Celine:

Welcome and thank you for joining us today. There are several registering and still signing in, so we'll pause for the next minute or so to allow everyone to get settled. Thank you for your patience.

Celine:

Good morning, and thanks for joining us for today's broadcast "Homomorphic Encryption for Your Business: Fact, Fiction, and Hype." A couple of housekeeping items first. Today's broadcast is being recorded, and all attendees are in listen mode only. During this broadcast, we'll be taking questions, and you can post your questions using the question and answer module in the communication panel. Your questions will be answered during the Q&A session at the end, and if we do not get to your question, we will follow up with you by email. Following today's broadcast, an email with a link to the recording will be sent to you for future viewing and to share with your colleagues. You can also download the Baffle solution brief or request a free demo under the attachments and links section. With that, I'd like to hand the mic over our presenter today, the VP of products at baffled Harold Byun.

Harold Byun:

Thanks, Celine. Thank you all for joining today. I'm looking forward to presenting our overview of homomorphic encryption, as well as what we'll call homomorphic-like encryption capabilities and where we believe that sits in the market. I welcome your questions and feedback. We'll also be showing you a live demo in this presentation as well. So, quick overview in terms of the agenda. We're going to provide a brief overview of homomorphic encryption, some of the key business and security benefits and challenges that we think that that method or methodology and approach helps address, what are some alternate methods of homomorphic-like encryption functionality, we'll then go through the live demo with a technical walkthrough in terms of some of the underlying workings or inner workings of how technologies types of technologies can work, cover some use cases, and open up for Q&A. We're not going to completely PowerPoint you to death. There is the demo as well, and we'll hopefully get out of here within 30 to 40 minutes, including the Q&A. So, maybe even shorter.

Harold Byun:

Again, welcome your questions, but we'll kick it off here in terms of what is homomorphic encryption. There's some common definitions out there in terms of the capability. The general gist of it, as many of you may already be aware based on your interest in this session, is that homomorphic encryption really allows for computation on encrypted data without decrypting the values and without keys. There're other definitions, obviously in the above. But the ability to convert data into ciphertext and analyze it as if it was in the clear. Effectively, also the underpinning parts of this based on Craig Gentry's research from the way back, I think starting out in the '70s and through the '90s at IBM, in terms of being able to perform mathematical computation on encrypted data values.

Harold Byun:

If you really look at what this might look like in a simplistic form, if somebody were to ask the question on encrypted values, totaling up expenses or tallying votes or running aggregate analysis or other types of computation, you could yield an answer either in the clear or in encrypted form that could then be consumed or passed by an application or another system. So, this is kind of the general gist of what homomorphic encryption might entail. Before we actually jump into some of these benefits, actually we did kick into this. I wanted to just go into the quick first poll that we have, which is in terms of where encryption is today do you think homomorphic encryption viable today? Do you think it's complete BS? This is basically something where you guys can start voting and just to get a sense of where people think this capability is in the industry today. We'll get into kind of some of the challenges and what's viable and what may not be so viable.

Harold Byun:

I'll just keep this open for a couple more seconds. We have people participating. Most people have no idea. Okay. Well, I think we'll walk through that and we'll see where we go from there.

Harold Byun:

What I wanted to cover today was actually what are the key benefits? Obviously, there's three major high-level areas where we think there are benefits for homomorphic encryption capabilities. Data privacy and compliance is an obvious one. Threat modeling in terms of the attack vectors that we're seeing today. Then, there's some alternate capabilities that this opens up for business acceleration and enablement. Some different information management models that may be of interest to folks. So, we'll start here with the data privacy and compliance area.

Harold Byun:

I don't think it'll be any surprise. I know everybody's heard of GDPR at this point, and I think the big thing that people have taken away from that is the 4% of revenue penalty that may be driven through GDPR compliance. But, what may also be of interest is that California also recently passed a Consumer Privacy Act, which you could argue is in some ways kind of a mini GDPR for the state of California, and it's expected that anywhere from 30 to 40 states in the US will pass similar requirements over the next couple of years. Singapore also has its own Personal Data Protection Act. The point being that the privacy standards that are governing data are only going to continue to see increased regulation as we go forward. So, what does that really mean in terms of how you choose to protect data? Are there mechanisms that you can enable to better protect data while still having your business be able to function and perform analysis?

Harold Byun:

One of the notions that homomorphic encryption actually opens up is this notion of privacy preserving encryption, and this is a notion where people can perform analytics, start building machine learning baselines or algorithm on opaque datasets, but still derive intelligence out of the underlying dataset. This example on the left is known as differential privacy. This is an example of how cloud AI or data analytics could be enabled without violating the underlying privacy or confidentiality compliance contract that you have on the data. The sample underneath it is an emoji analysis on frequency of usage and favorites. That's based on a publication or white paper from Apple. It's just an example of how you can, in theory, collect data without actually violating the underlying privacy contract and exposing roads, and that's more or less some of the promise of these capabilities like homomorphic encryption.

Harold Byun:

Labeled homomorphic encryption is another approach. This came out from Barbosa, Catalano, and Fiore. It's an anonymization of data scheme, particularly target around genome, but the gist of this is largely the same. You have multiple data providers that are funneling data into a data store (i.e. the cloud), and that a querying party could enact queries against that data set, run or derive intelligence, and derive results without ever actually seeing the underlying rows. Probably very relevant in today's world. One of the more recent examples where this would come into play is the Cambridge Analytica and Facebook scenario, where obviously there was harvesting of personal information, so really, what does that represent for big data? While there may be a lot of people that are suggesting that we stop collecting data, the reality is that many of us don't think that that is a reality and much of our personal data is already out there. The alternate question is how do you securely enable AI and aggregate analytics? This is one of the promises of the homomorphic-like capability.

Harold Byun:

Another scenario where we focus on threat modeling. Why do we focus on threat modeling and increased security around data? It comes as no surprise that the breaches continue to happen, and I think that there's been a lot of noise in the industry on data breaches. I think there's, quite frankly, data breach fatigue. The reason that these are referenced in particular is because two of them actually also represent insider threat breaches in the case of Anthem and the whole Elon Musk data exfiltration from Tesla that happened earlier this year. So, this notion of increased insider threat and privileged access monitoring and management has continued to rise to the forefront, and when you look at the threat model around data access, if you utilize a traditional encryption mechanism, as indicated on this slide, you can really see that a privileged user or a database admin effectively, with traditional encryption methods, basically gets access to the data and the clear. They can do a basic select statements. They can do a memory dump. An advanced attacker could certainly scrape memory, and they're going to get access to likely some prize crown jewels in your organization.

Harold Byun:

Why do they existing systems fail? Encryption at-rest. Well, our running joke around encryption at-rest here is that it's designed to protect against Tom Cruise dropping in from the ceiling of the data center. It's not how people steal data today, and in the case of transparent data encryption or database encryption or tablespace encryption, those are all containerized mechanisms for the database layer.

Harold Byun:

While they do provide a stepped up separation from the OS, again, any privileged user with access to the database can get access to all the data in the clear. The latest analyst and industry reports, generally depending on who you listen to, estimate roughly a 35 to 60% cause of breach via this insider threat risk. So, we really feel that these advanced encryption modes, whether they be homomorphic or alternate methods, there's a lot of progress being made in the homomorphic space, as well as in the in use or in process and in memory data protection space that can help mitigate a lot of these threats. It's also projected that a lot of organizations, especially in financial services, are also projecting by 2020 things like transparent data encryption will no longer be acceptable mitigation methods to drive compliance for data.

Harold Byun:

The last scenario that we wanted to talk from a key benefits or key use case areas is really around business enablement. When we look at business enablement, there's some alternate scenarios that we can enable using homomorphic encryption, cloud workload protection, being able to lift and shift increased numbers of workload to cloud where certain organizations may have said, "None of this stuff will ever live in the cloud." Being able to utilize an advanced encryption scheme would effectively mitigate a lot of the risks in terms of where the workload actually lives and potentially facilitate a much more nimble and flexible infrastructure management strategy. The privacy preserving encryption mode, or a cloud and aggregate analytics and AI that we talked about previously in the data privacy and compliance scenario, is another key use case for business to others that may be less obvious.

Harold Byun:

I'll touch on the third bullet shortly. So I'll skip over that right now, but automation and orchestration is one in particular, as organizations are trying to move faster. In this world, it seems like everybody is trying to move faster. There is a drive and a strong desire to automate as much as we can, and the challenge with orchestration and workflows or looking up certain values and triggering other downstream actions is that a if the data values are encrypted, the workflows, obviously, fundamentally break. With a homomorphic or advanced encryption strategy, you can effectively preserve the functionality of those machine-to-machine communications or API-based cloud-to-cloud communication models to enable automation without actually unveiling or revealing your data is something that this would support. Information supply chain consolidation is another example where there's some business value. In particular in that model, what we really look at is, as many of you may also be aware, third-party companies or third-party vendors typically introduce an increased data risk to your organization or an increased attack surface for your organization.

Harold Byun:

A model that actually enabled information supply chain consolidation would actually facilitate a method where rather than spending four to eight to 10 weeks or six months vetting vendors through questionnaires, and then determining that they were a viable vendor and then handing them your data, an alternate mechanism might be that you consolidate your data structure and footprint and have your vendors come talk to you. This is a little bit of an inverted model.

Harold Byun:

Obviously, the scenario that I described before is very common in terms of how people vet vendors today in the industry, but an alternate model could be you can consolidate your data footprint, control the access that vendors one, two, three to etc., get to your data and the analysis that they can run. You're still getting the benefits of sourcing out specific operations from your organization, but you're not running extracts. You're not batching things and farming them out to third parties so that your data is distributed all over the place. It's a different approach in terms of how you might look at what this might enable from your business, and also reduce some of the data security risks that you're carrying today.

Harold Byun:

In terms of some of what are the key challenges encryption, the biggest one that's been cited has been performance. Initially when the first kind of homomorphic encryption proofs came out, it was estimated that it'd be 100 trillion times slower. This is obviously a number of years ago. There's been some significant improvements since that time, but the latest statistic that I think I heard was something in the order of 50,000 end-to-end transactions being able to be performed in a range of time. In today's world, 50,000 transactions just isn't really very much, so I still think that there's quite a bit to go in terms of the performance that's required to power a homomorphic encryption method.

Harold Byun:

The second kind of challenge with this type of methodology is that it requires application modification. You need to understand in order to execute on a query on homomorphic encryption, you need to have prior knowledge of what type of query is going to be executed, whether that be an additive or summation or multiplication or a different computational operation, which kind of limits how you would retrofit this into your application scheme. Obviously, if something is incredibly repetitive from an operational state, then it introduces less overhead, but in a scenario where things are more freeform or where a user may be driving certain types of queries or an application may execute things that are less predictable in nature, it's going to require a significant application modification in order to retrofit and make this a viable strategy for your organization.

Harold Byun:

The third piece is really around encryption entropy or strength. There are some questions. Obviously, the encryption is exposing some malleable properties in terms of how you're actually achieving this mechanism without decrypting the data and without access to keys, and so there are still open questions around the overall encryption strength using a scheme like this.

Harold Byun:

Fact, fiction, or hype. Homomorphic encryption is ready and available today. We're in the vein where we believe that this is still a bit of fiction and a bit of hype. There are some folks that are starting to do things around differential privacy, and privacy preservation techniques are claiming fully homomorphic encryption solutions. We still think that this is a long way off from a real world scenario, but there is a fair amount of progress there. Those business benefits that we described previously, as well as the security mitigation methods that we described previously, still apply to this approach today.

Harold Byun:

What are some alternatives? From our perspective, there are other alternatives in the market that deliver what we would call homomorphic-like encryption capabilities or sometimes we'll say it's homomorphic-like but without the homomorphic part. What we mean by that is the ability to support mathematical computation on encrypted data but with a more performance model without changing the application without introducing breakage to the application and also using industry standard encryption. When we first started talking about this as an organization a number of years back when Baffle was founded, people thought the idea was crazy and impossible. People still think it's crazy and impossible, but we've had some significant progress, and actually, just recently were also awarded a patent on the approach that we've taken that allows for opaque or untrusted computation on encrypted data.

Harold Byun:

I know it sounds impossible, so we think that a picture is worth a thousand words. We're going to jump into a live demo here that will ideally demonstrate some of these capabilities and show you what kind of functionality you could expect from an alternate approach.

Harold Byun:

Let me just share the screen here. [inaudible 00:21:45] Share. Okay. You should see a standard database work bench tool here. In this case, this is a live data structure. You'll see that I just refreshed it. I'm not sure how this is rendering on your screen, but hopefully you can see these columns. These are traditional PII columns that somebody might deem sensitive in nature, and if I go into one of the column or values, you can see that the data is rendered as garbage because it's encrypted using an industry standard AES. Now what some of these alternate capabilities are able to enable is the ability to access the data through an application. In this case, we're using a commercial off-the-shelf application called Tableau to access that same data set, and what you can probably see is that I'm sorting on certain columns, which is an operation on the encrypted data that is fundamentally a compare operation. In addition to that, we can still also support aggregation and visual rendering of the data on an encrypted dataset.

Harold Byun:

There's a couple of proof points here. One being able to prove out in a performance manner that we're able to operate on the encrypted data set. Two, that there was no application modification because this is a commercial off-the-shelf application. In fact, I think we're still using the trial version here if you see in the upper left-hand corner. So, there was no source code modification, but the ability for us to, again, support mathematical aggregation, visualization of the data, and let the business derive intelligence without necessarily exposing the privacy of the underlying data rows is something that this could facilitate. Another scenario may be of interest to folks is and another classic performance use case is this is an example of a cloud-based application where we can get into this application, and you can see that there are certain datasets or columns that are actually encrypted. You can see that, as I scroll to the right, some of these columns are actually encrypted. You can also see that there's roughly 201,560 records. This is without this alternate encryption technology. You can see that the top record number in the upper left is ending in 6-0-2.

Harold Byun:

Now, if I go and run the solution through the advanced encryption scheme, you'll see that the top record number 6-0-2. You can see that the data is in the clear, and there's also the same number of records. But if I do things like sort on this data and execute the sort, you can see that I can sort ascending. I can sort descending. You can see here, I'll sort ascending. Again, I'll sort descending, and what the technology has been able to prove out, which is really interesting, is a wildcard search where I'll do a contained search for *A-R-O-L-D. When I go after that search, you can see in a very performance fashion, we're able to return a subset of the records.

Harold Byun:

I could do a more complex search. "N" begins with seven. Drill down into this record, and we're able to return those values. So, that is an execution of wildcard search on AES encrypted data in a non-deterministic mode. Fully randomized search on AES encrypted data, and you can see that Harold is in the text here. Celine, my colleague, is actually working with me on this session, and so I'm going to plug her name in here. When I update this and refresh, you can see that we'll bring all the records back, and I can do a search for E line, and I think it was followed by large, I'll go space L-A-R. We should get a singular record return.

Harold Byun:

So, this is a way that we've been able to enable a method for mathematical computation, wildcard search, sorting, comparative operations on encrypted data. The data is obviously encrypted at-rest, but it is also encrypted in memory, in process, and in the search index. It's using industry standard AES.

Harold Byun:

Let me pause there. I'm going to go back to just a couple more slides, and then we'll open up for Q&A. Actually, what I did want to do is walk you through. So, what did we just do? How does this all work from a technical walkthrough standpoint? Again, if you have questions, I encourage you to put them in the chat. Basically, if we take customer owned keys and apply AES encryption, what we're able to do is then take a computational operation, and we utilize a technique called "Secure Multi-Party Compute" to establish these stateless servlets. What the stainless servlets do is they take a mathematical computation. Let me go back a couple.

Harold Byun:

When we see a computational request, we hand the computational operation across to the secure multi-party compute servlets, break the operation apart so that no singular servlet knows the full operation, run the computation without ever decrypting the original values, and then returning an encrypted value to the data structure and passing it up to the application in the clear. That's obviously all happening in very rapid form, but if you go back to this slide, you can see that on the left-hand side of the screen, the data structure basically has encrypted data with no key presence. On the right-hand side, what we've done is we've broken apart any of the computational operation, so that none of those individual members have access to the original data and never get to see the entire computational operational request in its entirety.

Harold Byun:

What this does is create a split domain or security contract where no single member of the party can ever see the data in the clear or ever have the encrypted data co-mingled with the key. That's kind of the security contract that we've enacted here in a patented and highly performance fashion. Are there alternate methods that deliver homomorphic-like capabilities? A lot of people don't really believe that we're able to do what we can do, but we've proven it out across a number of customers and in some of the largest computing environments at scale. We would consider that obviously fact, but we welcome skepticism. We run into it pretty frequently in the industry. We also know that there've been a lot of bold claims made across the industry, and we're happy to get into a deeper dive with folks if you're interested in learning more.

Harold Byun:

Some use cases terms of where we're seeing organizations adopt this. We talked about lift and shift to cloud, being able to support data migration services, being able to operate in the cloud with an opaque or encrypted data set, and migrate more workload to an infrastructure as a service provider, like an AWS or Azure. That's one scenario that we've enacted on. Obviously you saw the SaaS application and the ability for us to support not just base-level computational operations, but also orchestration and automation workflows, which are critical to enterprises deriving more value out of their information technology investments. There's a scenario where we can enable those end-to-end workflows without any breakage at scale. This is a PHI use case in terms of Cassandra moving to an AWS environment so support for big data, no sequel environment protecting PHI in that type of environment.

Harold Byun:

This is a variant of the information supply chain consolidation slide that we were presenting earlier. In this particular model, what we're able to do is gate access to specific vendors and control which data sets they actually have access to. What this allows for is, again, a consolidated data structure where you are in control of who's accessing what data and the types of queries that they can run while still preserving the underlying privacy and confidentiality contract. The inverse of this is also something that we can facilitate, which is utilizing this advanced encryption mode to hide the identity or attribution of organizations that are participating in a shared data analysis structure. The use cases here typically around anonymized threat, intel sharing, or fraud detection where people don't necessarily want to raise their hand and say, "Hey, we experienced this type of fraud" or "We are tracking these IOCs associated with this threat actor because they're targeting us as an organization, but we still want to share it so that people are more aware." That's a scenario where we're able to facilitate aggregate analytics, but still hide attribution.

Harold Byun:

Some alternate use cases there. These are some resources I think Celine's posted these as well. There's a ton of additional articles on our website. Our CEO and co-founder and Ameesh Divatia has had a number of articles published on Forbes on secure computing and dark reading. There's a number of different perspectives that we can provide as well as on our own website. So, that's kind of the summary of homomorphic capabilities today, where we think the market is, as well as ideally some validation on some alternate strategies or alternative approaches that we think are out there. At this point in time, we want to open it up for any questions. Again, if you have questions, please feel free to type them in. I think we have few.

Celine:

Yeah. We have a few questions. The first one is what is the performance hit for this type of encryption?

Harold Byun:

So, it's a great question; it's typically one of the more common questions we get. The way that this solution is actually instrumented is first and foremost, not all of the data necessarily needs to be encrypted. Any data that's in the clear is just going to be passed through in this type of model, but in terms of operations on encrypted data, base-level store and retrieve are negligible and typically stub 5% overhead. The sorting and aggregation that we showed you in the live demo was roughly in the five to 10% range, so that's typically at or below transparent data encryption performance overhead. The one exception is the wildcard search, which we demonstrated. As far as we're aware, nobody else can do that on AES encrypted data, and that carries roughly a 25 to 30% performance penalty. We're not done optimizing, but we still think that that's in the millisecond range and it is wildcard search on unencrypted data. It's a great question there.

Celine:

Another question is what is key management and HSMs does it support?

Harold Byun:

The key management and HSMs are obviously where we're retrieving the encryption key material to drive encryption. We support KMIP, as well as PKCS #11 for HSMs, so it's virtually any key management solution or HSM that can communicate via those protocols and standards. We also support AWS KMS via a REST API, so all of those are available as a key management options.

Celine:

The last question we have for today is is this deterministic encryption?

Harold Byun:

Yeah. Again, we're using fully randomized AED, it's a counter mode AES, and the multi-party compute's been around for decades. We just happened to be the first application of that cryptographic method to do a general data protection strategy. It's based on a proof written by Mihir Bellare who is a mathematics and computer science professor UCSD, the co-inventor of HMAC, which is an implementation of that proof that basically is validating the multiparty compute cryptographic approach that we've taken to enable this computation on randomized AES. It's non-deterministic. It's not tokenized. We're not building a searchable index. All of these other types of approaches that other folks or other vendors have taken to try and tackle this problem or not what we're doing. It is computation in a secure fashion in randomizing encrypted data sets.

Harold Byun:

See if there's any other questions as we kind of roll this out. Again, appreciate all your participation. Hopefully, some of this information was useful. You can obviously reach us on the web. There's the general info@baffled.io. There's my personal info. I'm a little slow on email, but you're more than welcome to reach out as well. I want to thank you again for taking the time, and hopefully this was useful for you. Thank you.

Celine:

Thanks, Harold. On behalf of the entire Baffle team, thank you guys for joining us today. A little reminder that an email with the link to the recording will be sent to you for future viewing and also to share with your colleagues. We look forward to you joining us at our next webinar. Have a great day!