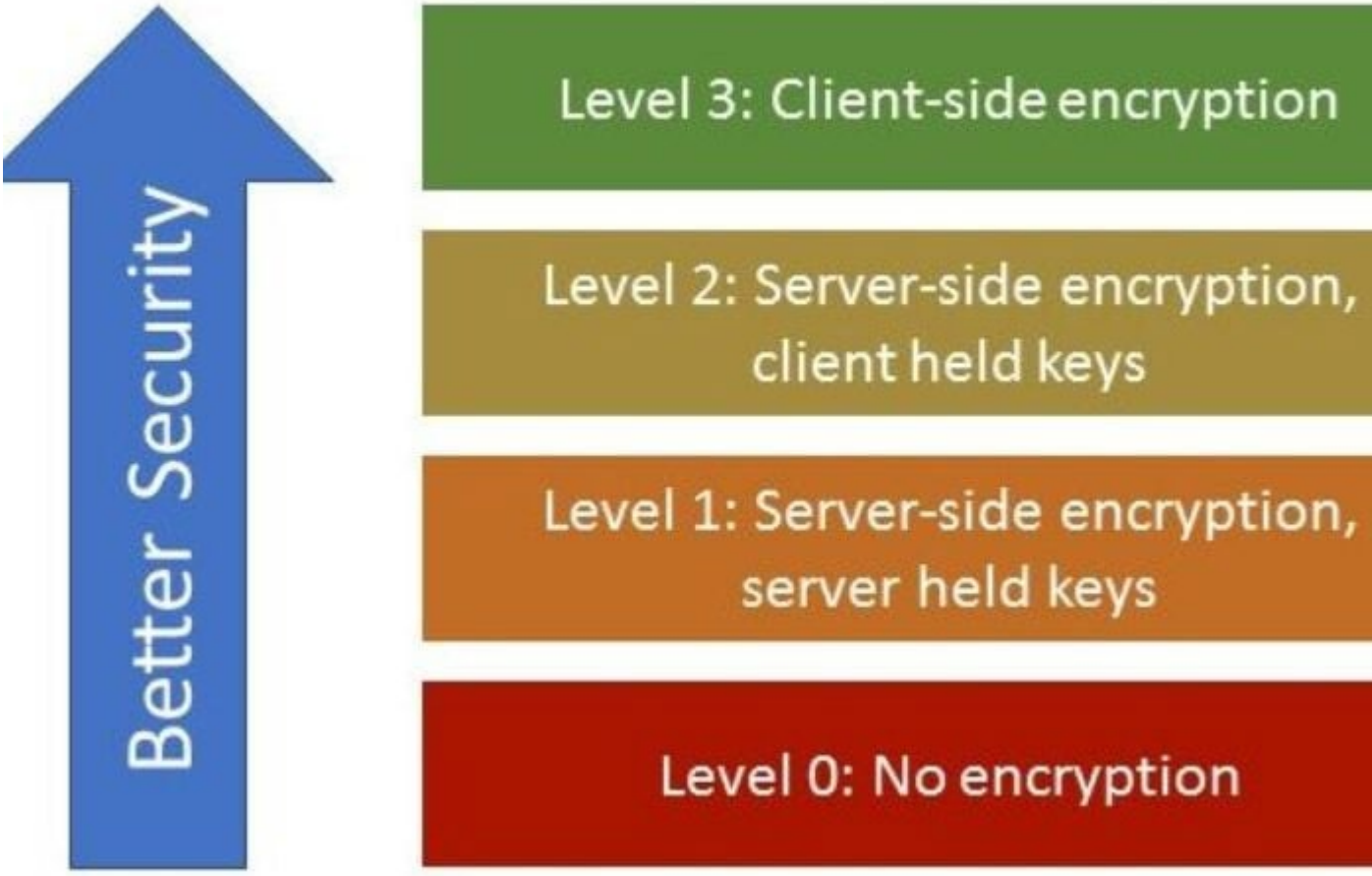


**Client-side vs server-side encryption – who holds the key?**

Technology News | May 14, 2018

By Julien Happich

---

To better understand encryption it is first necessary to consider the security of data in a state of transit and at rest.

Generally, data in transit is secure when TLS is used (in https, for example) to send data from A to B. In this scenario machines negotiate a secret encryption key between themselves and one-time keys are used only for that specific transmission. No person retains the key, which helps to keep the data secure.

When storing data in the long-term (data at rest), however, it is necessary to use a different type of encryption system; one which requires a secret key to decrypt the data. This is where users might encrypt but do so without achieving much security.

## The different types of encryption

The goal of encryption is to stop a security breach from becoming a data breach. It is designed to be an extra level of protection when there are privilege access-level breaches or accidental misconfigurations.

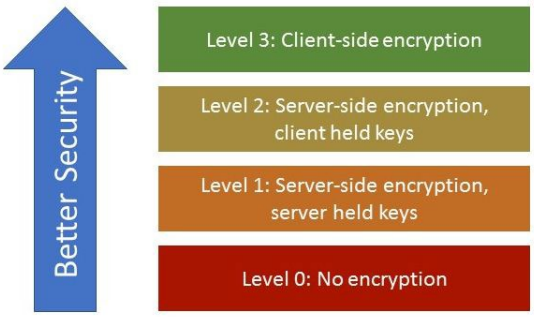


Fig. 1: The type of encryption chosen can make a
huge difference to the level of security provided.

To demonstrate why some forms of encryption offer better data security than others, let’s consider each type in turn:

- Client-side encryption – users encrypt their own data, with their own key.
- Server-side encryption with server held keys – users give regular (unencrypted) data to their cloud provider, with the latter encrypting it at their end. Users never see an encryption key and it's totally out of their hands.
- Server-side encryption with client held keys – users hold their own key but the server will encrypt/decrypt on their behalf.

The type of encryption chosen can make a huge difference to the level of security provided (see figure 1).

---

**Knowing your adversary**

When designing for security, it is important to know who your adversary is. Encryption protects data from three sets of parties:

- First party – malicious insider
- Second party – the cloud provider
- Third party – hackers/cybercriminals

When implementing multiple layers of security, it is best to put up each security barrier as high as possible, to minimise the potential for exposure. Client-side encryption is always favoured by cryptographers and security experts because it reduces the number of parties via which an attack or breach could happen. Only client-side encryption offers full protection against second and third parties.

Server-side encryption with server held keys is sometimes favoured by developers because it means that there are no changes required throughout the development process. Also,

traditionally client-side encryption has been difficult to implement and manage (although this is no longer the case) which has, unsurprisingly, put people off using it.

The reality is, however, that server-side encryption doesn't actually protect against third parties – and access-level misconfigurations can make it absolutely useless. This was demonstrated by the recent exposure of almost 200 million registered US voters by The Republican National Committee (RNC) data firm Deep Root Analytics and two other Republican contractors due to an access-control failure.

## Trade-off between cost and benefit


Fig. 2: What extra protections do different encryption types provide when regular access controls are breached?

So what do most people do? Most implement either no security (level 0) – which costs nothing but gives zero protection – or server-side encryption (levels 1 and 2), because it's simple and convenient (see Figure 2). This choice is reflected by research showing that 96% of breached data is not encrypted leaving organisations' valuable information open to manipulation by cybercriminals.

Level 2 security is, however, a good trade-off for embedded devices that run off long-life batteries. On such devices, it may be impractical to perform the encryption on the device due to battery drain or CPU slow-downs, so server-side encryption might be the best option, and better than none at all.

Level 3 security, client-side encryption, is the best for sufficiently powerful devices. For example, new encryption technologies such as ScramFS, which provides a library for developers to encrypt easily (for privacy) without needing to code crypto, can run on a Raspberry Pi device, encrypting HD video in real-time. It also provides authentication (detection of tampering) for each file saved through its API.

---

**Encryption's role in GDPR compliance**

Client-side encryption is the safest way to comply with certain aspects of the EU's new General Data Protection Regulation (GDPR) legislation.

As stated in Chapter 1, Article 4, for example, 'pseudonymisation' means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that **such additional information is kept separately** and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person.

Encryption is a form of pseudonymisation, with the encryption key as the "additional information"; "kept separately" is not clear (exactly how separate), but the intent is important because keeping the key alongside the data means a security

breach that reveals the encrypted data will also reveal the encryption key (see Figure 3).

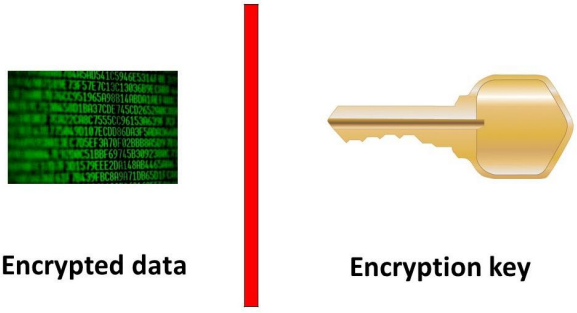


Fig. 3: Encrypted data and encryption key must be
kept separately to comply with Chapter 1, Article 4
of EU GDPR.

To exemplify this in a real-world scenario – say, due to a phishing attack, a company's cloud storage account credentials are compromised and their data is exposed. If the cloud server holds the keys, it will decrypt the data and serve it to the attacker. In order for encryption to be most effective, the encryption key must be kept secret from the server (which is only possible via client-side encryption).

## Client-side encryption = optimum data privacy

Dr Ron Steinfeld, a leader in post-quantum cryptography (Monash University, Australia), commented, "To eliminate trust in the server, I would recommend client-side encryption. The reason is that in server-side encryption methods, the user data and encryption key need to be communicated to the server to allow it to perform the encryption. Even if sensitive pieces of information are completely erased from server memory once encryption is completed, there are moments of time when they are stored there. During those times, the user's data is vulnerable to exposure if the server memory contents are

exposed (e.g. due to a server security breach, or intentional misuse of the data by the server).

Moreover, a breach into the server at any other time could initiate a hidden server process that waits until the server encrypts or decrypts user data and, when this is detected, dumps the server memory to the attacker to reveal the user's data and/or encryption keys. Consequently, server-side encryption still involves trust in the server to keep data private. In contrast, when using client-side encryption, the user's data and key are never revealed to the server, only the encrypted data is revealed."