Module 67

CSE201, Part 2 of 2: Introduction to Security Credential Management System (SCMS)

Ken Leonard: ITS Standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems.

I am Ken Leonard, director of the ITS Joint Program Office for USDOT and I want to welcome you to our newly redesigned ITS standards training program of which this module is a part. We are pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this new approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself.

This combined approach allows interested professionals to schedule training at your convenience, without the need to travel. After you complete this training, we hope that you will tell colleagues and customers about the latest ITS standards and encourage them to take advantage of the archived version of the webinars.

ITS Standards training is one of the first offerings of our updated Professional Capacity Training Program. Through the PCB program we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener which improves livability for us all. You can find information on additional modules and training programs on our website www.pcb.its.dot.gov.

Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. Thank you again for participating and we hope you find this module helpful.

Moderator (Nicola Tavares): Your first instructor is Virendra Kumar, a senior staff engineer, technical standards at Qualcomm Technology, Inc. Now Dr. Kumar is involved in consulting, research, and standardization efforts in the area of vehicle-to-everything communications security. He has had extensive involvement in the original design of the security credential management system and its subsequence security and privacy enhancements. He currently serves a chairperson of security and privacy workgroup at 5G Automotive Association, 5GAA, and has over 13 years of research experience in the area of cryptography and information security.

We also have with us Dr. William Whyte, who is senior director, technical standards at Qualcomm Technology, Inc., following the acquisition by Qualcomm of Onboard Security, where he was CTO. William is one of the world's leading experts in the design and deployment of security for connected vehicles and general mobile ad hoc networking systems. He is the editor of IEEE 1609.2, the baseline standard used worldwide for connected vehicle communication security, and of its related and successor standard. He was a key contributor to the design of the security credential management systems for connected vehicles in the United States and lead security consultant on the New York City connected vehicle pilot deployment.

It is now my pleasure to turn over the presentation to your first presenter, Dr. Kumar.

Virendra Kumar: Thank you, Nicola. And welcome back everyone for the part 2 of a two part course on security credential management system. As Nicola already said, if you have not already seen the first part of this course, we would highly recommend for you to do so before the second part. The learning objectives for this part of the course are first, identify the vehicle-to-everything, V2X, or V2X certification process for a vehicle, or a device to enroll in the security credential management system, or SCMS for short. And the second learning objective of this part of the course is to illustrate how to make a deployment plan that uses SCMS services.

So with that, having said, let's do a quick recap of the learning objectives of the first part of this course, the learning objectives from one through three. So in this whole V2X ecosystem, the main baseline standard is IEEE 1609.2, and this standard specifies security services, including cryptograph and data validation services, that can be used to protect data in transit. In this 1609.2 system, a receiver is sure that a sender is trusted to send a message or a command of a particular type, because the sender has a certificate, and that certificate states what a sender is allowed to do. And the 1609.2 processing cryptographically links the certificate to the message, to show that only that certificate holder could have generated the message.

These 1609.2 certificates—they are issued by the SCMS to any participant in the system. It's the SCMS primary responsibility to make sure that certificates are issued only to the actors who are entitled to them, by carrying out different checks. For example, making sure that the actor was entitled to the certificate in the first place and also making sure that the actor has not become malicious, untrustworthy, or otherwise unreliable, since the certificate was issued.

The SCMS and the 1609.2 certificate system, they are together designed to preserve privacy from eavesdroppers in the field, and also from the insiders at the SCMS. There are a few major challenges in the SCMS deployment, which include enrollment of the devices, in particular, establishing that the devices are entitled to certificates, especially for specialized applications. This is something that we'll cover in the next learning objective, learning objective number four. And the other challenges include keeping the devices provisioned with certificates. This requires regular access to the internet, and also understanding which devices should have their certificates withdrawn, or what is known as revocation. And these things will be covered in the next learning objective, number five, by my colleague, Dr. William Whyte.

Finally, as we recommended in the last part of this course, we reiterate that we highly recommend that the deployers—the deployment managers—they work with an SCMS provider rather than trying to run the SCMS services themselves because of the lots of complexities involved in running an SCMS securely. So with that said, let's jump right in to the learning objective four.

In this learning objective, we would understand and identify the vehicle-to-everything certification process for a device to enroll in the SCMS. So what are the security requirements for devices? When procuring a device for deployment, the procuring agency or the company needs to understand the security requirements for each device. The security requirements for a device are obtained by deciding what applications a device will run, determining the security requirements for each of those applications, and then setting the security requirements for the device, so that it meets the security requirements for each of those applications.

For some applications, the procurer can consult existing security analysis, and for new applications—for which there are no existing security analysis—the procurer can carry out a security analysis on their own, following the NIST cybersecurity framework. To make procurement easier, four device security classes have been defined. Please see the student supplement for where these definitions can be found. If a deployer wishes to deploy a new application on existing devices, they can carry out the security analysis for the new application and see if its requirements are already met by the deployed devices. If the requirements are already met, then the application can be deployed on the existing devices without any change. However, if the requirements—if there are new requirements on the application that are not already met by the device, then the device would have to have to be changed, to be replaced with devices with higher security requirements.

So when we talk about security requirements, there are some baseline security requirements that all devices have to meet. To run a specific application, devices may have to meet additional application-specific requirements. For example, although this is not a regulatory requirement, it could be the case that if a device runs single preemption. For example, it must require an operator to log in before the signal preemption message is sent.

The baseline security requirements are as follows. First, the device must protect its key material against being revealed by having a hardware and software protection against this. This is frequently done by protecting the keys in a special chip called hardware security module, or HSM for short. This is a chip that has protected memory and physical protection, to prevent an attacker from directly reading the keys from that memory. This protection is specified by a federal standard called FIPS 140. There have been several version of this FIPS 140 standard, starting from FIPS 140-1 to -2 and the most recent one, -3. These are all available from the website of National Institute of Standards and Technology, or NIST for short.

All the keys are often, in practice, protected by enclosing them in an HSM. There are other architectures that are permitted if they meet the same requirements. Additionally, the keys are used to sign messages when requested by an application. The application platform also has a security requirement to prevent malware from requesting the HSM to sign. As we noted, there are different architectures, including an integrated architecture, a connected architecture, and a network architecture. In an integrated architecture, the applications and crypto are done on the same processor and the whole thing is protected at the same level. Whereas, in a connected architecture, the HSM is separate from the application processor, but they're bound together so that it can only receive direct communication from that application processor.

And finally, in a network architecture, the HSM application processor and other processors, they call talk on a bus, and they're all connected to each other so the sources of each message need to be strongly authenticated. Further, on baseline security requirements, the devices may also need to enforce secure update mechanism and may require operator authentication. Different requirements document exists that will provide a guide to the security requirements in your environment. You should consult with your SCMS provider to understand exactly which requirements that SCMS is requiring for you.

There are different requirement documents: the OmniAir hardware/software/OS security requirement, the Car2Car communications security protection profiles. Please refer to these. The links to these requirements have been provided in the student supplement. Having said all of that, you can assume that the requirements that we are covering here as the baseline security requirements. They will always be required to be met. These are the baseline security requirements. Furthermore, there are some more additional requirements, security requirements. The device also needs to protect against malware, either malicious versions of valid applications, which could get installed, and then ask the crypto-processor to sign incorrect messages, or entirely distinct applications that shouldn't ask for messages to be signed, but in fact do.

A useful technique here is to require that only signed code can be installed on the device. Devices also need to protect against malicious firmware that might intercept messages between applications and the HSM. A technique that makes it difficult to install malicious firmware is known as secure boot. Secure boot ensures that the state that a device is in when it powers off is the state it is in when it next powers on. This means that if a root kit has been installed, it will be detected by the boot mechanism.

Secure boot is entirely local and doesn't require the device to interact with a network entity, making it suitable for the V2X setting.

As we mentioned earlier, the OmniAir consortium is defining our formalized security requirements for V2X devices. However, there is no formal or national policy that an SCMS operator must require this OmniAir certification, but in practice, existing SCMS operators respect OmniAir certification. OmniAir runs several plugfests every year, where interoperability can also be tested. Full conformance testing for security is likely to be available by early 2021.

So now let's talk about special permissions. The V2X ecosystem or community is comfortable with applications for which certificates can be issued at the factory. For example, most BSM sending devices (so BSM, for those of you who are not familiar, is basic safety message). Those BSM sending devices can be initially provisioned at the factory. This is because they are claiming a baseline set of permissions, and all that is needed to send BSMs is to be properly installed in a vehicle. However, there are some applications where there are stricter permissions, that must be met to ensure that it is appropriate for them to send messages. For example, in an emergency response vehicle, which might be able to request signal priority, it might need to be bound to an agency that owns them before they can get certificates. Further, there may be a requirement that a driver signs in with a password or a PIN before the devices can request prioritization or preemption.

Also, RSUs sending SPaTs, or signal phase and timing messages, might need to be properly installed on a network operated by an identified operator before they can send. So, as we see, for different applications, there may be different restrictions on when a device can get certificates and what it has to do to convince the SCMS that it is entitled to receiving those certificates. The requirements for these applications are still being developed. As a developer, you may have to work with your SCMS provider to understand what those requirements are.

Additionally, we may need to consider the case where a device is deployed, and later, a new application is installed on the device. For example, this slide shows a deployment manager that wasn't to deploy an RSU that sends SPaTs. To do this, the deployment manager may need to present a number of things to the SCMS; for instance, a certification lab report showing that the RSU is secure enough to run SPaT, and a proof of ownership. The SCMS then compares this against the policy and issues the certificate. Now the deployment manager decides to deploy another application, say SSM or SRM. The deployment manager goes back to the SCMS provider with a different set of evidence, as agreed with the SCMS provider. In this case, for example, since there is already a lab report associated with the device, maybe the SCMS provider only needs to know the identity of the device. Now the SCMS provider has all the information they need to issue a cert for the new application.

In the currently deployed SCMS interfaces—the CAMP interfaces—this flow is not possible, but it is enabled by the new standard for the interfaces, the IEEE 1609.2.1. Since this is a new functionality, processes for using it aren't fully mature. If you are a deployer and you are thinking you may want to install applications on your devices after they have been established in the field, make sure you talk to your SCMS provider before field deployment so you don't end up discovering that adding applications require extensive physical access.

So, this covers this learning objective. Let's take a short quiz on this learning objective.

Which of these is a requirement for a device to be secure enough to run V2X applications? And the options are first, the device requires a user to log in before it will send any V2X messages. Second, the device requires user permissions for updates. Third, the device supports virtualization, and fourth, the device protects its keys with a hardware security module. If you need time to answer this quiz, please pause this presentation.

So, let's review the answers. The correct answer is the d), option d) the device protects its keys with a hardware security module. This is correct because if the keys are not protected with a hardware security module, an attacker can get access to the device and potentially obtain a copy of the keys and use them to forge messages, signed messages. Let's review the other incorrect options. The first option is incorrect. The device requires a user to log in before it will send any V2X messages. There are many types of devices in the V2X ecosystem, such as the standard on board units in cars, which are expected to start broadcasting without requiring the user to log in. The second option is also incorrect. The option is, the device requires the user permission for updates. So, this is incorrect because although the updates much be secured, meaning that they must be authenticated as coming from a trusted source. They do not actually need the user's explicit permission. It is a courtesy to inform the user that an update is taking place, but user permission is not required as long as the device can ensure that the update is taking place under safe conditions. And finally, the third option, the device supports virtualization is also incorrect. While virtualization can improve security by sandboxing different applications and thereby preventing one application from interfering with another application's operations. It is not necessarily a requirement and that is especially because devices like standard onboard units that not only send one type of message.

So this concludes the first learning objective of this part, learning objective number four. And now I yield the virtual floor to my colleague, Dr. William Whyte, to cover the last remaining learning objective of this course. Thanks a lot for your attention.

William Whyte: Thanks very much, Virendra. Now we'll do learning objective five, which illustrates how to make a deployment plan that uses SCMS services. There’s a number of tools you can use when you're developing a deployment plan. One of those is the architecture diagrams from ARC-IT. The link to ARC-IT is in the student supplement, and this slide just denotes that SCMS services are fully described in ARC-IT and their deployment could be planned using the same tools that you are using to plan the deployment of other connected vehicle services.

So SCMS services are interesting and we'll talk about this in more detail later, because you have your ITS object over here that's in charge of operating within the network that's going to be carrying out the applications, but it needs to reach back to the SCMS to get its credentials, and that process of reaching back can be tricky to integrate within IOO traffic management network. So that's something we're going to talk about later on in this learning objective. But it's just to flag that if you're using ARC-IT to plan your network architecture, that observation will also become apparent through the use of the ARC-IT architecture diagrams.

Another useful tool is the security management operating concept. The security management operating concept is a document that you write that helps you understand the security requirements within your system, within your network. So deployments should develop a security management operating concept document that identifies all of the information flows that you're going to have within your deployment—identifies the security requirements. Those information flows use those to determine device security requirements, and then you use those to identify the security mechanisms for the information flows. So in other words, on this particular information flow, say, sending SPaTs, we're going to use 1609.2 security.

On this other information flow—say connecting between the traffic management center and RSU—we're not going to individually sign each message with 1609.2. We're going to need some different mechanism, like TLS, or a virtual private network. So this is the process you follow. It starts with analyzing the information flows and deriving the device security requirements in one stream and separately looking at communications security mechanisms, and if you're using 1609.2, defining what's called the 1609.2 security profile. That 1609.2 security profile, it's specified using a form that's available through 1609.2 and there are many different examples of good 1609.2 security profiles available through the SAE standards that already profile 1609.2. And then obviously, if you're using an existing application, a PSID will be defined for that application. If you're defining a new application, you'll need to observe a PSID for that application, which can be done via IEEE. And then this 1609.2 profile and the device security requirements become part of your certificate issuance policy.

So the device security requirements—are—tell the SCMS provider what security requirements the device needs to satisfy in order to be issued certificates and the 1609.2 profile tell the SCMS provider—when it's issuing the certificates—what to put into them.

Let’s do a case study, or a series of case studies. Let's look at some lessons learned from different deployments that have happened over the last few years. Some of these are part of the connected vehicle deployment project that was simply funded by USDOT; others are state initiatives. So one thing that was noted in New York City was that it was desirable for messages that are going to be applicable over a wide area, such as MAPs, which can be MAPs at multiple intersections, or TIMs, traveler information messages, they include information such as evacuation routes. Those messages should be generated centrally, rather than being generated on RSUs. A reason for that is that if an RSU can sign an evacuation warning, for example, then anyone—then you need every RSU along the evacuation route to be able to sign that warning, and then if any single one of those RSUs becomes compromised, that RSU can sign a fake evacuation warning. That's not a thing you want happening in your system, and so for security reasons, it was determined that it made most sense to sign those messages centrally, somewhere inside the network, and only have one instance of the sign in keys. And in that case, just like with the field devices, you need to be able to demonstrate to the SCMS provider that where you're generating the signature meets the security requirements for that application.

In New York City, in order to generate the TIMs and the MAPs centrally, they ended up purchasing a security appliance, a dedicated appliance with a hardware security module that stored the private keys. That was approved by the SCMS provider. And that protected the private key, but also did some basic format checking on the messages to be signed, to make sure that at the very least, they were valid instances of messages for an application. Obviously, the appliance couldn't check that data was correct, because that data's going to vary, depending on the exact thing that's being stated, but using the appliance gave an extra level of assurance that somebody wasn't just trying to submit rubbish and get it signed.

Meanwhile, the SPaTs were generated by the RSU, and the RSUs had to go through a separate certification process. CAMP, on the other hand, decided to generate SPaTs and MAPs at the RSU, or rather to generate MAPs centrally, but have them signed at the RSU. And so Tampa needed to have a different conversation with the SCMS provider to make sure that the RSUs were considered sufficiently secure not just to generate SPaTs, but also to generate these MAPs, which can contain information like regulatory speed limits that typically need to be protected to a higher level than signal timing information.

In Minneapolis, there's a plan to allow snowplows to request signal prioritization. That makes it particularly easy to coordinate snowplow fleet operations so that you can have some plow from the main line of a freeway, and others going up the exit ramp and down the entrance ramp, and making sure that any traffic signals at the top of the ramp are synchronized with the movements of the snowplows, so that that fleet can stay in formation.

And here we have the issue that we've talked about previously, where you want to make sure not just that the device that's going to request the certificate is secure or not. You want to make sure it's actually been installed in a snowplow. So in this case, we're not issuing the device with certificates while it's at the manufacturing plant. We're issuing the device with certificates when it arrives at the site where it's actually going to be deployed, and so some known contact at that site can be in charge of letting the SCMS know, yes this device that you've received a certificate request from, actually is on site, actually is in my possession.

Here’s some lessons learned. These have been learned as part of the different pilot deployments, particularly New York City. It's intended that this lesson learned document will be published. At the time that we're recording this presentation, the document currently isn't public. Hopefully, by the time you're listening to this, the document will be public. But if it's not, here's a preview of its contents.

So, one thing that's noted is that if you're deploying RSU certificates, you do need access to a production SCMS to update those RSU certificates. So there's a number of things to note about that. That's a short bullet point. The first is, production SCMS means that SCMS that will be trusted by vehicles in the area. So, you can't use a test SCMS. You have to use a production SCMS. The other thing is that the RSU needs to have access to the SCMS at least once a week. This will play into the topic we already mentioned where in order to get certificate updates, the devices need to reach outside the IOO network.

A production SCMS is also needed for certificate top-off for the OBUs. The OBUs need to be able to access the SCMS at least once a week. And here, if you're doing a deployment where you have dedicated vehicles, and if you are constraining those vehicles so that all their communications with the SCMS happen through the traffic management network, then you need to be very careful to test all those connections.

So, this isn't a particularly strict connectivity requirement, but it is a requirement for reliability that you're going to want to make sure it's been tested, and any edge cases have been knocked out.

And one particular operation that lends to issues was, what happens if a certificate download fails? The certificate download, as we mentioned, devices, especially OBUs, are going to have pseudonym certs. They're going to have batches of certificates, more than one certificate a week, and so it's conceivable that a certificate download can be interrupted in the middle, and the device won't successfully complete first time. So it's important to test what happens in that case.

And then, as we've mentioned as well, certificates have expiry dates. Certificates will have these validity periods, such that at the end of a validity period, the device is expected to have ready the certificate from the next validity period, and to transition seamlessly between using the old certificate and the new certificate. Commercial implementations of 1609.2 should do this, but again, this is a moment where the system is fragile. It's undergoing a transition, and so this is a case to also be tested.

Make sure that devices that tend to update certificates in a timely manner, in other words, they ensure they're provisioned with certificates some time before the current set of certificates expire. Make sure that they stop using the old certificates and start using the new certificates at the appropriate time, in other words, before the old ones expire, after the new ones become valid. And note that there's usually an overlap period between the old ones and the new ones, so that this change over isn't particularly stressful, so that there's a certain amount of wiggle room around the exact time that the transition takes place. And finally, if there are conditions that inhibit certificate change, make sure that the system transitions correctly after the conditions stop applying.

So for example, in SAE J2945/1, if a vehicle is in an alert state, then it shouldn't change the certificate in the middle of the alert state. The reason for that is that when certificates are changed, it introduces a certain amount of risk as to whether or not the receiver will be able to verify the message. They need to have the new certificate. They need to do some addition verification. Transmitting the new certificate makes the packet larger and so increases the risk of packet loss. And so when a vehicle's in an alert state, meaning that there's a significant imminent risk of a crash, typically vehicles are not meant to change their certificate. Obviously, that alert state isn't meant to be a persistent state. It's meant to be a momentary state, so this is another thing to be tested carefully. Make sure that the certificate doesn't change if it's meant to, and changes appropriately once that condition has been relaxed.

Another very important lesson learned is to integrate the connected vehicle cybersecurity needs very early into the agency's cybersecurity program. So this is making sure that the additional traffic on the network isn't at risk, but there's no risk that the contents of the traffic from vehicles or mistaken for any other kind of traffic, making sure that its devices have access to the SCMS without causing a security risk to the rest of the system.

To the greatest possible extent, any networking associated with connected vehicle pilots should be kept distinct and isolated from the general traffic management network. Again, this is a prudent risk management approach. Make sure that the over-the-air download and upload mechanisms have been fully tested before there's been extensive installation of the aftermarket safety devices, OBUs, and RSUs. If you are relying on over the air update, you need to make sure that it works, because otherwise, if an update is necessary and it needs to be done manually, that could end up being a significant expense. For multi-application devices, this calls back to a thing we were discussing and learning in objective two. Understand, will there be one certificate for application, one certificate for a device, or something else? And how you might like to do this might be constrained by what implementations support.

There are some implementations that only support a single certificate, covering all the PSIDs on the device. There are other implementations that support having more than one certificate and associating it with different PSIDs. And finally, if you're doing central signing of messages, for example, the MAPs and the TIMs, make sure that the RSU firmware supports both those modes of operation. It supports generating signatures locally for massages such as SPaT. It also supports receiving messages that are pre-timed and forwarding them without trying to sign them again. This is required by the RSU standards that is under development at the time of recording and should be published in early 2021. But obviously, it's worth checking for a specific deployment.

Are there lessons learned? First, make sure that certificate top-off is robust against losing connectivity mid top-off. Again, this calls back to a point we mentioned earlier, that the certificate updates process should be tested before failures that happen at different stages. The protocols are meant to be robust, but if you want to make sure that something is robust, always worth testing for failure cases. You want to ensure that certificate top-off succeeds, even if the device hasn't successfully topped off for some time. So, if a device has run out of valid certificates, if we're past the expiry time of all the certificates the device had, then again, the protocol should ensure that the device can successfully top off its certificates. But that's a case that's worth explicitly setting up and testing just in case there are any surprises.

If you have devices that request certificates individually for each time period—so we've talked earlier about the butterfly key mechanism that allows a device to send off a single certificate request, and then that turns into the generation of a large number of certificates. That's one model. Another model that is used more for RSUs, where butterfly keys are used typically more for OBUs. The model that's often used for RSUs is that every week, they want to get a new certificate for their application, and they send off a specific request for that certificate, so they don't use the butterfly keys approach. If you're doing that, make sure you explore what happens if the RSU sends off a request and fails to get a response, and then sends off another request. Does that result in two certificates being generated? Can the RSU download both of those certificates? The ideal outcome that you want is robustness, a device that's entitled to a single certificate only gets a single certificate. That provides protection against Sybil attacks. And finally, enrollment certificates.

Again, we've mentioned in one of the learning objectives in the first part of this course, enrollment certificates are the certificates that are used to authorize requests for authorization certs. And just like authorization certs, they have expiry times and lifetimes. Those lifetimes are typically a lot longer than authorization certs. They're typically a year or three years, but at some point, enrollment certs also expire and need to be rolled over. So, if your deployment is going to have a lifetime longer than the lifetime of the enrollment certs, you want to make sure that you test that transition as well, and make sure it goes smoothly.

So not only do your devices have but do the end entity devices have certificates, but of course, as we've mentioned, the CA have certificates as well, the enrollment CAs and the authorization CAs. Those certificates can also expire. So again, the expiry of certificates is a time when the system is fragile; it's in a transitional stage. You need to make sure that the certificate management software works across that expiry. It needs to work both for devices that are interacting directly with the CA and for devices that trust certificates issued by that CA. You don't want to accidentally get locked out because you no longer know the CA certificate that you will use to verify a message.

So, related to that, you need to make sure that new CA certificates can be distributed in a timely way. So, if I have a device that needs to access an access point, that access point uses a 1609.2 certificate to say, I'm a valid access point, and the CA certificate that issued the certificate for that access point is expired, then the reliant device can't trust the access point to use to update the CA certificates. We need to be certain that if authorization CA expiry is coming out, we've done everything that we can to make sure that the new authorization CA certificates are circulated in good time before the old ones expire. And finally, this is not so much technical and more of a process and institutional lesson learned.

Make sure there's a good way to get feedback to the SCMS manager and other government's body. If there are aspects of the design, if there are aspects of the deployment choices that are proving to be inconvenient or difficult to manage or causing significant expense or frustration, then the SCMS manager needs to know that. The SCMS supplier needs to know that so that information can be fed back into the system design and potentially into the standards themselves.

So now let's talk a little about misbehavior reporting and CRL download. We introduced misbehavior in learning objective two. Misbehavior is messages that have incorrect data in them that will cause bad outcomes. So, as we mentioned, if a vehicle says it's traveling at 300 miles an hour, clearly that's going to be a misbehavior. And in the pilot performance, there has been baseline behavior reporting mechanisms implemented. If particular types of constraint violations are detected in incoming messages, then the receiver will generate a misbehavior report, encrypt it for the misbehavior authority, and when they get an opportunity, they'll upload it to their RA, which will then forward it to the misbehavior authority. And some SCMS providers have implemented misbehavior authority that can receive those reports, do investigation, and potentially revoke or otherwise limit access to the system by devices that have been sending bad messages.

So we have this baseline misbehavior detection revocation mechanism. But if you're doing a deployment, you may want to have additional things that lead to revocation. Normally, if a vehicle is stolen, that isn't grounds for revocation of its BSM certificates because a stolen vehicle is still potentially a hazard and needs to be able to send these safety messages. It could be that for a specific deployment, there's a desire for only vehicles that are under the control of the deployment to send those BSMs. So you, as a deployer, may want to have a requirement that stolen vehicles have their vehicles revoked, and that should be communicated to the SCMS provider ahead of time. SCMS providers should be asked about performance metrics for revocation.

So what's the time between a report being submitted and the CRL being issued, based on the information in that report? The reason why that's important is, that gives you a sense of, if there's a device causing disruption in your system, how long is that disruption likely to persist before it's prevented by the SCMS, before the CRL comes out? So how long might you have to manage the effect of that disruption through local means?

Additionally, the protocols support devices individually requesting CRLs from their RA or from the distribution center, but CRLs are public information and can in principle be broadcast. So you may be interested in setting up some kind of mechanism that allows for CRLs to be broadcast from RSUs, or some other way of getting CRL information to the end entity devices, so they can be protected against misbehaving devices. There's no standard for this broadcast at the moment. You'd need to define it. The reason why there's no standard is that CRLs are typically large blobs of information and there's no native fragmentation or forward error correction mechanism in either DSRC or LTED decks. So some kind of fragmentation mechanism will need to be built on top of those lower level broadcast mechanisms.

So as you're thinking about misbehavior detection, misbehavior management, some questions you should discuss with the SCMS manager or SCMS provider are as follows: so if you're thinking about operational interactions with the SCMS, is there some other way to report devices that need to be revoked other than going through the misbehavior management channel? If you know that a device has been compromised and you don't want to have to wait for that compromised device to actually misbehave, is there a way you can jump the gun and get that device revoked before it's a position to cause damage?

Then there are questions you might want answers to, to help you understand the expected system behavior. So how much does the device need to misbehave in order to be revoked? Does a device need to be reported by multiple observers before it's revoked, or will a report from a single observer be enough? Are there different levels of trust in observers? If somebody gets reported as misbehaving by a fleet vehicle, owed by an IOO, does that report carry more weight inside the system than a report that comes from a privately owned vehicle? How quickly should it be possible to revoke a misbehaving device? Again, we mentioned this on the previous slide. What's the timeline likely to be between misbehavior starting to cause disruption in the system and that disruption being ended by the CRL publication? How frequently are CRLs published? And when there's a new CRL, how do devices know that that new CRL is available to be requested? And is there any process of repeal against revocation? If it turns out that a device is sending out bad messages, but it could be fixed by a software update, and that software update gets applied but the process is already in motion and the device gets revoked anyway, is there any way that that can be avoided? That the expense of reinstating a revoked device can be avoided and instead, the device can be reinstated by some other process?

And then questions for the device supplier and the SCMS provider to work on together. If a device has been revoked, so that means it's been seen to be misbehaving in some way that's hard to fix, how can it be trusted again? Is there a process to appeal against revocation that requires input from the supplier? And if a device is producing bad data because of a bad configuration, can we notify the supplier or the firmware provider and fix the device, rather than reporting it?

So now let's talk a little about these firewall issues that we've touched on previously: traffic on the infrastructure owners and operator network. So, most interactions between mobile devices and infrastructure in the V2X setting can naturally be confined to the traffic management network as they are primarily about exchanging local information for traffic management and planning purposes. But for the SCMS, the network traffic generated by that is different because its natural endpoint is outside the IOO network and remote on the internet. A device needs to contact the RA to download certificates. A device needs to contact the RA. So a device will need to go outside the IOO network to download certificates or to upload misbehavior reports. And in some cases, devices will have cellular connectivity and they can contact the SCMS directly without going through the IOO network. But for other devices, it may be necessary to provide internet access by the IOO network.

So RSUs will typically be only on the IOO network and won't have direct internet access for good management and anti-hacking reasons. And so, they will need specific consideration as to how traffic from them can be enabled to access the broader internet for purposes of reaching the RA. And since network connectivity is required for the certificate update, even devices that in principle do have cellular connectivity for updates, they may need the ability to use the IOO network connectivity to connect to the SCMS in case there's a failure of that cellular connectivity, for example, of the cellular subscription expires. These devices will need to contact the SCMS by the IOO network. The SCMS component will be outside the IOO network in general.

The way the SCMS architecture works, there's a lot of traffic between the RA and the ACA rather, the authorization CA. That authorization CA is typically not suited to being run inside an IOO network because of its physical security requirements. So it makes most sense for the entire set of SCMS operations to take place outside the IOO network, so there needs to be some way of accessing the SCMS from within the IOO network. And each IOO network will address this question in their own way, depending on local network configuration and security requirements, but you need to think through the implications early in the process in case re-architecture or reconfiguration of the network is going to be required.

Then we've talked about misbehavior detection, and that, when we say misbehavior, that's typically within the context of the actual V2X application, so bad BSMs, bad SPaTs, that kind of thing. But the network as a whole is subject to cyberattacks and those cyberattacks might result in bad V2X messages being sent. They might corrupt the MAP or the TIM or the SPaT data, and so a system where you're deploying any V2X applications, as well as having misbehavior reporting, you need to be aware of all the possible threats within the system, denial of service and standard network cyberattacks.

Any network should already have network monitoring and other security mechanisms in place. But part of the deployment process should be to review those network monitoring and other security mechanisms, make sure if the V2X deployment introduces new threats, either from V2X activity threatening the network or from a hacker on the network threatening V2X activities. If there are new threats, then the security posture, the security mechanisms need to be reviewed to make sure those new threats are protected against. It's conceivable that this is something that the SCMS provider will need to be in the loop on, but the SCMS provider will in general want to only issue certificates to devices that it has a good level of assurance will send out correct data. And so if there's outstanding unmitigated network threats, that's something the SCMS supplier is going to want to be aware of.

Finally, there's data management considerations to think about. This is not strictly within the purview of the SCMS, but it's part of the security review process, security design process that needs to be gone through, and coming up with a data management plan is one of the activities to be done when you're developing your security management operating concept, your SMOC. So many of these connected vehicle applications are going to result in generating large amounts of data that could potentially be personally identifying, particularly those applications that involve data originating from cars. You shouldn't be gathering that data and you shouldn't be deploying applications that will generate that data unless there's a side data management plan to ensure that the data is properly handled.

As I say, this isn't part of the SCMS necessarily, but one of the jobs that the SCMS provider is to be kind of a gatekeeper on the security front to make sure that all the i's are dotted and the t's are crossed, when it comes to implementing appropriate security mechanisms. So, it's conceivable that the SCMS provider will require to know that there's a data management plan before certificates will be issued. And so that's, again, another discussion that needs to be had between the deployment manager and the SCMS provider. There's a privacy analysis process that you can use, developed by NIST and DOT. There's a privacy analysis process that you can use, developed by NIST and DOT (Department of Transportation), that's for use with connected vehicle deployments, and the link to that will also be in the student supplement.

So that completes the content of the course. We have one more activity, one more question to answer. Which of these is the correct statement about data collection and management? Choices are, only vehicles can produce personally identifying information? Or individuals must give consent to their data being collected? Or if there is concern that data may reveal driver behavior that violates the law, it should be immediately shared with law enforcement? Or data must be managed in a manner consistent with local data protection regulations?

Let's have a look at the answers, and the correct answer is d) Data must be managed in a manner consistent with local data protection regulations, and that's kind of just a special case of the general case, that all activity needs to be consistent with local regulations, one way or another. A deployer must be aware of local data protection regulations and ensure that they are complied with. Looking at the wrong answers, we said only vehicles can produce personally identifying information, we gave that as an option. Obviously, that's incorrect. If you have pedestrian devices, they can generate personally identifying information, or fixed devices like cameras can generate information that might be linked to individuals. Individuals must give consent to their data being collected. That is incorrect. It's a good principle, but it can overridden, depending on the applicable local data protection regulations.

Erring on the side of requiring consent is not bad practice, but as I say, you need to review the regulations and see if there are other conditions, and also see, are there conditions under which data can be collected without explicit consent by individuals. And finally, if there's concern that they may reveal driver behavior or violate the law, it should be immediately shared with law enforcement. Again, this may be good practice, but it's not a requirement. This will depend on the applicable local data protection regulation and other laws, but in general, there's no requirement to be proactive about sharing data with law enforcement.

So, to summarize in this part of the course, we have looked at the V2X certification process for a device to enroll in the SCMS. Virendra led that part. We looked at the hardware security requirements and other requirements on the device. And we also talked about how to make a deployment plan that used SCMS services, and in that learning objective, we focused on security requirements on the network as a whole, security considerations for the network, and questions that you should be asking your SCMS provider well in advance of starting on the actual deployment.

To review the whole course, we've looked at these five different learning objectives: to define the communications security requirements in the connected vehicle environment; to describe how the SCMS uses cryptographic building blocks to provide trust; to understand how to get devices interacting with the SCMS in a deployment. And then the two learning objectives we addressed in this part, to identify the V2X certification process for a device to enroll in the SCMS and to illustrate how to make a deployment plan that uses SCMS services. So, this was part 2 of 2.

I hope you found both part 1 and this part, part 2, to be interesting and informative, and of use to you in your future work. Part 1, part 2. Thank you for completing the module. There's a feedback link below. Please use that to provide us with your thoughts and comments about the value of the training. But from myself, and from my colleague, Virendra Kumar, thank you very much and goodbye.