Mechanizmy i technologie bezpieczeństwa dla ograniczonych urządzeń IoT

Pojawiający się paradygmat Internetu rzeczy (IoT) jest powszechnie identyfikowany jako rosnący trend technologii informacyjnej w kierunku wszechobecnego społeczeństwa sieciowego, w którym wszystkie urządzenia, które mogą korzystać z połączenia, będą ze sobą połączone. Oznacza to, że oprócz tradycyjnych urządzeń i komputerów osobistych oczekuje się, że wszystkie rodzaje sprzętu elektronicznego, które mogą korzystać z połączenia, będą dostępne i dostępne online. W tak masowo połączonym scenariuszu bezpieczeństwo zostało natychmiast uznane za niezwykle ważny wymóg do spełnienia. Z jednej strony dobrze znane zagrożenia bezpieczeństwa, luki i ataki z tradycyjnych systemów informatycznych (IT) są w naturalny sposób dziedziczone. Dlatego protokoły komunikacji i zarządzania zaproponowane i przyjęte w IoT zapewniają już szereg podstawowych usług bezpieczeństwa. Z drugiej strony, wiele dodatkowych wektorów ataków na bezpieczeństwo jest dostępnych przeciwko łatwiejszym do namierzenia urządzeniom IoT, głównie z dwóch głównych powodów. Przede wszystkim długofalowa wizja IoT zakłada, że większość urządzeń będzie bezpośrednio podłączona do Internetu, aby była bezpośrednio dostępna. To również sprawia, że są one bezpośrednio narażone na kilka rodzajów ataków bezpieczeństwa, zwłaszcza ataków typu „odmowa usługi" (DoS), które w innym przypadku byłyby trudniejsze i mniej wygodne do przeprowadzenia przez przeciwnika. Po drugie, oczekuje się, że typowe środowisko IoT będzie zawierać znaczną liczbę urządzeń o ograniczonych zasobach, często zasilanych bateryjnie. To nie tylko sprawia, że duży zestaw urządzeń IoT jest znacznie bardziej podatny na ataki i mniej zdolny do radzenia sobie z atakami na bezpieczeństwo, ale także powoduje dodatkowe wyzwanie związane z projektowaniem, wdrażaniem i przyjmowaniem rozwiązań bezpieczeństwa, które są również przystępne cenowo dla ograniczonych urządzeń IoT i nie są znacząco wpływ na wydajność i skalowalność całego środowiska sieciowego i scenariusz aplikacji IoT. W tej części omówiono najważniejsze i najważniejsze praktyczne aspekty dotyczące bezpieczeństwa w IoT. W szczególności, najpierw omawiamy najważniejsze protokoły i mechanizmy bezpieczeństwa IoT, które są przyjęte w typowym stosie komunikacyjnym IoT i są obecnie uwzględniane w bieżących działaniach normalizacyjnych lub zostały już wydane jako standardy. Obejmują one na przykład podpisywanie i szyfrowanie obiektów CBOR, OAUTH 2.0, Object Security for CoAP, DTLS i skompresowany IPsec. W tej części przedstawiono również szereg konkretnych zagadnień związanych z bezpieczeństwem w Internecie Rzeczy oraz omówiono możliwe rozwiązania tych problemów w oparciu o najnowsze wyniki społeczności naukowej i działania w organach normalizacyjnych. Szczególną uwagę poświęca się przeciwdziałaniu atakom DoS na protokoły i sieci IoT, zachowaniu i poprawie wysokiej wydajności i skalowalności w obecności usług bezpieczeństwa, wydajnym i skalowalnym protokołom do zarządzania kluczami kryptograficznymi oraz wykrywaniu i przeciwdziałaniu atakom bezpieczeństwa.

Bezpieczeństwo w protokołach i technologiach IoT

Internet Rzeczy może zmienić sposób, w jaki pracujemy i żyjemy na co dzień. Jeśli istnieje jeden powód, który może ograniczyć to przejście, będzie to mędrzec w zakresie bezpieczeństwa i prywatności. Chociaż oferowanie silnego bezpieczeństwa jest już trudne w Internecie, w rzeczywistości jest znacznie trudniejsze w IoT, ponieważ „rzeczy" będą obsługiwać protokół internetowy w wersji 6 (IPv6), być globalnie osiągalne i niezwykle heterogeniczne (obejmujące urządzenia czujnikowe , smartfony, standardowe komputery, a nawet środowiska chmurowe) i zwykle są wdrażane w środowiskach niechronionych fizycznie i nienadzorowanych. Co więcej, większość z nich nie zapewnia żadnego konwencjonalnego interfejsu użytkownika, takiego jak wyświetlacz czy klawiatura. Jednocześnie środowiska IoTconstrained dziedziczą ograniczenia typowe dla konwencjonalnych bezprzewodowych sieci czujnikowych (WSN), na przykład ograniczone zasoby przetwarzania i energii, wieloskokowe topologie komunikacyjne i stratne łącza bezprzewodowe. Aby sprostać tym wyzwaniom,

opracowywane i standaryzowane są różne protokoły dla urządzeń i sieci IoT. W tej sekcji przedstawiono typowe protokoły i mechanizmy zabezpieczeń IoT, które są obecnie stosowane w scenariuszach aplikacji IoT i przypadkach użycia. W szczególności przechodzi przez typowy stos sieciowy zgodnie z podejściem odgórnym i przedstawia usługi bezpieczeństwa, na których opierają się protokoły IoT lub które zapewniają bezpośrednio. Nakreśla również typowy stos IoT z różnymi protokołami bezpieczeństwa i technologiami na różnych warstwach urządzenia IoT. W warstwie aplikacji OAuth 2.0 jest dostosowywany do urządzeń IoT w celu wprowadzenia precyzyjnych mechanizmów kontroli dostępu w IoT; bezpieczeństwo obiektów jest standaryzowane w celu ochrony poszczególnych elementów danych; a protokół Constrained Application Protocol (CoAP) stał się nowym standardem sieciowym dla IoT, który może przenosić komunikaty o zabezpieczeniach obiektów i OAuth. W warstwie transportowej dla urządzeń IoT preferowany jest bezpołączeniowy UDP, a protokół Datagram Transport Layer Security (DTLS) jest powiązany z CoAP, tworząc bezpieczny protokół CoAP (CoAPs). Alternatywnie możliwe jest użycie protokołu Internet Key Exchange w wersji 2 (IKEv2) do dynamicznego ustanawiania i zarządzania skojarzeniami bezpieczeństwa i powiązanym materiałem klucza, gdy preferowane jest przyjęcie IPsec (lub bezpieczeństwa sieci). Co więcej, IPv6 wyświetla potencjalnie nieograniczoną przestrzeń adresową i jest de facto protokołem sieciowym i adresowym dla IoT, podczas gdy protokół routingu IPv6 dla sieci o niskim poborze mocy i sieci stratnych (RPL) jest standardowym rozwiązaniem do routingu pakietów w ramach ograniczonej, niskiej moc i sieci stratne . Ponadto 6LoWPAN został ustandaryzowany w celu zapewnienia mechanizmów kompresji/dekompresji i fragmentacji/reasemblacji, a tym samym dopasowania większych pakietów do ramek IEEE 802.15.4 (lub podobnych). Wreszcie istnieje szereg możliwości w warstwie łącza i warstwy fizycznej. Ponieważ IEEE 802.15.4, a także Bluetooth Low Energy (BLE), mają ogromny potencjał, omawiane jest również bezpieczeństwo w ich obecności.

Lekkie formaty protokołów kryptobezpieczeństwa

Ponieważ oczekuje się, że wiele urządzeń IoT będzie ograniczonych zasobami, istnieje potrzeba zapewnienia usług bezpieczeństwa w szczególnie wydajny sposób, a tym samym zmniejszenia ilości informacji, które mają być przechowywane, przesyłane i przetwarzane. Oznacza to, że nie tylko zwykła zawartość na poziomie aplikacji, ale także zaszyfrowana zawartość, podpisy cyfrowe, a nawet klucze kryptograficzne muszą być reprezentowane i kodowane w zwarty i wydajny sposób. W tym celu CBOR Object Signing and Encryption (COSE) ma na celu tworzenie formatów kryptograficznych opartych na zwięzłej binarnej reprezentacji obiektów (CBOR). CBOR to zwięzły format binarny do serializacji struktur danych, odpowiedni dla urządzeń o ograniczonych zasobach. Znaczna część przewidywanych miliardów urządzeń IoT będzie składać się z urządzeń zasilanych bateryjnie lub gromadzących energię. W przypadku tych urządzeń IoT COSE ma ogromny potencjał do tworzenia lekkich reprezentacji kluczy kryptograficznych, szyfrowania, skrótów wiadomości i podpisów cyfrowych opartych na CBOR. COSE jest inspirowany JavaScript Object Signing and Encryption (JOSE), który już ustandaryzował formaty kryptograficzne przy użyciu JavaScript. Chociaż COSE próbuje ponownie wykorzystać funkcje JOSE, nie są one bezpośrednio kompatybilne do prostego przyjęcia. Jedną z głównych różnic, jakie przynosi CBOR w porównaniu z JOSE, jest bezpośrednie użycie formatu danych binarnych bez konieczności wcześniejszej konwersji ich na np. ciąg zakodowany w base64. Składnia zakodowanych komunikatów CBOR szczegółowo opisuje podstawową strukturę COSE i wspólne nagłówki COSE. Ponadto opracowuje obiekty COSE za pomocą dedykowanych materiałów kluczy kryptograficznych, a także za pomocą kryptoalgorytmów do szyfrowania/uwierzytelniania danych oraz do obliczania kodów uwierzytelniania wiadomości (MAC). Istnieją już propozycje kodowania parametrów, kluczy i wyników RSA oraz innych algorytmów jako komunikatów COSE. Również lekka reprezentacja COSE struktur danych kryptograficznych oraz ich implementacja i ocena w porównaniu z JOSE może być interesującym tematem badawczym.

CoAP i DTLS

CoAP to nowatorski protokół przesyłania danych przez Internet, zaprojektowany specjalnie dla urządzeń i sieci o ograniczonych zasobach. Opiera się na interakcjach typu żądanie/odpowiedź między punktami końcowymi i może łatwo zintegrować się z szeroko stosowanym protokołem Hyper Text Transfer Protocol (HTTP) w celu integracji z klasyczną siecią WWW. Chociaż protokół CoAP został zaprojektowany z myślą o ograniczonych urządzeniach i sieciach, z drugiej strony nie zapewnia sobie żadnych szczególnych prymitywów do bezpiecznej komunikacji. Zamiast tego, wiadomości CoAP mogą być chronione albo za pomocą zabezpieczenia obiektu, albo za pomocą rzeczywistych bezpiecznych protokołów komunikacyjnych. W takim przypadku specyfikacja CoAP zaleca przyjęcie protokołu DTLS. W szczególności Shelby i in. określ powiązanie CoAP z DTLS jako zestaw delt do zwykłego niezabezpieczonego protokołu CoAP. Z praktycznego punktu widzenia, jeśli wiadomość CoAP jest zabezpieczona przy użyciu DTLS, rozważany jest schemat coaps://URL, a nie coap://scheme w przypadku niezabezpieczonej komunikacji. W szczególności dostępne są trzy tryby zabezpieczeń PreSharedKey, RawPublicKey i Certificate. Oznacza to, że tryb PreSharedKey opiera się na wstępnie udostępnionych kluczach kryptograficznych, z których każdy zawiera listę urządzeń, które mogą używać tego klucza do bezpiecznej komunikacji. Jeśli rozważany jest RawPublicKey, urządzenie jest właścicielem pary asymetrycznych kluczy prywatno-publicznych bez skojarzonego certyfikatu, a zatem powierza weryfikację mechanizmom pozapasmowym. Ponadto urządzenie utrzymuje tożsamość obliczoną na podstawie klucza publicznego, a także listę urządzeń, z którymi może się komunikować. Wreszcie, w przypadku, gdy rozważa tryb certyfikatu, urządzenie przechowuje asymetryczną parę kluczy prywatny - publiczny wraz z certyfikatem X.509. Certyfikat wiąże klucz publiczny z jego podmiotem i zawiera podpis cyfrowy z dobrze znanego źródła zaufania (Cooper i in., 2008). W szczególności urządzenie dodatkowo utrzymuje listę głównych kotwic zaufania, do których można się odnieść w przypadku konieczności przeprowadzenia walidacji certyfikatu. W praktyce DTLS umożliwia dwóm urządzeniom ustanowienie powiązania bezpieczeństwa, wzajemne uwierzytelnianie i wymianę chronionych komunikatów CoAP. Internet Engineering Task Force (IETF) opublikował niedawno profil dla Transport Layer Security (TLS) i DTLS 1.2. Profil zapewnia bezpieczeństwo komunikacji dla urządzeń o ograniczonych zasobach wykorzystywanych do zbierania danych za pomocą czujników lub do sterowania siłownikami w aplikacjach przemysłowych systemów sterowania, automatyki domowej, inteligentnych miast i innych wdrożeń IoT. Protokół DTLS został dobrowolnie pomyślany jako bardzo podobny do protokołu TLS iw rzeczywistości spełnia równoważne wymagania bezpieczeństwa. Oznacza to, że umożliwia aplikacjom działającym na klientach i jednostkach serwerowych wymianę bezpiecznych wiadomości, czyli rekordów DTLS, w szczególności zapobiegających manipulacjom, fałszerstwo i podsłuchiwanie takich wiadomości. Jednocześnie DTLS wykazuje szereg różnic w porównaniu z TLS, dzięki czemu możliwe jest zapewnienie bezpiecznej wymiany komunikatów w oparciu o UDP i inne zawodne protokoły transportu datagramów. Na przykład nie można używać RC4 i innych szyfrów strumieniowych. Poza tym w każdym komunikacie DTLS jest wyraźnie zawarta wartość numeru sekwencyjnego. W ten sposób wiadomości DTLS nie są ze sobą powiązane, co umożliwia odbiorcom ich prawidłowe przetwarzanie, nawet w przypadku dostarczenia poza kolejnością. Ponadto DTLS rozwiązuje problem utraty pakietów poprzez retransmisję wiadomości i lokalne limity czasu. Po odebraniu nieprawidłowej wiadomości powiązana sesja DTLS może nadal zostać zachowana, a wiadomość ta może zostać po prostu po cichu odrzucona. W celu wymiany chronionych wiadomości przez DTLS dwa urządzenia klienckie i serwerowe wykonują proces uzgadniania w celu ustanowienia bezpiecznej sesji. Komunikaty uzgadniania przesyłane na tym samym etapie są grupowane razem w ramach jednego lotu DTLS. Komunikaty opcjonalne lub zależne od sytuacji są oznaczone w nawiasach kwadratowych i nie zawsze są wysyłane. Klient zazwyczaj przejmuje inicjatywę i wysyła wiadomość ClientHello do serwera w celu rozpoczęcia nawiązywania sesji DTLS. DTLS dopuszcza bezstanową

wartość cookie, która może być opcjonalnie wymieniana między serwerem a klientem za pomocą Fligth2 i Flight3, z zamiarem zaadresowania DoS przeciwko serwerowi. Następnie klient i serwer wymieniają się Flight4 i Flight5 w celu wzajemnego uwierzytelnienia, negocjowania zestawu szyfrów i parametrów bezpieczeństwa oraz ustalenia materiału klucza kryptograficznego. Wreszcie Flight6 potwierdza, że sesja DTLS została w pełni ustanowiona po obu stronach, a zatem klient i serwer mogą rozważyć ją w celu bezpiecznej komunikacji. Czytelnik może odwołać się do Dierks i Rescorla (2008) oraz (Rescorla i Modadugu), (2012) w celu uzyskania dodatkowych informacji dotyczących uścisku dłoni i specyficznego formatu lotów DTLS. Przed rozpoczęciem uzgadniania należy zapewnić dwóm partnerom DTLS wstępne materiały zabezpieczające. Obejmuje to na przykład preinstalowane klucze kryptograficzne, które są wykorzystywane do ustanowienia sekretu przedwzorcowego podczas uzgadniania. Następnie klient i serwer wykorzystują ten klucz tajny przedwzorcowy wraz z generowanymi przez siebie losowymi wartościami w celu uzyskania tajnego klucza głównego, z którego ostatecznie uzyskuje się rzeczywisty materiał zabezpieczający dla sesji DTLS. W celu dystrybucji preinstalowanych kluczy można rozważyć dwa główne podejścia. Pierwszy z nich wykorzystuje klucze asymetryczne. Oprócz typowego użycia certyfikatów X.509 (Cooper et al., 2008), dostępne są również surowe profile kluczy publicznych, w których klucze prywatno-publiczne są dostarczane bez certyfikatu. W szczególności takie klucze mogą być generowane przez producentów urządzeń, a następnie instalowane przed wdrożeniem urządzenia. W takim przypadku urządzenie weryfikuje surowe klucze publiczne otrzymane od innych partnerów za pomocą technik pozapasmowych. W szczególności zwykle przechowuje listę peerów, z którymi może się komunikować. Zamiast tego alternatywne podejście wykorzystuje symetryczne klucze wstępne (PSK). Oznacza to, że klient przechowuje pewną liczbę kluczy symetrycznych, z których każdy jest współdzielony z serwerem, z którym klient może się komunikować. Następnie, podczas uzgadniania, klient określa konkretny klucz do użycia, wskazując powiązaną tożsamość PSK. Na koniec klient i serwer uzyskują klucz tajny DTLS premastera z klucza symetrycznego, który współdzielili. Przyjęcie podejścia opartego na kluczu wstępnym jest szczególnie wygodne w środowiskach, które są zamknięte i w których łatwo można dostarczać klucze kryptograficzne do wdrożonych urządzeń. Co więcej, nie wymaga wysyłania i odbierania certyfikatów publicznych ani angażowania się w asymetryczne operacje kryptograficzne, które są kosztowne i szczególnie ważne w ograniczaniu serwerów DTLS o ograniczonych zasobach. Poza tym znacznie ułatwia operacje zarządzania kluczami, szczególnie we wdrożeniach IoT, w których ręczna i wczesna konfiguracja połączeń jest często powszechną praktyką, a zatem dostarczanie certyfikatów często okazuje się niepreferowaną lub nawet niewykonalną opcją.

Bezpieczeństwo obiektów w ograniczonych środowiskach REST

Standard CoAP określa możliwe użycie serwerów proxy jako podmiotów pośredniczących między klientami i serwerami CoAP w celu poprawy wydajności i skalowalności w wielu scenariuszach sieci i aplikacji. Jednocześnie, jak omówiono wcześniej w tej części, CoAP odwołuje się do protokołu DTLS w celu zapewnienia bezpiecznej komunikacji. Stanowi to problem w obecności pośredniczących serwerów proxy, ponieważ te ostatnie muszą być w stanie uzyskać dostęp i ewentualnie manipulować określonymi częściami komunikatu CoAP, aby wykonać zamierzoną funkcjonalność proxy. W konsekwencji operacje proxy na komunikatach CoAP wymagają, aby komunikacja DTLS od klienta CoAP kończyła się na proxy. Z drugiej strony, nadal jest bardzo ważne, aby uniemożliwić serwerowi proxy dostęp i/lub manipulowanie częściami komunikatów CoAP, które nie są ściśle konieczne do prawidłowego wykonania zamierzonej funkcjonalności proxy. W szczególności przyczyniłoby się to również do zmniejszenia i złagodzenia założeń dotyczących bezpieczeństwa i zaufania w jednostkach proxy rozmieszczonych w sieci. W tym celu wygodnie byłoby polegać na bezpiecznym protokole komunikacyjnym, który jest w stanie chronić komunikaty CoAP w sposób kompleksowy, nawet gdy obecne są węzły pośredniczące. Dokument opisywał odpowiednią propozycję o nazwie Object Security

for Constrained RESTful Environments (OSCORE). W zasadzie OSCORE jest protokołem bezpieczeństwa opartym na obiektach danych, umożliwiającym wymianę komunikatów CoAP, które są chronione od końca do końca między węzłami pośredniczącymi. OSCORE wykorzystuje obiekty COSE w celu zapewnienia integralności, pełnego szyfrowania i ochrony przed ponownym odtwarzaniem wiadomości CoAP. Użycie OSCORE sygnalizowane jest w pierwszej kolejności poprzez włączenie nowo zdefiniowanego Object SecurityCoAPoption do komunikatów secureCoAP. Jako możliwą alternatywę, ochrona komunikatów może być ograniczona tylko do ładunku pojedynczych komunikatów, zgodnie z podobnym podejściem o nazwie obiektowe zabezpieczenie treści (OSCON). Kompleksowe usługi bezpieczeństwa dostarczane przez OSCORE są również szczególnie pożądane w ograniczonych ustawieniach sieci, gdzie rozwiązania takie jak DTLS nie mogą być obsługiwane. Z drugiej strony OSCORE można łączyć i używać razem z DTLS. Oznacza to, że OSCORE może umożliwić kompleksowe zabezpieczenie komunikatów CoAP między serwerem CoAP a klientem CoAP, wraz z ochroną całych komunikatów CoAP (tj. łącznie z nagłówkiem) w trybie hop-byhop zapewnianym przez DTLS podczas transportu komunikatów między punktem końcowym a węzłem pośredniczącym, takim jak serwer proxy. Bardziej praktycznie, OSCORE zakłada, że kontekst bezpieczeństwa został wstępnie ustalony i uzgodniony między klientem CoAP a serwerem CoAP uważanym za punkty końcowe bezpiecznej komunikacji. W szczególności kontekst bezpieczeństwa to zestaw informacji i parametrów, których dwa punkty końcowe OSCORE używają do wykonywania określonych operacji kryptograficznych. Ponadto OSCORE jest bardzo podobny do DTLS i zapożycza wiele jego mechanizmów, takich jak wyprowadzanie kluczy i konstruowanie wartości nonce. Jednak OSCORE opiera się na COSE, a nie na warstwie rekordów DTLS, dzięki czemu może używać identyfikatorów kontekstu bezpieczeństwa i numerów sekwencyjnych o zmiennej długości.

Skompresowany protokół IPsec

Standard 6LoWPAN określa, w jaki sposób ciężkie datagramy IPv6 mogą być przesyłane w sieciach o małej mocy i stratnych w oparciu o IEEE 802.15.4 (IEEE Computer Society, 2011). Aby to osiągnąć, 6LoWPAN wprowadza szereg schematów kompresji nagłówków, które są w stanie znacznie zmniejszyć rozmiar nagłówków UDP, datagramów IP i rozszerzeń IP. W sieciach IP opartych na IEEE 802.15.4, zwanych również sieciami 6LoWPAN, bezpieczeństwo jest szczególnie ważne ze względu na dużą ekspozycję na Internet i wynikającą z tego podatność sieci. Obecnie zabezpieczenia IP (IPsec) to standardowe rozwiązanie zabezpieczające dla protokołu IPv6. W szczególności hosty IPv6 w Internecie powinny go implementować i być w stanie obsługiwać i przetwarzać pakiety chronione przez IPsec. W szczególności tryb transportu protokołu IPsec zapewnia bezpieczną komunikację typu end-to-end między dwoma węzłami w Internecie. Dlatego celowe jest dostosowanie 6LoWPAN w celu umożliwienia komunikacji IPsec między elementami obsługującymi IPv6 (np. węzłami czujników) w sieciach 6LoWPAN i ogólnymi węzłami IPv6 w Internecie. Przedstawiono wiele różnych propozycji kompresji nagłówków pakietów IPsec. Większość schematów kompresji jest dostosowana do ogólnych hostów internetowych i nie uwzględnia w szczególności sieci 6LoWPAN złożonych z urządzeń o ograniczonych zasobach. Migault i inni zaproponowali Diet-ESP, które określają sposób przeprowadzania kompresji pakietów IPsec, ale wymaga szeregu zmian i adaptacji w konwencjonalnych hostach w Internecie. Ponadto RObust Header Compression (ROHC) opiera się na elastycznej i wydajnej koncepcji kompresji nagłówków, ale jest przeznaczona do ogólnych hostów w Internecie, a zatem nie jest specjalnie przeznaczona dla sieci 6LoWPAN. Dlatego Raza i in. proponują skompresowany protokół IPsec 6LoWPAN, który jest przeznaczony przede wszystkim dla urządzeń i sieci IoT o ograniczonych zasobach. Schematy Diet-ESP i ROHC wraz z Generic Header Compression są dodatkiem do skompresowanego IPsec opisanego u Razya i inych , gdzie mechanizmy kompresji nagłówka są przeznaczone specjalnie dla sieci 6LoWPAN, działają z ogólnymi hostami internetowymi używającymi IPsec i nie powodują żadnej zmiany standardów Encapsulated Security Protocol (ESP) i

IPsec Authentication Header (AH). Obecnie najczęściej używany jest protokół ESP IPsec, który zapewnia zarówno poufność, jak i integralność ładunku ESP. W przypadku IoT przewidziano szereg przypadków użycia, w których przydatne jest, aby ESP, a także AH, chroniły integralność nagłówków IPv6 oprócz ładunków IPsec. Z różnych powodów korzystanie z AH w Internecie jest obecnie ograniczone. Oznacza to, że AH nie jest kompatybilny z translacją adresów sieciowych (NAT), która modyfikuje źródłowy adres IP podczas przesyłania, co powoduje niepowodzenie sprawdzania integralności pakietu i ostatecznie odrzucenie pakietu na hoście IPSec odbiorcy. Co więcej, ESP jest w stanie zapewnić zarówno szyfrowanie, jak i ochronę integralności pakietów IPsec, ale umożliwia tylko zabezpieczenie danych aplikacji i nagłówka ESP, pozostawiając nagłówek IP niezabezpieczony. Praktycznie jest to w porządku w przypadku korzystania z trybu tunelowego protokołu IPsec, ponieważ wewnętrzne dane aplikacji, nagłówek ESP i nagłówek IP są chronione zarówno poufnością, jak i integralnością. Ponieważ scenariusze IoT opierające się na łączności IPv6 nie obejmują translacji adresów sieciowych, możliwe i korzystne jest użycie trybu transportu w celu zapewnienia bezpieczeństwa typu end-to-end. Dlatego używanie AH wraz z ESP ma szczególnie sens w IoT. W szczególności, ponieważ kontrola integralności AH uwzględnia adres IP, protokół IPsec może być skutecznie wykorzystany do zapewnienia ochrony przed fałszowaniem adresów IP, które jest jednym z najczęstszych i najbardziej prawdopodobnych ataków na zabezpieczenia przeciwko urządzeniom o ograniczonych zasobach komunikujących się za pośrednictwem protokołu IPv6. Ponadto, mimo że zaproponowano bezstanową autokonfigurację adresów IPv6, nie uważa się tego za rzeczywisty wymóg do spełnienia. Oznacza to, że węzły o ograniczonych zasobach w sieciach 6LoWPAN są przypisywane do adresów IPv6 w czasie wdrażania i prawdopodobnie utrzymują ten sam adres przez cały okres ich istnienia, z wyjątkiem przypadków ręcznej interwencji za pomocą aktualizacji oprogramowania układowego/oprogramowania. Z drugiej strony, praktycznie niewykonalne jest zajęcie się autokonfiguracją w sieciach 6LoWPAN w sposób zapewniający łączność typu end-to-end, chyba że zostanie zaprojektowany i opracowany odpowiedni i wydajny mechanizm ukierunkowany specjalnie na sieci 6loWPAN; jest to w rzeczywistości interesujące wyzwanie badawcze. Zauważ, że nawet jeśli tylko jedna aplikacja działa na węźle 6LoWPAN, IPv6 potencjalnie zapewnia nieograniczoną przestrzeń adresową. Dlatego można sobie pozwolić na luksus zarezerwowania wielu różnych adresów IPv6 dla jednego pojedynczego węzła 6LoWPAN. W konsekwencji umożliwia to konfigurowanie unikalnych skojarzeń zabezpieczeń IPsec dla każdej aplikacji. Ponadto w przypadku, gdy protokół IKEv2 jest konkretnie uważany za protokół zarządzania kluczami, możliwe jest dynamiczne ustanawianie unikalnych skojarzeń zabezpieczeń dla poszczególnych aplikacji. Poza tym wiele przypadków użycia, takich jak uruchamianie alarmów i proste monitorowanie środowiska, wymaga jedynie ochrony integralności, dlatego użycie AHonly jest wygodnym wyborem. W chwili pisania tego tekstu istnieje oczekujący projekt standardu IETF, który definiuje faktyczne kodowanie IPsec AH i ESP.

IEEE 802.15.4 i Bluetooth Low Energy

Standard IEEE 802.15.4 zawiera specyfikacje dla warstwy fizycznej i warstwy kontroli dostępu do nośnika (MAC) przeznaczonej dla bezprzewodowych sieci osobistych o niskiej przepływności. Powierza innym protokołom, takim jak ZigBee lub WirelessHART, dodatkowe usługi w wyższych warstwach, lub alternatywnie może polegać na 6LoWPAN jako warstwie adaptacyjnej, która współpracuje ze standardowymi protokołami internetowymi. Ponadto warstwa MAC IEEE 802.15.4 bezpośrednio zapewnia szereg usług bezpieczeństwa, tj. autentyczność danych, poufność danych i ochronę powtórek na podstawie pakietu (IEEE Computer Society, 2011). W szczególności standard odnosi się do pakietu kryptograficznego opartego na 128-bitowej kryptografii klucza symetrycznego Advanced Encryption Standard (AES) (National Institute of Standards and Technology, 2001). W przypadku, gdy włączona jest bezpieczna komunikacja, pomocniczy nagłówek bezpieczeństwa (ASH) jest dołączany obok standardowego nagłówka MAC każdego pakietu, który jest zabezpieczony podczas transmisji na

podstawie tego, co określono w ASH. Węzeł odbiorczy odbiera i prawidłowo rozpina pakiet zgodnie z tymi samymi informacjami. W szczególności IEEE 802.15.4 zapewnia trzy różne tryby bezpieczeństwa, to jest CTR (tylko szyfrowanie), CBC MAC (tylko uwierzytelnianie) i CM (szyfrowanie i uwierzytelnianie). Tryby CCM i CBC MAC używają kodu integralności wiadomości (MIC), który może mieć rozmiar 4, 8 lub 16 bajtów. Operacje bezpieczeństwa w IEEE 802.15.4 wykorzystują nową wartość nonce. W szczególności jednorazowy generowany jest losowo po stronie nadawcy i zapewnia ochronę przed atakami typu powtórka. Różne dostępne tryby bezpieczeństwa i różne dostępne konfiguracje umożliwiają wymianę określonych ograniczeń aplikacji z wymaganiami dotyczącymi bezpieczeństwa i wydajności. Wreszcie, standard IEEE 802.15.4 nie opisuje, jak ustanowić i rozpowszechniać kluczowy materiał w sieci lub jak adresować uwierzytelnianie urządzenia. Zamiast tego zakłada, że obie takie usługi bezpieczeństwa są świadczone i wymuszane przez wyższe warstwy. W konsekwencji standard zakłada, że dana para węzłów nadawcy i odbiorcy pomyślnie uzgodniła te same ustawienia bezpieczeństwa i udostępnił wspólny materiał klucza, zanim będą mogli rozpocząć bezpieczną komunikację. Te same usługi bezpieczeństwa są dostarczane w rozszerzonym standardzie IEEE 802.15.4e, który rozszerza poprzedni standard IEEE 802.15.4 w celu ukierunkowania na aplikacje wbudowane o krytycznych wymaganiach (np. zastosowania medyczne lub przemysłowe) (IEEE Computer Society, 2012). W szczególności definiuje on tryb zachowania MAC w trybie Time Slotted ChannelHopping (TSCH), który łączy funkcje wielokanałowe i przeskakiwania kanałów z dostępem do szczelin czasowych. Umożliwia to nie tylko zapewnienie lepszej wydajności pod względem wysokiej niezawodności, dużej przepustowości sieci, efektywności energetycznej i przewidywalnych opóźnień, ale także przyczynia się do zwiększenia odporności sieci na ataki bezpieczeństwa montowane w warstwie fizycznej, takie jak zagłuszanie i zakłócenia radiowe. , dzięki mechanizmom przeskakiwania kanałów. Podczas gdy z jednej strony IEEE 802.15.4 jest obecnie de facto standardem obejmującym warstwy fizyczne i łącza dla sieci 6LoWPAN, inne nowe technologie również ewoluują. Na przykład BLE, sprzedawany praktycznie jako Bluetooth Smart, należy do obecnie dostępnych energooszczędnych technologii komunikacyjnych i stanowi atrakcyjną alternatywę. W szczególności BLE stał się lekką alternatywą dla urządzeń o ograniczonych zasobach w porównaniu z klasycznym Bluetoothem. Standard Bluetooth 4.0 obejmuje teraz również specyfikacje BLE, które obejmują tryb komunikacji rozgłoszeniowej oprócz energooszczędnych połączeń między urządzeniami Bluetooth. Bluetooth 4.2 został opublikowany w grudniu 2014 roku i zapewniał szereg nowatorskich funkcji, które sprawiają, że BLE jest obiecującą i wartościową technologią dla IoT (Bluetooth Special Interest Group, 2014). Najważniejszym dodatkiem wzbogacającym Bluetooth o możliwości IoT jest obsługa profilu IPSP (Bluetooth Special Interest Group, 2016). IPSP dodaje obsługę protokołu IPv6 w urządzeniu peryferyjnym Bluetooth oraz w centralnym urządzeniu nadrzędnym pełniącym rolę koordynatora sieci, a także określa, w jaki sposób urządzenia mają wykrywać się nawzajem i nawiązywać ze sobą połączenie w warstwie łącza. Poza tym, BLE Generic Attribute Profile (GATT) może być wykorzystany do określenia, czy urządzenie ma włączoną obsługę IPSP, co z kolei umożliwia wymianę pakietów IP za pomocą trybu kontroli przepływu opartego na kredytach Bluetooth L2CAP. Ponadto Nieminen i inni łączą BLE z 6LoWPAN, określając standardowy sposób transmisji skompresowanych pakietów IPv6 na górze BLE. W szczególności proponują zastosowanie mechanizmów kompresji nagłówków 6LoWPAN na szczycie BLE. Z drugiej strony sugerują, aby nie używać mechanizmów fragmentacji 6LoWPAN, ale raczej polegać na tych już opisanych w Bluetooth L2CAP. Ponadto jednym z najbardziej istotnych ulepszeń wprowadzonych przez Bluetooth 4.2 jest zwiększony rozmiar pakietów, co oznacza, że ładunek BLE w warstwie łącza został zwiększony z 27 do 251 bajtów. Ponadto Bluetooth 4.2 poprawia przepustowość do 2,5 razy. Te rozszerzone możliwości umożliwiają wydajną komunikację IPv6 przez BLE, częste i szybkie aktualizacje oprogramowania sprzętowego urządzeń, a także szybkie przesyłanie danych z urządzeń czujnikowych do jednostek zaplecza. Umożliwia także uruchomienie nawet w IoT szeregu złożonych protokołów bezpieczeństwa, na przykład DTLS i IPsec. Przed Bluetooth 4.2 standard

Bluetooth zapewnia silniejsze bezpieczeństwo w porównaniu z BLE, głównie w celu osiągnięcia efektywności energetycznej. Jednak wyświetla również mały rozmiar pakietu 27 bajtów, a zatem nie pozwala urządzeniom na używanie protokołów kryptograficznych z kluczem asymetrycznym. Zamiast tego Bluetooth 4.2 LE zapewnia BLE ten sam poziom bezpieczeństwa co standardowy Bluetooth, zalecając poleganie na kryptografii krzywych eliptycznych (ECC). W szczególności Narodowy Instytut Standardów i Technologii (NIST) zalecił stosowanie krzywych eliptycznych, podczas gdy Federalne Standardy Przetwarzania Informacji (FIPS) zaleciły stosowanie kryptografii symetrycznej AES-CCM do szyfrowania i ochrony integralności. Ponadto Bluetooth 4.2 zapewnia również zwiększoną prywatność, dodając rozdzielczość prywatnych adresów zarówno w hoście Bluetooth, jak i w segmentach kontrolera węzła Bluetooth. Ponadto Bluetooth 4.2 określa, że kontroler utrzymuje białą listę adresów prywatnych w warstwie łącza. W ten sposób kontroler jest w stanie generować i rozwiązywać adresy prywatne w warstwie łącza bez konieczności interakcji z hostem oraz odrzucać lub akceptować połączenia przychodzące zgodnie z utrzymywaną białą listą. To znacznie zmniejsza szybkość, z jaką host musi się „budzić", co znacznie ogranicza zużycie energii przez chip BLE. Większe bezpieczeństwo i prywatność dzięki nowym funkcjom o niskim poborze mocy oraz fakt, że BLE jest obsługiwany po wyjęciu z pudełka w większości popularnych smartfonów, sprawia, że BLE jest cenną i obiecującą technologią do wdrażania IoT, która obejmuje mobilność i umożliwia bezpośrednie podłączenie sieci 6LoWPAN do smartfon, na przykład we wdrożeniach związanych z urządzeniami do noszenia. Standard IEEE 802.15.4 jest nadal preferowanym i bardziej odpowiednim wyborem dla wdrożeń statycznych, takich jak inteligentne domy.

Autoryzacja oparta na protokole OAUTH w IoT

Według Internet Security Glossary autoryzacja jest definiowana jako proces udzielania klientowi zgody na dostęp do zasobu. Typowo przyjęte podejście polega na zarządzaniu autoryzacją użytkowników, usług i ich urządzeń za pomocą dedykowanych jednostek Serwera Autoryzacji (AS). Ramy autoryzacji OAuth 2.0 opierają się na tym podstawowym podejściu i zyskały miano jednego z najczęściej stosowanych standardów zarządzania procesami uwierzytelniania. W rzeczywistości umożliwia rozwiązanie wszystkich typowych problemów alternatywnych podejść opartych na współdzieleniu poświadczeń, dzięki wprowadzeniu warstwy autoryzacyjnej i oddzieleniu roli klienta od roli faktycznego właściciela zasobu. Zasadniczo struktura autoryzacji OAuth 2.0 umożliwia jednostce klienta (tj. hostowi, procesowi, użytkownikowi) żądanie i uzyskanie określonego, regulowanego i ograniczonego dostępu do zasobu dostępnego na serwerze zasobów (RS) poprzez egzekwowanie uprawnienia powiązanego właściciela zasobu. W szczególności właściciel zasobu udziela autoryzacji klientowi za pomocą pośredniczącego AS, który dostarcza klientowi rzeczywiste informacje związane z autoryzacją określone w tokenie dostępu. Mówiąc dokładniej, token dostępu jest praktycznie reprezentowany jako ciąg, który koduje decyzję autoryzacyjną wydaną dla określonego klienta i jest zwykle nieprzezroczysty dla tego klienta. Token dostępu określa między innymi czas trwania i zakres autoryzowanych dostępów do zasobów, które są ostatecznie określane przez AS i wymuszane przez RS. Ponadto AS zapewnia, że możliwe nieautoryzowane strony nie będą w stanie generować, modyfikować ani odgadywać żadnych wydanych tokenów dostępu w celu wygenerowania ważnych tokenów dostępu. Następnie klient może skontaktować się z RS w celu uzyskania dostępu do zamierzonego zasobu, zgodnie z poświadczeniami określonymi w tokenie dostępu. Oznacza to, że RS weryfikuje, czy token dostępu jest ważny i w takim przypadku kontynuuje obsługę żądania otrzymanego od żądającego klienta. Przed rozpoczęciem wykonywania protokołu autoryzacji klient musi zarejestrować się w AS, a wszystkie poświadczenia tokenu dostępu muszą być zawsze przesyłane przez bezpieczny kanał komunikacyjny, to znaczy zarówno z AS do klienta, jak i od klienta do RS. Ponadto, AS musi weryfikować tożsamość właściciela zasobu przy każdym żądaniu uwierzytelnienia w imieniu klienta i musi wcześniej ustanowić relację zaufania z zaangażowanym RS, który udostępnia

zasoby. Należy zauważyć, że ten sam AS może być powiązany z wieloma serwerami zasobów i wystawiać tokeny dostępu, które będą używane z wieloma serwerami zasobów. To powiedziawszy, bardziej niż uzasadnione jest założenie, że konsumenci końcowi i przedsiębiorstwa będą chcieli przyjąć takie samo podejście do zarządzania autoryzacją i kontrolą dostępu do zasobów w scenariuszach aplikacji IoT. Stanie się to również coraz bardziej prawdopodobne wraz z szybko rosnącą liczbą usług świadczonych przez aplikacje hostowane na urządzeniach IoT zorganizowanych we wdrożeniach na dużą skalę. Doprowadziło to do wniosku IETF mającego na celu ponowne wykorzystanie struktury OAuth 2.0 do rozszerzenia autoryzacji na urządzenia IoT z różnego rodzaju ograniczeniami, korzystając z podstawowych mechanizmów OAuth 2.0 tam, gdzie to możliwe, jednocześnie zapewniając realizatorom dodatkowe wskazówki, profile i rozszerzenia umożliwiające korzystanie z niego w bezpieczny i przyjazny sposób z zachowaniem prywatności. W szczególności Seitz i inni opisują ramy autoryzacji w IoT, które łączą w sobie kilka elementów konstrukcyjnych - (i) oryginalne ramy autoryzacji OAuth 2.0, (ii) protokół transferu CoAP oraz (iii) usługi zabezpieczeń warstwy aplikacji, oparte na zabezpieczeniach obiektowych danych zakodowanych w CBOR - wymagane, gdy zabezpieczenia warstwy transportowej zapewniane przez protokoły takie jak DTLS są niewystarczające, adekwatne lub wygodne.

Problemy i rozwiązania dotyczące bezpieczeństwa

Protokoły i mechanizmy przedstawione wcześniej są zwykle przyjmowane w celu spełnienia podstawowych wymagań bezpieczeństwa w Internecie Rzeczy, takich jak bezpieczna komunikacja na różnych warstwach, autoryzacja dostępu do zasobów i kompaktowe kodowanie dla formatów kryptograficznych. Niemniej jednak, nawet przy zapewnieniu takich podstawowych gwarancji bezpieczeństwa, systemy IoT są podatne na szereg luk w zabezpieczeniach i narażone na określone ataki bezpieczeństwa, wykorzystujące słabości i nieefektywności protokołów bezpieczeństwa, a także słabą skalowalność i wydajność typowych schematów zarządzania. Co więcej, takie wektory ataków mogą być szczególnie łatwe do wykorzystania w IoT z dwóch głównych powodów. Po pierwsze, większość urządzeń IoT ma być bezpośrednio podłączona do Internetu, aby była bezpośrednio osiągalna, a tym samym w pełni wystawiona na przeciwników zmotywowanych do przeprowadzania np. ataków DoS. Po drugie, wiele urządzeń IoT ma ograniczone zasoby, ponieważ są wyposażone w ograniczoną ilość pamięci, zasobów obliczeniowych i energii. Oznacza to, że ataki, z którymi normalnie można sobie poradzić, mogą mieć znacznie poważniejszy wpływ. Poza tym, nawet powiązane zabezpieczenia muszą być zaprojektowane tak, aby były wykonalne i możliwe do przyjęcia dla takich klas urządzeń ograniczanych. W tej sekcji przedstawiono szereg konkretnych problemów związanych z bezpieczeństwem związanych z systemami, protokołami i technologiami IoT oraz omówiono możliwe rozwiązania tych problemów w oparciu o najnowsze wyniki społeczności naukowej i działania w organach normalizacyjnych. Szczególny nacisk kładziony jest na różne formy ataków DoS, w tym (selektywne) zagłuszanie wykonywane w warstwie fizycznej, uwzględniając również potrzebę wydajnych i skalowalnych protokołów do zarządzania kluczami kryptograficznymi, bezpiecznej komunikacji w grupach multicastowych urządzeń IoT oraz adopcji systemów wykrywania włamań (IDS) i zapór sieciowych.

Odmowa usługi przeciwko CoAP

Ograniczone energetycznie, nawet zasilane bateryjnie urządzenia IoT są w znacznym stopniu narażone na szereg ataków bezpieczeństwa, zwłaszcza jeśli są bezpośrednio połączone z Internetem bez żadnej szczególnej ochrony zaimplementowanej za pośrednictwem dedykowanych bramek lub skrzynek zapory. Szczególnie dobrze znaną klasą ataków na zabezpieczenia są DoS, które z kolei mogą być montowane w celu utrzymania urządzeń IoT w ciągłym zajęciu odbieraniem i przetwarzaniem nieprawidłowych wiadomości, zapobiegając w ten sposób przełączeniu się w energooszczędny tryb

uśpienia i szybkiemu wyczerpaniu ich zasilanie bateryjne. Ten szczególny rodzaj ataków DoS jest często określany jako odmowa snu i jest uważany za szczególnie poważne zagrożenie bezpieczeństwa w przypadku urządzeń IoT o ograniczonym działaniu, zasilanych bateryjnie. Ponadto nie można całkowicie uniknąć ich wystąpienia. Oznacza to, że aktywny przeciwnik, który stale przesyła fałszywe wiadomości, może zawsze skłonić urządzenie IoT będące odbiorcą ofiary do ich odebrania i wykonania operacji przetwarzania na różnych warstwach. Oznacza to, że praktyczny środek zaradczy powinien zamiast tego skupiać się na jak największym zmniejszeniu wpływu ataku, ze szczególnym uwzględnieniem zużycia energii z powodu bezużytecznego przetwarzania wiadomości. Jednak przeciwdziałanie tej klasie ataków w skuteczny i wydajny sposób jest zdecydowanie trudnym zadaniem, szczególnie w obecności urządzeń IoT, które są bezpośrednio podłączone do Internetu, czyli globalnie dostępne, aby umożliwić zdalne zarządzanie i konfigurację, żądania operacji i wyszukiwanie informacji. Obecnie CoAP stał się de facto protokołem warstwy aplikacji dla IoT, ze szczególnym uwzględnieniem urządzeń o ograniczonych zasobach. Jednak jeśli chodzi o zapewnienie bezpiecznej komunikacji, CoAP nie zapewnia żadnych szczególnych prymitywów bezpieczeństwa ani środków zaradczych przeciwko atakom DoS. Zamiast tego CoAP sugeruje przyjęcie protokołu DTLS. który opiera się na kosztownym i złożonym procesie uzgadniania w celu skonfigurowania bezpiecznej sesji i skutkuje znacznym obciążeniem zapewniającym cały zestaw gwarancji bezpieczeństwa, z których niektóre mogą nawet nie być konieczne w niektórych scenariuszach aplikacji. W wielu pracach zaproponowano techniki wykrywania ataków typu „odmowa snu", zwykle z uwzględnieniem analizy ruchu lub innych mechanizmów wykrywania anomalii. Na przykład Bhattasali i Chaki proponują model probabilistyczny do wykrywania ataków odmowy snu, oparty na Absorbujących Łańcuchach Markowa. Zamiast tego Buennemeyer i inni opisują B-SIPS, system oparty na innowacyjnym algorytmie Dynamic Threshold Calculation, który zapewnia alerty po wykryciu zmian mocy. Jak wskazano w Raymond i in. (2009) te rozwiązania oparte na technikach wykrywania włamań zazwyczaj wymagają przechwycenia i przeanalizowania znacznego ruchu sieciowego, a zatem może być trudne lub nawet niemożliwe do wdrożenia i uruchomienia na urządzeniach IoT o ograniczonych zasobach. Różne proponowane podejścia uwzględniają tradycyjne usługi bezpieczeństwa, zwłaszcza w warstwie łącza. Na przykład Martin i in. (2004) proponują architekturę zasilania, która przetwarza tylko żądania, które zasługują na wysoki poziom zaufania, ale nie opisują żadnego konkretnego schematu poprawy odporności na rozładowanie baterii. Zamiast tego Raymond i Midkiff (2007) prezentują podejście do ograniczania szybkości, które opiera się na mechanizmach wykrywania włamań opartych na hoście, wymuszonych w warstwie łącza, uznając wiadomości za uzasadnione tylko wtedy, gdy są uwierzytelnione i nieodtwarzane, oraz podejmując działania w celu złagodzenia skutków ataku . Również Raymond i inni (2009) proponują strukturę opartą na klasycznym uwierzytelnianiu warstwy łącza, ochronie odtwarzania i identyfikacji zagłuszania. Takie podejścia do wykrywania, które opierają się na tradycyjnym zabezpieczeniu warstwy łącza, mogą jednak powodować znaczne koszty ogólne, zwłaszcza w zakresie zużycia energii . Oznaczają również, że ważność wiadomości musi być sprawdzana przez każdy węzeł sieci na ścieżce od źródła do urządzenia docelowego. To ewidentnie ma znaczny dodatkowy wpływ na wydajność sieci jako całości. Co więcej, te podejścia wymagają ustanowienia bezpiecznych relacji zaufania między wszystkimi urządzeniami na ścieżce komunikacyjnej (np. poprzez dystrybucję parami kluczy kryptograficznych), przez co prawdopodobnie nie zapewnią skutecznej ochrony przed atakami typu „odmowa snu" z perspektywy end-to-end. Ostatnio Gehrmann i in. (2015) zaproponowali usługę bezpieczeństwa opartą na krótkim kodzie uwierzytelniania wiadomości osadzonym bezpośrednio w nagłówku wiadomości CoAP, w celu wczesnego i skutecznego wykrywania nieprawidłowych wiadomości po ich odebraniu. Oznacza to, że umożliwia szybkie ustalenie, czy otrzymane wiadomości są ważne, czy nie, a mianowicie, czy zostały ewentualnie przesłane z nielegalnych źródeł. W takim przypadku ofiara urządzenia IoT może wcześniej odrzucić nieprawidłowe wiadomości i uniknąć wykonywania bezużytecznych dodatkowych operacji analizowania i

przetwarzania. To znacznie zmniejsza wpływ ataków odmowy snu na zużycie energii, a tym samym wydłuża żywotność baterii w Urządzeniu IoT.

Problemy z odmową usługi i skalowalnością w DTLS

Proces uzgadniania DTLS wyświetla dwa istotne problemy z bezpieczeństwem i wydajnością, które dotyczą głównie urządzeń działających jako serwer DTLS. Przede wszystkim, jak również podkreślono w specyfikacji DTLS, urządzenie pełniące rolę serwera DTLS jest szczególnie narażone na ataki DoS montowane podczas procesu uzgadniania. W szczególności aktywny przeciwnik ma możliwość ciągłego przesyłania komunikatów ClientHello do serwera DTLS i nakłaniania go do rozpoczęcia dużej liczby uścisków dłoni DTLS. To z kolei zmusza serwer do rozpoczęcia konfiguracji nowych (nieprawidłowych) sesji DTLS. Z serwera DTLS z punktu widzenia oznacza to przydzielanie zasobów sieciowych i pamięci oraz wykonywanie szeregu operacji wymagających dużej ilości zasobów. Co więcej, może to spowodować wyczerpanie dostępnych zasobów sieciowych i pamięci, przez co serwer będzie mniej responsywny lub, w najgorszym przypadku, nawet niedostępny do przetwarzania uzasadnionych żądań wysyłanych przez klientów DTLS. Kolejną wadą jest to, że ataki przeprowadzane za pośrednictwem sfałszowanych prawidłowych adresów IP zmuszają serwer DTLS do wysyłania wiadomości zwrotnych do „niewinnych" urządzeń, wyświetlając w ten sposób dobrze znany efekt wzmocnienia. Specyfikacja DTLS zapewnia wstępne rozwiązanie przeciwko wspomnianemu powyżej atakowi DoS, który wykorzystuje bezstanową i opcjonalną wymianę części informacji, a mianowicie pliku cookie, zachodzącej podczas pierwszych faz uzgadniania między klientem DTLS a serwerem. W szczególności serwer DTLS może odpowiedzieć na pierwszy komunikat ClientHello dodatkowym komunikatem HelloVerifyRequest, który zawiera plik cookie jako lokalnie wygenerowaną wartość. Następnie klient DTLS odpowiada kolejnym komunikatem ClientHello, który musi zawierać dokładnie to samo ciasteczko. Jednak proces wymiany plików cookie tylko komplikuje rozważany atak DoS, podczas gdy zasadniczo nie zapewnia żadnej rzeczywistej ochrony przed nim. Bardziej szczegółowo, przeciwnik, który jest w stanie przechwycić wiadomości wysyłane przez serwer DTLS podczas uzgadniania, nadal jest w stanie wykonać Atak DoS i skłonienie serwera do skonfigurowania znacznej liczby nieprawidłowych, półotwartych sesji DTLS. Wynika z tego, że środek zaradczy oparty na wymianie plików cookie między klientem a serwerem w rzeczywistości nie jest dobrym rozwiązaniem przeciwko atakom DoS montowanym przez zaradnego i dobrze zdeterminowanego przeciwnika on-path. Ponadto, jeśli uzgadnianie opiera się na podejściu PSK (Eronen i Tschofenig, 2005), serwer DTLS może wymagać przechowywania i zarządzania nieznaczną liczbą kluczy symetrycznych powiązanych z klientami DTLS. Serwer DTLS przechowuje w najgorszym przypadku jeden klucz wstępny dla każdego możliwego klienta DTLS. Oczywiście to podejście słabo skaluje się wraz z liczbą klientów DTLS w sieci. Ponadto znacznie komplikuje kluczowe operacje dostarczania i zarządzania, zwłaszcza w scenariuszach dynamicznych. Niemniej jednak schemat PSK jest bardzo przydatny, wygodny i szeroko stosowany w kilku dynamicznych środowiskach IoT, które obejmują urządzenia o ograniczonych zasobach, a także użytkowników końcowych bez możliwości bezpiecznego zarządzania bardziej złożoną infrastrukturą klucza publicznego. W Tiloca i inni zaproponowano architekturę bezpieczeństwa w celu rozwiązania obu omówionych już kwestii. W szczególności proponowane podejście umożliwia serwerowi DTLS wykrycie nieprawidłowego komunikatu ClientHello po jego odebraniu, a następnie natychmiastowe zakończenie uzgadniania na samym jego początku, co praktycznie neutralizuje atak i znacznie ogranicza jego oddziaływanie. Zapewnia również alternatywny schemat PSK, który znacznie zmniejsza liczbę kluczy wstępnych przechowywanych na serwerze do jednego, zapobiegając w ten sposób problemom ze skalowalnością i zarządzaniem. Ponadto proponowane rozwiązanie ma szereg zalet. Po pierwsze, opiera się na ustandaryzowanej metodzie rozszerzania komunikatów ClientHello, a zatem nie jest wymagana modyfikacja standardu DTLS. Po drugie, klient i serwer DTLS nie są wymagane do dalszej wymiany komunikatów, a nawet wymiana plików cookie nie jest już konieczna. Poza tym nie wpływa

to znacząco na obciążenie obliczeniowe klienta i serwera, ponieważ ta sama złożoność obliczeniowa jest zachowana dla procesu uzgadniania. Wreszcie można go łatwo ponownie zastosować w TLS, ponownie bez wymaganych zmian.

Komunikacja grupowa oparta na DTLS

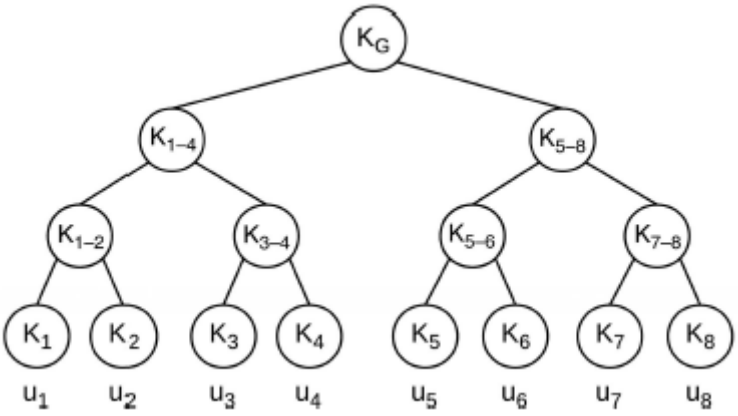
Jak omówiono, protokół CoAP nakazuje przyjęcie protokołu DTLS w celu zabezpieczenia komunikacji unicastowej między daną parą urządzeń IoT. Jednocześnie w kilku wdrożeniach IoT obejmujących grupy, których członkami są urządzenia IoT, korzystanie z komunikacji multiemisji przez IPv6 ma kilka zalet, w szczególności pod względem skalowalności i wydajności. Z tego powodu standard wyraźnie zdefiniował komunikację grupową opartą na protokole CoAP, odnosząc się do wielu różnych przypadków użycia, w których model komunikacji grupowej jest szczególnie korzystny dla urządzeń IoT. Na przykład komunikacja grupowa jest bardzo wygodna, a nawet nieunikniona w scenariuszach aplikacji IoT, takich jak aktualizacje oprogramowania układowego i oprogramowania, sterowanie oświetleniem, aktualizacje konfiguracji i parametrów, zintegrowane sterowanie budynkiem, transmisja alarmowa i uruchomienie sieci 6LoWPAN. W wielu wdrożeniach IoT te przypadki użycia wykazują szereg określonych wymagań bezpieczeństwa, które należy spełnić, w tym bezpieczną wymianę komunikatów multiemisji między urządzeniami IoT w tej samej grupie. Jednak w tym czasie CoAP nie zapewnia wsparcia ani nie zaleca interoperacyjnego sposobu zabezpieczania komunikacji grupowej multiemisji. Jednocześnie DTLS jest de facto protokołem bezpieczeństwa służącym do zabezpieczania komunikacji unicast IoT. Dlatego szczególnie wygodnie jest ulepszyć DTLS, aby mógł również obsługiwać bezpieczną komunikację grupową w IoT. Ten wybór byłby również szczególnie wygodny w porównaniu z możliwymi różnymi alternatywami. Na przykład wybór multiemisji IPsec wymagałby również użycia ciężkiego IPsec , a prawdopodobnie nawet dodatkowych protokołów wspierających, takich jak IKEv2 . Co więcej, DTLS ma być obecnie używany do ochrony komunikacji unicast w IoT, dlatego wybór rozwiązań IPsec byłby jeszcze mniej praktyczny i wygodny w rzeczywistych scenariuszach zastosowań. Jak dotąd w ramach działań normalizacyjnych prowadzonych w ramach IETF podjęto tylko kilka inicjatyw mających na celu zapewnienie bezpiecznej komunikacji grupowej opartej na technologii DTLS dla CoAP. W szczególności pierwsze podejścia do tego kierunku zostały przedstawione w dwóch dokumentach Keoh i inni oraz Kumar i Struik , którzy proponują ochronę komunikacji w grupie multiemisji urządzeń IoT poprzez wygodne dostosowanie warstwy rekordu DTLS, ale zostali zakwestionowani z wielu powodów. Wśród nich wskazano, że uwierzytelnianie źródła jest podstawowym wymogiem bezpieczeństwa, który należy spełnić, ponieważ samo uwierzytelnianie grupowe może nie wystarczyć w kilku scenariuszach aplikacji IoT. Ponadto, ochrona komunikatów odpowiedzi grupowych na komunikaty żądań multiemisji została zapewniona za pomocą tradycyjnych sesji DTLS w parach. Może to jednak skutkować wieloma praktycznymi problemami i może powodować znaczne pogorszenie wydajności, szczególnie w bardzo dynamicznych i dużych grupach urządzeń IoT.

Z drugiej strony metoda opisana w Tiloca (2014) omawia poprzednie podejścia i proponuje, jak zapewnić dwukierunkową bezpieczną komunikację grupową opartą na DTLS, jednocześnie odnosząc się do ograniczeń wykazanych przez poprzednio proponowane podejścia. W szczególności wszyscy członkowie grupy multiemisji biorą pod uwagę ten sam materiał klucza grupy lub wydajnie uzyskują indywidualny materiał klucza. Członkowie grupy polegają na posiadanym przez siebie ogólnym materiale kluczowym, aby chronić zarówno żądania wysyłane do grupy jako wiadomości multiemisji, jak i ewentualne odpowiedzi odsyłane jako wiadomości emisji pojedynczej. Wspólny materiał klucza jest dostarczany nowemu członkowi grupy po jego dołączeniu i nie jest konieczne wykonywanie uzgadniania DTLS między członkami grupy, aby umożliwić bezpieczną komunikację w grupie. Proponowane podejście zapewnia wreszcie uwierzytelnianie źródła lub grupowe uwierzytelnianie wiadomości wysyłanych w ramach grupy, zgodnie z wymaganiami konkretnego scenariusza aplikacji.

Wydajne i skalowalne zarządzanie kluczami grupowymi

Jak już wspomniano, komunikacja grupowa jest wygodnym podejściem do zastosowania w wielu scenariuszach aplikacji IoT w celu poprawy wydajności, szybkości reakcji i efektywności energetycznej. Wymaga to jednak również kompleksowego zarządzania materiałem klucza grupowego w zakresie generowania, unieważniania i (ponownej) dystrybucji. Ogólnie rzecz biorąc, materiał klucza grupy obejmuje sekrety współdzielone przez członków grupy, które umożliwiają bezpieczną komunikację tylko między urządzeniami w tej grupie, w danej warstwie komunikacji. Dla uproszczenia, poniżej opisano materiał klucza grupy jako pojedynczy symetryczny klucz grupy, który wszyscy członkowie grupy wspólnie współużytkują i używają do zabezpieczania komunikacji w grupie w warstwie aplikacji. Zgodnie z modelem komunikacji grupowej, urządzenie IoT jawnie dołącza do grupy multiemisji, aby stać się aktywnym członkiem. Od tej chwili urządzenie może odbierać (wysyłać) komunikaty rozgłoszeniowe od (do) innych urządzeń w tej samej grupie. W pewnym momencie urządzenie może chcieć opuścić grupę lub może zostać zmuszone do opuszczenia, jeśli zostanie naruszone lub podejrzewane. Aby zezwolić tylko rzeczywistym członkom grupy na uczestniczenie w komunikacji w grupie, konieczne jest unieważnienie i odnowienie bieżącego klucza grupy, czyli ponowne wprowadzenie klucza, zgodnie z następującymi zasadami. Gdy nowe urządzenie staje się członkiem grupy, należy uniemożliwić mu odszyfrowywanie i uzyskiwanie dostępu do wiadomości wymienianych przed jego dołączeniem, nawet jeśli je zarejestrowało (zabezpieczenie wsteczne). Ponadto, gdy urządzenie zdecyduje się opuścić grupę lub zostanie do tego zmuszone jako zagrożone lub podejrzewane, należy uniemożliwić mu odszyfrowanie i uzyskanie dostępu do dalszej komunikacji w grupie w przyszłości (bezpieczeństwo przekazywania). Praktycznie po zmianie przynależności do grupy konieczne jest unieważnienie aktualnego klucza grupy i rozesłanie nowego do obecnych/pozostałych urządzeń w grupie. Dlatego szczególnie ważne jest, aby ponowne kluczowanie odbywało się w sposób wydajny i wysoce skalowalny, zwłaszcza w grupach o dużej dynamice i dużej skali, gdzie dołączanie i opuszczanie może być bardzo częstymi zdarzeniami. Co więcej, szczególnie ważne jest skuteczne radzenie sobie z atakami polegającymi na zmowie, które występują, gdy wiele urządzeń w grupie zostaje zhakowanych, udostępniają swoje materiały zabezpieczające i mimo to biorą udział w procedurze ponownego wprowadzania kluczy. W ten sposób te zhakowane węzły współpracują w celu odzyskania wiedzy o kluczu grupy i nie zostaną skutecznie usunięte z grupy multiemisji. Nie istnieją protokoły zmiany klucza grupowego, które nie są w ogóle narażone na ataki polegające na zmowie, aczkolwiek charakteryzują się one różnymi poziomami odporności. Jednak tylko nieliczne zapewniają środki zaradcze w celu przeciwdziałania atakom polegającym na zmowie. Co więcej, kilka schematów ponownego wprowadzania kluczy pozwala odzyskać grupę przed atakami zmowy za pomocą całkowitej ponownej inicjalizacji członka, co oznacza, że każde nieskompromitowane urządzenie w grupie musi być ponownie inicjowane indywidualnie. W konsekwencji odzyskiwanie zmowy powoduje narzut komunikacji, który rośnie liniowo wraz z rozmiarem grupy i może bardzo negatywnie wpłynąć na skalowalność i wydajność sieci. Protokoły zarządzania kluczami grupowymi są zazwyczaj klasyfikowane jako scentralizowane, rozproszone i zdecentralizowane. Podejścia scentralizowane są szczególnie preferowane w przypadku grup o dużej dynamice i dużej skali, a szczególnie dobrze nadają się do minimalizowania kosztów komunikacji, przechowywania i obliczeń. Scentralizowane schematy wykorzystują zestaw materiałów kluczy administracyjnych, które są logicznie zorganizowane w celu zapewnienia skalowalnego ponownego wprowadzania kluczy po opuszczeniu węzła. Ponadto Logical Key Hierarchy (LKH) to scentralizowany protokół, który do organizowania kluczy administracyjnych opiera się na hierarchicznym drzewie logicznym. W szczególności klucz grupy jest korzeniem drzewa, indywidualne klucze członków grupy są liśćmi drzewa, a dodatkowe klucze administracyjne są wewnętrznymi węzłami drzewa. W grupie składającej się z n członków, przy założeniu, że kluczowy materiał jest zorganizowany w zrównoważone drzewo, zarówno narzut przechowywania po stronie

urządzenia, jak i narzut komunikacji urlopowej rośnie o (log n). Rysunek pokazuje kluczowy materiał zarządzany przez LKH w grupie złożonej z ośmiu urządzeń, a mianowicie u₁, u₂,… , u₈.



W szczególności, każde urządzenie posiada klucze na ścieżce od powiązanego węzła liścia w drzewie logicznym do węzła głównego skojarzonego z kluczem grupy KG. Na przykład urządzenie u3 posiada indywidualny klucz K3, dodatkowe klucze administracyjne K₁₋₄ i K₃₋₄ oraz klucz grupowy KG. Inne protokoły pochodzące z LKH nie osiągają lepszej wydajności. Ponadto scentralizowane protokoły Highly Scalable Scheme (HISS) i Group REkeying Protocol GREP wykorzystują logiczne podgrupy w celu zapewnienia wysoce skalowalnego i wydajnego ponownego wprowadzania klucza grupowego, wyświetlania tego samego narzutu pamięci masowej i obliczeniowego, czyli O(√n) oraz dystrybucji nowego klucz grupy przy użyciu pewnej liczby komunikatów o ponownym wprowadzaniu kluczy, która jest stała, mała i niezależna od rozmiaru grupy, czyli O(1). Ponadto protokół GREP opiera się na nowatorskim pomyśle historii połączeń członków, a w rzeczywistości wykorzystuje ją w celu przeprowadzenia odzyskiwania zmowy w znacznie bardziej skalowalny i wydajny sposób, bez uciekania się do całkowitej ponownej inicjalizacji członka. W szczególności HISS wyświetla narzut odzyskiwania, który zawsze rośnie jako O(n). W przeciwieństwie do tego, narzut odzyskiwania spowodowany GREP stopniowo wzrasta wraz z dotkliwością konkretnego ataku zmowy i płynnie rośnie jako O(√n), aczkolwiek tylko w bardzo mało prawdopodobnym najgorszym przypadku. Z drugiej strony protokoły rozproszone opierają się na czynnościach zarządzania kluczami wykonywanymi w sposób rozproszony, angażując samych członków grupy lub polegając na dodatkowych (pod)zarządzających kluczami. Są one jednak trudniejsze do wdrożenia, często mniej skalowalne i budzą obawy dotyczące bezpieczeństwa. Co więcej, niektóre nie zapewniają powrót do zdrowia po atakach zmowy , podczas gdy inne zawsze wymagają całkowitej reinicjalizacji członków lub handlu z oporem na zmowę

Selektywne zagłuszanie w sieciach bezprzewodowych

Wiadomo, że ataki zagłuszające należą do najczęstszych i najpoważniejszych rodzajów ataków DoS na sieci wykorzystujące technologie komunikacji bezprzewodowej. W szczególności ataki zagłuszające polegają na celowym ingerowaniu w częstotliwości operacyjne w sieci i są zazwyczaj klasyfikowane jako stałe, losowe, zwodnicze i reaktywne. Z jednej strony spektrum ciągłe zagłuszanie jest niezwykle łatwe do wykonania. Jego głównym celem jest uszkadzanie wszystkich przesyłanych pakietów sieciowych poprzez ciągłe nakładanie się transmisji losowych sygnałów. Należy zauważyć, że tę strategię zagłuszania można uznać za „zawsze aktywną”, ponieważ opiera się na stałej obecności wysokiego poziomu zakłóceń, co znacznie ułatwia wykrycie ataku. Z drugiej strony, zagłuszanie

reaktywne oznacza szczególnie inteligentne i energooszczędne podejście, zgodnie z którym przeciwnik aktywnie blokuje komunikację w sieci tylko podczas transmisji z innych urządzeń. Co więcej, bardzo prawdopodobne jest, że zagłuszanie reaktywne jest w rzeczywistości mylone z normalnymi kolizjami pakietów sieciowych, co znacznie utrudnia jego wykrycie niż inne formy ataku zagłuszającego. Mówiąc dokładniej, zagłuszanie selektywne jest specyficznym rodzajem zagłuszania reaktywnego i ma na celu zakłócanie komunikacji między urządzeniami sieciowymi, zgodnie z dobrze zdefiniowanymi i określonymi celami i kryteriami. W odniesieniu do innych rodzajów ataków i strategii zagłuszania, zagłuszanie selektywne jest bardziej wydajne pod względem mocy do wykonania, a nawet trudniejsze do przeciwdziałania i wykrywania, ze względu na zmniejszoną ekspozycję przeciwnika. Powszechnym kryterium branym pod uwagę przy selektywnym zagłuszaniu jest zakłócanie wszelkiej komunikacji jedno konkretne urządzenie w sieci. W szczególności atak ten jest szczególnie łatwy do przeprowadzenia w sieci bezprzewodowej wykorzystującej wielokrotny dostęp z podziałem czasu (TDMA). W szczególności, TDMA dzieli czas na okresową sekwencję superramek, z których każda składa się ze stałej liczby szczelin transmisyjnych. Jednak takie gniazda są zwykle przypisywane do urządzeń sieciowych w taki sposób, że każde urządzenie może pozostać aktywne tylko podczas swoich własnych gniazd, a zamiast tego może przejść w tryb uśpienia podczas innych gniazd, ograniczając w ten sposób zużycie energii. Z drugiej strony, urządzenie sieciowe zazwyczaj zachowuje dokładnie te same szczeliny dla kilku kolejnych superramek. W konsekwencji przeciwnik musi po prostu monitorować komunikację sieciową, określać szczeliny, do których mają dostęp urządzenia ofiary, i blokować je, aby całkowicie udaremnić komunikację urządzeń. Należy zauważyć, że taki atak jest również bardzo wydajny z punktu widzenia zużycia energii, ponieważ przeciwnik musi utrzymywać aktywność radiową tylko podczas szczelin transmisyjnych przydzielonych jej urządzeniom ofiary. Zaproponowane środki zaradcze mające na celu przeciwdziałanie atakowi zagłuszającemu zostały zazwyczaj podzielone na rozwiązania w warstwie fizycznej i cybernetyczne. W zasadzie rozwiązania warstwy fizycznej starają się uniemożliwić przeciwnikowi skuteczne ingerowanie w częstotliwości używane w sieci. Najważniejsze techniki należące do tej kategorii zostały omówione w Mpitziopoulos i inni i często opierają się na częstotliwości rozrzutu częstotliwości (FHSS), czyli metodzie, która wybiera inną nośną spośród dostępnych kanałów częstotliwości, zgodnie z algorytmem współdzielonym przez odbiornik i nadajnik. Jednak podejścia warstwy fizycznej nie są w stanie zasadniczo zneutralizować ataków zagłuszających. Z drugiej strony, rozwiązania cybernetyczne zakładają, że przeciwnik zawsze jest w stanie ingerować w warstwę fizyczną przy regularnych transmisjach w sieci. Dlatego rozwiązania cybernetyczne przeciwstawiają ataki zagłuszające za pomocą odpowiednich schematów bezpieczeństwa. Większość środków zaradczych w cyberprzestrzeni specjalizuje się w przeciwdziałaniu ciągłemu. Zamiast tego tylko kilka rozwiązań zostało specjalnie zaprojektowanych do ataków z selektywnym zagłuszaniem. Na przykład Wood i in. (2007) opisują eliminację Energy-Efficient JAMming w sieciach bezprzewodowych opartych na IEEE 802.15.4 (DEEJAM), to jest protokół MAC, który zapewnia ochronę przed selektywnymi atakami zagłuszania zamontowanymi za pomocą sprzętu obsługującego IEEE 802.15.4. Zasadniczo DEEJAM próbuje ukryć pakiety przed węzłem zakłócającym, aby uniknąć jego wyszukiwania, a tym samym ograniczyć wpływ pakietów, które i tak są uszkodzone. Jednak DEEJAM jest specjalnie dostosowany do sieci IEEE 802.15.4 i powoduje znaczne obciążenie obliczeniowe i zużycie energii. Ponadto Ashraf i in. (2012) zaproponowali niskonakładową platformę Jam-Buster, która wykorzystuje równy rozmiar pakietów, wieloblokowe ładunki i randomizację czasu budzenia urządzeń. W ten sposób zmniejsza przewidywalność czasu transmisji na urządzeniach sieciowych i eliminuje zróżnicowanie typów pakietów. W konsekwencji przeciwnik musi przesyłać dodatkowe sygnały zagłuszające, a zatem musi zużywać dodatkową energię, aby zachować skuteczność, i jest szybszy i łatwiejszy do wykrycia jako źródło ataku zagłuszającego. Jednak Jam-Buster zasadniczo nie przeciwdziała atakowi za pomocą rzeczywistego rozwiązania przeciwko zagłuszaniu, ale zamiast tego stara się, aby selektywne zagłuszanie było mniej wydajne i wygodne w wykonaniu. Proano i Lazos rozważają szczególny rodzaj

zagłuszania selektywnego, w którym tylko szczególnie ważne typy pakietów są zagłuszane podczas ich transmisji, i proponują pewne techniki łagodzenia skutków ataku, oparte na prymitywach kryptograficznych. Chociaż może to być dobre rozwiązanie do przeciwdziałania klasyfikacji pakietów, wymaga również, aby pakiety były w całości zaszyfrowane, dzięki czemu urządzenia odbiorcze nie mogą wcześniej przerwać odbioru pakietów, które nie są do nich zaadresowane. Daidone i inni autorzy proponują rozwiązanie przeciwdziałające selektywnemu zagłuszaniu w sieciach IEEE 802.15.4 oraz wykorzystują mechanizm Guaranteed Time Slot (GTS). W trybie GTS centralny koordynator może przydzielić co najwyżej siedem zarezerwowanych szczelin czasowych do węzłów czujnikowych w sieci, w każdej superramce. Proponowane rozwiązanie odnosi się do poważnej podatności GTS na selektywne zagłuszanie, opisanej wcześniej w Sokullu i całkowicie opiera się na węźle centralnego koordynatora, który oblicza i losowo zmienia wzorzec wykorzystania gniazd GTS w każdej superramce. Pozwala to na zmniejszenie skuteczności ataku do 1/7, ale jednocześnie sprawia, że centralny koordynator jest pojedynczym punktem awarii. Co więcej, proponowany środek zaradczy uwzględnia w szczególności standard IEEE 802.15.4, a zatem nie jest ogólny. Wreszcie Tiloca i inni przedstawiają JAMMY, dynamiczne i rozproszone rozwiązanie, które przeciwdziała selektywnemu zagłuszaniu w ogólnych sieciach bezprzewodowych TDMA. W zasadzie JAMMY permutuje wzorzec wykorzystania szczelin urządzeń sieciowych w sposób losowy, w każdej superramce. Wynika z tego, że szczeliny powiązane z każdym urządzeniem sieciowym zmieniają się w sposób nieprzewidywalny na podstawie superramki. W rezultacie przeciwnik ma jedyną możliwość zacięcia wybranych losowo szczelin, mając nadzieję na odgadnięcie tych, które zamierzone urządzenie ofiary wykorzystuje do transmisji w tej superramce. W przypadku, gdy tylko jedna szczelina na urządzenie jest używana w każdej superramce, wtedy udany atak selektywnego zagłuszania ma prawdopodobieństwo 1/N, przy danym N liczbie szczelin tworzących superramkę. Co ważniejsze, JAMMY nie opiera się na żadnej jednostce centralnej i jest całkowicie rozproszony, to znaczy, że każde urządzenie sieciowe określa szczeliny do wykorzystania do transmisji w następnej superramce w sposób autonomiczny (tj. bez wymiany danych z innymi urządzeniami) i w spójny sposób (tj. bez kolizji z innymi urządzeniami). Co więcej, JAMMY jest dynamiczny, ponieważ wiele urządzeń może w dowolnym momencie dołączyć i opuścić sieć. Wreszcie, w zasadzie nadaje się do użytku w dowolnej sieci bezprzewodowej TDMA, niezależnie od konkretnej technologii.

Systemy wykrywania włamań i zapory ogniowe

Chociaż bezpieczeństwo komunikacji może chronić przed atakami, których celem jest poufność i integralność wrażliwych danych, istnieje szereg ataków mających na celu zakłócanie sieci czujników jako całości, takich jak ataki DoS. Zwłaszcza w przypadku sieci o znaczeniu krytycznym konieczne jest zatem przyjęcie środków zaradczych przeciwko tego typu atakom. Istnieje wiele podejść, które mogą chronić sieci czujników przed atakami DoS poprzez wykrywanie i ewentualnie przeciwdziałanie włamaniom i innym anomaliom. Te systemy wykrywania włamań (IDS) dla sieci czujników można sklasyfikować jako oparte na watchdogu, w których wyznaczony komponent jest używany do wykrywania samolubnych węzłów i złośliwych napastników; oparte na sygnaturach lub regułach, które wykrywają znane typy włamań poprzez rozpoznawanie złych wzorców; oparte na anomaliach, które wykrywa odchylenia od modelu normalnego lub dobrego zachowania; lub wykrywanie oparte na klastrach, gdzie grupa węzłów wybiera lidera, który uczestniczy w procesie wykrywania. Te konwencjonalne podejścia są ukierunkowane na bezprzewodowe sieci czujników i nie mają bezpośredniego zastosowania w IoT. W rzeczywistości zakładają, że nie jest dostępny żaden centralny sterownik ani jednostka zarządzająca, wiadomości w sieci czujników nie są zabezpieczone, a węzły nie są globalnie identyfikowalne. W przeciwieństwie do tradycyjnej sieci czujników, sieć 6LoWPAN zawiera router graniczny 6LoWPAN (6BR), który jest zawsze osiągalny, węzły czujników są globalnie identyfikowane za pomocą adresu IPv6, a bezpieczeństwo komunikacji jest podstawowym wymogiem

do spełnienia. Ponadto zbudowanie systemu IDS odpowiedniego dla IoT pozostaje bardzo trudnym zadaniem, ponieważ urządzenia 6LoWPAN mają być globalnie osiągalne, są połączone za pomocą stratnych łączy bezprzewodowych, są w większości jednostkami o ograniczonych zasobach i wykorzystują niezbyt dobrze przetestowane nowe protokoły IoT , na przykład CoAP, 6LoWPAN i RPL. Wykorzystując te szanse i zagrożenia, opracowano system IDS odpowiedni dla IoT o nazwie SVELTE. SVELTE przede wszystkim broni się przed atakami routingu na sieci 6LoWPAN z protokołem RPL. Kluczowa decyzja w projekcie IDS dotyczy umieszczenia modułów IDS w sieci. SVELTE to hybrydowy system, w którym ciężkie moduły wykrywania są hostowane w 6BR, a w węzłach czujników dodano lekkie funkcje powiadamiania. SVELTE ma dwa scentralizowane moduły hostowane w 6BR: (i) 6LoWPAN Mapper (6Mapper), który zbiera informacje specyficzne dla RPL z poszczególnych węzłów czujników i odbudowuje sieć RPL w 6BR oraz (ii) moduły detekcji, które analizują zmapowane dane, a następnie wykonują rzeczywiste wykrywanie włamań. Moduł 6Mapper ma odpowiednią funkcję w każdym indywidualnym węźle, który dostarcza informacje o mapowaniu do 6BR. Chociaż SVELTE może chronić sieci 6LoWPAN przed atakami routingu w sieci, fundamentalne jest, aby urządzenia czujnikowe były chronione przed globalnymi, potężnymi intruzami. Na przykład host w Internecie może z łatwością przeprowadzać ataki DoS mające na celu wyczerpanie ograniczonych zasobów węzła czujnika. Zapora jest zwykle używana do filtrowania wiadomości z hostów internetowych przeznaczonych do sieci lokalnych. W przypadku wdrożeń IoT, w których integralność i poufność wiadomości od końca do końca są wymuszane między urządzeniem czujnika a hostem w Internecie, zapora nie może zbadać zaszyfrowanej zawartości. SVELTE posiada rozproszoną mini-firewall dla sieci 6LoWPAN. Zarówno 6BR, jak i każdy węzeł z ograniczeniami mają moduł zapory. Oprócz oferowania klasycznych funkcji blokowania przed zewnętrznymi atakami określonymi przez administratora systemu, zapora może filtrować i blokować złośliwe hosty zewnętrzne, powiadamiane w czasie rzeczywistym przez legalne węzły w sieci 6LoWPAN.

Wniosek

Omówiliśmy najważniejsze protokoły i technologie bezpieczeństwa, które są obecnie dostępne do przyjęcia w IoT, na różnych warstwach typowego stosu komunikacyjnego. Ponadto w tym rozdziale omówiono konkretne problemy związane z bezpieczeństwem w IoT oraz rozwiązania, które mają je rozwiązać, wykorzystując najnowsze badania i wysiłki standaryzacyjne. Oczekuje się, że zostanie wykonanych więcej pracy i należy podjąć dalsze wyzwania, aby osiągnąć de facto bezpieczny Internet Rzeczy, a ostatecznie utorować drogę do jego wszechobecnego i szerokiego przyjęcia. Następujące trendy i tematy będą szczególnie odgrywać kluczową rolę w bieżącym programie badań i standaryzacji dotyczących bezpieczeństwa w IoT. Po pierwsze, coraz więcej scenariuszy aplikacji wymaga wyraźnej obsługi bezpiecznej komunikacji grupowej, wbudowanej w różne warstwy stosu. Obejmuje to zarówno rzeczywistą ochronę wiadomości, jak i zarządzanie materiałami zabezpieczającymi i oczekuje się, że skupi się zwłaszcza na warstwie aplikacji (poprzez podejścia oparte na bezpieczeństwie obiektów) oraz na warstwie sieci (poprzez protokoły IPsec i IKEv2). Po drugie, obszerny zestaw profili dla IoT należy określić, aby zdefiniować, w jaki sposób skorzystać z ram autoryzacji opartych na OAuth 2.0, zarówno do obsługi faktycznego autoryzowanego dostępu do zdalnych zasobów, jak i do wspomagania kontekstowego ustanawiania bezpiecznych kanałów z serwerami zasobów IoT. Wreszcie, potrzebne są nowatorskie podejścia do poprawy odporności jednostek IoT na ataki typu „odmowa usług", których celem jest udaremnienie ich dostępności, a nawet wyczerpanie ich zasobów energetycznych. Takie podejście powinno lepiej wykorzystywać informacje kontekstowe i dostosowywać się do zmiennych warunków ataku. Wreszcie, te otwarte problemy wymagają wydajnych rozwiązań, które wykazują trwały wpływ na wydajność i wrażenia użytkownika oraz które można realistycznie wdrażać i łatwo konserwować nawet w przypadku dużych, dynamicznych wdrożeń urządzeń o ograniczonych zasobach.