December 2022

Enhancing Cryptographic Security by Partial Key Management

Omer Berkman

Marcel M. Moti Yung

Enhancing Cryptographic Security by Partial Key Management

ABSTRACT

Cryptographic security can degrade over time due to attackers using more powerful hardware or more sophisticated software. To maintain security, cryptographic machinery is replaced or strengthened as and when weaknesses are found. However, updating certain cryptographic components is infeasible or expensive, resulting in updates that either don't occur or are delayed. This disclosure describes techniques to enhance cryptographic security by updating portions of a cryptographic system when updating cryptographic parameters is only partially possible. Authenticating data (auth-data) sent by the un-updateable component during normal operation is used to deliver new and upgraded security parameters to secure communication. Security degradation resulting from the inability to effect an end-to-end update is limited to the immediate vicinity of the un-updateable component. The described techniques can be used to improve security of Internet-of-Things (IoT) device communication.

KEYWORDS

- Internet of things (IoT)
- Bluetooth beacon
- Accessory tracking
- Cryptographic key
- Diffie-Hellman public key
- Ephemeral public key
- Cryptographic protocol
- Public-key cryptography

BACKGROUND

Security based on cryptographic protocols can degrade over time due to advances in cryptanalysis or due to attackers using more powerful hardware and/or more sophisticated software. To maintain security, cryptographic machinery is replaced or strengthened as and when cryptographic weaknesses are found. However, updating certain cryptographic components may be infeasible or expensive, resulting in updates that either don't occur or are delayed. Examples include cryptographic components in the field; components that belong to end users; components that lack remote secure update capability; field components that require a hardware patch or replacement; etc.

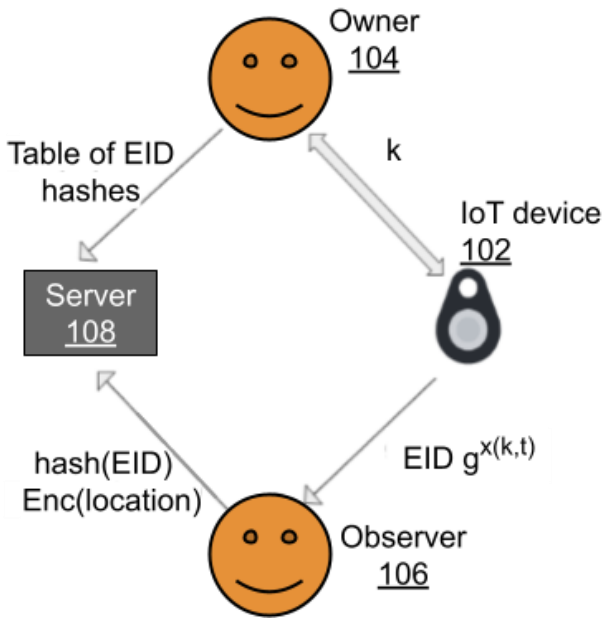

Fig. 1: Example cryptographic application with a relatively weak field device

Consider the example of Fig. 1, which illustrates a wireless Internet-of-Things (IoT) device (102) owned by an owner (104). During setup, the owner shares a key *k* with the IoT device. The IoT device uses *k* and the time *t* to generate and advertise over a beacon an ephemeral public key $g^{x(k,t)}$, which also serves as an ephemeral ID (EID), e.g., a Diffie-Hellman

(DH) public key whose secret part is known to the owner. The IoT device emits the beacon (which can be, e.g., a Bluetooth low energy (BLE) beacon) periodically, enabling it to be located if it gets lost.

When in close proximity to the IoT device, the owner can readily locate the device using its beacon. However, once the IoT device is deployed in the field (or is lost), the owner may be too far away from the IoT device to be able to directly receive its beacon. An observer (106, e.g., a passerby with a wireless connection) can receive the beacon and assist in locating the lost IoT device as follows. At periodic intervals, e.g., once a day, the owner shares with a server (108) a table of EID hashes. The nearby observer uses the EID received from the IoT device and their own ephemeral public key to anonymously encrypt the location and send a hash of the EID and the encrypted location to the server. Using the table of EID hashes, the server associates the received hashed EID with the owner but, not knowing *k*, is unable to decrypt the encrypted location. The owner accesses the server, downloads the encrypted location, and decrypts it. The identity of the observer is protected from the owner and the server, while the location of the observer (and hence of the IoT device) is protected from decryption at the server.

As explained earlier, based on the capabilities of potential attackers, the cryptographic specification for the beacon can periodically be upgraded. For example, the size of the elliptic-curve-based ephemeral public key advertised by the IoT device on its beacon can be increased to make the IoT device and the encrypted location resistant to attacks. However, the IoT device, being in the field, may not be amenable to being updated to a more robust cryptographic standard.

In general, cryptographic techniques evolve, and cryptography can become out-of-date. A cryptographic system is changed to accommodate new cryptography, e.g., with stronger, longer

keys, but not all computing elements (e.g., IoT devices) in the system can be changed (updated) at the same time. To enable old, un-updated systems to work with new, up-to-date systems, the conventional strategy is to enable backward compatibility: The new system is expected to interact with the old system and adjust to it.

However, the current practice of backward compatibility makes it so that the new system degrades itself to work with the old cryptography of the old system, e.g., in an old-system mode. This means that weak cryptography continues to be used, well past the system update, due to asynchrony in the update across components in the system. This is often unacceptable as a security principle.

DESCRIPTION

This disclosure describes partial key management, e.g., techniques to enhance cryptographic security by updating portions of a cryptographic system when updating cryptographic parameters is only partially possible. Security degradation that can result from the inability to effect an end-to-end update is limited to the immediate vicinity of the un-updateable component. The techniques are applicable in particular to the Internet-of-Things (IoT), where an un-updateable device often communicates with another component (the owner) via another nearby component (the gateway or observer) and the cloud. In such scenarios, security degradation is limited to the communication between the un-updateable device and the gateway.

Effectively, the techniques achieve backward compatibility, but backward compatibility does not merely revert the system to an older security level; rather, data sent from an un-updated component during its normal operation is utilized to enhance the security of the rest of the system so that only the part of the system communicating directly with the non-updateable

component remains with the weaker security while the security in the rest of the system is enhanced.

Unlike conventional approaches, end-to-end system security isn’t degraded to accommodate an un-updated component. The described techniques enable the sustenance and advance of cryptographic security for longer durations, even across cryptographic updates that don’t reach every component in the system.

Specifically, referring to the example of Fig. 1, the owner sends periodically to the server for each EID that the beacon advertises within that period a new and stronger ephemeral public key signed with the private key associated with the EID, alongside the EID hash. When an observer (serving as gateway) receives the EID of the IoT device, it sends the hash of the EID to the server. The server locates the respective signed new key and sends it to the observer. The observer verifies the signature using the EID, encrypts their location with the new key and sends it to the server. Thus, rather than using the EID to encrypt the observer's location (as is conventional), the EID is used to verify the signature on a new and stronger key. To prevent an attacker from forging a signature using a brute force attack on the weaker old key used to generate the signature, the observer sets a short timeout after which it refuses to accept the new key. In this manner, cryptographic security is maintained without modifying the IoT device or its beacon.

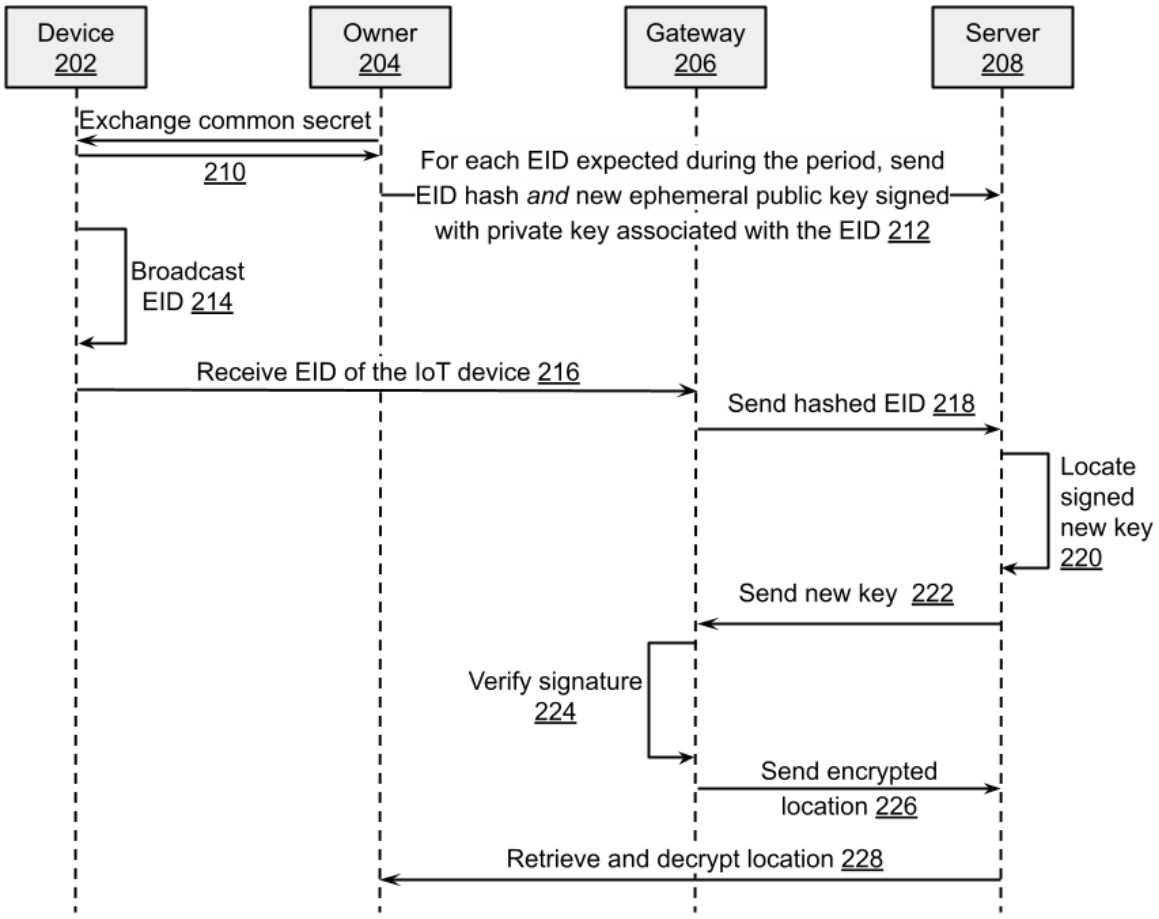

Fig. 2: Enhancing cryptographic security by updating portions of a cryptographic system

Fig. 2 illustrates enhancing cryptographic security by updating portions of a cryptographic system when updating cryptographic parameters is only partially possible. A device (202) is in the field and lacks internet connectivity. Therefore, it cannot be updated to the latest cryptographic standard. The device can, however, broadcast wirelessly over a limited range, using, e.g., BLE beacons. The device is owned by an owner (204) device that has internet connectivity and is capable of the latest cryptographic standard. The owner and the device communicate securely using one-way or two-way communication, which is readily possible when they are in close proximity, but is only indirectly possible (if at all) when they are not.

Initially, when the owner and the device are in close proximity, common secrets are exchanged or otherwise generated (210). Once deployed in the field (or lost), the device and the owner are no longer in close proximity to each other and cannot communicate directly. Indirect device-owner communication is made possible via a gateway (206) that is in proximity to the device. The gateway has internet connectivity and the latest cryptographic standard. The gateway can communicate with the device using its communication medium, e.g., BLE. The gateway may be ad hoc, e.g., not fixed, and there may be times when there is no gateway near the device. For example, the gateway can be a Bluetooth-enabled smartphone of a person who happens to be close to or walk by the device. The gateway cannot share secrets with the owner or the device nor can it access their public keys, if such public keys are at all used. Thus, there is no a priori secure communication channel between the gateway and the owner.

At some point, to forestall attackers with newer and more powerful technology, cryptographic standards are generally upgraded. However, upgrading or replacing the device is infeasible or expensive. As shown below, the inability to upgrade the device need not result in a degradation of end-to-end security. Rather, degradation can be limited to the immediate proximity of the device, e.g., to the device-gateway communication, while maintaining upgraded, secure gateway-owner communication as described below.

The owner periodically sends to a server (208) for each EID the beacon is expected to advertise during the period a new and stronger ephemeral public key signed with the private key associated with the EID alongside the EID hash (212). The device uses the key k and the time t to generate and broadcast over its beacon (214) an ephemeral public key $g^{x(k,t)}$, which also serves as an ephemeral ID (EID).

When the gateway receives an EID from the IoT device (216), it sends a hash of the EID to the server (218). The server locates the respective signed new key (220) and sends it to the gateway (222). The gateway verifies the signature using the EID (224), encrypts its location with the new key, and sends the encrypted location to the server (226). The owner retrieves the encrypted location from the server and decrypts it (228).

To prevent an attacker from forging a signature using a brute force attack on the weaker old key used to generate the signature, the gateway sets a short timeout after which it refuses to accept the new key. Effectively, the owner uses the auth-data, e.g., the ephemeral DH public-key, which is common to itself and the device, to authenticate and possibly also encrypt a new key, and to send it to the gateway (or, e.g., apply secure key exchange with the gateway). In this manner, cryptographic security is maintained without modifying the device.

Generally, the techniques limit degradation to the immediate proximity of an un-updateable component or device by:

- identifying data, known as auth-data, which is sent during normal operation (or can be sent upon interactive command) by the un-updateable device such that auth-data enables the authentication of other data;
- delivering to the gateway new security parameters authenticated by auth-data or by using auth-data to securely negotiate new security parameters; and
- securing the communication between the gateway and the owner using the new security parameters.

Auth-data can be based on cryptographic keys that do not provide long-term security. In such cases, authentication is performed within a specified, relatively short, time period. In addition to authenticating new security parameters, it is also possible to encrypt them.

Auth-data can exist even if authentication is a priori not needed or used. For example, in some cases, encrypted data using keys known to the owner may be used as auth-data.

In many real systems, such auth-data exist as a result of the system using some form of cryptography. For concreteness, consider a system where two entities A and B communicate with each other, where A which is not updateable sends and receives data through another entity G (G stands for gateway), where G may not be fixed or known in advance. In this case data sent by A as part of its normal operation and which can be considered as Auth-data include a public key (or ephemeral public key) for which B knows the private portion or a pseudorandom string that B can generate by itself. Examples of the latter include symmetrically encrypted data or message authentication code (if B knows the plaintext data or if the plaintext data is from a small domain), randomly generated ephemeral IDs, etc.

Attack model

Consider again for concreteness the scenario above with A, B and G (that is, where A and B communicate by the gateway G near A and where A is non-updateable). Assume also that the system is upgraded and because A is non-updateable, the technique disclosed above has been implemented into the system using some form of Auth-data . Below, attacks on the disclosed system in different scenarios and different types of Auth-data are considered.

If there is no attack on the A-G link, the end-to-end communication between A and B via G is secure. This is the typical case since A and G are in proximity to each other.

Attacks on the A-G link can also be thwarted, as follows. Consider the case of an eavesdropping attacker on the A-G link which tries to use the data it observes on the A-G link to learn auth-data in order to deceive G by playing the role of B. Note that such an eavesdropping

attacker can only attack the system if auth-data is based on symmetric cryptography. Even so, there are ways to thwart such an attack, as follows:

1. B can send authenticated new security parameters (or apply its part of the key exchange protocol) before auth-data is broadcast.
2. B can prepare authenticated new security parameters before auth-data is broadcast and request a secure time-server to sign the new security parameters.

The system may also be able to thwart attacks in which the attacker is capable of applying man-in-the-middle (MITM) attacks on the A-G link. For example, this is possible if G is able to verify authenticity of Auth-data.

Note also that for the device-finding system such as the one described above, an active MITM attack is not really useful since the attacker is close to the beacon and may have other means (such as GPS) to find out the location (which in the device-finding application is the secret the system protects).

Below are further example applications, additional to the above-described, device-finding application.

Example application: telemetric data

A crowdsourced system enables a beacon to send telemetric data and other information to the beacon's owner when the owner and the beacon are not in proximity to each other as follows.

- The beacon advertises an ephemeral ID (EID) that changes every few minutes (so as to not be tracked) and encrypted data. Both the ephemeral ID and encrypted data are generated using symmetric cryptography.
- A nearby smartphone (or other device) acting as a gateway picks the EID and the encrypted data and forwards them to a central server. Effectively, this is a mobile ad-hoc

network, typically implemented in a layer away from the user and automatically performed by the phone.

- The server, in turn, associates the EID with the beacon's owner (using a table containing all EIDs expected to be advertised by all beacons that day).
- The owner can thereafter access the server, get the encrypted data of its beacon and decrypt it.

Modifications to handle un-updateable beacons: If one or more beacons are un-updateable due to their being in the field or being away from their owners, the following can be implemented.

- Being pseudorandom, the encrypted data and/or the EID can be used as the auth-data. (If the EID is used as part of auth-data then the server must be made to work with hashed EIDs rather than the EIDs, so as to break symmetry between the producing devices and the checking server.) For example, once a day, the owner can upload to the server a table including, for each expected EID, a new key (or other parameters) encrypted and authenticated using the secret associated with the auth-data associated with that EID.
- Upon getting an EID and encrypted data from a beacon, a gateway now forwards only the EID (or the hashed EID) to the server. The server finds the respective new key (in encrypted and authenticated form) and sends it to the gateway. The gateway authenticates and decrypts the new key using auth-data, uses it to secure the beacon's encrypted data (so the data would be doubly encrypted) and sends the result to the server.

Example application: transport layer security (TLS)

A cipher suite agreed upon between a client A and server B in TLS may not necessarily be strong enough since the server, the client or both may not support a cipher suite of a desired

strength, as a result of flawed implementation or for other reasons. Suppose client A doesn't support a sufficiently strong cipher suite, and that it is un-updateable.

Modifications to handle an un-updateable client: Leveraging the fact that each data packet exchanged between the client and server goes through multiple hops, the server G (gateway) in the first hop from A can negotiate with B a stronger cipher suite (or agree on encryption/authentication outside the TLS protocol). To authenticate (and possibly encrypt) the new cipher suite, B and G may be able to use the session's encrypted data or MAC as auth-data. The result is that each packet is protected by the weaker cipher suite only in the first hop (between A and G) and is protected by both the weaker inner cipher suite and the stronger outer suite between G and B. If G is unable to negotiate a suitable cipher suite with B as well, then the server G′, G′ being the next hop after G, may be able to negotiate a suitable cipher suite with B. The goal is to minimize the prefix of exposure. Unlike services that break TLS to serve as content readers (or middleboxes [3]), security is increased on the suffix of the route (rest of the system) without decrypting any ciphertext, old or new.

CONCLUSION

This disclosure describes techniques to enhance cryptographic security by updating portions of a cryptographic system when updating cryptographic parameters is only partially possible. Authenticating data (auth-data) sent by the un-updateable component during normal operation is used to deliver new and upgraded security parameters to secure communication. Security degradation resulting from the inability to effect an end-to-end update is limited to the immediate vicinity of the un-updateable component. The described techniques can be used to improve security of Internet-of-Things (IoT) device communication.

REFERENCES

[1] Sousa, Patrícia R., Luís Magalhães, João S. Resende, Rolando Martins, and Luís Antunes. "Provisioning, Authentication and Secure Communications for IoT Devices on FIWARE." *Sensors* 21, no. 17 (2021): 5898.

[2] Melo, Wilson S., Raphael Machado, and Luiz FRC Carmo. "Using physical context-based authentication against external attacks: Models and protocols." *Security and Communication Networks* 2018 (2018).

[3] "Middlebox," https://en.wikipedia.org/wiki/Middlebox accessed Nov. 23, 2022.

[4] Liron David, Avinatan Hassidim, Yossi Matias, Moti Yung, and Alon Ziv. "Eddystone-EID: secure and private infrastructural protocol for BLE beacons." In *IEEE Transactions on Information Forensics and Security*.

[5] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. "Who can find my devices? Security and privacy of Apple's crowd-sourced bluetooth location tracking system." *arXiv preprint arXiv:2103.02282*, 2021.

[6] Tyson Macaulay and Richard Henderson. "Cryptographic agility in practice." https://uploads-ssl.webflow.com/612fec6a451c71c9308f4b69/614b712e53ce8f7fad0c3c4a_ISG_AgilityUseCases_Whitepaper-FINAL.pdf. accessed Nov. 23, 2022.

[7] Tingfeng Yu, James Henderson, Alwen Tiu, and Thomas Haines. "Privacy analysis of Samsung's crowd-sourced Bluetooth location tracking system." *arXiv preprint arXiv:2210.14702,* 2022.