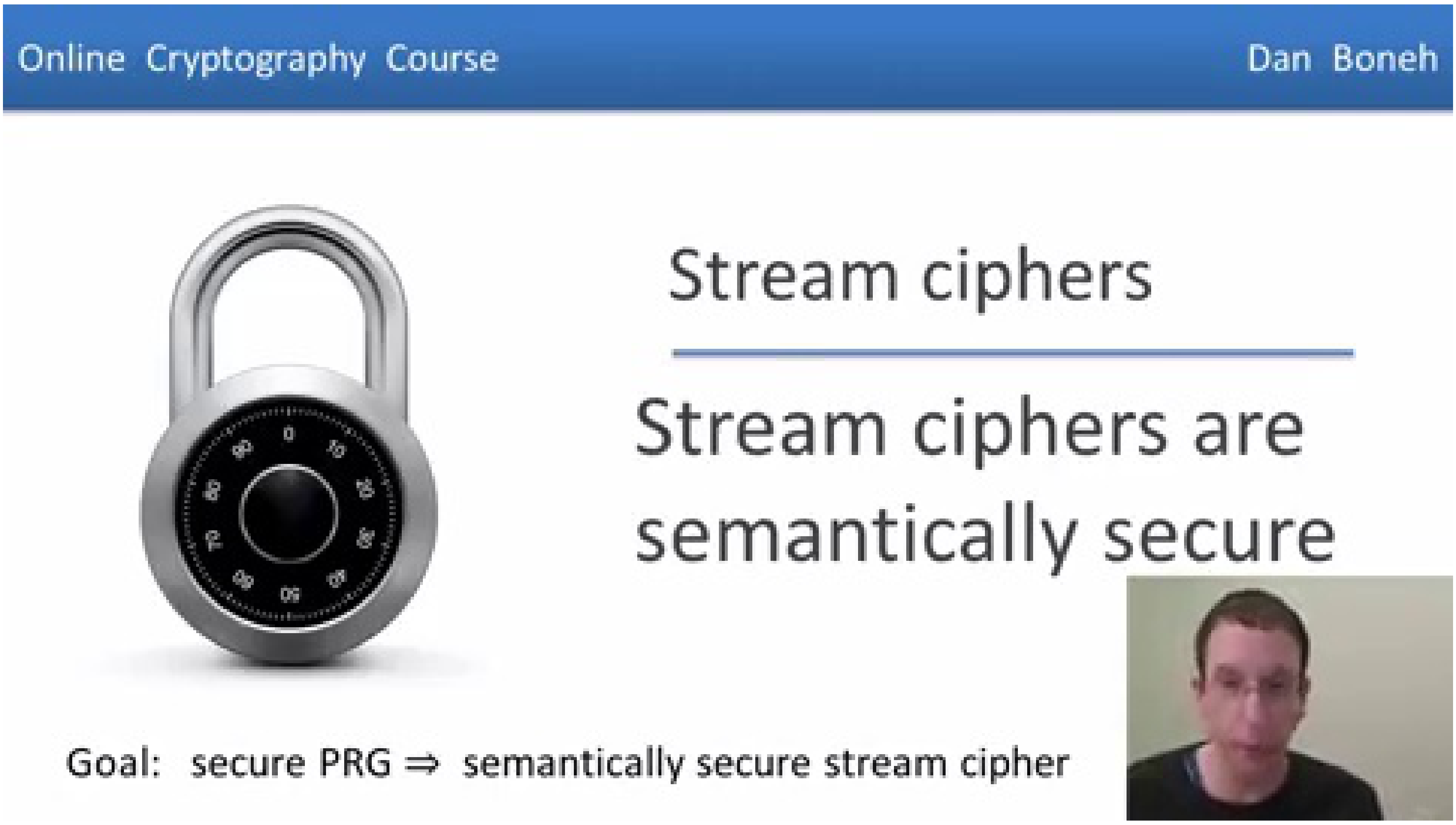


Online Cryptography Course

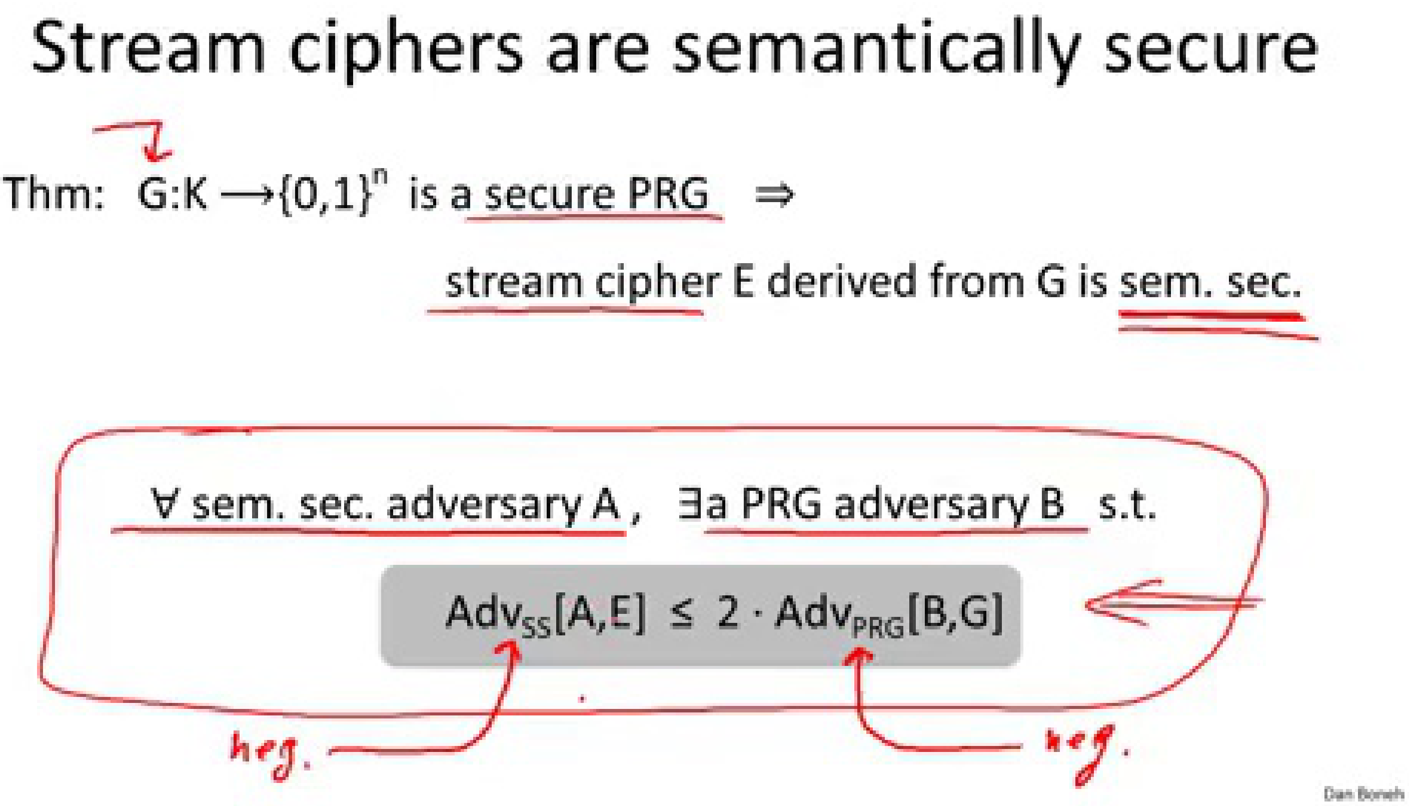
Dan Boneh

Stream ciphers

Stream ciphers are semantically secure

Goal: secure PRG ⇒ semantically secure stream cipher

So now that we understand what a secure PRG is, and we understand what semantic security means, we can actually argue that a stream cipher with a secure PRG is, in fact, a semantically secure. So that's our goal for this, segment. It's a fairly straightforward proof, and we'll see how it goes.



Stream ciphers are semantically secure

Thm: $G:K \longrightarrow \{0,1\}^n$ is a secure PRG ⇒

stream cipher E derived from G is sem. sec.

∀ sem. sec. adversary A , ∃a PRG adversary B s.t.

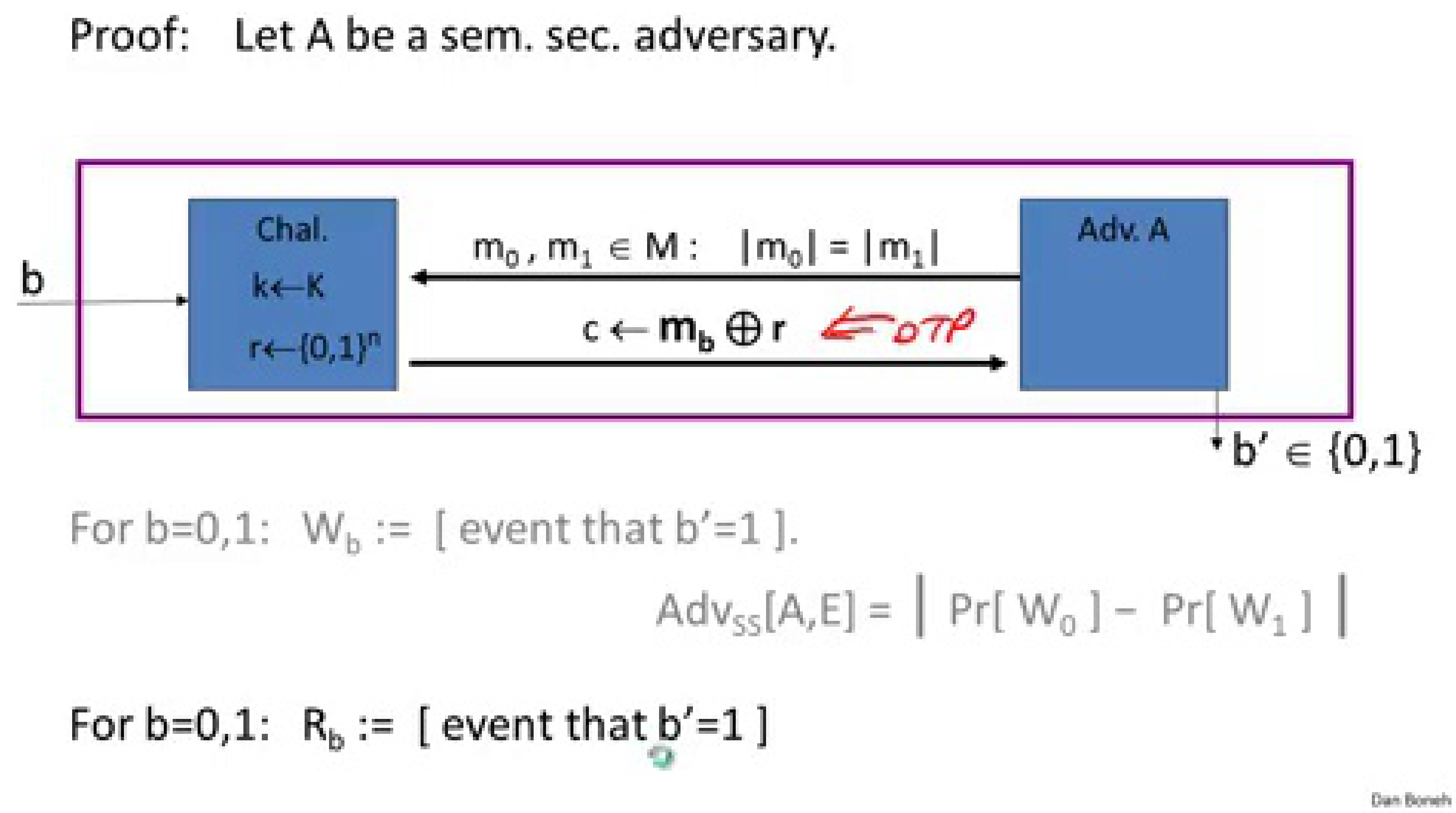
$$\text{Adv}_{SS}[A,E] \le 2 \cdot \text{Adv}_{PRG}[B,G]$$

neg.

neg.

Dan Boneh

So the theory we wanna prove is that, basically, given a generator G that happens to be a secured, psedo-random generator. In fact, the stream cipher that's derived from this generator is going to be semantically secure. Okay and I want to emphasize. That there was no hope of proving a theorem like this for perfect secrecy. For Shannons concept of perfect secrecy. Because we know that a stream cipher can not be perfectly secure because it has short keys. And perfect secrecy requires the keys to be as long as the message. So this is really kind of the first example the we see where we're able to prove that a cipher with short keys has security. The concept of security is semantic security. And this actually validates that, really, this is a very useful concept. And in fact, you know, we'll be using semantic security many, many times throughout the course. Okay, so how do we prove a theory like this? What we're actually gonna be doing, is we're gonna be proving the contrapositive. What we're gonna show is the following. So we're gonna prove this statement down here, but let me parse it for you. Suppose. You give me a semantic security adversary A. What we'll do is we'll build PRG adversary B to satisfy this inequality here. Now why is this inequality useful? Basically what do we know? We know that if B is an efficient adversary. Then we know that since G is a secure generator, we know that this advantage is negligible, right? A secure generator has a negligible advantage against any efficient statistical test. So the right hand side, basically, is gonna be negligible. But because the right hand side is negligible, we can deduce that the left hand side is negligible. And therefore, the adversary that you looked at actually has negligible advantage in attacking the stream cipher E. Okay. So this is how this, this will work. Basically all we have to do is given an adversary A we're going to build an adversary B. We know that B has negligible advantage against generator but that implies that A has negligible advantage against the stream cipher. So let's do that. So all we have to do again is given A, we have to build B.



Proof: Let A be a sem. sec. adversary.

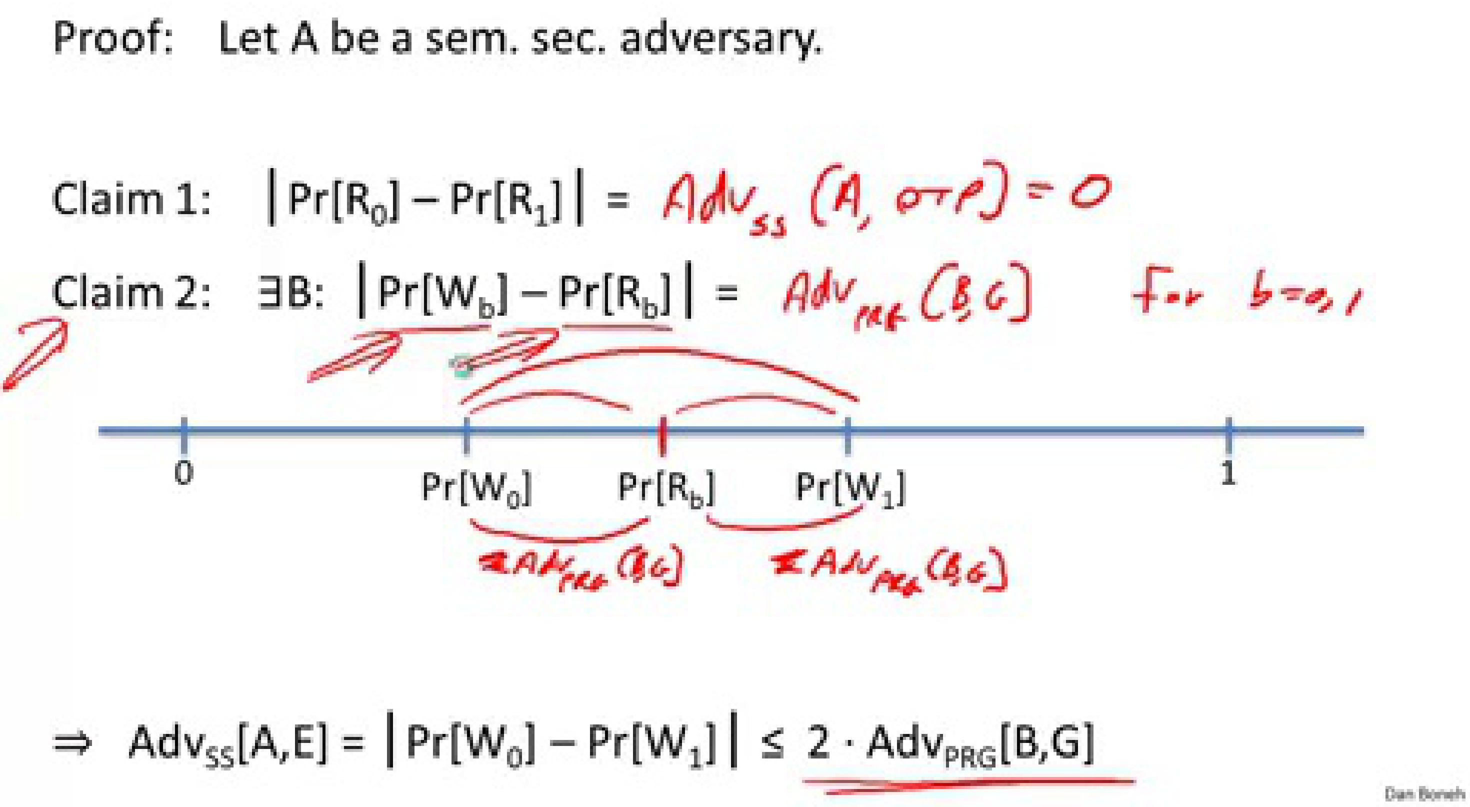
For b=0,1: W_b := [event that b'=1].

$$\mathrm{Adv}_{SS}[A,E] = \big|\, \Pr[W_0] - \Pr[W_1] \,\big|$$

For b=0,1: R_b := [event that b'=1]

Dan Boneh

So let A be a semantic security adversary against the stream cipher. So let me remind you what that means. Basically, there's a challenger. The challenger starts off by choosing the key K. And then the adversary is gonna output two messages, two equal length messages. And he's gonna receive the encryption of M0 or M1 and outputs B1. Okay, that's what a semantic security adversary is going to do. So now we're going to start playing games with this adversary. And that's how we're going to prove our lemma. Alright, so the first thing we're going to do is we're going to make the challenger. Also choose a random R. Okay, a random string R. So, well you know the adversary doesn't really care what the challenger does internally. The challenger never uses R, so this doesn't affect the adversary's advantage at all. The adversary just doesn't care that the challenger also picks R. But now comes the trick. What we're going to do is we're going to, instead of encrypting using GK. We're going to encrypt using R. You can see basically what we're doing here. Essentially we're changing the challenger so now the challenge cipher text is encrypted using a truly random pad. As opposed to just pseudo random pad GK. Okay. Now, the property of the pseudo-random generator is that its output is indistinguishable from truly random. So, because the PRG is secure, the adversary can't tell that we made this change. The adversary can't tell that we switched from a pseudo-random string to a truly random string. Again, because the generator is secure. Well, but now look at the game that we ended up with. So the adversary's advantage couldn't have changed by much, because he can't tell the difference. But now look at the game that we ended up with. Now this game is truly a one time pad game. This a semantic security game against the one time pad. Because now the adversary is getting a one time pad encryption of M0 or M1 But in the one time pad we know that the adversaries advantage is zero, because you can't beat the one time pad. The one time pad is secure Unconditionally secure. And as a result, because of this. Essentially because the adversary couldn't have told the difference when we moved from pseudo random to random. But he couldn't win the random game. That also means that he couldn't win the sudo random game. And as a result, the stream cipher, the original stream cipher must be secure. So that's the intuition for how the proof is gonna go. But I wanna do it rigorously once. From now on, we're just gonna argue by playing games with our challenger. And, we won't be doing things as formal as I'm gonna do next. But I wanna do formally and precisely once, just so that you see how these proofs actually work. Okay, so I'm gonna have to introduce some notation. And I'll do the usual notation, basically. If the original semantics are here at the beginning, when we're actually using a pseudo-random pad, I'm gonna use W0 and W1 to denote the event that the adversary outputs one, when it gets the encryption of M0, or gets the encryption of M1, respectively. Okay? So W0 corresponds to outputting 1 when receiving the encryption of M0. And W1 corresponds to outputting 1 when receiving the encryption of M1. So that's the standard definition of semantic security. Now once we flip to the random pad. I'm gonna use R0 and R1 to denote the event that the adversary outputs 1 when receiving the one-type pad encryption of M0 or the one-time pad encryption of M1. So we have four events, W0, W1 from the original semmantics security game, and R0 and R1 from the semmantics security game once we switch over to the one-time pad.



Proof: Let A be a sem. sec. adversary.

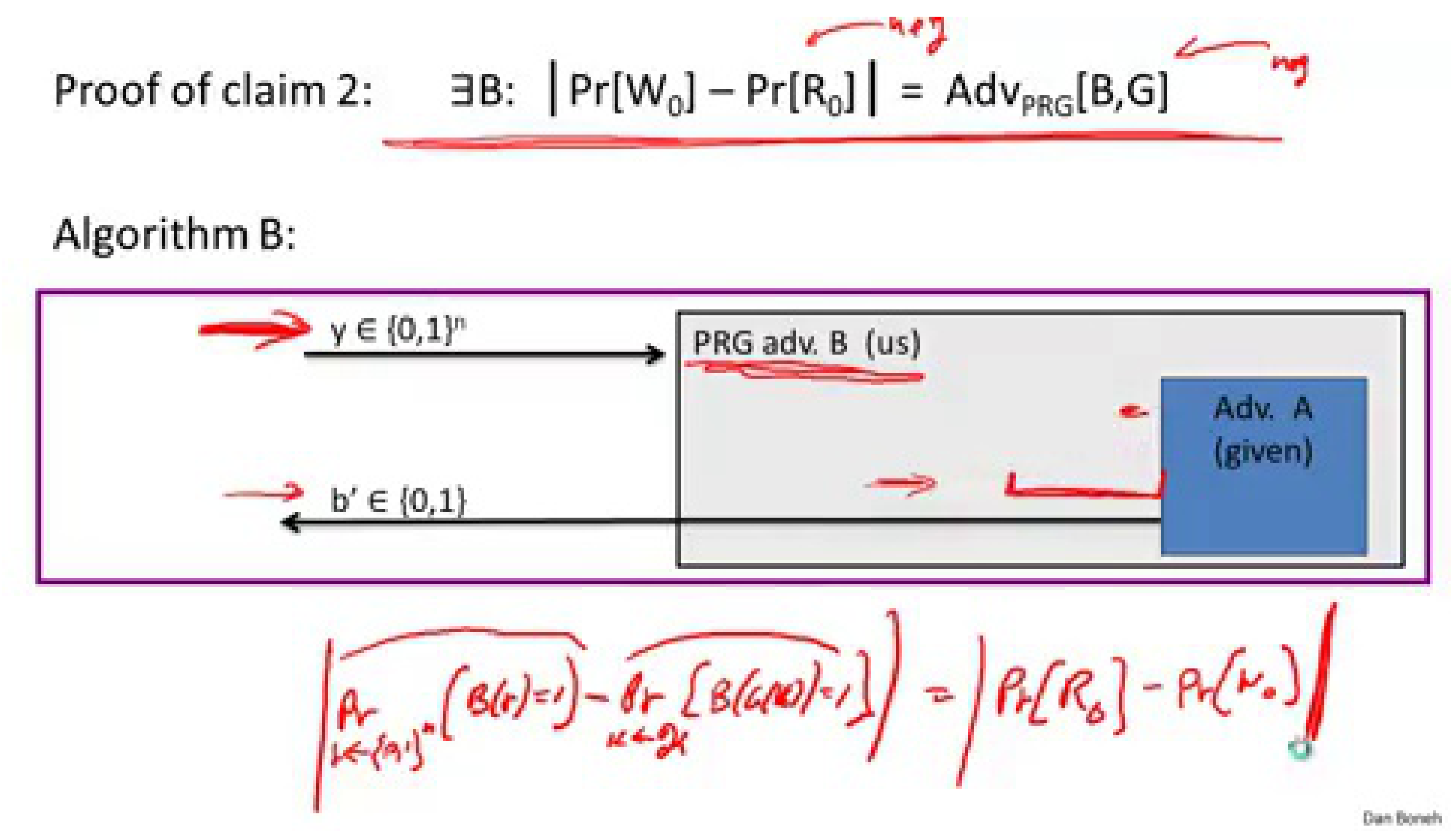
Claim 1: $\left|\Pr[R_0] - \Pr[R_1]\right| = Adv_{SS}(A, OTP) = 0$

Claim 2: $\exists B$: $\left|\underline{\Pr[W_b]} - \underline{\Pr[R_b]}\right| = Adv_{PRG}(B,G)$ for $b=0,1$

$\Rightarrow$ $Adv_{SS}[A,E] = \left|\Pr[W_0] - \Pr[W_1]\right| \leq \underline{2 \cdot Adv_{PRG}[B,G]}$

Dan Boneh

So now let's look at relations between these variables. So first of all, R0 and R1 are basically events from a semmantics security game against a one-time pad. So the difference between these probabilities is that, as we said, basically the advantage of algorithm A, of adversary A, against the one-time pad. Which we know is zero. Okay, so that's great. So that basically means that probability of, of R0 is equal to the probability of R1. So now, let's put these events on a line, on a line segment between zero and one. So here are the events. W0 and W1 are the events we're interested in. We wanna show that these two are close. Okay. And the way we're going to do it is basically by showing, oh and I should say, here is probability R0 and R1, it says they're both same, I just put them in the same place. What we're gonna do is we're gonna show that both W0 and W1 are actually close to the probability of RB and as a result they must be close to one another. Okay, so the way we do that is using a second claim, so now we're interested in the distance between probability of Wb and the probability of Rb. Okay so we'll prove the claim in a second. Let me just state the claim. The claim says that there exists in adversary B. Such that the difference of these two probabilities is basically the advantage of B against the generator G and this is for both b's. Okay? So given these two claims, like the theorem is done because basically what do we know. We know this distance is less than the advantage of B against G. That's from claim two and similarly, this distance actually is even equal to, I'm not gonna say less but is equal to the advantage. Of B against G, and as a result you can see that the distance between W0 and W1 is basically almost twice the advantage of B against G. That's basically the thing that we are trying to prove. Okay the only thing that remains is just proving this claim two and if you think about what claim two says, it basically captures the question of what happens in experiment zero what happens when we replace the pseudo random pad GK, by truly random pad R. Here in experiment zero say we're using the pseudo random pad and here in experiment zero we are using a Truly random pad and we are asking can the adversary tell the difference between these two and we wanna argue that he cannot because the generator is secure.



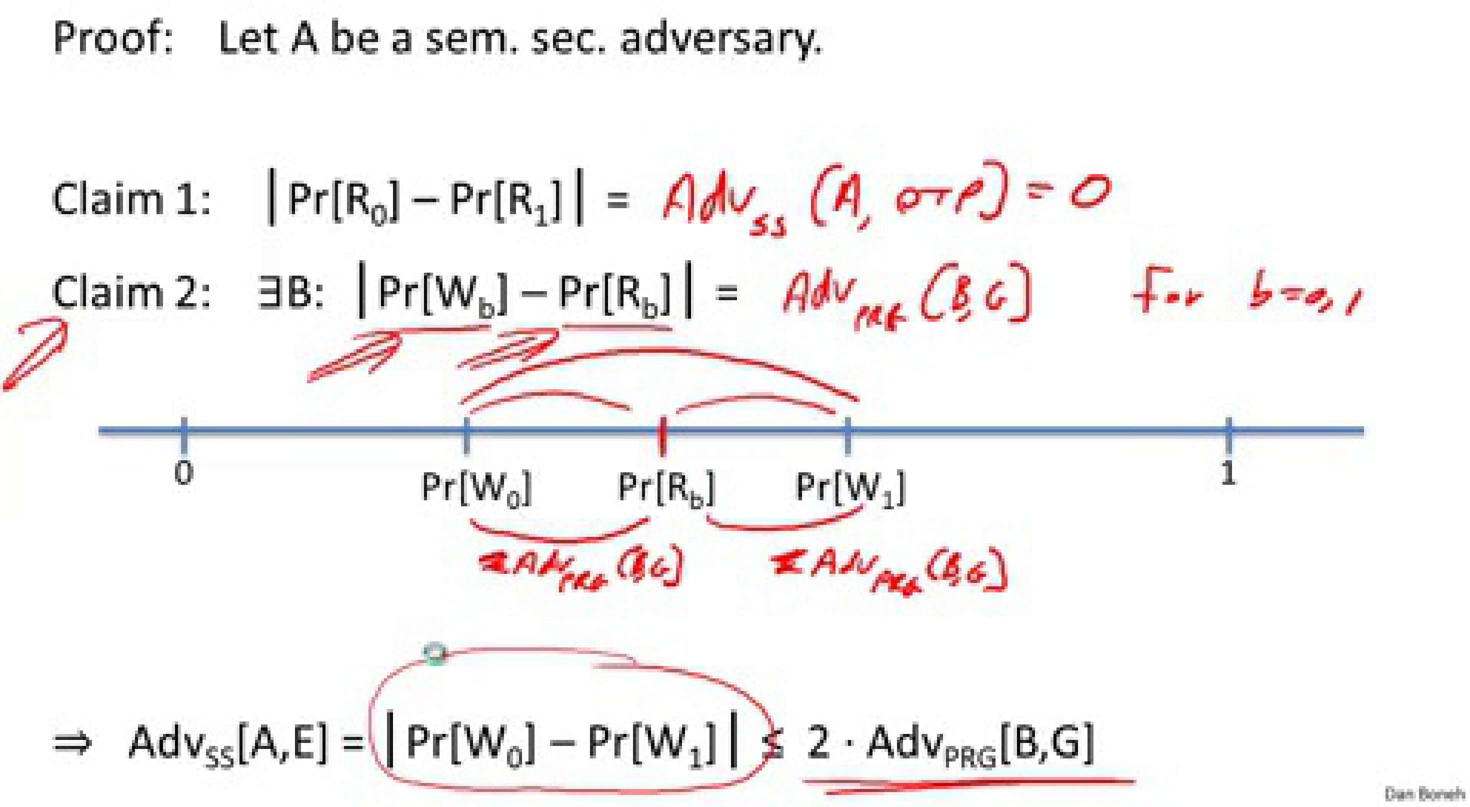
Proof of claim 2: $\exists B: \left| \Pr[W_0] - \Pr[R_0] \right| = \mathrm{Adv}_{\mathrm{PRG}}[B,G]$ neg neg

Algorithm B:

$$\left| \overbrace{\Pr_{r \leftarrow \{0,1\}^n}(B(r)=1)} - \overbrace{\Pr_{k \leftarrow K}[B(G(k))=1]} \right| = \left| \Pr[R_0] - \Pr(r_0) \right|$$

Dan Boneh

Okay so here's what we are gonna do. So let's prove claim two. So we are gonna argue that in fact there is a PRG adversary B that has exactly the difference of the two probabilities as it's advantage. Okay and since the point is since this is negligible this is negligible. And that's basically what we wanted to prove. Okay, so let's look at the statistical test b. So, what, our statistical test b is gonna use adversary A in his belly, so we get to build statistical test b however we want. As we said, it's gonna use adversary A inside of it, for its operation, and it's a regular statistical test, so it takes an n-bit string as inputs, and it's supposed to output, you know, random or non-random, zero or one. Okay, so let's see. So it's, first thing it's gonna do, is it's gonna run adversary A, and adversary A is gonna output two messages, M0 and M1. And then, what adversary b's gonna do, is basically gonna respond. With M0 XOR or the string that it was given as inputs. Alright? That's the statistical lesson, then. Whenever A outputs, it's gonna output, its output. And now let's look at its advantage. So what can we say about the advantage of this statistical test against the generator? Well, so by definition, it's the probability that, if you choose a truly random string. So here are 01 to the N, so probability that R, that B outputs 1 minus the probability, is that when we choose a pseudo random string, B outputs 1, okay? Okay, but let's think about what this is. What can you tell me about the first expressions? What can you tell me about this expression over here? Well, by the definition that's exactly if you think about what's going on here, that's this is exactly the probability R0 right? Because this game that we are playing with the adversary here is basically he helped us M0 and M1 right here he helped add M0 and m1 and he got the encryption of M0 under truly one time pad. Okay, so this is basically a [inaudible]. Here let me write this a little better. That's the basic level probability of R0. Now, what can we say about the next expression, well what can we say about when B is given a pseudo random string Y as input. Well in that case, this is exactly experiment zero and true stream cipher game because now we're computing M XOR M0, XOR GK. This is exactly W0. Okay, that's exactly what we have to prove. So it's kind of a trivial proof. Okay, so that completes the proof of claim two.



Proof: Let A be a sem. sec. adversary.

Claim 1: $\left|\Pr[R_0] - \Pr[R_1]\right| = Adv_{SS}(A, OTP) = 0$

Claim 2: $\exists B: \left|\Pr[W_b] - \Pr[R_b]\right| = Adv_{PRG}(B,G)$ for $b=0,1$

$$\Rightarrow \quad Adv_{SS}[A,E] = \left|\Pr[W_0] - \Pr[W_1]\right| \leq 2 \cdot Adv_{PRG}[B,G]$$

Dan Boneh

And again, just to make sure this is all clear, once we have claim two, we know that W0 must be close to W1, and that's the theorem. That's what we have to prove. Okay, so now we've established that a stream cypher is in fact symmantically secure, assuming that the PRG is secure.

End of Segment