Will End to End Encryption Save Us All?

Whitepaper | January 2010

BrandenWilliams
SECURE BUSINESS GROWTH

Table of contents

Information must move for our economy to function. Technology innovations over the last decade have done wonders to eliminate questions on how to connect devices over an increasingly mobile and on-demand workforce. For some reason, network security in these instances focuses on protecting the medium or means of communication, and have lacked when looking at protecting the actual device, yet endpoints are where the hacks occur. It's foolish to assume that a device connected to a trusted network is immune to attack.

End to End Encryption (E2EE) is touted by many vendors today as the panacea for numerous security and compliance initiatives, most notably PCI DSS. Is this technology, which is not new by any means, really the solution to all of our problems? Those inside the industry have heard claims like this before, and those who buy into them often find themselves committed to a serious sum of money without a complete solution.

This article will explore various forms of E2EE and their benefits and drawbacks, with the majority of the examples built on the challenges faced by those complying with PCI DSS.

What is End to End Encryption (E2EE)?

E2EE is a concept describing a method to secure data while in flight from one device to another. The simplest example that we use every single day is Secure Sockets Layer (SSL) inside of a web browser. When you sit down at your laptop and casually start browsing your favorite e-shop, your traffic may be able to be snooped by just about anyone sitting in between or near your traffic as it leaves your laptop, traverses the Internet, and lands on a server somewhere on the planet. That's why when it comes time to enter in our payment details, we are trained to look for the lock in the toolbar, https in the URL, or the new green bar in the location field. To the average consumer, this means that their payment data is safe, and cannot be stolen while transmitting over the untrusted Internet[1].

SSL was necessary to ensure the success of the Internet in the mid-1990s. Without it, people would have refrained from shopping online, thus removing the cost advantages eTailers see and significantly reducing the commercial appeal of a network that was originally designed to survive a nuclear attack. SSL is considered so essential today that businesses now protect basic things like information exchange by adding an SSL certificate to their non-commerce websites. How else would they protect the significant Internet population that chooses to do business over Wi-Fi links—especially those found in coffee shops, hotels, and airports. Sure, it does not guarantee that either endpoint is safe, but it does take the worry out of snooping eyes reading the traffic once it leaves said endpoint to traverse network equipment throughout the globe.

For the purpose of this article, I would like to loosely define E2EE as a method to protect data in flight over a network such that only each end of the transaction has the ability to see the plaintext.

FOOTNOTES

[1] *As security professionals, we understand the fallacy of this statement. Sure, provided controls are correctly implemented, the data is generally safe from Point A to Point B, but as we will explore in this article, that only guarantees that data in flight is safe, not once it reaches the endpoint.*

End to End Encryption Variants

One of the biggest challenges with E2EE is that it is a term thrown around to mean a multitude of different things. In a pure networking aspect, E2EE is simply a term to describe data protection between two endpoints. Users get in trouble when they throw the term around without understanding or defining what the ends actually are. If I were to tell you we are going to spend the day frolicking among sand dunes, doesn't your participation hinge on whether those dunes are on a beautiful beach overlooking an ocean or if they are in the middle of the Sahara?

The lack of definition about the ends is considerably important when using it in reference to PCI DSS—specifically when using E2EE as a scope reduction technique. Without clearly defining which ends you are encrypting between, you cannot determine the impact that will have on protecting your networks or reducing your compliance scope. Over the next few sections, I want to propose some common vernacular that we can use as an industry to describe some of the more common flavors of E2EE.

POS/PED to Acquirer Encryption (P2AE)

P2AE is the most complete form of E2EE that would theoretically give merchants the most relief from dealing with PCI DSS. In this case, the PIN Entry Device (PED) would encrypt the payment data before sending it through the Merchant's network to the Acquirer for processing. At no time during the transaction should any Merchant system (aside from the PED) see the decrypted payment data. The Acquirer can return the approval status with some unique way to identify either the payment instrument or the transaction itself. Settlement would need to occur with this process in mind, considering that the settlement files will not have the payment instrument included with the transaction. This is considerably important when the authorization and settlement amounts could differ, such as in the fuel or hospitality industries[2].

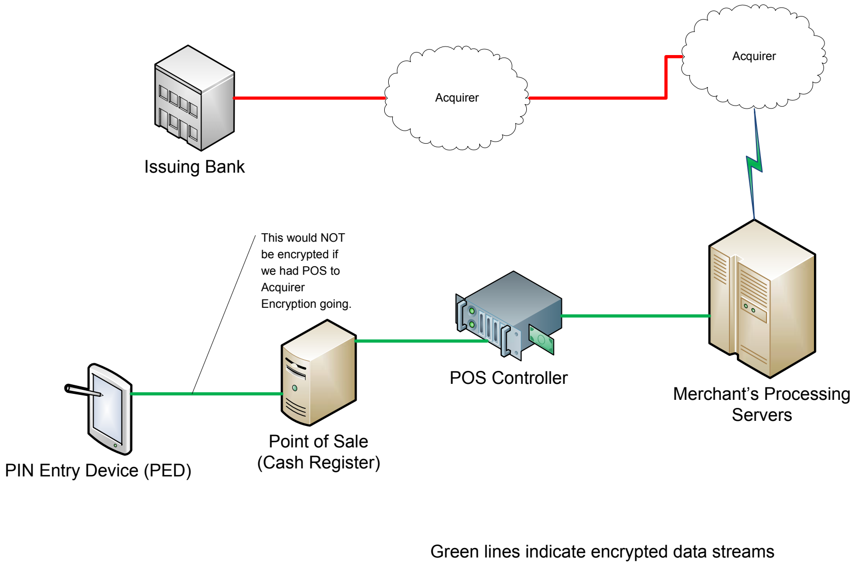

Issuing Bank
Acquirer
Acquirer
This would NOT be encrypted if we had POS to Acquirer Encryption going.
POS Controller
Merchant's Processing Servers
PIN Entry Device (PED)
Point of Sale (Cash Register)
Green lines indicate encrypted data streams

Figure 1: POS/PED to Acquirer Encryption (P2AE)

FOOTNOTES

[2] *Merchants belonging to those groups typically authorize or pre-authorize an amount larger than the transaction would likely be to ensure funds are available after the transaction finishes, then settle for the actual amount.*

For merchants deploying and E2EE setup like this, the only scope issues for PCI DSS would be the PED devices themselves, and any backend processes that might require the full card number later. A list of such processes will be proposed and explored later in this article.

The above scope statement, of course, must be validated by a QSA (or internal audit where applicable) for each Merchant. Merchants still have a responsibility to secure data in accordance with PCI DSS, but the amount of work to be done is dramatically reduced. Merchants could argue that if they outsourced the management of their PEDs and had no direct access (other than physical of course) to the devices, they might only have to validate the absence of data in their environment for PCI DSS compliance. Imagine how much cheaper that would be than disrupting your operations for a few weeks to conduct an assessment with an outside party!

One slight variant of this method (not different enough to get its own acronym or classification) is Point of Sale (POS) to Acquirer Encryption. Some POS devices have integrated PED equipment, and some PEDs cannot do encryption natively. As long as the POS network is adequately segmented from the rest of the network, a Merchant's scope may be limited to just the POS devices and network.

POS/PED to Processor Encryption (P2PE)

Some merchants, namely smaller ones, don't process transactions directly with an acquiring bank. Instead, they work with interim processing houses or giant Independent Sales Organizations (ISOs) like Heartland Payment Systems or First Data Merchant Services. Once transactions hit your processor, it becomes their responsibility to keep said transaction secure.

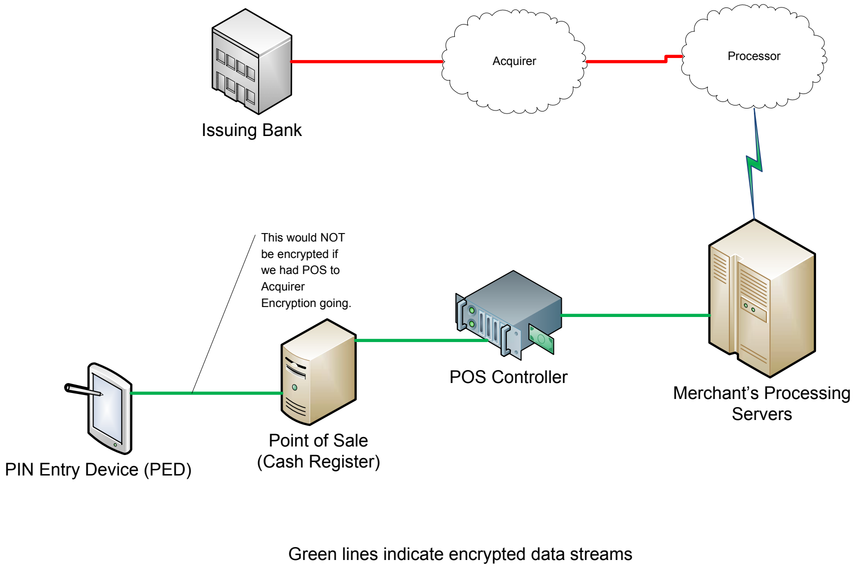
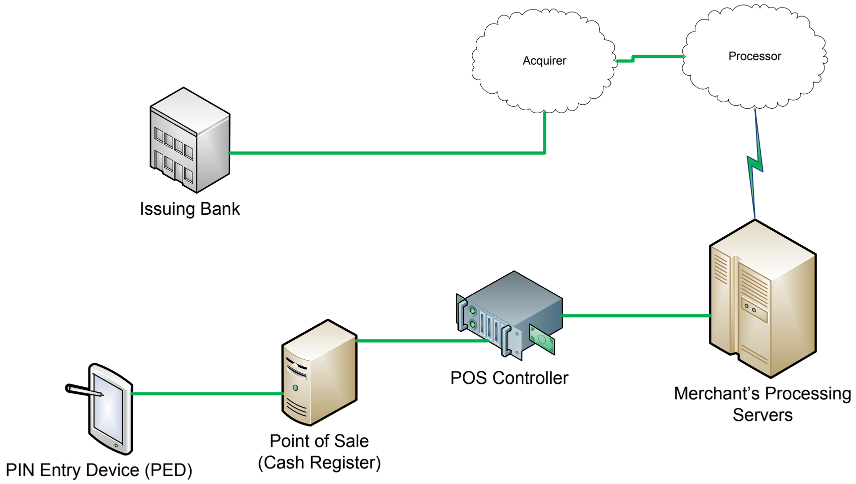

Acquirer
Processor
Issuing Bank
This would NOT be encrypted if we had POS to Acquirer Encryption going.
POS Controller
Merchant's Processing Servers
Point of Sale (Cash Register)
PIN Entry Device (PED)
Green lines indicate encrypted data streams

Figure 2: POS/PED to Processor Encryption (P2PE)

Processors like this can be interesting groups to work with while looking at your security and compliance posture. Some large processors act nearly identical to acquiring banks, while other processors are not much more than a few guys and servers in a garage somewhere. Either group can work well with your setup, but as a Merchant you must know what you are working with on the other side of the copper line that feeds your bank account.

PED to Issuer Encryption (P2IE)

P2IE is the ideal form of protecting information while processing payments. Unfortunately, it's not only a fallacy in some regards, but unlikely to happen without substantial investment by all parties that touch payment data. The closest thing to P2IE encryption that exists today is PIN-Debit transactions. Information collected from the magnetic stripe in conjunction with the PIN entered by the user creates an encrypted value (PIN block) that makes its way back to the Issuer for approval through various networks. Each time the transaction hops to the next provider, part of the PIN block is decrypted, and then re-encrypted with the next working key. This means that each hop potentially can compromise the confidentiality of the transaction if the device doing the crypto operations is compromised. Another variant would be EMV, commonly known as Chip & PIN.



Green lines indicate encrypted data streams

Figure 3: PED to Issuer Encryption (P2IE)

To get to a P2IE model, encryption algorithms and key exchanges would need to be standardized across the entire payment ecosystem, or at least would have to provide both the encrypted and the standard methods of passing the data for providers that do not have the capability, thus negating the whole point. It certainly CAN be done, but would require substantial investment from all groups operating inside the ecosystem.

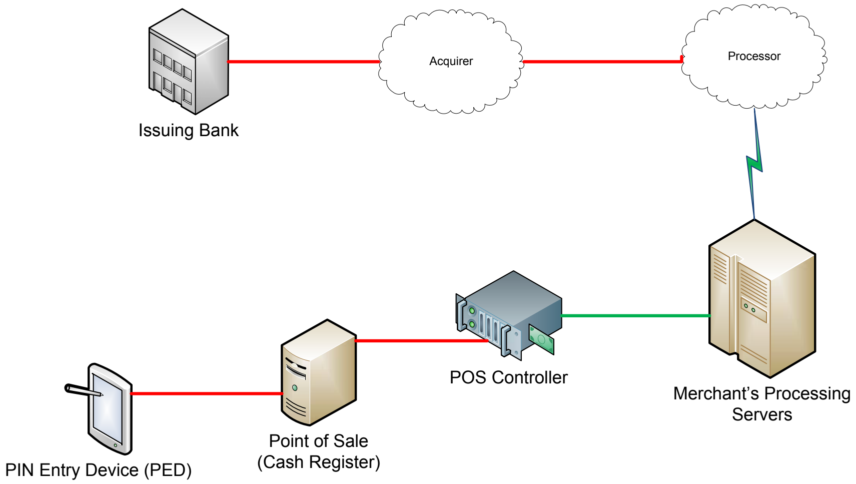
Controller to Processor Encryption (C2PE)

This is an odd one that I've seen more than once in small Level 1 or large Level 2 merchants. It usually requires there to be some kind of third party application to be used

FOOTNOTES

[3] The only reason why I am specifically using processor and not Acquirer here, is I've not seen a setup by which an Acquirer was doing the decryption. Instead, some entity that sits between the two parties performs the decryption and sends along to an Acquirer for processing.

for processing payments, or even the entire POS system. In this case, transactions coming into the POS controller (which controls the various POS terminals or registers that you bring your goods to for payment) are not encrypted, but the controller itself will perform an encryption routine to protect the data until it hands off to the processor[3].



Green lines indicate encrypted data streams

Figure 4: Controller to Processor Encryption (C2PE)

This setup can be quite effective depending on the architecture of your POS network. For example, if all transaction are segmented from the rest of the store, or if the only thing on the store network are the POS terminals and controller, merchants could take a significant portion of their network out of scope. One major drawback to this setup is that the POS controller becomes one of those juicy targets that a hacker might want to attack. It's really a simple exercise in math.

Let's say a hacker targets a popular retail chain in mid-September with the goal of capturing the increase in payment transactions caused by holiday shopping. He targets a discount retailer with 500 stores, with each store typically having at least ten registers. If architected to his advantage, he would be able to get much more bang for his buck by placing malware on the 500 controllers over the 5,000 terminals.

If you deploy this type of architecture, be sure you perform detailed security testing on your controllers, and pay close attention to new threats that could affect their security. Several breaches in the last two years resulted from attackers successfully targeting POS controllers.

Merchants can uniquely identify any transaction in their environment without the entire PAN with the following four pieces of data:

1. *Truncated PAN (first six and last four, or just last four)*
2. *Date and time of purchase*
3. *Purchase amount*
4. *Authorization code*

Take advantage of this method where you can to avoid storing full PAN data.

What else do you need to consider?

Is it really this simple (relatively speaking)? Can I just throw a little bit of technology at this problem and make my life an order of magnitude easier?

There is **always** a catch, isn't there? I learned from Mrs. Adamson in high school economics that there is no such thing as a free lunch, and this situation is no different.

Settlement and Reconciliation

As long as transactions are authorizing normally, this process will work quite nicely. You still need to consider how settlement and reconciliation will be handled. If the decrypt operations only occur at the acquirer or processor, they will need to provide you with a method to handle settlement and reconciliation such that you do not need the original card data.

Failed Swipes

What happens if a customer presents a payment card that does not work when it is swiped? That's the next obstacle. If you are going with a P2AE option where the POS is doing the encrypting, then hand-keying the card into the POS device should give you the same protection as if the card was swiped. If the encryption happens at the PED, however, you will now be passing unencrypted card data through to your acquirer if you hand key (unless you can somehow route the card through the PED), thus undoing all of your hard work for that one customer who swears their card always works everywhere else. If you choose to take an imprint of the card, or call the details in, just remember that those processes need to be secured in their own way.

Law Enforcement

Another often overlooked area is working with law enforcement. Several retailers have complained that they cannot secure card data in certain areas of their network because of their work with law enforcement. I disagree completely on this approach, and believe that there are many ways to work with law enforcement to properly do investigations and preserve data, while keeping things secure at the same time. The important thing to remember here is that if you work with law enforcement and they give you PANs for research purposes, you still need to secure that data. You can't control if a Sheriff in College Station sends you 10 credit cards over email, but what you CAN control is what you do with them once you receive them. In fact, I would argue that if you wanted to truly keep your networks clean, have them send the data to a GMail account that you set up, and secure the data once you receive it.

Paper Records

Don't forget paper records in this mess either. Faxed or mailed orders with PANs, preferred customer lists[4], business continuity actions for power or other outages, and reports from the finance group. Paper is somewhat easier to physically protect provided that you focus on physical security, but in the same regard, that type of security is much more expensive to maintain on a daily basis. Regardless, don't forget about it.

Refunds

Finally, how do you deal with refunds? Retailers have policies in place on refunds to prevent fraud. Many retailers require that you provide the same card to a refund clerk that you paid with the original merchandise with in order to get a cash refund. How can you check this on the backend without storing the actual card? Well, that's definitely a challenge. I propose that you work with your acquirer to use some kind of token or transaction identifier, then let them perform this check for you. Provided it matches, issue the refund. Better yet, include a transaction identifier like many big box retailers now do so you can just tell the acquirer to refund Transaction X to the original payment method. Yes, these services will cost you more money, but is that money spent per transaction better than spending thousands or millions on building infrastructure and hiring experts to support and maintain it?

These are a few examples of processes that may undo your E2EE strategy, and there are

FOOTNOTES

[4] I once had a customer that stored imprints of their largest customers in a rolodex that sat on a floor manager's desk for convenience.

definitely many more. If you are thinking about E2EE outside of PCI, think about the same types of process you might use for the data and apply the same concepts. As with most things related to information security, it ultimately comes down to the information. Know where it is, where it goes, and how long you need to keep it around.

Attention Shift

Some of you reading this article may be thinking, “Branden, this is great but if I take the network out of this, aren’t hackers just going to find some other way to take my data?” I’m glad you are thinking this way. It means that you realize that E2EE is not the end all solution to your security issues.

The way we exchange data is constantly changing and improving (hopefully) as technology and process evolve. In order for us to keep up, try doing an exercise where you trust no network link provided to your application or data. Assume everything is like an open Wi-Fi at your local coffee shop. E2EE solves the first issue, data security over the network, but does not cover the second and more important issue, endpoint security. If machines are allowed to participate freely on a network (which, let’s be honest, are pretty much allowed to at any given time on most networks), they must defend themselves from an occasional malicious user that pops up to start probing around.

Retailers have dealt with this problem for years. The minute that a network shows up in the store to power a POS system, they now realize that depending on their store configuration, someone with a little knowledge (or ingenuity) could easily plug a laptop in and start probing around, leading to a data compromise. Most of the time, this is handled with physical security. But what about something like a satellite office? Hopefully a satellite office for a major healthcare provider will have similar security to the main headquarters, but often this is not the case simply from a scale issue. If the headquarters houses 3,000 employees, and the satellite office only has ten, which do you think will have better security?

Securing the network will force attackers to concentrate on endpoints. This means that systems truly have to be hardened, patches must be deployed consistently and timely, and the amount of data storage should be scrutinized and monitored. As networks continue to evolve and a clear perimeter ceases to exist, focus on securing the transmission between devices and the devices themselves and ignore the perceived security of the medium used to enable the exchange.

Go Save Us All!

E2EE fills a useful purpose in today’s compliance and security landscape, and should be considered as part of the overall security strategy for any organization that moves data from one endpoint to another. As we increase the scope and complexity of our networks, we should focus less on the medium by which data moves, and more on securing data from endpoint to endpoint. Computing power today is substantial, and with the exception of certain types of bloatware (we all know which ones they are), most of the endpoints today have enough resources available to perform basic crypto routines in which to protect data.

The important part of deploying this type of strategy is ensuring that the standards you use are open and cheaply to deploy (SSL is certainly a good candidate here), and paying attention to endpoint security after you deploy. Networks are not nearly as juicy of a target as endpoints are, and encrypting cleartext data over network links forces hackers to focus all of their efforts on endpoints.

Take some time to review the types of data that openly traverse your network, and don't forget to think about batch processes or things that only happen every so often when you do your review. Building an impressive E2EE strategy will surely look bad if you forget the massive backup and data dump that occurs once per quarter, exposing millions of cleartext records to a patient hacker.

To answer the question posed in the title, will E2EE save us all? Not by itself, but implemented in a method that focuses on protecting the data as it leaves the endpoint, it will be a critical component of the plan that will "save us all."

Whatever you do, be sure you make your strategy as scalable as you can. Mobile computing isn't going away, and you may need to be able to handle that mobile workforce in developing countries and markets. E2EE is a good strategy when used properly, but as with most security controls, it should be used as an important component to your overall strategy, not the strategy itself.

Contact information for requests for permission to reproduce or distribute materials available through this course are listed below.

About the Author:

Branden R. Williams, CISSP, CISM, CPISA/M, has been making a name for himself in the Information Technology and Security arena since 1994, as a high school Junior. Now, a graduate of University of Texas, Arlington earning his BBA in 2000 with a concentration in Marketing and the University of Dallas, where he earned an MBA in Supply Chain Management & Market Logistics, in 2004, Williams is sought after as both an Adjunct Professor and Information Technology & Security Strategy Leader in the corporate world.

Williams regularly assists top global retailers, financial institutions, and multinationals with their information security initiatives. Read his blog, buy his book, or reach him directly at http://www.brandenwilliams.com/.

TEL 214 727 8227
FAX 214 432 6174

BLOG brandenwilliams.com

EMAIL brw@brandenwilliams.com