EPISODE 1248

[INTRODUCTION]

[00:00:00] JM: Encryption algorithms provide the means to secure and transfer sensitive information by taking input and transforming it into an unreadable output. Usually, a special key or multiple keys are needed to unscramble the information back to the original input. These algorithms power the security of everything from our cellphone lock screens to fortune 500 company servers. The company Skiff is protecting data privacy with their first product, the only end-to-end encrypted document collaboration platform with password protected folders, expiring links and secure workplaces.

Skiff's document platform has all the traditional features of a typical document editor, making it feel familiar and comfortable. There end-to-end encryption and built in password protection are two of several methods that make collaborating on documents more safe and within user control than on any other platform. In this episode, we talk with Andrew Milich, CEO of Skiff. Andrew was previously an Associate Product Manager at Schmidt Futures. We discuss data privacy and security, the Skiff document collaboration platform, and potential future security products.

[INTERVIEW]

[00:01:03] JM: Andrew, welcome to the show.

[00:01:04] AM: Thanks so much for having me.

[00:01:06] JM: You are working on a privacy-focused document collaboration platform. And when I think about privacy focus documents, it seems like Google Docs does just fine. I mean, if I make a new Google Doc, I don't share it with the world. It's private. Right? What is not private about that?

[00:01:26] AM: Yeah, that's a really good question. And you're coming to the right people for that. So I'm one of the creators of Skiff, which is a collaboration platform, a bit like Google Docs designed to be private from the ground up. So even if you don't share Google Docs, you're still effectively working on a platform that is not private to the technology provider, and possibly people listening to your network or who have access to data through the provider as well. And so we've seen this migration to use end-to-

end encrypted messaging in Signal, in WhatsApp, in effectively all the communication apps we use today. And so Skiff is rebuilding documents and collaboration around that same principle to kind of give you absolute and total privacy around the way you work and the things you write and share.

[00:02:11] JM: So in the Google Docs world, if I'm writing to a document, is that vulnerable to security holes?

[00:02:20] AM: It's a great question. I think Google Docs built out a lot of the collaboration apps that we kind of now use every day for work and everything else. So it's a great product. But it's just like, now we send text messages. And we don't really want Apple or Android and Google or Facebook to read our messages, sequence them, use them in their models, possibly leave them vulnerable to other governments and authorities and other people to access. And so we've switched to using end-to-end encryption. And so, from the places I've worked and the people I've worked with, we really want that to be the way everything is by default. So not just secure in the network connection and things, but completely private only to you and the people you work with. Does that make sense?

[00:03:03] JM: Of course. So what is the actual end user product of Skiff?

[00:03:09] AM: Yeah. So it's a coming about right now. And we've just started onboarding users to our public beta. So it's effectively a collaboration platform. So you can write documents, you can share them, you can share links to them. And then you can kind of build out your own personal workspace as well. So in sub-pages, in tables, and adding all sorts of little widgets and things to your documents too. And so it's designed to kind of be that private workspace that you can use to collaborate with other people. And privacy is not just in the platform itself. It's also in the features we built on top of it and the icons we use and the copy we use in the product, and kind of all throughout pervasively inside Skiff. So that's the user-facing element right now. But I also know you talk a lot about software engineering and the backend, and that's where kind of a lot of the innovation and stuff really came about.

[00:03:59] JM: So I guess, table stakes is the document collaboration stuff that you would expect from Google Docs, like just a word processor basically with commenting and all that kind of stuff. But the real heavy lifting that you're doing, the real differentiated part of what you've built is this end-to-end encryption system. Is that right?

[00:04:24] AM: Absolutely. Yeah. So, we have commenting, we have all the kind of nice flashy Word processor stuff. But I feel like in the world few years ago, and even today, there's been this narrative that you can't have really beautifully designed, performant, easy to use security products. And we spend an enormous amount of time and energy on features and design and how the product feels and making it not just table stakes, like you're saying, but also leaning into privacy and leaning into all these new collaboration platforms that have introduced newer features. So I don't know what comes to mind when you think of using a collaboration platform or features that you'd love to see in a product like this.

[00:05:05] JM: Are you asking me?

[00:05:07] AM: I'd love to know. I think we hear from a lot of people using different products. And I'm sure you've seen all them at this point.

[00:05:13] JM: Well, I have only ever really used Google Docs. I have not gotten on like the notion train. I never used whatever – What was the one that was acquired by Salesforce? Quip, I think? So Google Docs has always seemed like the best to me. But I mean, I don't know what else there is. I just need bulleted lists and font sizes and stuff.

[00:05:39] AM: I'll be honest, the engineering of bulleted lists is actually much trickier than I had imagined, where every single tab issue that you've found in a real product comes down to lines of code that have a bug in them. So don't underestimate the bulleted lists.

[00:05:56] JM: Gotcha. Well, as I understand the intent encryption part is actually pretty tricky, because there's a lot of data in in a doc, and every change that you would make to the document to send to the server, it could actually be – There's a lot of data that you would have to send over the wire. And so encryption can get kind of tricky. Can you talk about why it is somewhat work intensive to encrypt a document over the wire like that?

[00:06:27] AM: Absolutely. So effectively, when you're doing end-to-end encryption, it means that the centralized server and the technology provider can know nothing about what you're working on and what you're writing. So, in the traditional model, the Google Docs model, the server can merge and kind of save every single keystroke. With our model, that's not possible. We don't even know the keystrokes you're typing. So what happens instead is basically the document is decentralized when you're working

on it. You have a copy in your browser. Your collaborator has a copy in their browser. And every time you press a key, you encrypt information about the key you pressed, send it to your collaborator, who then is the only person who can decrypt it and applies it to their copy of the document.

And you're right, that is pretty sophisticated stuff that has only really been coming about into scalable apps in the last couple years. And so if you check out our white paper and want to chat about the technology there, it's using some stuff called CRDT's and replicated data types that live in multiple people's browsers, but can store the same copy of a data structure decentralized. So you're right. And it's kind of just nascent stuff coming about, but that's the way you have to do it if it's end-to-end encrypted.

[00:07:39] JM: Why can't Google Docs just bolt this on?

[00:07:44] AM: I love talking about that. So, I'll say there're two things I think about here. One is it's really hard, like you said, to build end-to-end encrypted products. So if you've already built something, and Google Drive, I think has about a billion users, you've you built something that has a billion users in exabytes or more of their data, effectively rewriting that and turning it inside out to make all of that data private to you is an enormous engineering lift. It's like I gave you a book for a holiday and then said, "I want you to share that book with everyone. But you can no longer know the contents of that." And so it takes a lot of enormous refactoring and redesigned to do something like that. And I think actually, Facebook Messenger is trying to move to total end-to-end encryption, but it has taken three year may take more years. So three or more years just to rebuild the messaging product.

That, I think, really stands for how tough it is to build end-to-end encrypted products. But then we've also seen smaller and encrypted products like Signal become mainstream, especially in the last few months. And so it's really exciting to see that this is another way that new technology and new kind of creators can build products that more people can use and not just incumbents kind of rewriting their existing products.

[00:08:58] JM: Can you tell me a little bit more about what kinds of novel encryption protocols you've had to develop?

[00:09:06] AM: Absolutely. So my thoughts here are mostly that we're working on the usability and scalability of end-to-end encrypted products. And so we're using a lot of tried and true cryptography that is pervasive in WhatsApp signal, other encrypted messengers and other products, but building it to be scalable for collaboration. And so examples of that are we've built end-to-end encrypted link sharing, where if you turn on link sharing in your browser, it effectively generates a key in your browser in private that gets stuck in a document link that you can then send to your collaborators. And so that link is not known by us. It's not known by Skiff, but allows you to share an end-to-end encrypted file seamlessly.

A couple more things that we've really tried to focus on are building scalability and performance into the basis of what we've created. So our platform really doesn't care about documents that much. It's pretty agnostic to what you're working on. So we have prototypes of sheets and calendars and other stuff that we're prototyping, but really built around the same private document model. So I think those are two of the key properties. The last is just scalability. It's expensive to encrypt and decrypt data all the time. And there are places where we do that lazily or changing an encryption key when you access a file, instead of on all the files all the time. And so that lets you create workspaces with a lot of people and more files that really kind of satisfy the same performance requirements of great products like Google Docs.

[00:10:44] JM: We've done some shows about distributed document systems. And you've already mentioned CRDT's. Can you talk about the difference between operational transforms and CRDTs?

[00:11:02] AM: Absolutely. I love both piece of technology. So, operational transforms are the kind of birthplace of collaborative tools on the web. And they're effectively done by letting people express changes to a document as operations. So like a key press is an operation or a deletion is an operation. In the kind of other model that you described of CRDT's are conflict-free replicated data types. We store changes to a document as, let's say, time-sequenced events. And everyone main maintains their own sequence of all the events that have happened to a document and can replay that and add other people's events and arrive at a single source of truth on the document.

So let's now just dig into the operational transforms a bit and then the CRDT's a bit. With operational transforms, you're expressing operations to a document that can be composed and combined with other operations. So two people type the letter A and the letter B on different computers around the world. A server sequences and absorbs both of those operations and says, "Oh, this operation at this

point in the document plus this operation compose together means I should append A or B," or depending on where you typed, maybe B or A.

In the completely different model of CRDT's, both users around the world typing A and B append an operation to their own copy of a document. So instead of sending it to a server, they append the A or the B to their version of the document with a timestamp and then send that operation to each other. So user one gets user two's B letter that they've added, and the other user gets the letter A. And based on that timestamp, they can each arrive at the same version of the document. So in that first model of operational transforms, we need the server to resolve conflicts and compose these two operations. In the CRDT model, both users can do it separately, which is awesome. It's more private. It's more scalable. And it's this awesome piece of technology that we're now seeing built into real products.

[00:13:08] JM: So can you take me through what is happening as I am typing on a document that is open on my computer and somebody else's computer? Like how is the data being propagated?

[00:13:22] AM: Absolutely. So let's start at the key press. So you type the letter A in the document on your computer. Inside your web browser, we have a version of that document that converts the letter A into a change to the document. And so that'll store where you typed it, what time you typed it, and maybe the current copy your version number of your document. So now that little package of information is encrypted with the documents encryption key. That gets then actually directly sent out to everyone else who is online at the document at the same time. So let's say you have two collaborators also on that document in their web browsers. You take that little packet of information, the letter A, the time you typed it, the version of the document, and you send that to the other two users.

Now they decrypt that little package. They now know exactly what you knew when you typed the letter A, which is the letter you typed, the time you typed it and the version. So they decrypted that information. And they then merge it into each of their copies of the document. And using that timestamp and the version that you had, it means everyone ends up at the same copy of the document. And you can do that basically decentralized, where you're all talking to each other, and private, which is through using that encryption key. Does that make sense?

[00:14:45] JM: What are the biggest engineering challenges that you've run into so far?

[00:14:50] AM: I'd say the largest theme here is nothing is impossible with end-to-end encryption, but it's typically harder. So link sharing is probably the best example of this. Our link sharing model works exactly like Google Docs. I copy and paste a link and you're good to go, edit, save, comment, etc. But behind the scenes, it's doing really fancy cryptography where there's an encryption key in the link itself. And when you load up the link, you take that encryption key and you decrypt the document. And then every time you type a key, you're encrypting and decrypting other people's changes.

So a lot of the user experience feels exactly the same as a normal product, but it's a lot more sophisticated engineering-wise. So to highlight kind of the most serious ones, link sharing is really up there where you want to build the most simple and intuitive interface, but with a lot of cryptography behind it. The other is actually search. And this comes up quite a bit. But when you're typing in Google Docs, or in Google Drive, Google servers will index every word, every letter you type into your personalized search index. And so then when you go back and you search apples inside your Google Drive, that query goes up to Google, they find all your documents, and they've kind of pre-indexed that for you, and it comes right back to you. We can't do any of that. We don't know what's in your documents. We don't even know the titles, or most of the time, the last time you edited them. So we have a lot more work on our hands to build search. And there are ways we can do it. We can actually index your documents as you're typing in them in your browser. And so it stays private to us. But it's still a lot more complicated and it takes a little more engineering and design to build a feature like that than if we weren't worried about privacy or encryption. So I think the general theme is end-to-end encryption is hard. But it honestly requires the type of deliberate design you would need to protect people's privacy. And we care a lot about that.

[00:16:51] JM: How does your problem set compared to end-to-end encrypted services like WhatsApp or Signal?

[00:16:59] AM: Really interesting question. And I highly encourage checking out our white paper. I'll say the boil down technical model is imagine taking an encrypted messaging app, WhatsApp or Signal, and effectively crashing it into a password manager, which is really good at storing secrets in your web browser. And you take the password manager, the lessons from a password manager, which can store all your login information and your personal secrets, and then use that to share information about documents, like you can do in a group message in WhatsApp or Signal. So when you log into our

platform, you have a place to store your personal encryption keys and your documents and document keys. And that's all for you kind of like a password manager.

But when you need to share things, you need a way to send information to people securely and privately. And so Signal on WhatsApp, using the Signal protocol and elements of public key cryptography, let you share encryption keys with other people privately. And so we need that part of the group messaging and communication app combined with kind of storing information about you that only you should access. Does that makes sense too?

[00:18:12] JM: What else do you cover in your white paper that I might not think to ask you?

[00:18:17] AM: Well, it's really the surface level stuff that's important to us, which is design, usability and performance. So if there are common standards now of having responsive queries in under 100, or 200 milliseconds. So when you make a new document, which for us means making a new encryption key, encrypting some data, sending that to the server, we want that to be done performantly in a way where you feel like you're not using some difficult or painful security app. That's true of a lot of the features we've built on top of encryption as well. So expiring access to documents, where you can set a time for someone's viewing access to expire and they can no longer see it. Or adding in watermarks or dynamic watermarks to documents where you may be worried about screenshots for the devices they're worried on, or just letting you have more control over the information and how you share it.

So I know it's a little less technically flashy than end-to-end encryption and end-to-end encrypted link sharing. But I think we're seeing privacy products turn this corner on usability and design. And that's really important to us. And the last thing I'll say is a lot of that's also open source. So a lot of the modules we've written here and the client code and the UI library and a lot of that is open source and on our GitHub. And so we're trying to push that forward too.

[00:19:42] JM: So you mentioned the bulleted list difficulties. I'm wondering just what else is difficult about creating simply a document sharing system.

[00:19:56] AM: Documents are tricky, because we have such a high bar for the collaboration platforms and writing platforms we use today. So I'll probably split it up into in your question about the document editor itself, let's say the editor framework, and then collaboration. And we have in the last few years a

lot of wonderful editor frameworks that are open source and have great communities around them. So we use an editor right now called Prosemirror. And we've also used editors called Slate, and Quill, and Draft. And all of these are built by individuals or teams or companies and open source so you can basically build editors right into your product. But I'll say there's a difference between a basic editor that has bold, underline and italics, and then building in fonts and spacing and alignment, and all those bullet points little bugs and colors, and all this sort of stuff. Just because representing data in a text editor is really complicated. So we typically store data as an HTML page or the JSON format, where it's kind of cataloguing different nodes in a document. But with fonts and underline and all this stuff, you just have to keep track of so much information for each unit of the document. That means things can get messy or conflicts. And that's where a lot of bugs come about. So that's complicated.

The second area, which I think you kind of covered really clearly in your questions is the collaboration models we choose. And there's a bit of a divergence between some editors and communities that build operational transforms and others that build CRDT's into their frameworks, and choosing the right one, setting it up, and making it performance is tricky as well.

[00:21:45] JM: Do you plan to try to make the full document suite like presentations and spreadsheets also?

[00:21:55] AM: We're excited about it. I think we're seeing in a lot of new productivity products a somewhat simpler, horizontal, or horizontally integrated platform people use. So you can build tables and integrations directly into documents, or into simple versions of sheets, where people can customize their own integrations and hooks on top of that. I mean, that's really exciting from an engineering perspective. Frankly, we could build all the features of the Google products in this space or in the desktop products we use. But it takes an enormous amount of time. And just like building bulleted lists is a challenge. Building tables into a document and fonts and all of that is time consuming and difficult. And so dragging table edges and adding rows and columns, and formatting borders, all of that is just hundreds or 1000s of lines of code per feature. And when they start to interplay, things get complicated. So we're ambitious. We want to build it all because they can't build end-to-end encryption, and we can build their document editors. So we want to do that. But it takes time.

[00:23:12] JM: Who's the target market?

[00:23:15] AM: We're focusing on in our early launch, I'd say two different personas. One is a lot of people who have security as a mindset in the work they do and the professional career they're living in. And that means a lot of researchers, journalists, private practice, individuals who run private practices, and those people are all around the world, or they kind of communicate and work globally. And I spent a lot of time before starting Skiff working on research in that category and around people where security is existential. And they need and use secure products. And they're already using, let's just say encrypted messaging to share documents and encrypted email to conduct interviews. And so having a platform like ours means they can work more efficiently and more freely.

The other persona that's interesting is the growing privacy enthusiast and kind of privacy vegetarian type. And I would also put myself in that category, where we choose to use private products because they are safer, they're better designed, and they're more trustworthy. And it takes a movement of people to build that community. But it's been awesome to see that grow on many subreddits, many Discord channels, Slacks and other communities where people choose and try to support privacy-first products. And they have been a wonderful group of early adopters and people we interview, research, communicate with about the product. So those are the two early personas.

One quick sentence down the line, there are a lot of times in the professional world where security is also existential. So the C-suites of technology companies, and we work with a few of them, communicating about board documents, and board committees and strategic decisions they're making. They already use products with some of the superficial security that we've built in, watermarks, expiration, etc. But these products should be end-to-end encrypted and they should be well designed and built to the standard of modern privacy. And that's where I see us going in the future as well.

[00:25:33] JM: Back to the protocol of making updates to the collaborative document. Do you have the protocol basically worked out and completely implemented or are there improvements that you still need to make to your collaboration protocol?

[00:25:53] AM: We're definitely always improving it. And there's a lot of new fields and research in both the spaces we're in, collaboration and security. So end-to-end encryption becoming more efficient and scalable and also CRDT's becoming more efficient and scalable means that we're always improving it. But we're onboarding users to the platform. We have a great community built around it right now. And I could log on and in 20 seconds send you a link that you could just click on and edit an end-to-end

encrypted document in real time, from your phone, from your computer, from your tablet. So it's working and the product is built. But we definitely see a long road ahead, which is exciting.

[00:26:41] JM: And what kinds of improvements Do you think could be made?

[00:26:44] AM: Totally. That's interesting. On the security side, there are some new protocols being developed around some new security properties and efficient group messaging. So there's this protocol called the MLS protocol, I think message layer security, that's designed to efficiently – So I think in like logarithmic time, manage the encryption keys around group chats. And so that's something that would have quite a bit of application to our product where we have a lot of encryption keys for a lot of documents and managing them efficiently as you share or unshare people is essential.

On the collaboration side, I will say I love CRDT's, but they are also complicated performance-wise. So because they're storing those timestamps of when everyone typed into a document, they can end up accumulating a lot of old information. So they might have a few timestamps from a day ago or a week ago. Some large paragraph or chunk was deleted, or a large section was copy and pasted into the document. And I hope, and we're active in those communities, we can slim that information down significantly. So you can collect a lot of the garbage still floating in that document from when changes were made in times past.

A lot of that is there for a good reason. We want to be able to merge changes from people with all different copies of the document. But it means also that you have to be careful performance-wise, where if you're holding a document that has basically previous versions of itself stored internally, you can hit a storage problem pretty quickly.

[00:28:24] JM: When you were writing the white paper and implementing the protocol, did you get it vetted by like distributed systems experts? Or did you run it by professors or anything?

[00:28:38] AM: Absolutely. And the white paper on the site is many versions from what we originally wrote. So we, I guess, both in the platform and the white paper, we've gone through many third-party security audits. So we're about a year old. And I think we've done three to four at this point. And we have more scheduled for the upcoming months. And those are meticulous and quite they're out. So they read every line of code you've written, compare it against the white paper and the properties you

want to promise your product. Those are enormously helpful. And on the more cryptography and design side, we work with a couple of fantastic advisors on that front, from some senior people at the Signal Foundation, to cryptography professors at Stanford, to some graduate students have been a little more active in developing the protocols with us. So kind of all of that group plus our community of users constantly getting feedback, and it's truly an iterative product that the white paper today is not the white paper six months ago. So definitely, and that's essential for any product in this space. We want people to know what they're using and know what the properties are because threat model really matters for a lot of users.

[00:30:01] JM: What are the other features around security and privacy that you'd like to build into Skiff?

[00:30:07] AM: I'm really excited about this. We're starting to see differential privacy in data sets and data sharing become important to not just regulation, but how companies want to share their data. And so not just building an end-to-end encryption and privacy in that respect, but building all of the dimensions of privacy into a workspace. And so when you think of using private products, they're not just encrypted, but they have all the future is around controlling access responsibly and efficiently. So I think differential privacy around data sets is really cool.

I think we're starting to see the rise of some of the – Or actually research-wise, some of the watermarking and protection in that sense, and how do you – Even when you share things, how can you make sure they're still yours and still data that you can retain some amount of control over? So I'm excited about that.

I think the last is advancing this area of products on a protocol level as well. And so like I was saying, you're updating encryption models and collaboration models. I think we're starting to see those be built into platforms from the ground up, too. And so like the rise of Filecoin and IPFS as a storage method for building distributed products, and this new message layer security product or protocol being built to build more products like ours. So I'm really excited to build that into our platform, to really just become a privacy platform in the features, the icons, the copy, the way you write documents, and then also, of course, the end-to-end encryption as the basis of it.

[00:31:52] JM: And of all those things, what is the next – Are you focused on building any of those products next year or are you completely focused on the document product right now?

[00:32:01] AM: Actually, we certainly are. And we're currently – We've built an experimental version of Sheets, just because I think we're really excited about all those features, differential privacy, and most importantly, a lot of users ask for privacy focused spreadsheets and end-to-end encrypted spreadsheets. So we're hard at work on that. I think beyond that, we want to expand it to a bigger ecosystem of products like this. And once you – Every ounce of work that goes into the platform, the better all the products can get. And so we are hard at work on Sheets now, but more in the future, too.

[00:32:40] JM: Are you surprised that no company has done this before? Hasn't done end-to-end encryption with document collaboration?

[00:32:47] AM: The answers are yes and no. Surprised because big companies have fantastic engineering teams and a lot of money to spend on products like this. Also not surprised because it's a big engineering lift. Like I said, you'd have to kind of turn an existing product inside out to make it end-to-end encrypted. And so from that perspective, big companies and incumbents will struggle to do that.

The last area here is end-to-end encryption is also a philosophy, and you have to believe in privacy to do that. And I think we've seen a lot of ambivalence from large, large technology companies on how deeply they want to integrate privacy into their platforms. And so for us, we're that's the existential reason why we exist to build privacy products. It's a natural choice. When you are managing millions to billions of users, stakeholders and governments and big companies and all that sort of stuff, choosing end-to-end encryption is a lot more difficult, because it really means choosing privacy and half-baking that doesn't make sense, technically, procedurally, ethically. And we'll see. I do hope big technology companies start to make their products more private. But it's a huge effort ideologically and technically.

[00:34:18] JM: You've worked in a variety of domains prior to starting Skiff. You've worked in autonomous cars and SpaceX. Why did you choose document collaboration of all the things you could work on?

[00:34:34] AM: It actually was quite a natural transition. And I feel at home building this stuff. For me, it's really been about human safety and security as the deepest engineering calling. And you're right, I

got to work on the Dragon 2 human capsule at SpaceX and on autonomous robo taxis at Zoox. And in both cases, you're writing software where every line is tested, every line is unit tested. You are simulating 1000s to millions of events that could happen to the vehicle or capsule, all to protect human safety. And so it was wonderful working with users for both of those products. People who would ride in a taxi, or astronauts who would live in this capsule. And for end-to-end encrypted collaboration, it feels just as existential, where I know what it's like to think about the security of the products you use, and where the information you're writing or communicating is going, and what that means to human life.

And so we apply the same engineering practices of extensive unit testing, all sorts of simulations and integration tests. And then also just speaking with our users a lot, and understanding their threat models and how they work and who they work with and where they work and what their concerns are. And it's just the most fun engineering environment I could imagine. I think I love the people who work on this stuff, who work with us, and just love the rigor and the intensity of thinking about human safety in the products you built. And that is really addictive.

[00:36:16] JM: Cool. Well, Andrew, is there anything else you'd like to discuss before we close off?

[00:36:24] AM: I think that's it. I'm so grateful to be here. And it's exciting to see, not just us, but a lot of engineers start to care about privacy a lot and the world's thinking about private products. And so thank you so much for having me.

[00:36:40] JM: Absolutely. Thanks for come on the show.

[END]