Randomness in Cryptography January 21, 2013

Lecture 7: Privacy ⇒ Extraction Continued

Lecturer: Yevgeniy Dodis *Scribe: Travis Mayberry*

We have seen that secure encryption, using a source S, implies extraction on that source. This week we will try to extend that result to other privacy primitives.

1 Last Time

Previously, we showed that encrypting b bits with an n bit key implies extracting $b - \log n$ bits. To prove this, we first constructed a special extractor Ext' and defined $\mathsf{Ext}(k) = \mathsf{Ext}'(\mathsf{Enc}_k(0))$. Showing this construction adheres to the properties of an extractor required the following steps:

1. **Security**: By the security property of Enc, we switched $\mathsf{Enc}_k(0)$ with $\mathsf{Enc}_k(U_b)$, since the statistical distance between them is less than ε.
2. **Correctness**: Since all messages must have a unique encryption, $\forall k : \mathbf{H}_\infty(\mathsf{Enc}_k(U_b)) \geq b$. This shows us that a ciphertext possesses enough min-entropy for extraction.
3. For all b-sources of size $\leq N = 2^n$, $\exists\varepsilon$-secure Ext with output $\ell \approx b - 2\log\frac{1}{\varepsilon} - \log n$.

These steps will be useful when we consider other privacy primitives. We can use this proof as a blueprint with other primitives, as long as they have similar properties we can use in the security and correctness steps.

The main question we will answer today is: to what extent can we generalize this result from encryption to other privacy primitives? We will also explore the limits of this proof and show that it is possible to have encryption on a small number of bits without implying extraction.

2 Decryption Error

In some settings, allowing for a decryption error γ can circumvent some impossibility results. We will show that this is not true for our extraction setting.

DEFINITION 1 $(\mathsf{Enc}, \mathsf{Dec})$ has (S, γ)-decryption-error if $\forall k \in S, \forall m \Pr_{k \leftarrow K}[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) \neq m] \leq \gamma$. ♢

That is, we allow a γ probability that a particular pair, message m and key k, will not be decrypted correctly.

Lemma 1 $(\mathsf{Enc}, \mathsf{Dec})$ *with* (S, ε) *security and* (S, γ) *decryption error* $\Rightarrow \exists(\mathsf{Enc}', \mathsf{Dec}')$ *which has* $(S, \varepsilon + 2\gamma)$ *security and* $(S, 0)$ *decryption error.*

Proof: Let $Z = (k, m) = Dec(k, \mathsf{Enc}_k(m)) \neq m$, that is, all the key-message pairs which do not decrypt correctly. Define Enc' as:

$$\mathsf{Enc}'_k(m) = \begin{cases} 0||\mathsf{Enc}_k(m) & \text{if}(k,m) \in Z \\ 1||m & \text{if } (k,m) \notin Z \end{cases}$$

We know $\forall m, \forall k \in S : \Pr_{k \leftarrow K}[(k, m) \in Z] \leq \gamma$. Therefore,

$$\begin{aligned} \forall m' : \mathsf{SD}(\mathsf{Enc}'_k(m'), Enc'_k(m)) &\leq \Pr[(k,m) \in Z] \\ &+ \Pr[(k,m') \in Z] \\ &+ \mathsf{SD}(\mathsf{Enc}_k(m), \mathsf{Enc}(m')) \\ &\leq 2\gamma + \varepsilon \end{aligned}$$

□

By changing all cases where decryption error would occur into cases where no error occurs, but security fails, we can transform encryption with decryption error into a less secure encryption with none. Now, we can apply our main theorem to Enc' and construct an extractor. If ε and γ are negligible in the security parameter, then their sum is also negligible and we obtain the same impossibility results. Also note, this rules out public key encryption as well, as long as K is the local randomness used for key generation and encryption.

3 Commitment

A commitment scheme is a function $\mathsf{Com}_k(m) = C$. The first part of our theorem for Enc works for Com as well if we substitute the secrecy property of commitment for the security property of encryption. With commitment schemes we call the error δ, so we lose a factor of δ in this step.

For the second step of the proof, we must show that commitments have enough min-entropy. What we will actually do is show that the statistical distance between the output of commitment and another distribution with enough min-entropy is very small We can use the "weak binding" property of commitments to do this. Recall, a commitment scheme is (S, τ)-weakly-binding if $\forall K \leftarrow S : Pr_{k \leftarrow K, (m1,m2) \leftarrow U_b}[\mathsf{Com}_k(m_1) = \mathsf{Com}_k(m_2) \& m_1 \neq m_2] \leq \tau$. That is, the probability that two messages which are not equal will commit to the same value is less than τ. This also implies that $Pr_{k \leftarrow K, (m1,m2) \leftarrow U_b}[\mathsf{Com}_k(m_1) = \mathsf{Com}_k(m_2)] \leq \tau + 2^{-b}$, because two randomly chosen messages will be equal only with probability 2^{-b}. Putting this in terms of collision entropy, we can say $\mathsf{Col}(\mathsf{Com}_k(U_b)|k) \leq \tau + 2^{-b}$. Define $\tau' = \tau + 2^{-b}$.

Lemma 2 $\mathsf{Col}(C|K) \leq \tau' \Rightarrow \forall \varepsilon > 0, Pr_{k \leftarrow K}[\mathsf{Col}(C|K = k) \geq \frac{\tau'}{2}] \leq \varepsilon$

Proof: $\tau' \geq \mathsf{Col}(C|K) \geq \Pr_{k \leftarrow K}[\mathsf{Col}(C|K = k) > \frac{\tau'}{\varepsilon}] \cdot \frac{\tau'}{\varepsilon}$ □

Take X to be $C|K$.

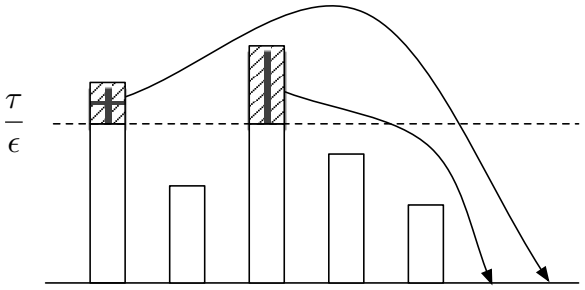

Figure 1: Truncate probabilities greater than $\frac{\tau}{\varepsilon}$ and move to new values.

Lemma 3 $\mathsf{Col}(X) \leq \tau \Rightarrow \forall \varepsilon > 0, \exists X'$ *such that* $\mathsf{Pred}(X') \leq \frac{\tau}{\varepsilon}$ *and* $\mathsf{SD}(X, X') \leq \varepsilon$. *Equivalently,* $\exists X'$ *such that* $\mathsf{SD}(X, X') \leq \varepsilon$ *and* $\mathbf{H}_\infty(X') \geq \mathbf{H}_2(X) - \log \frac{1}{\varepsilon}$.

Proof: Create a new distribution X' where all probabilities greater than $\frac{\tau}{\varepsilon}$ are truncated to $\frac{\tau}{\varepsilon}$ and the extra probability is redistributed to new events. Such a distribution has $\mathsf{Pred}(X') \leq \frac{\tau}{\varepsilon}$ because no event happens with probability greater than $\frac{\tau}{\varepsilon}$.

By Markov's inequality, the fraction of events that must be truncated is equal to $\Pr_{x \leftarrow X}[Pr[X = x] > \frac{\tau}{\varepsilon}] \leq \varepsilon$. Additionally, this means that $\mathsf{SD}(X, X') \leq \varepsilon$. □

Corollary 4 $\forall \varepsilon > 0, \exists \{C'_k : k \in \{0,1\}^n\}$ *such that:*

(a) $\mathsf{SD}(C, C'_k|K) \leq 2\varepsilon$

(b) $\mathsf{Pred}(C'_k) \leq \mathsf{Col}(C|K)/\varepsilon^2$

Theorem 1 *If* Enc *is* (S, ε)*-hiding and* (S, ε)*-weakly-binding,* S *is* $(\ell, \delta + 3\varepsilon)$*-extractable where* $\ell = \log \frac{1}{\tau + 2^{-b}} - 4 \log \frac{1}{\varepsilon} - \log n - O(1)$

Proof: First, we will define two sets of keys: "good" and "bad". All good keys k will satisfy $\mathsf{Col}(C|K = k) \leq \frac{\tau'}{\varepsilon}$, while bad keys will not. By lemma 2, the probability that a key will be bad is at most ε.

$$\begin{aligned}
\mathsf{SD}(\mathsf{Ext}(\mathsf{Com}(K, 0)), U) &\leq \delta + \mathsf{SD}(\mathsf{Ext}(\mathsf{Com}(K, U_b)), U) \\
&\leq \delta + \Pr[\text{K is bad}] + \mathsf{SD}(\mathsf{Ext}(\mathsf{Com}(K, U_b|\text{K is good}))) \\
&\leq \delta + \varepsilon + \mathsf{SD}(\mathsf{Com}(K, U_b), C'_K|\text{K is good}) + \mathsf{SD}(\mathsf{Ext}(C'_K), U_\ell|\text{K is good}) \\
&\leq \delta + 2\varepsilon + \varepsilon
\end{aligned}$$

This holds as long as $\ell \leq \log \frac{1}{\tau'} - 4 \log \frac{1}{\varepsilon} - \log n - O(1)$. □

If δ and τ are both negligible in the security parameter, and $b = \omega(\log \lambda)$ (where λ is the security parameter) we can choose $\varepsilon' = \delta + 3\varepsilon$ which is still negligible and $\ell = \Omega(\log \frac{1}{\tau}) = \omega(\log \lambda)$.

4 Secret Sharing

Recall that secret sharing is the problem of dividing a secret into two or more shares, where the original value can only be recovered by combining all of those shares back together. We will consider just two-out-of-two secret sharing, where the secret is split into two shares.

In order to use our proof that $\mathsf{Enc} \Rightarrow \mathsf{Ext}$, we need something that plays the role of $\mathsf{Enc}_k(0)$.

4.0.1 Option 1: One share

Use $\mathsf{Share}_1(k, m)$. This works for our security step since we can go from $\mathsf{Share}_1(k, 0)$ to $\mathsf{Share}_1(k, U_b)$, but it does not satisfy our correctness step. There is no guarantee that $\mathbf{H}_\infty(\mathsf{Share}_1(k, U_b)|K) \geq \frac{b}{2}$ as all the entropy could be in one share, i.e. $\mathsf{Share}_1 = k, \mathsf{Share}_2 = k + m$.

4.0.2 Option 2: Both shares

Use $(\mathsf{Share}_1(k, m), \mathsf{Share}_2(k_m))$. This works for correctness, but fails security because, having both shares, there is none.

4.0.3 Option 3: Dynamically choose

$$\text{Choose } \mathsf{Enc}_k(m) = \begin{cases} \mathsf{Share}_1(k, m) & \text{if } P(k) = 1 \\ \mathsf{Share}_2(k, m) \end{cases}$$

Question 1 *Can we use option 3 to prove that* $\mathsf{Share} \Rightarrow \mathsf{Ext}$*?*

5 Limits of Extraction

Our general proof that $\mathsf{Enc} \Rightarrow \mathsf{Ext}$ has a loss of $\log n$ bits in the extractor. Does this mean that we cannot extract with encryption less than $\log n$ bits? We will show that, in fact, encryption of less than $\log n - \log\log n$ bits does not imply extraction of even one nearly unbiased bit.

We begin by showing that one-bit encryption does not imply a one-bit extractor. We will do this by proving that, given an extractor, there exists a source which allows for secure encryption but which causes that extractor to be almost completely biased.

First, define this source $S = \{K\}$ on $\{0,1\}^n$ and one-bit encryption $(\mathsf{Enc}, \mathsf{Dec})$ such that:

(a) Enc is perfectly secure i.e. $\mathsf{Enc}(K, 1) \equiv \mathsf{Enc}(K, 0)$

(b) No $(S, 1 - \varepsilon)$-bit extractor for small $\varepsilon \approx 2^{-n/2}$

Equivalently, $\exists(\mathsf{Enc}, \mathsf{Dec})$ s.t. $\mathsf{Good}(\mathsf{Enc})$ is not $(1 - \varepsilon)$-extractable, where $\mathit{Good}(\mathsf{Enc}) = \{K : \mathsf{Enc}(K, 0) \equiv \mathsf{Enc}(K, 1)\}$.

For our proof, it will be useful to envision this encryption as a graph. Given $(\mathsf{Enc}, \mathsf{Dec})$, define $G = (V, E)$ where V is the set of ciphertexts ($|V| = S$) and E is the set of keys

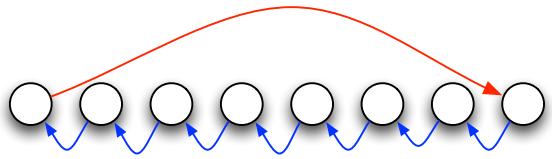

Figure 2: Cycle includes at most one red edge.

($|E| = 2^n = N$). $\forall k \in E$ define $e_k = [\mathsf{Enc}_k(0) \to \mathsf{Enc}_k(1)]$. This way, the edges of the graph describe all possible encryptions under all keys. Our only constraint on this graph is that it must have no self-loops (otherwise decryption would be impossible).

Having defined our encryption as a graph, what graph should we use for our proof? The first thing to note is that multiple edges from one node to another are redundant, they can always be merged and have their probabilities added. More edges are better because it enlarges our set of encryptions $\mathsf{Good}(\mathsf{Enc})$. Therefore, the best graph to use is the complete, directed graph on $|V|$ vertices.

With this graph, we have $V = \{1, ..., S\}$, $E = \{(c_1, c_2) | \forall c_1 \neq c_2\}$ and $S = \sqrt{N} = 2^{n/2}$. We call this graph $\mathsf{Comp}(S)$. Now, we will show that $\mathsf{Good}(\mathsf{Comp}(S))$ is not $(1, 1-\varepsilon)$-extractable for small ε.

Theorem 2 $\mathsf{Good}(\mathsf{Comp}(S))$ *is not* $(1, 1 - \frac{2}{S})$*-extractable.*

Hence, there exists S such that S is $(1, 0)$-encryptable but not $(1, 1 - \frac{2}{2^{n/2}})$-extractable.

Proof: Take any extractor $E \to \{0, 1\}$. This is equivalent to a two-coloring of $\mathsf{Comp}(S)$, because the extractor must take a ciphertext and output a bit (color).

Note that, for all cycles $C = (c_1 \to c_2 \to ... \to c_t \to c_1)$, $k = U_c \in \mathsf{Good}(G)$, because $\mathsf{Enc}(U_c, 0) \equiv \mathsf{Enc}(U_c, 1) \equiv U_{\{c_1,...,c_t\}}$.

Define $\mathsf{Cycles}(G) = \{U_c | c - cycle\} \subseteq \mathsf{Good}(G)$. It suffices now to show that $\mathsf{Cycles}(G)$ is not $(1, 1 - \frac{2}{S})$-extractable.

Assuming the extractor colors edges red and blue (for zero and one bits respectively). There are two cases which could exists:

(a) There exists a cycle which is completely red.

(b) There does not exists a cycle which is completely red.

If a red cycle exists, then we are done because, using this encryption, the extractor will always choose zero and be completely biased. If a red cycle does not exist, then the subgraph G' of red edges acyclic. This means that it can be topologically sorted, so that lining up vertices from left to right all edges will go in one direction only. Consequently, there exists an ordering of vertices $c_1, ..., c_S$ such that $\forall i > j : \mathsf{Ext}((c_i, c_j)) = 1$. This follows directly from above because our graph is complete (has edges between all vertices) and all of the red edges go left to right. Therefore, any edge going right to left must be blue. Knowing this, we can select the cycle which goes from c_S to c_1, visiting every node in between, and then going back to c_S. All edges going leftward will be blue. There is single edge going rightward, (c_1, c_S), which may be blue or red. If it is blue, then we have a blue cycle and

the extractor can be always forced to output one. If it is red, then the extractor outputs one with probability $1 - \frac{1}{S}$. □

Question 2 *Some distributions in S had $\mathbf{H}_\infty(1)$. Can we make $S \subseteq \mathsf{Weak}(k)$? Yes!*

Theorem 3 *[1] $\forall \varepsilon \geq 2^{-n/2+1}, \exists S \subseteq sfWeak_n(n - \log \frac{1}{\varepsilon} - O(1))$ such that S is $(1,0)$-encryptable but not $(1, \frac{1}{2} - \varepsilon)$-extractable.*

Question 3 *Can we really show $\log n$ loss is necessary? Yes*

Theorem 4 *[2] $\forall b \leq \log n - \log\log n 02, \exists S$ which is $(b,0)$-encryptable but not $(1, \frac{1}{2} - \varepsilon)$-extractable, where $\varepsilon = 2^{2b - \frac{n}{2^b}+1} \leq \frac{1}{16n^2}$.*

Question 4 (Open) *We have shown $b \leftrightarrow 1$ separation for $b \approx \log n$. Can we show $b \leftrightarrow b - \log n$ separation?*

References

[1] Yevgeniy Dodis, and Joel Spencer. "On the (non) universality of the one-time pad." *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on.* IEEE, 2002.

[2] Carl Bosley, and Yevgeniy Dodis. "Does privacy require true randomness?." *Theory of Cryptography* (2007): 1-20.