MIT CSAIL Alliances | Henry Corrigan Gibbs Podcast Export 3

Welcome to MIT'S *Computer Science and Artificial Intelligence Labs Alliances* podcast series. My name is Steve Lewis. I'm the Assistant Director of Global Strategic Alliances or CSAIL at MIT. In this podcast series, I will interview researchers at CSAIL to discover what they're working on, and how it will impact society.

Henry Corrigan Gibbs is an Assistant Professor in MIT'S Department of Electrical Engineering and Computer Science and is a principal investigator in MIT's CSAIL. Henry builds computer systems that provide strong security and privacy properties using ideas from cryptography, computer security, and computer systems.

Henry completed his PhD in the applied cryptography group at Stanford. After that, he was a postdoc in the research group at EPFL in Lausanne, Switzerland. For their research efforts, Henry and his collaborators have received an ACM Doctoral Dissertation Honorable Mention Award, three IACR Best Young Researcher Paper Awards, the 2016 Caspar Bowden Award for Outstanding Research and Privacy Enhancing Technologies. And in 2015, IEEE Security and Privacy Distinguished Paper Award.

Henry's work has been cited by the IETF and NIST. And his Prio system for privacy preserving telemetry data collection is used today in the Firefox web browser, Apple's iOS, and Google's Android operating system. Can you bring our listeners up to speed on the landscape of privacy research today?

Sure. So I'd say there's at least three big categories of privacy research that are going on today. The first one is what I'll call something like data stewardship. So once you've collected user data, say, you're a company or you're a government agency, how do you ensure that the way you're using this data doesn't lead to more privacy problems? So, for example, you could imagine collecting a bunch of images from users training a machine learning model on it. And then using that machine learning model on a product.

You could worry that the way that you're using that machine learning model would actually leak information about people's images. So one strand of research is looking at how do you ensure that we can avoid that leakage. Another line of research is on, essentially, protecting us from data breaches. So once a company has your data, how can you prevent attackers from getting in there and stealing it? And that's more traditional computer security work.

Hardening systems, figuring out ways to defend databases, defend web services from attack, and so on. And then the last line of work is what I'll call data protection, which is how you can get the benefits of an online service, if you're a user without ever having to give your data away to a company? So you can imagine being able to message with your friends without Facebook being able to see what messages you're writing, or maybe searching on Google or something without Google ever learning what your search query is.

And that line of work is using more advanced cryptographic techniques. So those are the three.

Let's talk about data breaches. Because certainly, that's been in the news a lot for some time now. And it directly impacts people's privacy. So tell us what your group is doing? What your approach to protecting against data breaches?

So there's two ways we're trying to attack the problem. So the first one is based on the idea that if a company doesn't have your data, they can't lose your data. So the principle is that we're trying to design computer systems in such a way that the user's data can remain on the user's device, and not ever have to be stored in the cloud in unencrypted form. So the old way of doing-- or I should say the way that people do this often today build web services, is they basically collect a bunch of user data, do something interesting with it, and then give the results back to the end user.

And what we're trying to do is figure out ways where the user can say, encrypt their data, give an encrypted copy of their data to a web service. And even though the service can't actually see what's going on what this encryption is containing, the service can still operate on it. It can still compute on this encrypted data. And return the encrypted result back to the client. So the point of that is that even if the company gets breached, they don't actually have any unencrypted data to leak.

Everything that the service is holding is an encrypted form. So we're a long way from getting to that vision. But we're building up by showing that you can do this in a small scale, and for certain types of services.

We talk about encrypted data at REST and data in transport. But when you're encrypting data, doesn't that have some type of computational overhead associated with it? And could you talk about your work on reducing that computational overhead?

Yeah. This is the big problem is that if you want to compute on encrypted data, you pay something gigantic in terms of the computational costs, say, 1000x. 10,000x, even, 100,000x in terms of compute cost. And that's really unacceptable for any real world use case. So a big part of the work that we're doing now is figuring out how we can drive down that computational cost. Some of that is using systems engineering principles. So you can imagine taking advantage of hardware acceleration, taking advantage of GPUs.

But another piece of it is figuring out how to custom design the cryptographic tools to the application that we're interested in. So if you're doing machine learning, how can you customize the cryptography so that machine learning on encrypted data is more efficient than it would be, if you were just doing general purpose things? But that is one of the big challenges is figuring out how to make this stuff go fast.

And how far away do you think you are from some breakthroughs?

So I guess there's a few ways to answer that question. I think in terms of particular applications, I think we're actually pretty close. I would, say, five years, maybe three years. So what applications are those? I think if you want to, say, perform a simple query on a database without the database server learning what your query is. So this is a problem called private information retrieval. I think we're pretty close, actually, to having systems that do this in practice at scale.

I think for more complicated applications like collecting a bunch of encrypted images from users, training, neural network on those images in encrypted form, I think we're very far from figuring out how to do that it in scale. I think basically, these techniques don't tend to work super well when the computation is complicated, or when the data set is very, very large.

Well, five years would be good for me. Can you talk about your work on SafetyPin and True2F, and how do we approach these projects?

Yeah. So these projects are looking at a different way to protect against data breaches. So the idea is that if an attacker gets into your data center, you want to set things up in such a way that the attacker has to compromise many components of the system before it can actually extract any sensitive user data.

So this project called SafetyPin, the system that we built, is a system for pin-based backup. Or you can think of it for Android devices or iPhones, where the system stores users backups in encrypted form in a data center. But the data is split across many, many machines in such a way that if the attacker wants to basically, get at this data, it's going to need to compromise hundreds or thousands of the machines in the data center.

The challenge is at the space of pins is very short. So you have a four-digit pin. It's easy to brute force. So if you just encrypted the backup with that pin, and stored it on a server, anyone who could get access to that encrypted backup could brute force the pin and decrypt the data. So our idea was to split up the ciphertext. So split up the backup even in encrypted form. Spread it out over a very large number of machines in such a way that if the attacker doesn't know what the pin is, it can't even figure out which machines it needs to compromise.

So, basically, you need to compromise a very large number of machines before you can even do anything useful in terms of brute forcing the pin. And I think that's one instance of a bigger picture, which is really figuring out how to design computer systems in a way that even if many of the components fail or get compromised, you still get some security out of the system as a whole.

And when you talk about designing the systems, but what about policies? What about some of the effects of privacy research and policies? What can businesses do to help in that regard?

Yeah. So I think one of the most interesting developments in the last few years in terms of policies, particularly as affect on business is this regulation in the EU called the General Data Protection Regulation, or GDPR, that many of you may have heard of. And the idea is that it requires companies that are collecting user data to comply with general principles about how that data should be stored. And what privacy properties you need to guarantee of your users?

And I think the interesting thing, for me, is that that's actually had a big impact on the way that US-based businesses deal with privacy. We think about privacy because this regulation applies to even to US-based businesses that operate in Europe. And I think the interesting thing or the exciting thing is that it's really forced people to think more carefully about what types of data they're collecting, how they're storing it, whether they really need to collect certain types of information and stored forever, and architecting their systems in such a way that they can make a defensible case that they're doing things in line with best practices in terms of privacy protection.

So I think regulation and policy has really driven a lot of interest in privacy. And I'm hoping that that translates into more uptake of advanced privacy technologies by industry.

Yeah. And there are some substantial fines that can be levied against a business, if that privacy is breached. If they have a data breach, I think it's up to 4% of their annual turnover or something like that. But some pundits have GDPR would say, it's really not enforceable. And we really need to have the right to be forgotten. How can we do that in a database that logs everything?

Yeah. That's a great question. These are not, I think, what the policy specifies. Or I should say there's two things going on with GDPR. One is that it's quite vague in terms of what it requires. And I think it's written to be vague. Because, I think the people who were writing the regulation knew that the technology would be changing so quickly. The regulation could never keep up. So it's not clear. I think a lot of companies are unclear on what it actually requires them to do.

And the other thing as you point out is even if it requires a certain thing, there's a question about, is that technically feasible? And I think all of this, we're still really figuring out. So you mentioned the right to be forgotten, which is the idea that you, as a user, should be able to go to a tech company and say, I want you to erase all the information that you have about me. And they should be able to do that.

And it turns out in a big computer system, data is cached and stored all over the place. And the company may not even know where all the data it has about you is stored. And I think one thing that's happening is that companies are paying more attention to this. And architecting their systems in such a way that they can handle those requests. And the other thing is that I think companies may just end up making mistakes and getting caught and having to deal with the consequences.

Yeah. I had just flown on Delta Airlines. And they're taking your picture as you're boarding the plane now. So I felt like saying to them, OK, what are you doing with my date? What are you doing with this picture? How long are you storing it? Who do I write to get it deleted? What would you like to see more attention put on by businesses with regard to privacy?

It's a challenge in the US. I think in Europe, there's more of a sense. And I think more of a legal framework that thinks about user data as the property of the person whose data it is. So your photo, when you get on the plane, would still belong to you, even though Delta had a copy of it. And you could ask them to delete it, or to tell you what they're going to do with it, and so on. In the US, my sense is that really is not the case.

So when they take your photo, it's their photo. And they get to decide how long they're going to keep it. Of course, there's limitations to that in many cases. But really, the data that, say, an ad network collects about you or a webmail provider collects about you is more or less, there's to do with what they want. One thing I think we're seeing more of. But I would like to see even more of is the idea that the user's data is really owned by that user.

And I think if we expect that of the products that we use, I hope that the companies are responsive to that. So they give you the option to say, download all the data that they have about you, or delete data that you don't want them to store, or explain to you how they're going to use your data, whether they're going to sell it, give you the option to opt out of these things, and so on. I think that's something I'd really like to see.

What do you feel is some of the misconceptions surrounding privacy technology today?

I think one misconception is that if you're encrypting data, that's good enough to protect your privacy. So a lot of people, when they visit a website, they see this HTTPS in the URL bar. They see the padlock, the green padlock. And they think, OK, I'm good. And many companies feel this way also. That if they set up a secure website for getting their customer data, that that's a very strong privacy protection for their users. And it is true that that's one thing you need to do is encrypt the data between the user and the endpoint, between the user and your server.

But that's really insufficient. And I think people need to recognize this that even if you have a very strong encrypted pipe between your user and your server, if you're storing that data on your server in unencrypted form, it's still at risk. And in particular, I think people should get used to the idea that if you're storing user data on your servers, eventually, that data will get breached. And so you just have to plan for that. And architect your system in a way that you can recover from that.

So you're saying just because the transport of data is secure, being HTTPS doesn't mean where it resides the landing place, the parking lot is. Although, I would assume most enterprises are encrypting data at REST. But you say that's not good enough.

Transport security is necessary, but it's not sufficient for privacy. And encrypting data at REST is also a fantastic thing to do. I think the problem is that because companies need to operate on that data, it can't be encrypted at REST for very long. Data encrypted at REST makes sense, say, if an employee is taking home a laptop that has a bunch of health information on it. And they're worried about their laptop getting stolen from their car.

So that's a situation in which the attacker steals the laptop. If it's encrypted, then you're feeling pretty good. Oh. you're out of laptop. But you're at least not worried about the data getting compromised. But if you have a server that's doing something interesting with user data, you can't really encrypt it on the server. Because the server needs to look at the data to do whatever it's going to do. So if an attacker gets on to that machine, compromises that server, it's going to have access to all the data.

So I think really the only protection against that is minimizing the data that you're collecting, and keeping the really sensitive stuff off of servers that are, say, exposed to the internet.

So when someone hacks into a server, they just have to break one encryption key. And then they get the everybody's data? Or is it better to have everybody's data and their repository has their own key? So there would be a million keys to break versus one key to break. Is that something that's a feasible approach?

So typically, it depends on how the system is set up. But in, say, a standard web application, you have a front end server that's basically talking to the user. And that server is talking on the backend to a database. And in many, many cases, that database has tons of different customers' data in it. And that front end server has access to data from every customer that the business has. I'm describing a simple architecture. And so what the attacker has to do is not to compromise any encryption keys necessarily.

The attacker just needs to find a bug in the code that's talking out to the user side of things. Once it finds a bug in that code, it can say, take over the process that's running the web server. And the attacker now can talk to the database and ask questions about any user's data.

So the way you defend against that is there's many ways. But you can partition the database. You can add multiple layers of security. You can auditing. You can add intrusion. You can do a bunch of things on top. But fundamentally, if your web application has access to a bunch of user data because you're going to say compute statistics over it or have users talk to each other, that data is exposed to the world.

What do you view as the most important privacy technologies in development now, or let's say in the next five or 10 years? How do we get out of this mess here?

Yeah. That's a great question. I think one of the most interesting and successful privacy technologies that I've seen of late are these encrypted messaging applications, where in very short period of time we went from all of our text messages being unencrypted essentially. For most people if you're using Signal or WhatsApp, or iMessage, your messages with your friends and family are end-to-end encrypted, which is very cool. I don't know. To me, that's a really big success of privacy technology.

And I try to look at that success and understand why it was successful. So reverse engineer what happened. What made end-to-end messaging take off in that way? And I think one of the big things is that it's a privacy technology or security technology that's completely transparent to the end user. So most people don't even know that WhatsApp is not that encrypted. They don't care. It just silently does what it's supposed to do, which is protect your data.

And I think until we get to the point where these more sophisticated privacy technologies have that property, where they're completely transparent. The user doesn't even know what's going on. I think we won't be able to get to that scale. So you ask what are some of the most promising privacy technology. I mentioned private search. So that's one thing that I've been super excited about is searching over databases without revealing your search query to the database.

And I think that's one that is both technically feasible. We're not that far away from making it work. And also, super important from a privacy perspective. Your Google search query is like a ton of information about what you're thinking about, where you're traveling to, what your political beliefs are, what health issues you're having. And so if we can protect that information I think we'll be on the way to better privacy on the web.

Where can people go to find out more about your research?

Our research group's web page, which is linked from the CSAIL home page is a good place to start. And all of our code is open source. also. So if you're interested in some of these, you check out one of our papers that you're interested in. Using some of the tools I'm playing around with it, you can definitely grab the source code too.

Cool. Well, Henry, it's been a fascinating conversation. We appreciate your time today. Thank you very much.

Yeah Thanks so much for having me.

If you're interested in learning more about the CSAIL Alliance Program and the latest research at CSAIL. Please visit our website at cap.csail.mit.edu. And listen to our podcast series on Spotify, Apple Music, or wherever you listen to your podcasts. Tune in next month for a brand new edition of the *CSAIL Alliance* podcast. And stay ahead of the curve.