Produced by
Samuel Lewis, YouFibre - samuel@substantial.group
Josef Karthauser, Truespeed - joe@truespeed.com

End-to-End Encryption and Message Signing for One Touch Switching

Revision 5 | 28 March 2023

Summary

In various calls including Ofcom, TOTSCo, and several industry members, the need for encrypting data in transit has been discussed but not yet agreed upon. There have also been several instances where message signing has been discussed for sender verification. This document sets out some of the reasons why it is incredibly important that TOTSCo and the rest of the industry work together on a standard for End-to-End Encryption and Message Signing within the existing TOTSCo framework.

Key Risks

By failing to implement End-to-End Encryption, the industry is presented with the following risks:

1. **Personally Identifiable Information (PII) is readable by TOTSCo and anyone who gains access to TOTSCo's servers.** Examples are addresses, names, phone numbers, and partial email addresses. The result of this is that TOTSCo then becomes a Data Controller[1] in the case of GDPR.
2. **Detailed Commercially Sensitive Information could be stored by TOTSCo which is market sensitive if exposed. Additionally, if TOTSCo's servers are compromised then this risk extends to data in transit.** This includes addresses, details about the services being lost/gained, and the relevant providers. It is therefore possible to visualise each provider's performance within different geographical regions, along with exposing who each provider's customers are moving towards.
3. **Man-in-the-Middle (MITM) attacks are possible if a hacker compromises TOTSCo servers.** Messages can be inspected and easily modified during transit.
4. **The solution is not compliant with the Telecoms Security Act** (see TSA 2.80 as an example).
5. **Simply put, TOTSCo becomes a GDPR Data Controller, and the data can be compromised. This means that we run the risk of repeating the 2015 TalkTalk Cyber Incident where all CPs may be impacted without having control over the TOTSCo Security Policy.**

By failing to implement Message Signing, the industry is presented with the following risks:

1. **Man-in-the-Middle (MITM) attacks are possible if a hacker compromises TOTSCo servers.** Messages can be modified during transit.
2. **Man-in-the-Middle (MITM) attacks are possible if a hacker compromises a Retail Communication Provider's API/Hub credentials.** Fraudulent messages can be sent, posing as the "source" RCP, trivially.
3. **Simply put, it is not possible to verify that the sender specified in the message header is the genuine sender.**

By failing to implement both E2EE and Message Signing, we risk repeating the mistake TalkTalk made back in 2015, by not controlling data processors for PII such as: names, addresses, phone numbers, and (partial) email addresses (see: https://ico.org.uk/about-the-ico/media-centre/talktalk-cyber-attack-how-the-ico-investigation-unfolded/). As an industry, we should follow the best practice of designing a system to be secure and only exposing what is essential. TOTSCo does not need to process PII.

We also run the risk of data breaches with similar effects as T-Mobile in 2021, mentioned in the TSA 2.12, "As an example, on 17 August 2021 it was confirmed that T-Mobile was subject to a data breach which saw the personal data of nearly 50 million customers being exposed. […] This enabled a single hacker to access customer data within a number of weeks." (see also: https://www.t-mobile.com/news/network/cyberattack-against-tmobile-and-our-customers).

Produced by
Samuel Lewis, YouFibre - samuel@substantial.group
Josef Karthauser, Truespeed - joe@truespeed.com

End-to-End Encryption (E2EE)

Below is a comparison table to help understand the differences between the current solution (No E2EE) and the proposed solution (E2EE).

No E2EE	With E2EE
CPs and TOTSCo are GDPR Data Controllers (big risk for TOTSCo)	TOTSCo is **not** a GDPR Data Controller, only CPs
TOTSCo is a singular target for hackers to gain access to industry wide Personally Identifiable Information	TOTSCo is **not** a singular target for hackers to gain access to industry wide Personally Identifiable Information
TOTSCo is a singular target for hackers to gain access to industry wide Commercially Sensitive Information	TOTSCo is **not** a singular target for hackers to gain access to industry wide Commercially Sensitive Information
Message Recipient **cannot** be confident that a message was destined for them, thus making it contractually unsound	Message Recipient **can** be confident that a message was destined for them, thus making it contractually sound
TOTSCo will need to implement a comparatively complicated set of systems, security frameworks, and audit policies to protect PII.	Easy to implement, GDPR-compliant at launch.
Not compliant with Telecoms Security Act	Compliant with Telecoms Security Act
Might meet the currently undefined Ofcom reporting requirements	The message type can remain exposed via the message header to meet Ofcom reporting requirements. Ofcom are likely to put data security over reporting requirements.

Message Signing

Below is a comparison table to help understand the differences between the current solution (No Message Signing) and the proposed solution (Message Signing).

No Message Signing	With Message Signing
Messages **can** be forged	Messages **cannot** be forged
Recipients **cannot** be certain that the sender is as stated in the header	Recipients **can** be certain that the sender is as stated in the header
MITM attacks are possible via the TOTSCo Hub	MITM attacks are **not** possible via the TOTSCo Hub
MITM attacks are possible via a Third Party using an RCP's credentials	MITM attacks are **not** possible via a Third Party using an RCP's credentials
Not contractually sound	Contractually sound
Current solution, no changes required.	Easy to implement

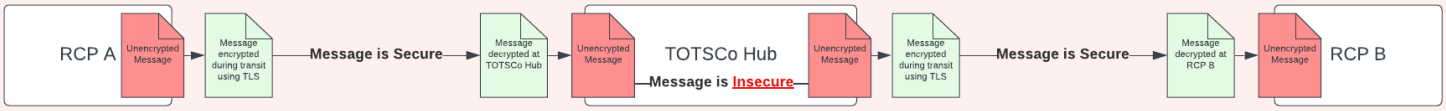
Message Signing works by using the sender's private key to sign a hash of the message body. Then, recipients of the message can verify that the sender is genuine by using the sender's public key to decrypt the signature[2].

Produced by
Samuel Lewis, YouFibre - samuel@substantial.group
Josef Karthauser, Truespeed - joe@truespeed.com

Differences as a Diagram

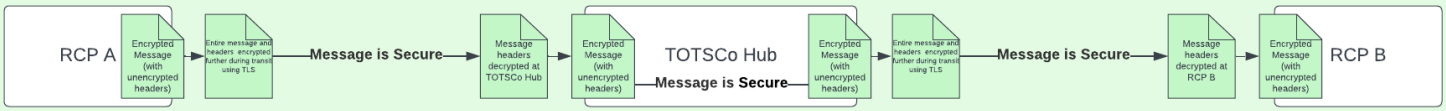
PII = Personally Identifiable Information
CSI = Commercial Sensitive Information

No E2EE



Unencrypted PII/CSI processed, possibly retained, TOTSCo is a GDPR Data Controller and privy to highly sensitive commercial information.

With E2EE



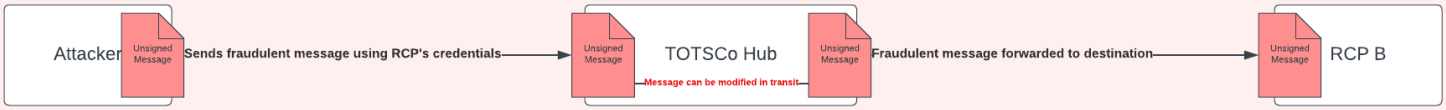
RCP A can be sure that only RCP B can read the contents.

TOTSCo is not privy to PII/CSI and is therefore not a GDPR Data Controller. TOTSCo can still report on the metadata of the messages.

RCP B has the ability to decrypt this Encrypted Message entirely.

No Message Signing

*Switch Orders are **not** contractually sound.*

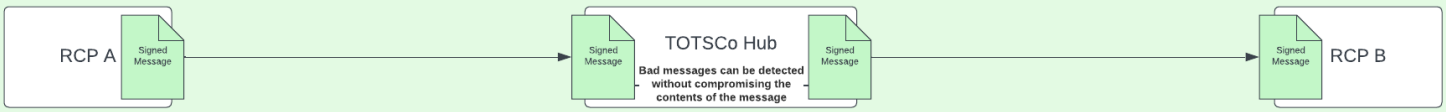


An attacker can pose as a CP if they are able to locate the CP's credentials for the TOTSCo hub. This can be used to send fraudulent switch requests.

RCP B has no means of determining if the message is genuinely from RCP A, or if it has been fraudulently created.

With Messaging Signing

*Switch Orders are contractually sound. Messages can be verified with access to only the sender's **Public Key**.*



RCP A can sign the contents of the message to provide RCP B with a means of verifying sender authenticity. Attackers would need RCP A's private key to send fraudulent messages.

RCP B has the ability to verify sender authenticity and ignore fraudulent messages.

Produced by
Samuel Lewis, YouFibre - samuel@substantial.group
Josef Karthauser, Truespeed - joe@truespeed.com

Payload Differences

Current Payload

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "string",
      "correlationID": "string"
    },
    "destination": {
      "type": "RCPID",
      "identity": "string",
      "correlationID": "string"
    }
  },
  "residentialSwitchMatchRequest": { "messageBodyHere": "..." }
}
```

Proposed Payload

```
{
  "envelope": {
    "source": {
      "type": "RCPID",
      "identity": "string",
      "correlationID": "string"
    },
    "destination": {
      "type": "RCPID",
      "identity": "string",
      "correlationID": "string"
    },
    "messageType": "residentialSwitchMatchRequest"
  },
  "encryptedPayload": "&&&%%%$$$EXAMPLEencryptedPAYLOAD$$$%%%&&&",
  "signature": "$KNL4n-poejek%%EXAMPLEmessageSIGNATURE%%dkl;me£{P'mnek[sopo'm[p2o;kled"
}
```

Produced by
Samuel Lewis, YouFibre - samuel@substantial.group
Josef Karthauser, Truespeed - joe@truespeed.com

Conclusion

By adding E2EE, senders can be certain that messages are only read by the intended recipient. By introducing Message Signing, recipients can be certain that the sender is as stated in the message header. With both, the interaction between the GCP and LCP is secure, verifiable and forms the basis of an end-to-end contract without requiring trust of a third party.

In addition to the data privacy concerns, TOTSCo is responsible for providing accurate reporting to Ofcom. There is no way for TOTSCo to be accurate if they cannot verify that messages are genuine, which is a problem solved by Message Signing. This can be aided by moving the Message Type into the envelope/header.

This proposal does not specify a specific encryption scheme, but by using open-source approaches, TOTSCo can choose from a variety of suitable implementation methods.

In conclusion, the changes proposed for the TOTSCo protocol vastly simplify the hub's security design whilst also reducing costs associated with security, initial development, and maintenance. With E2EE, TOTSCo does not need to become a GDPR Data Controller, simplifying legal requirements on both the CP side and TOTSCo's side of OTS.

Notes

1. Per TOTSCo Technical Design V.0.3.0: 3.4 "The archive is a persistent data store of all messages being processed through the post office. The information is held for a period before being purged. Storage policies may differ depending on the message type but GDPR must be adhered to."
2. For an example of how Message Signing encryption/decryption works, see this useful StackOverflow post which walks you through the procedure: https://stackoverflow.com/questions/18257185/how-does-a-public-key-verify-a-signature