(X) \leftrightarrow Eve \text{ accepts}] \leq \varepsilon$. In other words, Eve should only be able to impersonate Bob with probability less than ε.

Trivially, we could have Bob send his sample x' to Alice and have her compare with her value x. Since Eve does not have the source X, she could only win with probability 2^{-k} by guessing the most likely value for x. However, after this protocol there is no entropy remaining in X, since Eve has seen everything. This means that, using this protocol, we could only attest to Bob's liveness one time; additionally, we cannot further use X for extraction or any other protocol which relies on some entropy, so it is not composable.

Knowing this, we can try to achieve a better protocol which tests liveness while retaining maximum *residual entropy*. If a protocol achieves $H_{\inf}(X|\text{ transcript}) \geq k - \alpha$, we would like to minimize α while still achieving ε security.

From our results last week, we can show that any one-round protocol will have poor parameters, namely $\alpha \geq n - k + \log \frac{1}{\varepsilon}$. Particularly, if $k < \frac{n}{2}$, then we might as well send x because we cannot possibly have any leftover entropy.

We can, however, use a two-round challenge-response protocol to get better parameters. Making use of a min-entropy condenser $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^{seed} \to \{0,1\}^l$, consider the following:



Alice will accept if $R = R''$ and reject otherwise.

Lemma 1 *Cond is a (k, k', ε')-condenser $\Rightarrow$ (k,ε)-secure liveness test where $\varepsilon \leq \varepsilon' + 2^{-k'}$.*

Eve is able to win if she can beat the condenser (probability ε') or if she can guess the output of the condenser for some seed $Y' \neq Y$. Therefore, the total error of our liveness test is $\varepsilon = \varepsilon' + 2^{-k'}$. For our condenser then, it suffices to set $\varepsilon' = \frac{\varepsilon}{2}$ and $k' = \log \frac{1}{\varepsilon} + 1$

(which can be much less than k). Since we can construct a condenser even for very small values of k', this allows us to beat the previous $k \geq \frac{n}{2}$ bound.

Theorem 1 $\forall \varepsilon, \forall k \geq (2\log\frac{1}{\varepsilon}+1), \exists$ *an efficient 2-round* (k,ε)*-liveness test with* $\alpha = O(\log\frac{1}{\varepsilon}+\log n)$

Proof: It suffices that $H_2(\mathsf{Cond}(X;Y)|Y) \geq 2\log\frac{1}{\varepsilon}$. For our condenser, we can use a δ-universal hash h (with l bit output), which has the property $\forall y, x' \neq x : Pr_y(h_y(x) = h_y(x')) \leq \delta$. We then have:

$$\begin{aligned}\mathsf{Col}(h_y(X)|y) &\leq \mathsf{Col}(X) + \delta \\ &\leq 2^{-k} + \delta\end{aligned}$$

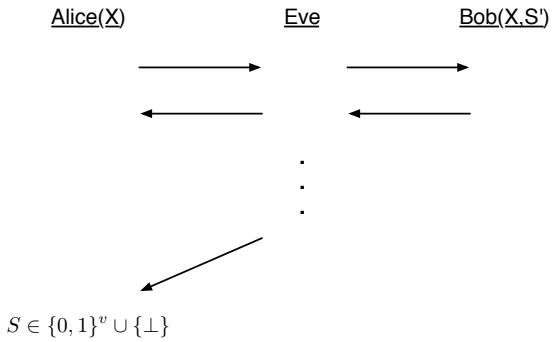
If $k \geq 2log\frac{1}{\varepsilon}+1$ and $\delta \leq \frac{\varepsilon^2}{2}$, then $\delta + 2^{-k} \leq \varepsilon$ and our condition is met. Recall from lecture 1 that we can construct a δ-universal hash such that $l, |y| = O(\log\frac{1}{\delta}+\log n) = O(\log\frac{1}{\varepsilon}+\log n)$.

□

Extension: What if Y is not uniform, but an (N,c)-source (independent of X)? Trivially, we can do it with $\varepsilon' = \varepsilon * 2^{N-c}$. However, we can also do it with a 2-source strong extractor, having the property $(Y, 2\mathsf{Ext}(X;Y)) \overset{\approx}{\varepsilon} (Y, U_l)$, $\forall (N,c)-$source Y and $(n,k)-$source X. Such extractors are known for $k, c = \Omega(\log n + \log N + \log\frac{1}{\varepsilon}$. The best known 2-source extractor, due to Raz[5], requires that $\frac{c}{N} \geq \frac{1}{2}$ (i.e. the entropy rate of the seed must be at least one half).

2.2 Interactive MAC

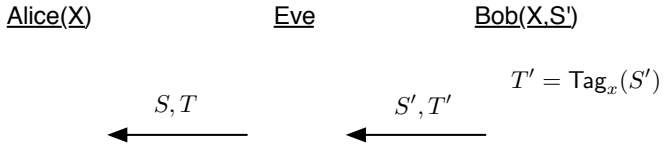
We can also consider how the interactive setting applies to Message Authentication Codes. Such a protocol would look like this:



Alice outputs the message S, if it has been unmodified, or $\perp$ otherwise. An interesting attacker to consider in this setting is one which has oracle access to $A(X)$ and $B(X,S')$ (where S' is the input to Bob and S is the message output by Alice). For such scheme to be

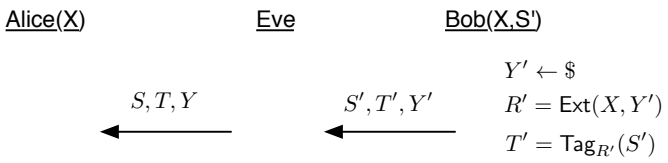
correct, if Eve is passive, $\forall S', Pr[S = S'] = 1$. We say that an interactive MAC has (k, δ)-security if $Pr[S \notin \{S', \perp\}] \leq \delta$. It is easy to see that a liveness test can be implemented with an interactive MAC, so we also know that for $k \leq \frac{n}{2}$ we must have at least two rounds.

Idea 0: We have said that one round cannot work, but let's start with a simple one-round idea to set the stage:



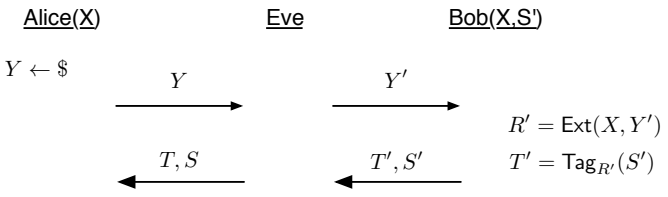
Alice would output S if the tag matches. This cannot work because we know that we can't build MACs with $k < \frac{n}{2}$.

Idea 1: Extract first on X and then apply Tag.



Again, Alice can compute the tag and output S if it matches. In this case, however, Eve can change the seed Y to whatever she wishes. Our extractor is only guaranteed to work if Y is uniform, and Even can cause it to have no entropy at all. She could, for instance, provide a degenerate seed which causes the extractor to always output a fixed string (regardless of X). She would then be able to predict R and forge any tag she wishes.

Idea2: Let Alice choose Y.



Alice then calculates $R = \mathsf{Ext}(X; Y)$ and $T = \mathsf{Tag}_R(S)$ and compares $T \stackrel{?}{=} T'$. On the surface, this seems like a much better idea. It is the first protocol which might actually work, because it is doesn't violate the impossibility result we already had (at least two rounds). Alice knows that R must be good (unknown to Eve) because she chooses Y herself. Additionally, Eve must change Y' before actually seeing the tag. However, Y' can still be changed by Eve and could case R' to be weak. This might cause the tag to leak some information about X. The security of our MAC relies on the fact that $R = R'$, and Eve can correlate these two variables by changing Y'.

Solution: Make Ext and/or Tag non-malleable.

1. Extractor non-malleable and tag normal: will work.
2. Extractor normal and tag non-malleable: impossible to have a fully non-malleable MAC information theoretically
3. Extractor and tag both partially non-malleable

Option 1: We have already constructed nm-Ext.

DEFINITION 2 $\mathsf{nm}-\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^{seed} \to \{0,1\}^l$ is a (k,ε)-nm-Ext if $\forall (m,k)$-source X, $\forall A$ s.t. $Y' \stackrel{\triangle}{=}$ is not equal to Y, $\forall Y$ $(Y, \mathsf{nm}-\mathsf{Ext}(X,Y'), \mathsf{nm}-\mathsf{Ext}(X,Y)) \stackrel{\approx}{\varepsilon}$ $(Y, \mathsf{nm}-\mathsf{Ext}(X,Y'), U_l)$. ♢

Known results for non-malleable extractors:

1. **Existential:** Due to DW09[3]. Works for $k = \Omega(\log n + \log \frac{1}{\varepsilon})$ and $\ell = \frac{k}{2} \cdot O(\log n + \log \frac{1}{k})$.
2. **Constructive:** Previously we constructed a non-malleable extractor (using a four-wise independent hash function with the "double run" trick) with $\ell = \frac{n}{4} \forall k \geq \frac{n}{2} + 2\log \frac{1}{\varepsilon} + 1$.
3. **Best Known:** k slightly less than $\frac{n}{2}$ (i.e. entropy rate $\approx .49999$).

Lemma 2 *If nmExt is $(k, \frac{\delta}{2})$-secure with output $\ell \geq O(\log v + \log \frac{1}{\delta})$, then (k,δ)-secure 2-round MAC for v-bit messages exists.*

Proof:

There are two possibilities: Eve either leaves Y unmodified and $Y = Y'$, or she changes it and $Y \neq Y'$. If $Y = Y'$, then $R = R'$ and we can rely on standard tag security. If $Y \neq Y'$, then R' is uncorrelated with R (by the non-malleable property of our extractor) and T is uncorrelated with T'.

□

Corollary 3 $\exists$ *(inefficient) 2-round (k,δ) MAC for v-bit messages, $\forall \delta < \frac{1}{n}$, $\forall k = \Theta(\log \frac{1}{\delta})$ where $\alpha = O(\log \frac{1}{\delta} + \log v)$.*

Additionally, we can have efficient interative MACs if $k > \frac{n}{2}$. We can have $H_{\inf}(X|$ transcript$) = k - O(\log v + \log \frac{1}{\delta})$, making $\alpha = O(\log v + \log \frac{1}{\delta})$. This is much better than our one round solution which worked for $k > \frac{n}{2}$, so an additional round not only allows us to circumvent an impossibility result but improve communication (and residual entropy) in cases which were already possible.

Li [4] gives evidence that achieving $k << \frac{n}{2}$ is hard, and related to 2-source extractors.

3 Look-ahead extractors/MACs

We have seen that non-malleable extractors allow is to do privacy amplification, but perhaps we can have a weaker notion of non-malleability which also achieves our goals but which is easier to construct. We define this "look-ahead" notion as follows:

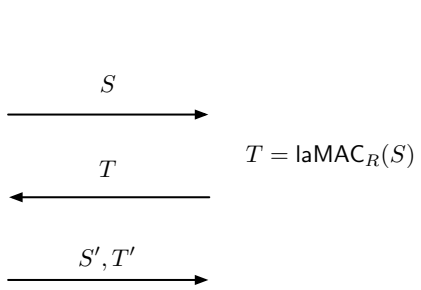
DEFINITION 3 $R = (R_1, ..., R_t), R' = (R'_1, ..., R'_t)$ are ε-look-ahead given Y if $\forall i = i...t$, $(R'_1, ..., R'_{i-1}, R_i, ..., R_t) \stackrel{\approx}{\varepsilon} (R'_1, ..., R'_{i-1}, U_i, ..., U_t)$. ♢

This means that $\mathsf{SD}([R]_{i+1}^t, [U]_{i+1}^t | [R']_1^{i-1}, Y) \leq \varepsilon$. Additionally, if this holds for all i, it means that $R, R' \approx U$.

Now, based on that notion, we can define look-ahead extractors.

DEFINITION 4 $\mathsf{laExt} : \{0,1\}^N \times \{0,1\}^{seed} \to (\{0,1\}^\ell)^t$ is a (k, ε)-look-ahead extractor if $\forall A(Y) \to Y'$, $R = \mathsf{laExt}(X; Y)$ and $R' = \mathsf{laExt}(X, Y')$ are ε-look-ahead. ♢

Similarly, we can define a look-ahead MAC with the following game:



A wins if $S' \neq S$ and $T' = \mathsf{laMAC}'_R(S')$.

DEFINITION 5 $\mathsf{laMAC} : (\{0,1\}^\ell)^t \times \{0,1\}^v \to \{0,1,\}^\lambda$ is an (ε, δ)-look-ahead MAC of v-bit messages S if $\forall R, R', Y$ s.t. R, R' are ε-LA given Y, $\forall A$, $Adv(A) \leq \delta$. ♢

3.1 Look-ahead MAC construction

We can start by construction laMAC for 1-bit messages, with $t = 4$.

$$\mathsf{laMAC}_{R_1,R_2,R_3,R_4}(b) = \left\{ \begin{array}{cccc|l} R_1 & & & R_4 & \text{if } b = 0 \\ & R_2 & R_3 & & \text{if } b = 1 \end{array} \right.$$

Now, if A wants to change the message from 1 to 0, she knows R_1, R_2 and R_3, but must guess R_4. This is hard because of our look-ahead property (the R values give no advantage to guessing the ending R' values). Changing 0 to 1, she knows R_1 (which is useless because of our look-ahead property) and R_4, but must guess R_2 and R_3 (2ℓ bits). R_4 is only ℓ bits so it cannot "give away" both R_2 and R_3 which are 2ℓ bits combined. She must guess at least ℓ bits in this case as well.

The probability she can correctly guess these is less than $\varepsilon + \frac{2\ell}{2^{2\ell}} - \varepsilon + 2^{-\ell}$. If we set $\ell = \log \frac{1}{\delta} + 1$ then we get our desired security.

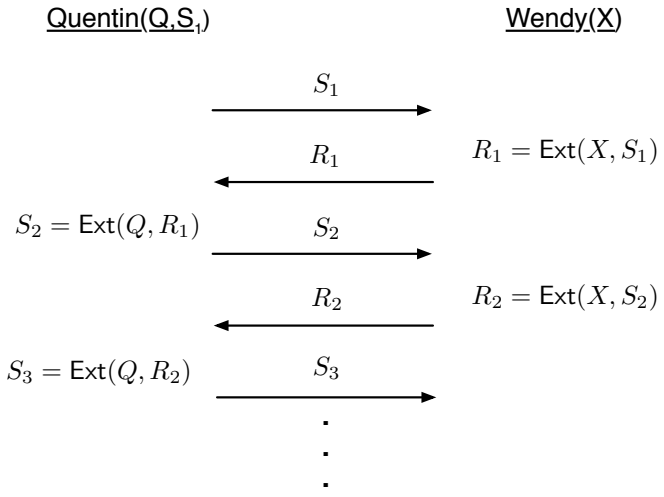
Corollary 4 *If we MAC individual bits of a v-bit message this way, we can also achieve a multi-bit MAC for v-bit messages with $\ell = \log 1\delta + 1$ and $t = 4v$. This way, the length of the key and the length of the tag are both $O(v \log \frac{1}{\delta})$.*

If $v = O(\log \frac{1}{\delta})$ (like in our privacy amplification), then this also means that the key and tag are $O(\log^2 \frac{1}{\delta})$.

Conjecture 2 *This quadratic entropy loss using laMAC is optimal.*

3.2 Look-ahead extractor construction

A look-ahead extractor can be created by playing a "game" as follows between Quentin and Wendy:



No matter what strategy the two players employ, the values $[R_{i+1}, ..., R_t]$ look random to Quentin after he has only seen $[R_1, ..., R_i]$. Because X is "secret" from him, he cannot predict what the output of the next extractor will be, even though he knows all the seeds he is sending. This is exactly the property of a look-ahead extractor, so this game can be played in Alice or Bob's head to create laExt, with the seed $Y = (Q, S_1)$.

Lemma 5 *laMAC and laExt $\Rightarrow$ 2-round (k, ε)-secure MAC for v − bit messages.*

An interactive MAC using the above laMAC and laExt constructions will have $\alpha = O(\log^2(n) + \log^2 \frac{1}{\delta})$.

References

[1] Mark Braverman, Avinatan Hassidim, Yael Tauman Kalai: Leaky Pseudo-Entropy Functions. ICS 2011: 353-366

[2] Yevgeniy Dodis, Yael Tauman Kalai, Shachar Lovett: On cryptography with auxiliary input. STOC 2009: 621-630

[3] Yevgeniy Dodis, Daniel Wichs: Non-malleable extractors and symmetric key cryptography from weak secrets. STOC 2009: 601-610

[4] Xin Li: Non-malleable Extractors, Two-Source Extractors and Privacy Amplification. FOCS 2012: 688-697

[5] Ran Raz: Extractors with weak random seeds. STOC 2005: 11-20