What are Trusted Third Parties? (Part 2)

Hamsa Hassan, Data Protection Practitioner with London Ambulance Service NHS Trust, Sophie Stalla-Bourdillon, Senior Privacy Counsel & Legal Engineer, and Alfred Rossi, Research Scientist, both with Immuta, explore the potential of coded TTP

In Part 1, we unpacked three types of bona fide Trusted Third Parties ('TTP') which can be used as pseudonymisation services, master table escrow services or linking and de-identification services. In our terminology, bona fide TTP means an organisation or entity that is independent or functionally separated from the relying parties. In Part 2, we explore the potential of coded TTP, which leverage technology such as secure multi-party computation or trusted execution environments to perform specific TTP functions without the need to resort to a bona fide third party.

Multi-party data processing activities often involve queries over 'joins' of confidential information. For example, two hospitals may wish to determine which patients they treat in common without revealing to each other the names of other patients.

Bona fide TTPs may serve as clearing houses in such workflows, securely collecting inputs from the respective parties and following the agreed processing directives. Absent a bona fide TTP, there are two approaches to implementing confidential joins: secure multi-party computation ('SMC') which simulates a TTP without needing one, and trusted execution environments ('TEEs') which aim to implement TTPs in hardware.

Secure multi-party computation

Secure multi-party computation is a branch of cryptography concerned with designing protocols that enable parties to jointly perform a computation while keeping their respective inputs secret from each other. The SMC model is often described with the aid of two models of multi-party computation known as 'real-world' and 'ideal-world'.

The ideal-world model posits that there exists an incorruptible and universally trusted third party who is willing to carry out joint data processing tasks on behalf of the other parties. The TTP is provided with mutually-agreed instructions for carrying out processing activities, as well as the private input of each party; carries out the computation; and releases the results back to the contributing parties in accordance with their mutually-agreed instructions.

In contrast, the real-world model posits that all joint processing activities take place through messages exchanged directly between party members. A real-world multi-party computation (processing activity) is a secure multi-party computation provided that the real-world party members learn no more information than they would have under the ideal-world model, where all communication, processing, and distribution of results takes place via the TTP.

SMC offers at least the same guarantees as a traditional TTP: namely, that processing is faithfully performed in strict accordance with the mutual directives of the parties who remain unable to read each other's input. As a result, as with using a traditional TTP, SMC does not guarantee that data are de-identified or that relying parties are otherwise unable to infer personal information. Additional techniques and safeguards, such as the written agreement for the TTP to incorporate the use of privacy mechanisms, are required to achieve this.

Any processing activity can be carried out under SMC, however, not always efficiently. SMC protocols are typically slower than their ideal-world counterparts. Further, all of the parties must be online in order to participate in the protocol, which can present a significant technical barrier in practice.

Trusted execution environments

An alternative approach to implementing confidential joins of information is found among computer processor features for secure process isolation.

The resulting computing environments, known as Trusted Execution Environments ('TEEs'), isolate code and data from other processes — even privileged ones. As such, it is not possible for other processes, or even the operating system, to examine the intermediate state of computation.

The TEE processes the code and data independently of the broader computing environment, and can provide cryptographically verifiable attesta-

(Continued on page 16)

(Continued from page 15)

tions regarding code and data, allowing parties to ensure that the intended processing is occurring over their data as provided.

In this way, the traditional TTP is replaced by a piece of hardware designed so that processing activities cannot be observed from the outside.

As a result, if a TEE is utilised in a cloud computing scenario, the cloud provider remains unable to exfiltrate data from the trusted processing environment. At the same time, cryptographic attestations from the TEE ensure that the controller remains in control of processing, despite having delegated this activity to the cloud provider.

If the scenario is merely for delegation of processing activities with the output to be received by the controller, then no further steps (beyond encrypting the results prior to exiting the TEE) are required to protect the data from the cloud provider.

However, if the TTP is being used in a multi-party scenario, then as with traditional TTPs and SMC, there is no guarantee that data are de-identified or protected from inference attacks. Additional techniques and safeguards will have to be put in place to achieve this.

Open-query environments

One interesting extension to confidential multi-party joins allows any party member to initiate a query without seeking prior approval from the others.

As mentioned above, it is worth noting that while TTPs and their proxies (SMC, TEEs) can be used to implement open-query confidential join workflows, the use of these techniques alone does not automatically ensure that the data accessed through these environments are de-identified.

The root of the issue is that these mechanisms only safeguard against parties reading each other's inputs and the confidentiality of processing. Any party who receives the data is in the position to learn not only the output, but everything inferable about other parties' inputs when analysed in conjunction with the data and their own input. In effect, there is the possibility for exfiltration by query or by inference.

In the non-open model, a party uncomfortable with such possible exfiltration may simply withhold processing consent. In an open-query system, confidentiality must be guaranteed through agreement to use mechanisms like differential privacy, post-processing, or rules to abort query processing whenever the results would violate a parties' declared privacy objectives.

Whilst the imposition of such agreements might seem onerous, it allows for significantly better protection.

Conclusion

The use of a bona fide TTP does not guarantee that the architecture of the data analytics environment has been built in accordance with best practices. In reality, often the compliance burden is partially shifted to the bona fide TTP. A TTP acting as a controller will therefore have to demonstrate compliance and perform a comprehensive risk analysis.

Whilst process firewalls may offer support, they are only one piece of the puzzle. Data protection requirements such as the requirements for data minimisation, data security, and intervenability must still be adhered to. In particular, it is essential to understand that sending more data than necessary to the TTP could undermine data minimisation (i.e., the ability to tailor the amount of data to the purpose of the re-usage), and disconnecting data owners from the data analytics platform run by the TTP could undermine the ability for data subjects to exercise their rights over their data.

Because coded TTPs only protect the 'input side' of processing activities, it remains necessary to add additional data protection guarantees. SMC and TEEs provide technological alternatives equivalent to bona fide TTPs without an actual third party, provided that cryptographic assumptions hold, and that hardware remains bug and defect free.

Hamsa Hassan
Sophie Stalla-Bourdillon and
Alfred Rossi
hamsa.hassan@nhs.net
sstalla-bourdillon@immuta.com
arossi@immuta.com