July 05, 2017

Trust Chain Recovery Method For Secured Connected Devices Solutions

Olivier KREET
ALE International

Georges NOGUEIRA
ALE International

Docket Number	**FR82017003**
Title	**“Trust Chain Recovery Method For Secured Connected Devices Solutions”**
Contributors	**Olivier KREET, Georges NOGUEIRA**
Company	**ALE International**

Description of the technical solution:

Computer networks are facing unprecedented security risks. Dramatically impacting threats are now targeting our critical infrastructures, such as :

- massive denial of services in the Internet because of lacking security in the Internet of Things (IoT),
- or, the recent ransomware campaign that spreads all over the world.

The need of security mechanisms is now inevitable to maintain the system. This target can be performed thanks to a set of now mature security protocols, but it will only succeed if we achieve the required security management of the multiplying seamless endpoints that do not allow any human interaction.
The following solution addresses this requirement of trust management.

Secured communications of IP based protocols are usually performed with the help of TLS transport layer protocol. TLS requires the use of digital certificates to enable (at least) the server authentication from connecting clients, which in turn must have a trust store containing the corresponding server public certificate prior to the client-server connection.

While this became the standard architecture, we will illustrate the solution in the field of IP Telephony which is a core business market for some companies. In VoIP technology, TLS based application protocols are used to secure the signaling link, while media link is usually secured thanks to SRTP protocol. TLS protocol requires the use of digital certificates as developed in the following.

In this description, we will use the following definitions:

- **Client** is a device connecting to a server with a TLS based protocol; clients must have trust stores containing public certificates thus enabling TLS connection to succeed after a server certificate verification.
- **Server** is a device that receives connection requests from clients; server must have a digital server certificate for enabling the TLS connection and proving its identity thanks to TLS authentication mechanisms.
- **CTL**, Certificate Trust List, is a content of public certificates to be imported in clients trust store and that is used by clients to perform servers authentication.
- **Secure installation** denotes a network of connecting clients and a server that rely on the use of any kind of TLS based protocols where the server is authenticated by clients.

- On the contrary, an **unsecure installation** denotes a network where no security protocol – TLS– is enabled.

- A **locked-in secure** installation, is a secure installation where clients have trust store configured to contain a given server public certificate, and by extension will not trust any other unknown server certificate, nor accept trust store modification as explained here below.

In a **locked-in secure** installation, clients are not supposed to accept any modification of their trust store from unknown sources for security reasons. Without such protection an attack can be performed by tampering with the client trust store in order to inject trust with a rogue server. Such attack is thus prevented by restricting trust store modifications with CTL that are only originated by the already known secure server/context. This principle forms a chain of trust where the CTL are all linked with the previous and the next one within a consistent security context.

This trust store protection however sets the following drawback: if the security configuration information is lost on server side and no backup is available, the security context that forms the chain of trust is lost and all clients configured with it are stuck with a lost configuration. Any modification of their trust store would not be possible with any CTL that could not be linked to their previous one.

In consequence, solutions must be found to allow recovery (or at least reset), especially for large networks of client devices. The capability to allow quick recovery and to avoid manual actions on each client is paramount. Both these properties, secure and automated, must be guaranteed so this procedure cannot be used undeservedly to set back the security of valid secured systems. Massive updates of clients trust store must be considered as a last-resort and a controlled solution for networks being blocked because of a loss of server security configuration or context.

Again, the solution to this problem must be under the strict control of the customer and possibly a technical support resource from the device vendor; only the customer knows that its network is in a situation that requires performing this massive update, and prevents any attacker to mess with its locked-in secure installation.

In situations where the chain of trust is lost and all clients configured with it cannot connect anymore to their communication server, most common solutions range from cumbersome procedures that do not easily scale, to solutions that sacrifice solution security to ease managing such cases. These solutions can be detailed as follows:

- Remove security context thru a manual action performed by an administrator locally on the phone. This is generally done from a local configuration menu on the terminal, which access it restricted to authorized admin persons so security cannot be disabled trivially. Access to such configuration typically requires entering an admin password. A menu to erase the security context is then proposed and results in the terminal to come back to its initial mode. In case of large site experiencing the loss-of-trust-chain issue, this operation requires the admin to intervene on all terminals, possibly on a large amount of geographical sites. This makes the recovery procedure long and costly, and leaves the telephony solution out of service for a duration that can cause significant business impacts for the end-customer. Such procedures are applied in some solutions.

- As a variant, it is possible to reset the phone to factory settings thru more of less constraining procedures. This can be done by accessing a local menu as on the previous solution, keeping the same level of constraint and relative security against unauthorized security deactivation. It can alternatively be based on a hardware procedure consisting in either a key sequence, or reset button, that can however quite easily be known by anyone. This still requires an individual action on each set, and in the second case opens the solution to security threats as anyone could easily deactivate security and possibly insert as man-in-the-middle when activating security again, making terminal trust rogue authorities.

- Finally, a theoretical possibility is to authorize endpoint to automatically accept any new authority even if it cannot be verified by the chain of trust stored locally. This solution of course does not protect against man-in-the-middle intrusions, and cannot be considered relevant on security standpoint. Prompting end-user on acceptance of unknown server certificate, as may happen on web browsers, cannot be considered a valid generic solution neither, considering that the terminals UI can be limited and capability to display relevant information might not be present and moreover that users might simply accept whatever message gets displayed by lack of understanding of what is exactly at stake.

The proposed procedure addresses shortcoming of existing solutions.

It enables an automated distribution of a recovery token resetting “stale” security context to all endpoints, requiring no per-set intervention. Instead of intervening on all endpoint, the solution administrator interfaces with the terminals’ vendor by providing a piece of information that is signed by the vendor, and returned to the admin. This recovery token is then injected in a configuration file that all endpoints automatically get during their standard startup procedures. The time to recover is limited to this interaction with the vendor support services, i.e. a matter of minutes rather than hours or days.

Although automatized, it maintains security properties of the solution by ensuring that the recovery token can only be emitted by the endpoint vendor and for authorized entities that the vendor can authenticate. It also limits the scope of the recovery token to a specific customer installation only, so re-using the recovery token on another system than the one is was requested for has no impact and brings no risk of degrading security.

Our solution provides a way to manage security context loss and recover a locked-in secure installation without the server administrator having to modify manually each clients of the deployment. This recovery process allows forcing clients to forget a specific CTL and go back to initial start-up (e.g. trust on first use mode if enabled). The solution relies on the following principles:

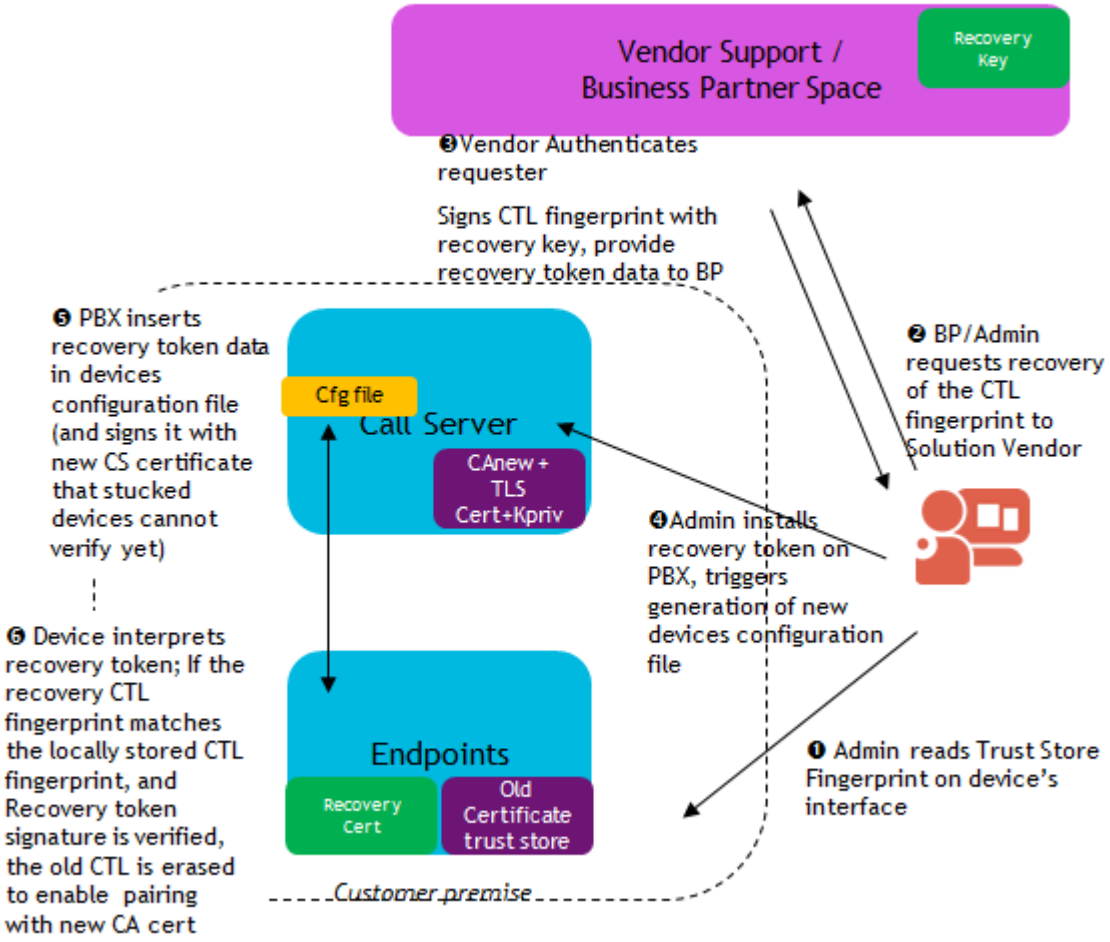
Precondition:

- The vendor manages a recovery key and its associated digital certificate, which is trusted natively by vendor clients (VoIP phones). This digital certificate is dedicated to TLS recovery procedures. This so-called recovery key remains under full control of the vendor and is not distributed to any third party, including the customers.

On loss of trust chain:

- The network administrator (i.e. customer or installer) must retrieve a fingerprint provided by client devices that describes the security context they are stuck with, i.e. the old and lost CTL. This action is performed by providing information obtained from the current CTL passed through cryptographic function that delivers a hash and signed fingerprint. This output information can be accessed by any means on a given device of the installation (this info is read-only and does not have to be protected by admin password, as considered not sensitive).
- The network administrator connects and authenticates on the vendor support service platform, so that the vendor is able to ensure the administrator is an allowed customer requester. This is also used for operation traceability purposes.
- The network administrator (i.e. customer or installer) provides the information to vendor support services, e.g. with a dedicated web services requiring the administrator to register, or by sharing thru any out of band method (e.g. mail, phone).
- The vendor uses its secret recovery key to sign with asymmetric cryptography the fingerprint input. The resulting information is denoted as the recovery token. The recovery token is then sent back to the network administrator (i.e. customer or installer).
- The recovery token is installed on the server to be deployed with available provisioning mechanisms that allow mass distribution services to the clients (e.g. it can be inserted as an optional parameter of the configuration file that devices automatically download at startup).
- Clients receive the recovery token from provisioning and verify the signature of the recovery token with asymmetric cryptography procedures and the embedded public vendor recovery key. The fingerprint of the locally stored CTL is compared to the one of the underlying recovery token. If matching occurs, the client allows erasing the locally stored CTL and gets back to an initial state allowing reconfiguring the security of the device.

The following diagram depicts the procedure as implemented in this solution with VoIP phones as clients and PBX as server:


Vendor Support / Business Partner Space
Recovery Key
❸Vendor Authenticates requester
Signs CTL fingerprint with recovery key, provide recovery token data to BP
❺ PBX inserts recovery token data in devices configuration file (and signs it with new CS certificate that stucked devices cannot verify yet)
Cfg file
Call Server
CAnew + TLS Cert+Kpriv
❷ BP/Admin requests recovery of the CTL fingerprint to Solution Vendor
❹Admin installs recovery token on PBX, triggers generation of new devices configuration file
❻ Device interprets recovery token; If the recovery CTL fingerprint matches the locally stored CTL fingerprint, and Recovery token signature is verified, the old CTL is erased to enable pairing with new CA cert
Endpoints
Recovery Cert
Old Certificate trust store
❶ Admin reads Trust Store Fingerprint on device's interface
Customer premise