GARY GENSLER: I just want to say how touched I am that you are all still here. I really-- you know, there's a lot of shopping opportunities in the MIT courses. And that you have come back and not shaken loose after reading Satoshi Nakamoto's peer-to-peer Bitcoin paper, or maybe you just came back to see whether I was going to crash and burn describing it.

But what we're going to try to do in the next three classes, just to frame it, is really give you some of the technical underpinnings of blockchain technology through the lens of Bitcoin. Bitcoin is just the first use case of blockchain technology. So if I often say Bitcoin this or Bitcoin that, it's really largely-- not entirely-- largely applicable to blockchain technology.

My feeling is I'm only about eight or nine months ahead of all of you. I may have spent my whole professional life around finance and public service, and I can talk a lot about markets and about public policy, but MIT has given me the gift of thinking about blockchain technology. And I'm trying to return that gift a little bit for you all.

And I have a few computer scientists in the room that are going to bail me out if I don't get this right. Sabrina, and then, oh, I see Alin is putting up his-- do you all know Alin? He's actually a PhD student at MIT, computer science. So somebody gets to that part of their life--

AUDIENCE: Terrible life choice.

GARY GENSLER: Yeah, yeah. What was that?

AUDIENCE: Terrible life choice.

GARY GENSLER: Terrible life choice. Yeah. But he's going to bail us all out.

But the reason that I think it's relevant not to just belabor it, is I really believe the only way that any of us can get to ground truths is to know a little bit about how the inner workings of this technology are. You're not going to have to do an algorithm or actually do a hash function, but to know underneath it. And then you can step away and say I no longer need to know how the

carburetor on the car works, but I know what a carburetor is. Or, you know, whatever analogy you want.

So with that little bit, as opposed to sort of all of that Socratic cold calling that I did last class, because money, Fiat currency is something at the core, and ledgers is at the core of a Sloan student's either education or background, this a little less of the core. If today's and the next couple of lectures, if you can work with me then I want you to interrupt me anytime you've got a question.

I'm not going to do much cold calling. I don't want you to relax too much. I still want you to do the readings the next three classes. But just raise your hand, stop me, say, well, but what is that all about. And that just sort of we can work a little bit different on these next classes.

So, as I'm always going to be doing, consistency. What are the study questions? So really, what are the design features? What are the key design features of this new technology, blockchain. And I put a few on the syllabus. And we're going to go through all this today and next week. Cryptography, append-only, timestamps blocks, distributed consensus algorithms, and networking.

I list four. Later in this lecture, you'll see 8 or 10 that-- I guess it's 10 that we're going to really dig dig into.

Can I just get a sense of the class and this is not for Talita or Sabrina to write down notes about participation. Is it a decent assumption, did most or all of you at least read Nakamoto's paper? All right. Good. All right, great. Just a sense, how many of you felt you got at least half of it, maybe less than 2/3, but at least half of it? All right, pretty good.

When I first read it, I was right with you. So it's all right. Alin you got more than half of it, right?

AUDIENCE: I read it five years ago, so.

GARY GENSLER: You read it five years ago. Yeah, yeah, yeah. Yeah, life choices, talk about it. All right. And you're taking this class. Good, good. So we'll go through each of those.

And then more specifically, we're going to peel back the cryptography. The two main cryptographic algorithms, or these words that you'll hear sometimes, cryptographic primitives-- Alin, what is a cryptographic primitive?

AUDIENCE: Oh, it's a wild beasts. There are so many of them.

GARY GENSLER: Yeah, but what's the two words together mean?

AUDIENCE: Well, that's I'm saying. It could be anything. It could be a hash function, could be encryption function, could be a very powerful computation scheme, it could be a data outsourcing scheme, could be a data access privacy access.

GARY GENSLER: But it's anything that basically protects the communication in the presence of adversaries.

AUDIENCE: Well it's also something that you can use to prove that computation was done correctly on trusted servers. It's not just communication, it's also computation.

GARY GENSLER: So communications and computation that needs to be protected or verified, have some form of cryptographic algorithm, which happens to be called a cryptographic primitive. The two main ones-- and there's a third one we'll talk about later in the semester-- but the two main ones, hash functions, just as a working knowledge of blockchain is worthy to know, and we're going to get-- everybody's going to get there. We're going to all get there to where you have some sense of what a hash function is. And then this whole concept of digital signatures, which relates to asymmetric cryptography.

Those two are very fundamental to blockchain technology. Later in the semester, we'll talk a little bit about zero knowledge proofs, but they're not as fundamental to the first application. And so that's why they're kind of-- and they help make things verifiable and immutable. And that's the business side, the market side. Why does it matter? Otherwise, like, who cares what's in the carburetor if it doesn't matter?

And then how does this all relate to the double spend problem? I can cold call on this. Isabella, do you remember what the double spending problem was from?

AUDIENCE: It was when they would use the same coin, I guess, and they would use it multiple places and other digital wallets [INAUDIBLE].

GARY GENSLER: All right. So in essence, a double spend is when you have a piece of information and you use it twice. And we happen to call this piece of information "money," but you use it twice.

You can send an email to two people and that's OK. I mean, it's a little embarrassing if you're sending it to one friend telling them you're available for dinner and the other friend thought you

told them you weren't available. But you can still send it to two places. But in the system of money, it's a critical thing that you don't use it twice.

The readings, was the demo helpful? I mean, we're going to do a lot more on that. I watched that demo last November, December. That was one of the first things I watched. From an MIT student. I don't know if you knew Bosworth. And I found it very helpful, so I'm glad. And I see it's actually that demo is on a Stanford blockchain course as well, so the West Coast, one of our competitors is using an MIT product.

And so we're going to just do a slight review of what we did in class 2. And then we're going to talk about the key design features, hash functions, as I mentioned, what is an append-only log, block headers and Merkle trees, and asymmetric cryptography and digital signatures. Crazy. We're going to cover all five of those today. And then you're going to tell me how we did. Oh, Bitcoin addresses, which is just a small thing. Six, actually.

So last time, for those of you that weren't with us, we talked about money. And again, money is just a social construct, or an economic consensus mechanism. We're going to talk a lot about consensus next Tuesday when we talk about the consensus protocol on Bitcoin. But remember, money itself is just a consensus.

There was a question on Tuesday, I think Alin actually had asked this question about well, what does it mean to be a liability in the central bank? Why is money, what does that actually mean?

And I said it just means that somebody else will accept it. It's a social consensus because it's not that they're going to give you anything else. It's just that you can get a bank deposit, you can pay your taxes, you can use it at Starbucks, if in fact, you've already gotten a cup of coffee. If you remember, it's only legal tender for a debt. And so forth.

Fiat money is just in that long line. But it's had its challenges and instabilities. It doesn't mean it's going to go away. I'm not a Bitcoin maximalist who thinks that Fiat currencies are going to go away. But Fiat currencies have their instabilities, particularly around weak monetary policy. In essence, when you debase a currency and allow a lot of it to be issued, or usually around unstable fiscal policy.

So either the government is spending a lot, the King is off to foreign wars, and the Bank of England was actually set up in the late 17th century in essence to control the currency when

the King was-- of England, I think-- was in wars with France, if I can recall. A lot of banks, central banks, were set up right about when a sovereign was off debasing a currency and spending too much at war.

Ledgers, we talked about ledgers, how critical ledgers are. In essence, ledgers are a way to keep records. And those records could either be transaction records or balance records. We'll see that Bitcoin is set up as a transaction ledger system.

Later we're going to be talking about other blockchain technologies that are set up as balance ledgers. So one should not just think immutability that there's only one way to do this. But transactions and ledgers are at the core of Bitcoin.

And central banking is of course, built on ledgers. The master ledger of the central bank, and then the commercial banks have sort of the sub-ledgers. And then you can think sometimes your digital wallet, maybe Starbucks has yet a third tier ledger.

We obviously live in an electronic age already. We know this. There's been many efforts, they've all died until Bitcoin to crack that riddle that we talked about, peer-to-peer money without a central authority.

And later in the semester when we talk about what are the use cases, that's going to be the core thing. It's why I'm not a maximalist. I'm not sure in every circumstance a central intermediary isn't necessarily so bad. And this is not a value judgment. It's just pure money and markets and so forth. But in some circumstances, decentralization really will compete and beat the centralized intermediary.

So let's talk about his little paper, which of course he was modest, or she was modest. Please remind me, we don't know who Nakamoto is or was, or a group of people. "I've been working on a new electronic cash system that's fully peer-to-peer with no trusted third party."

So you've seen this slide before. But a time stamped append-only log. Just think blocks of data. To kind of oversimplify, but it's got a name, blockchain.

And I don't think-- did Satoshi's paper, you all read it in the last few days, I of course read it again yesterday just to make sure I remembered it, I don't remember that he ever used the word blockchain. Am I right about that? Right. So the words blockchain are really-- have been sort of layered over his innovation.

So information, blocks going on. And that leads to basically a database. But it's the blocks of data. Bitcoin right now is about 550,000 blocks, and the blocks are added on average every 10 minutes. And we'll talk about why it's every 10 minutes, and not only why Satoshi Nakamoto made it every 10 minutes but how they maintain that.

Other blockchains like Ethereum it's about every seven seconds. So don't get too caught up that it's all the same. And there's some technologists, here Silvio Micali is working on Algorand and that's even tighter, less than seven seconds. So there's not one way. There's multiple designs on how often blocks are added. But let's start with Bitcoin.

Secured by yes, guess what, those two cryptographic primitives, hash functions and digital signatures. Lose anybody yet? Yeah? Maybe. And then there's a consensus for agreement.

The whole debate usually about databases is who gets to change the data. And this is true in all databases. In its essence, it's usually centralized. But in blockchain, it's all a sudden, well, maybe it's not centralized. Who gets to add that next bit of information, that next block? And the consensus agreement is-- which we'll discuss next Tuesday-- is about that very issue.

And I think there was a little pretty picture of that done in slides before. But I'm going to I'm going to delay that discussion until next Tuesday. And hopefully you'll all come back.

So what are the key features? And I might do a little cold calling. Do you remember any key feature, Tom? From the papers?

AUDIENCE: Oh, boy.

GARY GENSLER: It's all right.

AUDIENCE: Yeah. You know, the hash function.

GARY GENSLER: Hash function. Any other key features? Let's see how many. I'm going to have 10 on this page.

AUDIENCE: A private and a public key.

GARY GENSLER: What is that?

AUDIENCE: Private and public keys.

Private and public--

GARY GENSLER: Oh, private and public key. Yes. So asymmetric cryptography, or private and public keying. Yes, hash functions, yes, private and public key. Any other kind of key design features, or words you didn't understand? Maybe that's another way to put it. Leandro.

AUDIENCE: Addresses.

GARY GENSLER: What's that?

AUDIENCE: Addresses.

GARY GENSLER: Bitcoin addresses. Three.

AUDIENCE: Timestamp server.

GARY GENSLER: Timestamp server. That's four of the things. This is going well. [INAUDIBLE].

AUDIENCE: Double payments.

GARY GENSLER: Double payment is something that it's trying to address. It's not really a design feature, but it's a-- they have a solution for double payment, so I'll give you a credit for it. But it's--

AUDIENCE: Miners.

GARY GENSLER: All right. So Hugo says miners, which is really the consensus. So I'll say that the design feature is the consensus or proof of work. Kelly.

AUDIENCE: The full node versus the lightweight node.

GARY GENSLER: Right. So very interesting, this concept of nodes. And Satoshi actually talks about full nodes or lightweight nodes. In essence, how much information has to be stored. I want to reserve that. Kelly, please remind me when we talk about block headers to come back to that. But nodes in the network is a very important design feature. Over here.

AUDIENCE: The Merkle tree structure. The Merkle tree structure.

GARY GENSLER: Merkle tree structure. So Merkle tree structure is a way to compress a lot of data, and also to sort through that data. Uh-oh. No, Sabrina's not going to clean me out here. Merkle tree structure is there. We're going to talk about that. Two more.

AUDIENCE: Nonce.

GARY GENSLER: What's that? The

AUDIENCE: Nonce.

GARY GENSLER: Nodes. All right. What's that?

AUDIENCE: Nonce.

GARY GENSLER: Nonce. The nonce. OK. So a nonce. Anybody know what the word nonce is? A year ago I didn't. So this-- so we're all getting there. What, do I have a look, do you know what a nonce is? Yeah.

AUDIENCE: In the actual protocol, it's essentially a guess for the miners to kind of--

GARY GENSLER: So the word "nonce" means a random number that is used once. N for number, and "once." It's a number that's random and it's used once. That's how I've learned it. Whew.

And so one more, because this is great, actually.

AUDIENCE: Peer-to-peer.

GARY GENSLER: Remind me your first name.

AUDIENCE: Pria.

GARY GENSLER: Pria. Peer-to-peer. All right. So this is what I have. Cryptographic hash functions. We're going to go through these in more detail. Timestamped append-only logs, block headers and Merkle trees. So Merkle trees were discussed. But we need to actually say what information is kept at the head of the block as opposed to all the body. And some of that's just to make it more manageable.

Asymmetric cryptography, which is this public key, private key, and signatures. The Bitcoin addresses themselves, which interestingly are a little bit different than public keys. And then I breach break because in the next, we're going to talk about next Tuesday, the proof of work, the miners, the then the nodes, the nonces, they're are all in that little topic.

There's actually in Bitcoin a really important protocol is how information gets propagated on the internet. Just the network communication. It's not written about a lot. You won't read a lot about it in Nathaniel Popper's *Digital Gold* or all the other popular books, but it is an important

thing to remind ourselves that information has to propagate around the internet and all these transactions have to communicate with each other.

There's currently about 10,000 nodes on the Bitcoin network. We don't know where all of them are, but they're probably in 180 different countries. And so it's just-- also the networking and communication matters. And it matters to the economics a lot.

There's a native currency. This is interesting that it was the one thing that no one said. That's an actual technological design feature. It's not only that he created a currency, but the native currency is part of the economic incentive system. And we'll have some fun with that.

In essence, he said that when you mine and did the proof of work, you created and you've got some native currency called Bitcoin. So he created an economic incentive system. Whomever Satoshi Nakamoto was or is knew a lot about economics, as well as technology. Yes.

AUDIENCE: I just wanted to quickly add to what you said. So it's not only that he created this native currency, but wants the finite supply has reached, the currency can be distributed as a transaction fee, which I think is very important in [INAUDIBLE].

GARY GENSLER: And remind me your first name?

AUDIENCE: Daniel.

GARY GENSLER: So what Daniel just said is really interesting. Not only to take light of this individual or individuals that did this. But this world of Bitcoin and other cryptocurrencies creates a unit of account that could be valued. And once it's valued, you have sort of a native currency.

But as Daniel said, Nakamoto also said there would be a finite limit. It happens to be 21 million Bitcoin is the most that it can be, and we'll get there around the year 2040. Does anyone know how many Bitcoin there are right now? About half of you were investing in it. Hugh? Hugo? About 17 million Bitcoin right now. And all 17 million have come from this process of proof of work and mining.

Initially it was 50 Bitcoin every 10 minutes, roughly every 10 minutes. Then it went down to 25, and we're now at 12 and a half Bitcoin. And does anyone know what today's value purported-- I always should say purported value of Bitcoin, because I don't know if we can trust some of those websites that say with the values are. What is it?

AUDIENCE: $6,500.

GARY GENSLER: So $6,500 of Bitcoin at 12 and a half Bitcoin to mine a block. So you see that it's about $80,000 US is the reward to mine a block, right? So he created an incentive system that initially, if you got 50 Bitcoin and they weren't worth a penny, you would not commit that much.

You had to be a hobbyist, basically, in 2009, or a cyberpunk, or just kind of curious. Because you weren't getting much incentive. If in fact it's worth 6,500 today, you're getting $80,000 if you actually successfully mine a block.

And then there's the transaction inputs and outputs. Think about a check, who signs it, where you move money. There's something called the unspent transaction ledger. So this is the ledger part.

So when you think-- I think of the technology, I think of cryptography, which is kind of all that stuff at the top which we're going to discuss today. Secondly, the consensus mechanism. In essence, that's that key question of any database, who gets to amend the database? Who gets to decide to change the state of what we all agreed to?

And then thirdly, is the ledger, or the transaction ledger, which we're not going to deep dive into the scripting language, but we are next Thursday going to talk a little bit about the underlying scripting. Does that give you a path that's all this cryptography, the consensus, and then the transactions. Yes.

AUDIENCE: I have a question.

GARY GENSLER: And your first name? If everybody just says first name.

AUDIENCE: Oh. I'm just curious, so you mentioned that--

GARY GENSLER: I'm curious about your first name.

AUDIENCE: Sean.

GARY GENSLER: All right.

AUDIENCE: So just curious, you mentioned that the block value is roughly $80,000 US as of now. So just curious, in terms of the CPU power, the electricity that will be consumed to mine the block, how much does that translate to equivalent US dollar terms?

GARY GENSLER: So the question that's asked is how much electricity is being consumed for that miner to get that reward, that $80,000. And I'm going to try to answer in one minute. But we'll come back to this later in the semester about economics, and blockchain economics, and mining economics. But what has happened over these 10 years is more and more computers are being used, or are trying to mine for the Bitcoin.

And so today in the most recent research I've seen is that the probability of winning a block-- there's so much-- is it measured in terahashes? I can't remember the numbers. But it's how many terahashes, which, is it 15 zeros Is a terahash? Is it that, or is it 12? Well, in any event, there's so many hashes being done a second, x number of terahashes, that your probability of winning is quite low.

And so what's happened is most nodes and miners have entered into agreements called mining pools, where they smooth out the risk and everybody shares in the rewards. But those economics we'll talk about later, it's thought to be that you need electricity cost around $0.03 a kilowatt hour to be successful. And in most parts of the world you can't get electricity for $0.03 a kilowatt hour.

So you would put your mining rigs where you can get low cost electricity or where you possibly can-- you can get it legally low cost or illegally low cost. So there are a lot of mining rigs and in jurisdictions where there may be local officials that are allowing those mining rigs, and instead of $0.03 a kilowatt hour to the electric company it's $0.01 to $0.02 cents a kilowatt hour to the local government officials.

And the two largest mining pools are in China. And the third is in Russia. But we'll get into the sort of economics and at least some theories about why some are where they are.

So cryptography. So Alin's probably going to clean me up. It's not just communication in the presence of adversaries, it's also computation in the presence of adversaries. That would be good.

And we talked about-- we're not going to deep dive. If you remember, even in ancient times if you were going to war there was this wonderful little way that you could do cryptography. And then anybody who's seen imitation games about the British breaking into the German codes, even though they should have probably given more credit to the Polish government that had probably broken into it in the 1930s, but Turing did great work. And then we're going to talk about asymmetric cryptography today.

All right. What is a hash function? A hash function, and these are just words that I think of it, I think of it as a fingerprint for data. But it has certain properties. The one that you'll see throughout is that it takes inputs of input x. It maps that input of any size to a fixed size.

So one that we use here in the US, one hash function we all use is zip codes, in a way. It's five digits, it's a fixed size. I know I'm doing this as a loose hand, how can I think of it. But zip codes. You might have 50,000 people or 5,000 people all living in one postal district. And you can map them to zip codes, and it's a fixed let.

Now, I don't know whether my friends in the computer science departments-- but it's an early sense of a hash function. I just wanted to say there are tangible things in our life that act like hash functions. Problem with zip codes is it will not in any way be a secure hash function. And you'll see that in a minute. But it does take-- you can be a 300-pound person or a 30-pound kid and you still map into the same zip code.

It's deterministic. It's always the same. So if you take a certain set of data, it will always give you the same hash. And that's relevant to the background. And you can efficiently compute it. You don't want to take a year to do this. You've got to do it in short periods of time. And in Bitcoin's case, it's done in nanoseconds or less, because they're one computer, one CPU can do-- can't remember, probably-- how many millions a second?

AUDIENCE: Couple of terahashes a second.

GARY GENSLER: Couple of terahashes a second. So it's a remarkably efficient algorithm. And so a bunch of mathematicians-- and hashing started in the 1950s and '60s, but the ones that we're talking about here are much more recent.

But it's really terrifically talented scientists, mathematicians, computer scientists, and sometimes the National Institute Standards of Technology here in the US working on hash functions. So it takes a array of any size, puts it into a fixed number-- I think zip codes for a minute-- it's deterministic. It's always-- you only live in one zip code, in a sense. And it's very efficient.

But now what are its cryptographic properties? Because a zip code wouldn't make it. It just wouldn't. Well, the computer scientists use this term preimage resistant. I would just say it's one way, you can only go one way, meaning it's infeasible to determine the input from the

output. It's infeasible to determine the x from the hash of x.

Does anybody know why I use the word infeasible rather than impossible?

AUDIENCE: [INAUDIBLE]

GARY GENSLER: First name?

AUDIENCE: Brotish.

GARY GENSLER: Brotish

AUDIENCE: Because we can do it with brute force.

GARY GENSLER: So you might be able to use it brute force. What do you mean by brute force, just so everybody--

AUDIENCE: Try all the options.

GARY GENSLER: Try all options. But as I understand it, a sort of tenet of cryptography for centuries is not to have it mathematically impossible, the point is getting it so infeasible that your adversary can't either get the communication or so forth. So hash functions, I just say this because you can't assume that Bitcoin can't be broken. We all call it immutable. It is immutable. Until the hash functions that are inside of Bitcoin might be broken.

And even Satoshi wrote about this in 2010. He got emails. There's this wonderful book if any of you want that I mentioned in the bookshelf at the end of the syllabus, he said, well what if a SHA-256, which is the hash function, gets broken?

And his answer, by the way, was well, there will be a better hash function at that time. Whatever that is, we'll hash the entire system, whatever that is. Because remember, you can take something of any size, hash it with a new system, and move forward.

And so he or she felt in this wonderful email is that Bitcoin actually could transition to a new hash function as long as you had a little bit of time before it was all corrupted. Kelly.

AUDIENCE: Is this what his article called the Gambler's Ruin problem? Is that we you're describing?

GARY GENSLER: The Gambler's Ruin problem.

AUDIENCE: The probability that an attacker could catch up to recreating it.

GARY GENSLER: OK.

AUDIENCE: That's something else. That's--

GARY GENSLER: Will you speak a little louder?

AUDIENCE: Yeah. So that's the-- you want to sort of assess how hard it is to fork Bitcoin. If I have a lot of computational power, how hard is it for me to create a fork? And Satoshi does an analysis at the end of the paper--

GARY GENSLER: Oh, I apologize. You're talking about in his paper. Yes. In his paper, he's talking about how hard it is computationally to do what some people call a 51% attack, to basically take over all the nodes. And that part of his paper we're going to talk about next Tuesday. But it's basically, can you take over the nodes? I was talking about a separate thing, can you break the cryptography. And he doesn't write about that in his paper. He writes about it in an email about 10 months later or so.

Second key cryptographic thing. So we said one is it's one way. The other thing is this concept of collision resistant. I presume if everybody in this room told me your birthdays, there's multiple people in this room who have the same birthday. And in fact, if we got it past 26 people in a room it's over 50% chance that two of you have the same birthday. We don't need to get to 183 people in the room, which is half of the days of the year. We can get to about 26 or 7.

And similarly, the key thing is is that two sets of data are-- it's again, infeasible that x and y would hash to the same thing. It's not impossible. It's infeasible. And if you look at the history of hash functions, this is usually the thing, that at some point in time these hash functions will not be collision resistant. Some quantum computing will come along, or something will come along. But for now you can put something of any size in and they're independent.

They also look terribly random. It's called an avalanche effect, meaning you change one little difference and the whole thing looks different. So when you noticed on that little video, if you changed one thing, it all looked so different. And why that's important is it makes it more secure.

And then there's something called puzzle friendliness. Even if you know a little bit of the input, it doesn't mean that you're going to get the output. I put these up here not for you to know

them. You're not going to get tested. If you go into business, as Elon, you've started, when you probably haven't thought, well, collision resistant this or that.

But I just wanted you to know there is a bunch of cryptography underneath this. And the key is it is not 100% immutable. It's probably one in, you know, I don't know, a quadrillion immutable. But there's still-- these things could be broken. And quantum computing and something else might-- Alin.

AUDIENCE: The actual probability should be actually 1 over 2 to the power of 128. So much more than one quadrillion.

GARY GENSLER: So it's 1 over 10 to about the 40th. How'd I do? My math all right? All right. And anybody who's interested can come to office hours. So it's highly unlikely to be broken. But I think it's always worthwhile to say, well, no, there's some outward-- it's not as bounded as you think.

So what is it used for? In many places it's used for names, and references, and pointers, and in something called commitments. In Bitcoin, it's used for pointers because one block points to another block. But it's also used in commitments.

You'll hear these words. We're not going to delve into them. But the headers and the Merkle trees use something called SHA 256, which is a standard which is literally 256 bits long. That's like zeros and ones for 256 registries.

But a Bitcoin address actually-- Satoshi Nakamoto threw on a loop. I'm glad to debate why, but he uses two hash functions for Bitcoin addresses. The one thing I saw that he actually wrote about it is he said if one of them is broken at least the other one is less likely to be broken. So as I've read about it, I think in his own voice is you have to hash something twice. And he was just making it that much more secure, even knowing it was one out of 10 to the 40th chance.

AUDIENCE: Which is astronomically low, so.

GARY GENSLER: Right. So. So remember, where's Caroline? I remember-- there we are. You asked me about, I thought I had set it up for today, which you were good to remind me for Tuesday, what's the longest running hash, time stamped hash?

AUDIENCE: That is a great question.

GARY GENSLER: Thank you for the compliment.

AUDIENCE: The answer is-- yeah, I don't know that phonetically, so I'm not sure if I'm totally butchering this one. But it came out of Bell Labs with Stuart Haber and Surety.

GARY GENSLER: There he is. Yeah. So Haber and his colleague-- yes. You got it.

AUDIENCE: That's my roommate.

GARY GENSLER: That's you roommate. Terrific. So I'm just trying to say it wasn't Bitcoin that had it. He did this in 1991. But by 1995, they started a company called Surety. I don't think it took off that much. It's not competing with Apple for the largest market cap or anything like that or Facebook.

But every week in the notices section, you can see a hash literally. It's time stamped because it's in the *New York Times.* And it's a hash, all those funky digits and everything of all the information came before it. And they're basically hashing any document. Any document that you want a timestamp in that week, you put it in. One follows another, and that's a blockchain. It's not about money. There's no native currency and so forth.

I believe that Haber and Stornetta are three of the eight or nine footnotes in the Satoshi paper. Maybe it's four of them. So he gets his credit. And if you go to his website, Stuart Haber, I think he says, blockchain's co-founder on his personal website. Who knew?

So here, we get-- this was in the National Institute, the NIST paper. But timestamp append-only logs in Bitcoin or blockchain. What is put together is the header, the top information. And if I can go past the visual and just say, what's there? There's five pieces of key information. The version, it doesn't change that often. But there is a version number. The previous block's hash, so it's some information about all the blocks that came before it. The Merkle Root hash, which does anybody want to tell me what that does, the Merkle Root?

AUDIENCE: So it essentially posts the transactions in the bottom most layer of the tree and then creates the [INAUDIBLE] hash of each of the transactions.

GARY GENSLER: So if I go back to this nice little picture, the yellow box at the bottom up each of these blocks is all the transactions. There could be upwards to 1,000, 2,000 transactions in a block. So there's blockchain concept, 1,000, 2,000.

There's means and methods well before Nakamoto's paper about how to compress that, how to keep that information a little bit tidier. And that uses this thing called Merkle Roots. The five items right at the top, what's called the block header, doesn't have the 1,000 transactions.

And earlier, Kelly, you had asked me about full nodes and light nodes. A light node or a wallet that anyone here could download on your cell phone probably does not download the millions of transactions that have happened in the history of Bitcoin. You are unlikely to download what's called a full node. But you might download all the headers, this bit of information that's all of the headers.

All of the information in Bitcoin is still not that large. It's less than 200 gigs. But all of the headers, I think, is single digit gigs. I can't remember if it's four or six gigabytes right now. What is the number?

AUDIENCE: The header is 80 bytes. So it's 80 bytes times 500,000, which is 50 megabytes, 60 megabytes of headers.

GARY GENSLER: So it's 60 megabytes, so it's much smaller as opposed to like 180 gig. So Satoshi was thinking in advance. And every blockchain that you're going to work on, likely, I mean, there might be some, this concept of it's really keeping the security by a little bit of information in something called a header and then pushing all the meat of the transaction and data down. And this is really important when you get to like Ethereum where there's a lot of data, a lot of computation down in each of these blocks.

It's sort of like if Stuart Haber had a lot of documents and pictures and everything. You don't have to have all the picture quality and a whole movie. You can actually hash a whole movie, and you still get these 256 bits.

So whoops. So the header has the previous hash, this Merkle Root, which is just a way to get all the transactions. Just think of a Merkle Root as a way to grab 2,000 transactions in a way. A timestamp, that one's easy. We can get that. Difficulty target, anybody know what blockchain, Bitcoin tried to do to make it more or less difficult over time? No. Brodish, we've heard.

AUDIENCE: [INAUDIBLE] time but such that it stays with creating a block every 10 minutes. So with more computational power, it gets harder to find a block.

GARY GENSLER: So it's harder to find a block, the more miners there are. So every block header needs to have some what's called a difficulty target. How difficult is the mining going to be? Since we're talking about mining next Tuesday, these all bring me back to difficulty target. And then what's

a nonce?

AUDIENCE: [INAUDIBLE]

GARY GENSLER: What's that?

AUDIENCE: Just a random number.

GARY GENSLER: A random number that's used one. Number once, nonce. And that's hash functions. How'd we do? We're a little off the skids. We are MIT. Yes?

AUDIENCE: I have a question. The number of characters in the hash is equal to your--

GARY GENSLER: The output, not the input.

AUDIENCE: No. No. They put the number of characters in the hash is limited, right? So that's a pool of functions that you have. When you have many, many transactions, that's like a flow, right? So internally, you're just consuming and consuming hashes up to a point where you're going going to repeat that hash, right? So how do you know for the same has, you have two different information, to which information you're referring to?

GARY GENSLER: So could you help me pronounce your first name?

AUDIENCE: Diermo.

GARY GENSLER: Diermo, has asked the right question. He's say, well, how do you know? Especially as you have more and more time and more and more time, you might get the same output of a hash from different inputs. And if you recall-- wait. Somebody does recall. Now before Brodish, in front of Brodish.

AUDIENCE: The papers mentioned that it's possible that two the hash of x equal to hash of y. But if the miners are working at the same time, if the same information are not treated at the same exact time, it won't be a problem because then they just continue just like two different--

GARY GENSLER: So you're correct as it relates to mining. But there is another piece of it as well is that the hash function, if it's a good cryptographic secure hash function, is what's called collision resistant where what you're saying is so infeasible, in fact, 1 divided by 10 to the 40th, that's a 1 with 40 zeroes after it. It's so infeasible to happen, it's possible but infeasible to happen. What you're referencing is what if two parties solve the cryptographic puzzle as opposed to a collision. And

because of the difficulty, they just got at the same time. Please.

AUDIENCE: It seems like a dumb question but--

GARY GENSLER: No. There's no dumb questions when it comes to this. I really mean that.

AUDIENCE: The timestamps attributed, so is it from the whole system or?

GARY GENSLER: So timestamps are not a particularly important part of Bitcoin. They are timestamped. But sometimes if somebody puts something off and it's off by a few minutes or even up to two hours, there's a check in the technology in the scripting function if the timestamp's off more than a couple hours. So literally, it's not that precise.

Having said that, the real way that timestamping happens is if a block is mined and it's the 540,000th block and it's sort of accepted in all the nodes, these 10,000 nodes start mining the 540,000 and 1st block, in essence, it's just think of it as almost like a stack. And so what's, in essence, more relevant than the actual time that's in the header, and they all have a timestamp in the header, but what's more relevant is the order of the blocks, and, most importantly, the previous block hash. Yes?

AUDIENCE: I would say that without the timestamps, you cannot do this difficulty readjustment. The timestamps are very important. If you don't have timestamps on the block, you cannot do the difficulty readjustment, which is necessary to keep the rate of blocks 10 minutes after [INAUDIBLE].

GARY GENSLER: I'm going to partially agree with you because the difficulty adjustment happens every two weeks. So even if any one individual or five or six timestamps are a little goofed up in the two weeks, the algorithm is basically looking over the course of about 2,000 blocks.

AUDIENCE: Yeah. So a little goofed up is fine. But you need the timestamp.

GARY GENSLER: You need the timestamps. But it's more important is basically the-- here, I'll go back a slide. It's the order of the blocks. Please.

AUDIENCE: Going back to when we talked about collisions. The paper didn't really go into detail, but it said like in addition to how unlikely it is with to the power of 128 that even if there were two that hashed to the same kind of has digest that it would be unlikely that they'd both be valid in the context. So given what's a valid blockchain transaction that that could even further reduce the

likelihood of any problems, which there wasn't a lot of detail as to why the blockchain context would even make two hashes of the same value even more unlikely because of the context.

GARY GENSLER: I want to hold that question for Tuesday. But it has to do with rather than the collision issue, what the paper is talking about is if two miners solve the puzzle. And that doesn't mean that they got identical hashes because the puzzle is not geared to getting an exact hash. The Bitcoin puzzle is having a certain number of leading zeros. So it's literally started, I think, it was nine or 10 leading zeros. I'm talking about 10 years ago. And now, you have to hash to something with, I think, it's about 20 or 26 leading zeros. Meaning it's gotten more and more difficult, and the result of the hash has to have a bunch of leading zeros, what you saw in that video. I'm sorry.

AUDIENCE: I have a question on how the hash, the [INAUDIBLE] hash comes about. So if it's only hashing the transactions, how does it change when the hash of the previous block changes?

GARY GENSLER: OK, so, Addy. It reminds me of that old television show with Johnny Carson. And you just did a great setup for the comedian. So thank you. So I'm going to go to Merkle Roots.

So Merkle Roots, which are a binary data tree, looks something like this. If one had 1,000 transactions, I wouldn't have a pretty slide. So this only goes to four levels. But think of four transactions at the bottom. They're each hashed.

And then you concatenate. You put the two hashes together. You hash that. You keep going up the tray. If you had 1,000 transactions, because that's 2 to the 10th roughly, then you'd have 10 levels of this tray. And so that's what happens.

And literally, the mining pull operators are doing this a lot for the nodes. But in the Bitcoin core application, in software that anybody in this room could download the software if you wished. There is software that helps, takes transactions, puts them basically into this binary tree called a Merkle tree, uses hash functions, and basically skinnies it all the way up to the top. Does that--

AUDIENCE: I think what my question was that given that this structure exists, how does the root hash change with the previous block? So basically, we saw that if you change the hash of the previous block, all the blocks forward will get invalidated because the hash changes. But it doesn't seem to use the previous hash.

GARY GENSLER: So I'm going to repeat the question. Does the Merkle Root that is basically a summary of the

10,000 transactions that are in a block change if the rest of the header changes or the previous block change? And the answer is no. It only changes if some of the data in the 10,000 transactions change.

And so a Merkle Root will change if you put different transactions in the mix or, as is really important, one of the incentives. You get your 12 and 1/2 bitcoins today in what's called a Coinbase transaction. And so one of these 1,000 transactions is the payment to the miner. So the Merkle Root would be different depending upon who wins. But that wasn't your question. I'm just saying. But Merkle Roots are a very efficient way to take thousands of transactions, store it up, have one spot. Please.

AUDIENCE: So the order of the different transaction has to be exactly the same for everyone that is hashing, right?

GARY GENSLER: No, actually not. So if you're hashing, and you're running a mining rig, and Elon's running a mining rig, if Elon solves the puzzle and propagates it out on the network, and people start mining on top of Elon's block because they say, well, he's finished. You're not-- you're just going to probably start mining on the top of his block and look in something called the mem pull. The memory pull is this network of all the free floating transactions. You'll scoop up the next set of transactions.

AUDIENCE: And so how can we validate that all the transaction he wrote are the real ones?

GARY GENSLER: All right, so validation, which is more next Thursday, but I'll give it a shot. No, no, no. It's a good question. Every transaction-- or actually, you're setting me up, digital signatures. There you go. Thank you. Did you have a question or I'm going to on to digital.

So the second cryptographic thing, and we're going to keep going back and forth, hash functions are basically a way to compress a lot of data, have a fingerprint, make sure that it's basically commitment. Digital signatures, well, remember that little graph that we had Alice and Bob? Alice wants to send a note to Bob and just say, hello, Bob.

She wants to encrypt it. She encrypts it with Bob's public key, sends it to him. He decrypts it with his private key. You might say, oh my god, Gensler, what's a private key? What's a public key? In cryptography, it's a way to kind of scramble information. I know. I'm really making this like--

So if we went back to that little mechanism the Romans used or we used what the Germans used in the Enigma machine, they were symmetric cryptography. Both people had the key. The key was the Enigma machine with five rotors. In the 1970s, some wonderful technologist here and elsewhere basically said, well, what if the key isn't the same? Because the adversary could steal the key.

What if it's not symmetric but it's asymmetric? There's a private key and a public key. In essence, there's two keys that have some mathematical relationship. And the math between these two keys don't matter for a class like this. But know that the public key and the private key link together. They're bonded together.

But the critical thing is about digital signatures, there's three functions. You have to generate a key pair. And when a key pair is generated, a public key and a private key are generated at the same time. And they need a random number to go into it.

And one of the things that makes a lot of Bitcoin and other wallets insecure, and it's probably why some have been hacked, the wallets, not Bitcoin, is because they don't have good random number generation. Yes, Brodish? I saw-- I was at a conference last week where a technologist from the University of Pennsylvania had done a survey of 150 hedge funds, mining companies, and Bitcoin wallet companies and the like.

So they actually let a cybersecurity individual get inside and do a survey of 150 what you would consider really committed, high end users of Bitcoin, miners and hedge funds and crypto exchanges. And it was horrifying, their cyber security as to what they're doing with their private keys. Before he even got to the private keys, many of them didn't really have a secure way to create the random numbers to create their private keys. So it's just a piece. When somebody says they have really good private key, public key, in the back of your mind, just know there's got to be some way to do a random number generation. That's the only math that I'm going to ask you to remember of that.

There is a signature function. And the key thing is a signature creates. You can create a digital signature from a message and a private key. So if Kelly has a private key and wants to send a secret message to somebody across the room-- Isabella, you want a message from Kelly? Kelly's going to take the message. You got this, Kelly? You're going to take the message, and you're going to sign it with a private key. You send it over to Isabella. How's Isabella know that it was from you?

AUDIENCE: She has to decrypt it with her key.

GARY GENSLER: She's got to verify it. So there's a function called a verification function, and it comes back just yes, no. I mean, it might say it differently. But it's just a yes, no. It's a verification function. Isabella-- you want to do this with me-- is going to verify your signature is valid for this message because you have the public key. So you're right.

Isabella has your public key. But using your public key, she can verify that the signature. It's magical math. Well, it's not magical math. It's real math. But it's not math we need to study in this class. Yes, Hugo?

AUDIENCE: Back to generating the key pair.

GARY GENSLER: Yeah?

AUDIENCE: So they're both generated from the random number? One is not-- like the private is not determined by the public key or the other way around?

GARY GENSLER: The public-- you can think of it-- in Bitcoin, it uses an elliptic curve cryptography. And you can think of it as that the private key is based on the random number. To be more technical, the random number is what gets you to the public key. But I think of it as the private key is almost the random number, and then the public key is generated along with it.

AUDIENCE: So [INAUDIBLE].

GARY GENSLER: Yes.

AUDIENCE: So you pick a random number actually between 0 and 256, that's your private key. To pick a public key, you derive it directly from the private key. In fact, all you do is you exponentiate another number by the private key. So you can think of the public key as a one way function of the private key. So given a public key, you cannot recover the private key. If you could, then you could sign, potentially disastrous.

GARY GENSLER: And instead of exponentiation, in Bitcoin, it uses a function called the elliptic curve. So what properties? And these are the key economic properties as well as cryptographic properties. Basically, it's infeasible. And again, I use the word infeasible. I didn't say impossible, even though Eileen might want to tell me that it's 1 over 10 to the 40th of something. But it's infeasible to find a private key from a public key, so reverse engineer.

AUDIENCE: So even if you can't find the private key, like in the case of Kelly and Isabella, if I knew Kelly's public key, could I send a message to Isabella impersonating Kelly?

GARY GENSLER: No. You need to do a signature-- if you please just run your eye up there. To do a digital signature, you need a private key and a message. And it's a function of the message and the private key.

Let's call it complex math. That digital signature was created from the private key. And the public key was created from the private key. And to oversimplify the reason that the verify function works is because both the digital signature and the public key that Isabella has-- Isabella has this digital signature, and she has the public key, and she has the message.

The math is such that, basically, the private key, if you wish, almost like factors out. But think of two functions. Isabella has Kelly's public key, the message, the digital signature. It either verifies or it doesn't. But she never has to see the private key. And in fact, Kelly does not want her to ever see the private key.

AUDIENCE: Eric, maybe just to simplify the way the validation of the digital signature works is Kelly's message is run through a hash function which generates a hash. And it's encrypted with her private key. Then the message encrypted and the digital signature goes to Isabella. Isabella, what she does is using the same hash function to run it with the document to generate the hash function and uses the public key of Kelly to unencrypt the signature and compare those two hashes. If those two hashes correspond that means that the message belongs to Kelly and it hasn't been tampered with. So that's the more or less the simplification of the digital signature process.

AUDIENCE: I don't know if--

GARY GENSLER: So I mean, the key is basically that there's a scheme unrelated to Bitcoin that exists for many other reasons on the internet, many other reasons in commerce and at war that this public key, private key cryptography. And it's not simply just going back, it's not just simply Alice sending something. It's also digital signatures. You generate the key pair. Everything in Bitcoin, everything in Ethereum has key pairs, public key and private key, a digital signature. But, Kelly, never lose your private key. You got that? Do not. And by the way, you have to create it with a good random number generator because most sophisticated hedge funds around the world aren't. So you're going to be better than those. That's what I learned at a

conference I was at recently. And then there's a verification function.

AUDIENCE: A quick question about the random number generator and the verification function. So is there any third party generating the generator or the generator is a function already existing and already there?

GARY GENSLER: So the question is, if random number generation is so important, are there outside parties that have good software, in essence, to produce the random number generation? And the answer is yes, and there's some that are not so good. And yes, some good laptops have it.

At the heart, I want to skip ahead. Elliptic curve digital signature algorithm, that's the actual algorithm that Bitcoin uses to take the private key and so forth. But many of the wallets, if you download a wallet application to hold your Bitcoin, to hold your Litecoin, to hold some other coin, that wallet application has a random number generation software. I can't attest to all the random number generation software. I'm not a cyber security expert. But there's probably a range of some that are a little bit more. There's stronger ones. The key to random number generation is if you're generating any length that it truly is not clumpier, that there's let's say it's what maximum entropy, and that you really don't have any clumps. If it all clumps in one area, then that's not great randomness.

So I just want to finish because there's one other thing we're going to chat about to lay the groundwork is Bitcoin addresses. I put that up. You can look at the slides later. The details don't matter much. But the key thing is that when you hear somebody talk about public keys and Bitcoin addresses, colloquially, we all reference them the same. They're actually not.

The technology that Nakamoto did was he uses the public key. He literally hashed it twice, once with this hash function called SHA256, another hash function, then concatenates, and puts a little check sum at the end, and then uses something called a base 58 to make it even shorter. I've gone back and read some of Nakamoto's emails for the two years after he published all this and I've read other things. My understanding is the reason there is two hash functions and actually two different ones was just to make everything a bit more secure.

Also, a public key is very long. It's about 512 bits. And so you can shrink the data and make the data more compressed by hashing it, which took it to 256 bits. He hashes it twice, and then he does this base 58 and makes it even a little tighter. So for all purposes, you could go ahead and just use public key and Bitcoin address is the same. But remember back in the mind, oh, actually, they're a little different. Bitcoin addresses are a little bit more secure supposedly,

unless of course somebody has hacked into your wallet and figured out all these little details.

A Bitcoin address is a little bit like the signatures on these notes we talked about, right? Remember what an-- half of you don't use checking accounts. But these are early forms of checks. And there's a signature on the bottom. That's really kind of a Bitcoin address. I'm sorry, the signature is the digital signature. The address, the Bitcoin address is who it's paid for.

And I promise last slide. We're going to be talking about this next week. Transactions, all that stuff that rolls up into the Merkle trees. All that little itty bitty important information, they basically have an input and an output, the input and a lock time.

But the input is a previous transaction. This uniquely identifies, basically, money. And you're going to send value in Satoshis. He named the unit of count for himself. There's a lot of Satoshis in every one Bitcoin. That's why we don't hear much about Satoshis. But there's 10 to the 8th Satoshis in every one Bitcoin. So when you actually enter in the computer code in a transaction, you're doing it in Satoshis.

And it's sent to a public key. That's a coin. That is what the incentive system's all about. Any other questions? And this is just I know. There's a lot. I wonder how many of you are going to come back on Thursday.

No. Let me say this. It's not just that we're at MIT. But we are at MIT. Come on. Everybody in this room can get these kind of key concepts. The key questions that we talked about were timestamped append-only logs. Does anybody want to tell me what a-- if this class here in the next seven minutes can get these two concepts, that's all we talked about for the last hour. So I don't know your name in the orange shirt.

AUDIENCE: Andrew.

GARY GENSLER: What's that? Andrew? Andrew, what's time append-only logs?

AUDIENCE: Timestamped append-only logs is essentially a record of transactions or a block as blockchain uses it with a time. And that can't be changed in the future. So you can only add on transactions.

GARY GENSLER: So it's kind of immutable because of all this cryptography. Stuart Haber was making it in a timestamped append-only log. And he was placing it where? Carolyn, you still with me? Where

was Haber putting it?

AUDIENCE: *New York Times.*

GARY GENSLER: *New York Times.* There you go in the classified section. So it's just it's a bunch of blocks of data compressed up. So we talked about something called Merkle trees and Merkle Roots. Just think about as that's a way to take a lot of information and compress it but also make it searchable later because 1,000 transactions, when we talk next week, you have to be able to verify. Somebody asked me about how to verify, right? When you go back to verify, you need an index number to find it in that Merkle tree situation. And it's secured through hash functions. Anybody want to tell me that easiest lay definition of the hash function? Jennifer?

AUDIENCE: It's like a mapping can be so members can get to just one.

GARY GENSLER: Right. You could take a picture of this classroom and everybody exactly and they could map it into something. I don't know. Would a QR code be a form of a hash? Not cryptographically secure. But is it a hash?

AUDIENCE: It's more of a different representation of some data rather than binary you're using.

GARY GENSLER: All right, so I failed that one.

AUDIENCE: It often stores hashes.

GARY GENSLER: So cryptographic hash function is a way to take not only a lot of information and put it into a fixed form, but the key thing here is the hash functions are what tie the blocks together because hash functions can point to previous information. And as the video showed, if you change any of the underlying information, the hash changes. So what does that give you? It basically secures the data. You know if somebody has tampered. So the only reason to really learn about hash functions is it's to say, oh, I get it. This is one of the ways to make this data tamper proof. Go on.

AUDIENCE: I have a question about a theoretical event where a better hash function is found than the SHA256. How would that be implemented into the Bitcoin network practically? There needs to be a consensus and--

GARY GENSLER: So how would any relevant change be adopted into Bitcoin is always a challenge because it's a decentralized network. And all decentralized networks have a little bit of a governance

challenge. The governance challenge is, how do you do software updates?

We all know that on our laptops, our iPhones, there's probably software updates going on here now unbeknownst to me, right? They're probably just Apple has dropped. I mean, who knows what they're doing in here, right? And Uber, I really, one of my favorites, who knows what's happening inside this phone. But the commercial enterprise, the central authority has a way to update the software. We probably sign some terms of use that allows them to do that.

In a decentralized network like this, there has to be consensus. And so the only way really to update the software for a new hash function or for most everything else is, in essence, that the nodes, the operators of the software collectively in a consensus form adopt it. So it's another way that not only is the data immutable because of these hash functions but the software is. And that comes both with benefits and costs.

Some people would say that's a bug of blockchain. Some people would say it's a feature. You can come to your own judgment over the course of this semester. But the software is harder to update than software in centralized authorities because centralized authorities just say-- they just push the-- now sometimes you have to click and say update. But don't be naive. Not every software do you click. I mean, there's some that's just happening. But here, you've got to have consensus.

I know it didn't answer your question about the hash function. But if it were a hash function that had to be updated and everybody said they had to quickly update it, there's interesting debates about this, but you wouldn't need to go back over all 540,000 previous blocks. You could just hash all 540,000 blocks, 180 gigabytes to one 256 or maybe it's then a different, and then you'd have that. And it would be tamper proof.

So those are the key things. That's what we covered really. What we're going to cover next Tuesday is consensus protocol. We've talked a lot about proof of work here because everybody thinks of Bitcoin about proof of work. But we're going to talk about proof of work, the nodes, and the native currency.

And then next Thursday, we're going to talk about transactions. Again, I try to break down this technology. If you want to forget about this lecture, and you're going to go, oh my god, it was like going to the dentist, you can tell your friends that you actually know something about cryptography. It is called cryptocurrencies. So how could we not know something about cryptography?

But it's basically those three things. It's cryptography. It's a consensus mechanism and the transactions. So right? Cryptography, consensus mechanism, transactions. And we will get through it. And then you'll see this matters to finance and whether it's got any use cases. So thank you.