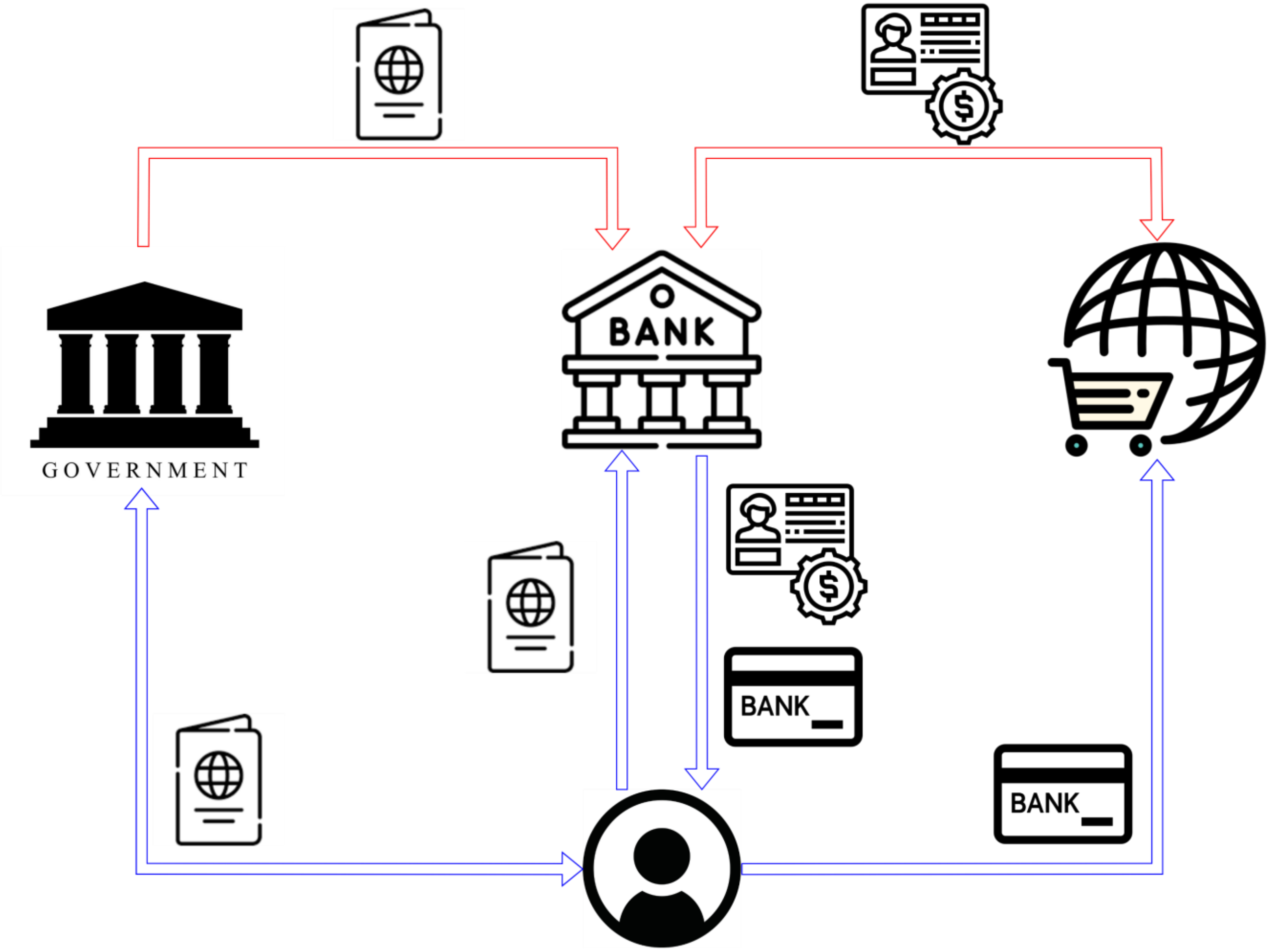
Anonymous authentication for mitigation extended attack surface in zero trust systems

Kryptonite

Zero Trust security model for personal data protection

Federated model (OpenID (2005), OAuth (2006), FIDO UAF, U2F (2014))

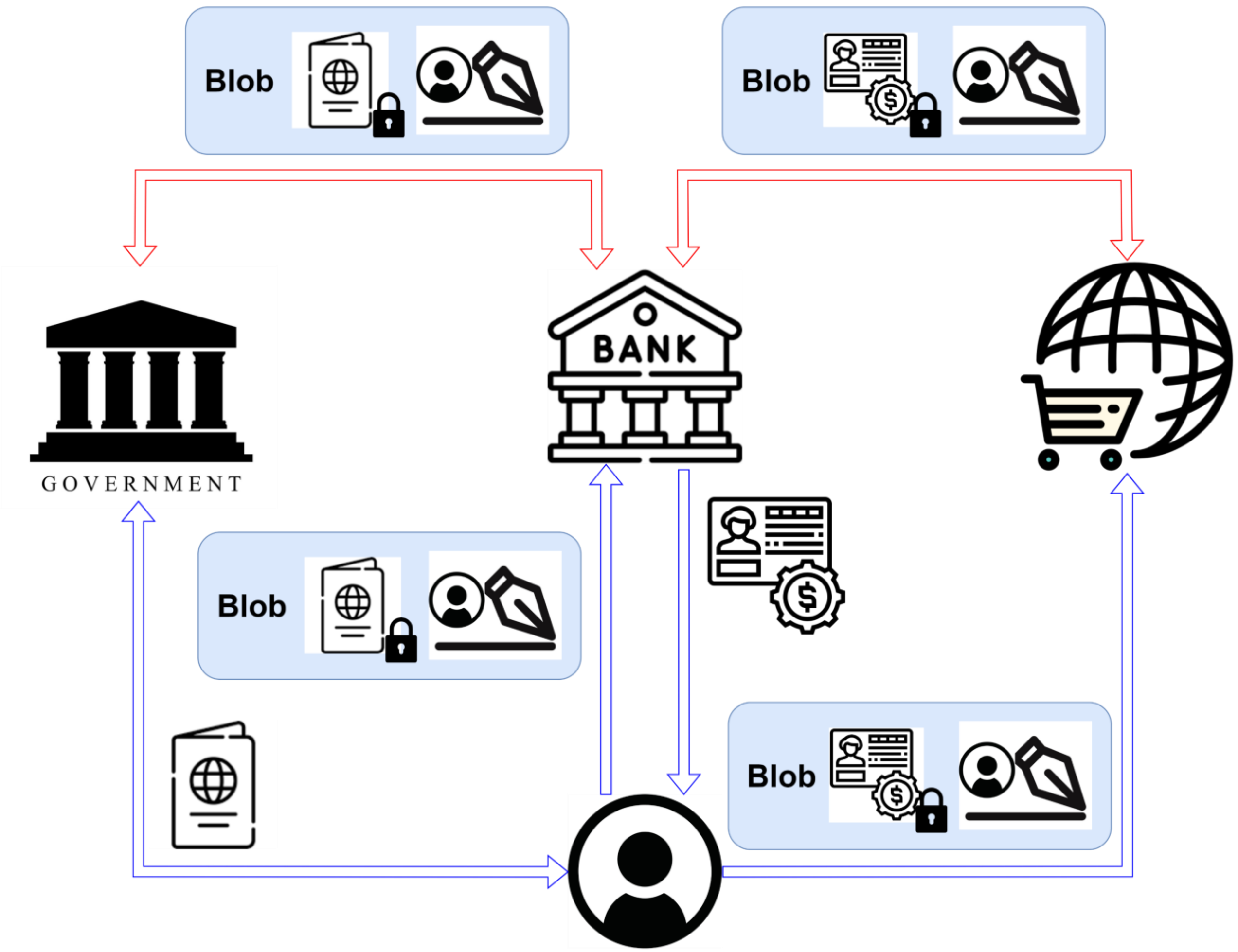


The same digital identity can be used multiple times in different contexts through the use of identity providers.

However, a user has to trust that the service provider will not disclose the user's data. In the federated model, users lose the ability to control personal data distribution.

Zero Trust security model for personal data protection

User-centric model



The idea of the user-centric model is to provide such an ability to the user. Instead of providing the user's data for authentication, the user provides a message of a special type (a blob), which contains authentication conditions (who is authenticated, to whom, when, and based on what type of data), encrypted authentication data, and the user's data.

This message cannot be used for any other authentication process, and the data context is unknown to the service. In order to authenticate the user, this message must be transmitted to the identity provider.

X-protocol

Personal data exchange protocol: X

V. Belsky[1][0000−0002−4546−5464], I. Gerasimov[1][0000−0003−1921−823X],
K. Tsaregorodtsev[1][0000−0002−9281−417X] and I. Chizhov[1][0000−0001−9126−6442]

[1]Cryptography Laboratory, SPC «Kryptonite», Moscow, Russia cryptolab@kryptonite.ru

Abstract. Personal data exchange and disclosure prevention are widespread problems in our digital world. There are a couple of information technologies embedded in the commercial and government processes. People need to exchange their personal information while using these technologies. And therefore, It is essential to make this exchange is secure. Despite many legal regulations, there are many cases of personal data breaches that lead to undesirable consequences. Reasons for personal data leakage may be adversary attack or data administration error. At the same time, creating complex service interaction and multilayer information security may lead to many inconveniences for the user. Personal data exchange protocol has the following tasks: participant's data transfer, ensuring information security, providing participants with trust in each other and ensuring service availability. In this paper, we represent a personal data exchange protocol called X[1]. The main idea is to provide personal data encryption on the user side and thus to prevent personal data disclosure and publication. This approach allows us to transfer personal data from user to service only in the form of an encrypted data packet — blob. Each blob can be validated and certified by a personal data inspector who had approved user's information. It can be any government department or a commercial organization, for example, passport issuing authority, banks, etc. It implies that we can provide several key features for personal data exchange. A requesting service cannot publish the user personal data. It still can perform a validation protocol with an inspector to validate user data. We do not depend on service data administration infrastructure and do not complicate the inspector's processes by adding additional information about the personal data request. The personal data package has a link between the personal data owner and a service request. Each blob is generated for a single request and has a time limit for a provided encrypted personal data. After this limit, the service can not use a received package. The user cannot provide invalid personal data or use the personal data of another person. We don't restrict specified cryptographic algorithms usage The X protocol can be implemented with any encryption, digital signature, key generation algorithms which are secure in our adversary model. For protocol description, Russian standardized cryptographic protocols are used. The paper also contains several useful examples of how the X protocol can be implemented in real information systems.

Keywords: X· personal data · VKO GOST · symmetric cryptography.

1 Introduction

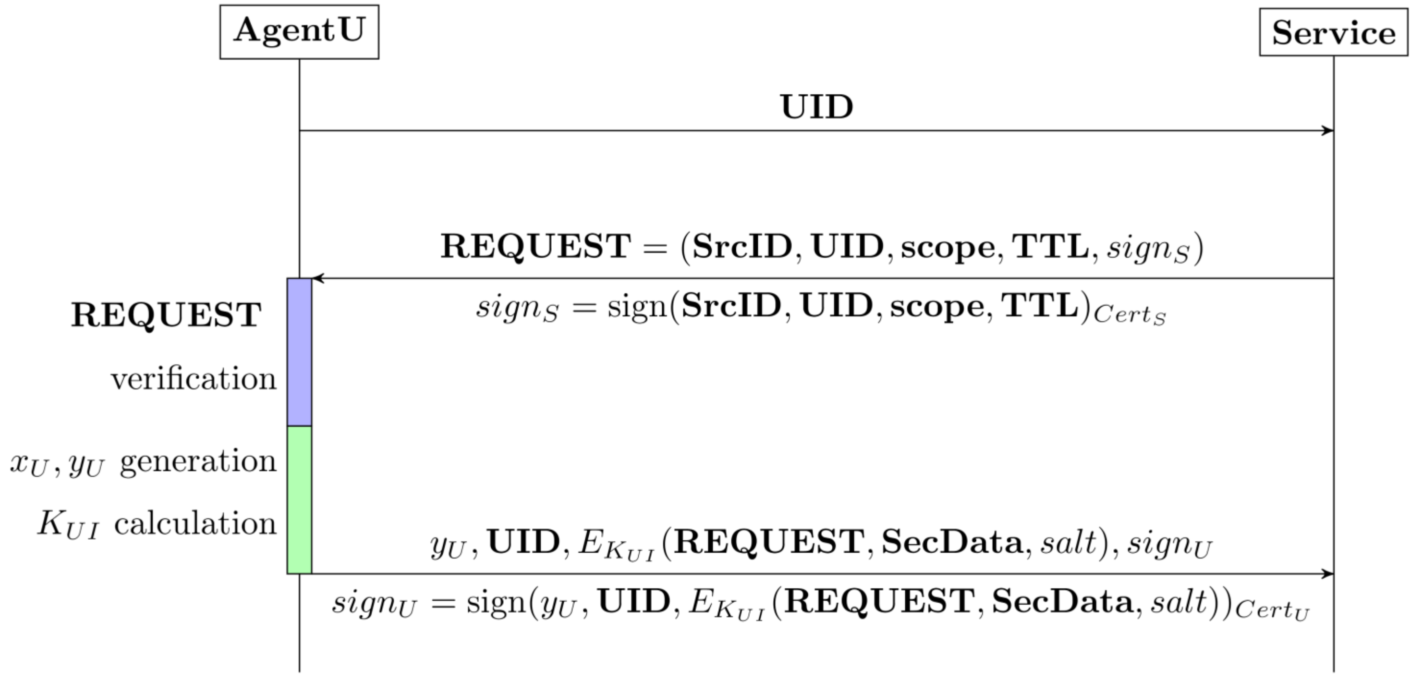
Recently information technology is being actively implemented in public service delivery processes. Digital government, public service portals and similar information systems are becoming more and more familiar in modern society. A lot of useful tasks such as taking a loan, applying for a passport, sign a contract can be done without leaving home using a computer or a mobile phone. Generally, it is necessary to provide personal data for performing these operations. Despite the rather strict regulation of personal data processing in many countries, there always occur data leakage cases as the result of administration errors or hacker attacks. It leads to undesirable consequences for people, for example, money, property and reputation loss.

Some countries increase the restriction of information security policies, but it leads to the creation of significant inconveniences for such users as services, which in turn lowers their attractiveness to citizens. Thus the creation of useful and secure personal data processing system — one of the main problems for public information services developers.

In the most general case, the following tasks are set for personal data processing systems:

[1] The paper was published in Russian in International Journal of Open Information Technologies ISSN: 2307-8162 vol. 8, no. 6, 2020

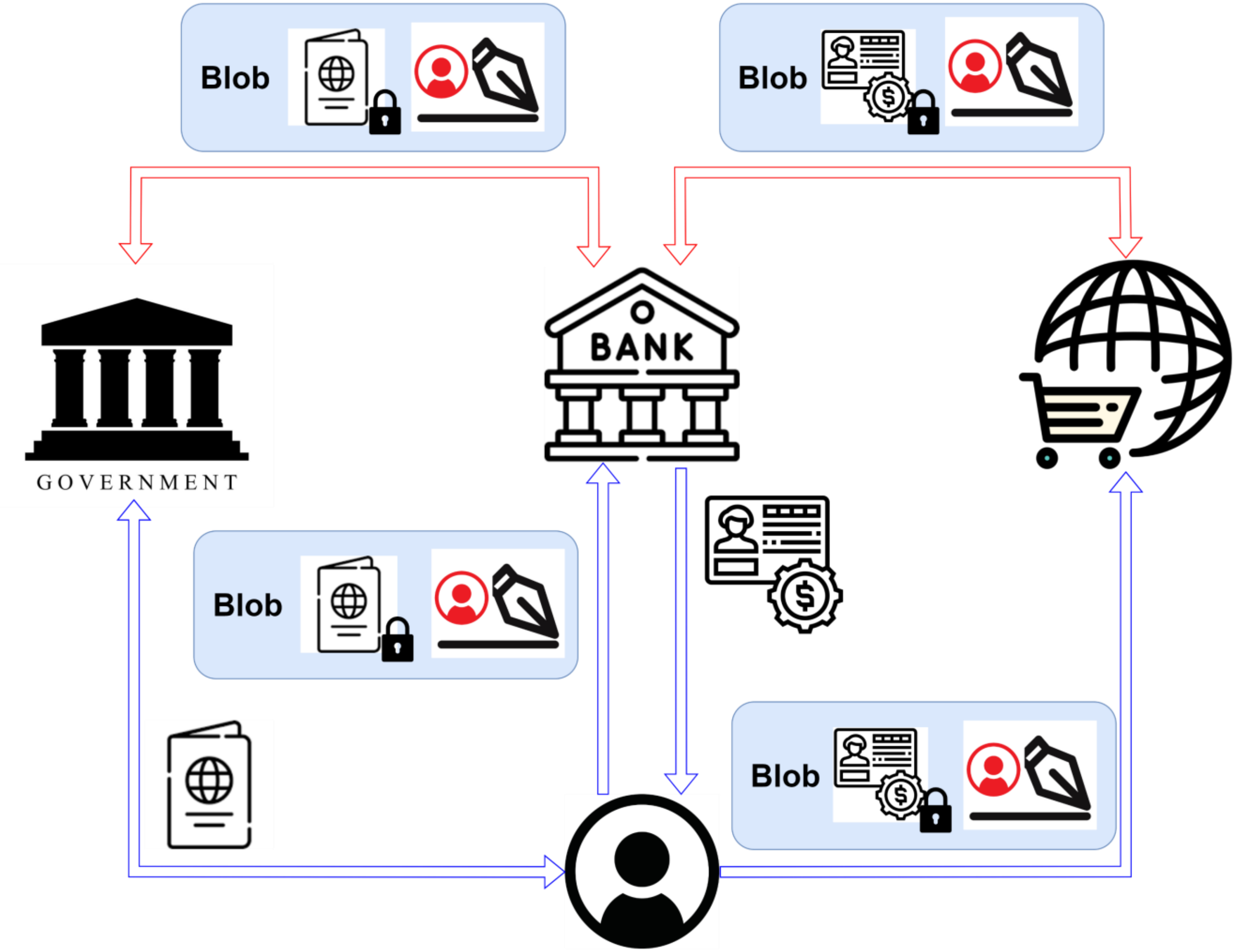
The X-protocol allows using cryptographic methods to ensure the transfer of personal data from the user to the service in the form of an encrypted block. The service can check the validity of the transmitted data with the help of a personal data inspector who created (registered) this data for the user.


AgentU
Service
UID
$\textbf{REQUEST} = (\textbf{SrcID}, \textbf{UID}, \textbf{scope}, \textbf{TTL}, sign_S)$
$sign_S = \text{sign}(\textbf{SrcID}, \textbf{UID}, \textbf{scope}, \textbf{TTL})_{Cert_S}$
REQUEST
verification
x_U, y_U generation
K_{UI} calculation
$y_U, \textbf{UID}, E_{K_{UI}}(\textbf{REQUEST}, \textbf{SecData}, salt), sign_U$
$sign_U = \text{sign}(y_U, \textbf{UID}, E_{K_{UI}}(\textbf{REQUEST}, \textbf{SecData}, salt))_{Cert_U}$

Belsky V. et al. Personal data exchange protocol: X //Cryptology ePrint Archive. – 2020.

Message linkability for extending attack surface: problem

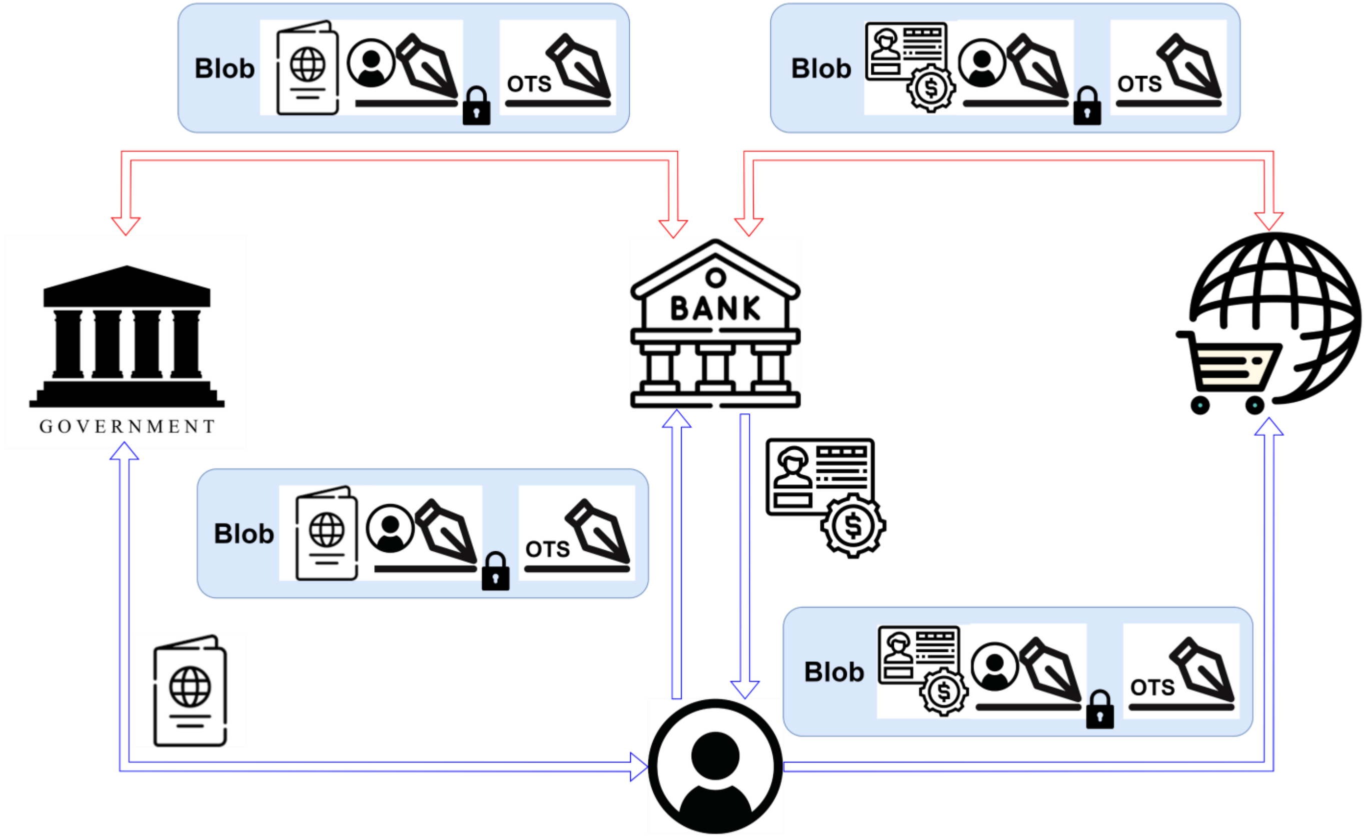
User-centric model



An authentication message requires a user signature in order to prevent message corruption and repudiation. However, if an adversary has two or more messages for different purposes, there is a possibility to link them using their digital signatures.

Message linkability for extending attack surface: solution

User-centric model



The idea is to divide two properties by transferring a property of nonrepudiation in the encrypted part and preserving message integrity in the public part.

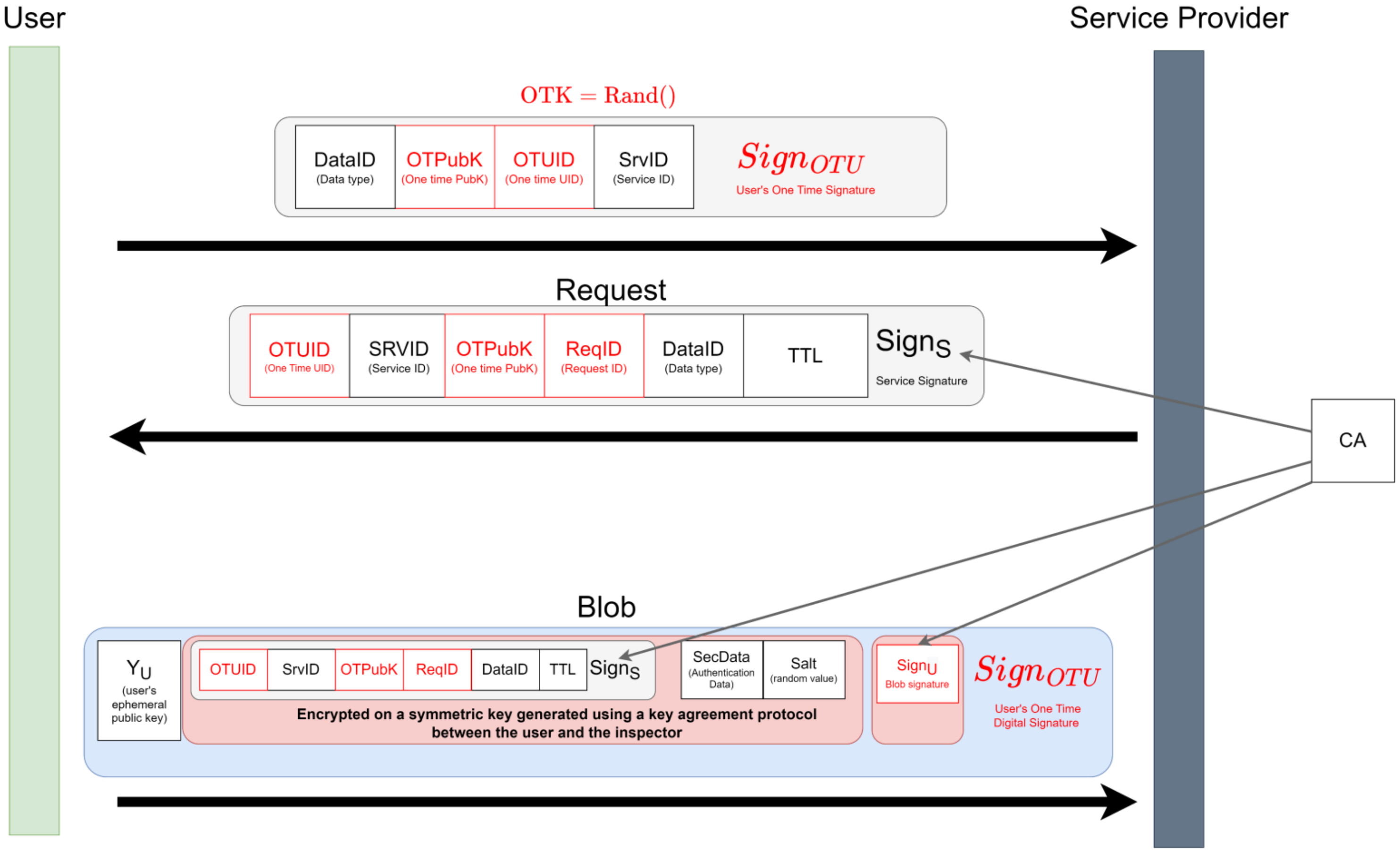
Implementing the solution using X-protocol

The user, in order to get a service, provides his identification data, which contains:

Data ID – the data type that will be used for authentication. This field is necessary in order to obtain the inspector's identity;

One-time public key and user ID – the user identities for this session. A one-time public key is generated from a one-time private key, which is randomly generated;

Service ID – describes to whom the user is going to communicate.



Implementing the solution using X-protocol

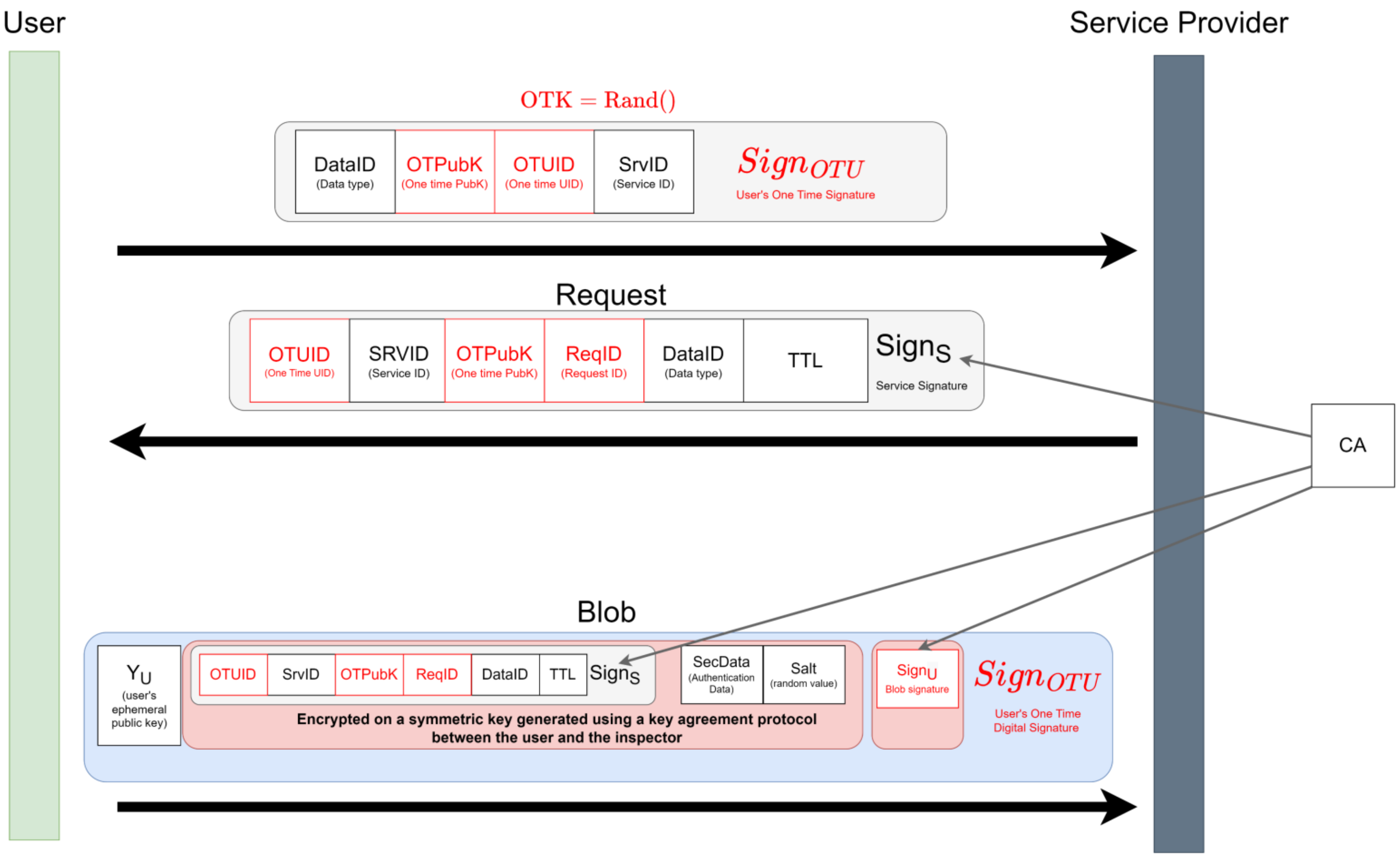
A signed service request contains:

Service ID;

User identity for this session;

Request ID – unique value for each service's request;

TTL – for how long the user's authentication is legitimate.



Implementing the solution using X-protocol

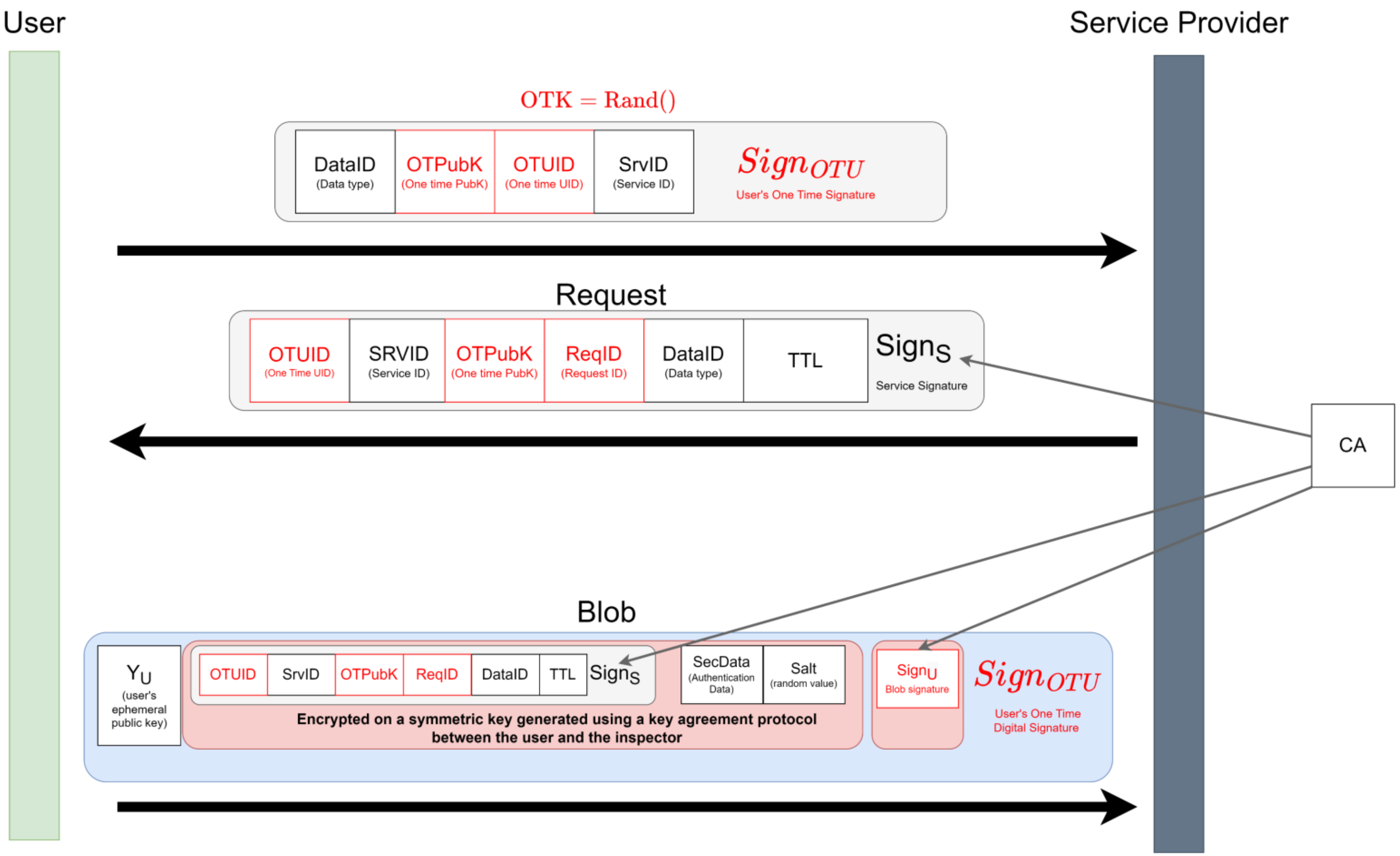
Kryptonite

In order to provide an authentication message called Blob, user needs to form an encryption key with the inspector. He performs a key agreement protocol using an ephemeral key pair and the inspector's public key. In the blob, an ephemeral public key is specified.

The blob contains:

Reply – an encrypted part that contains the service request, authentication data, and signature generated by the user using a key pair that is registered with the certificate authority;

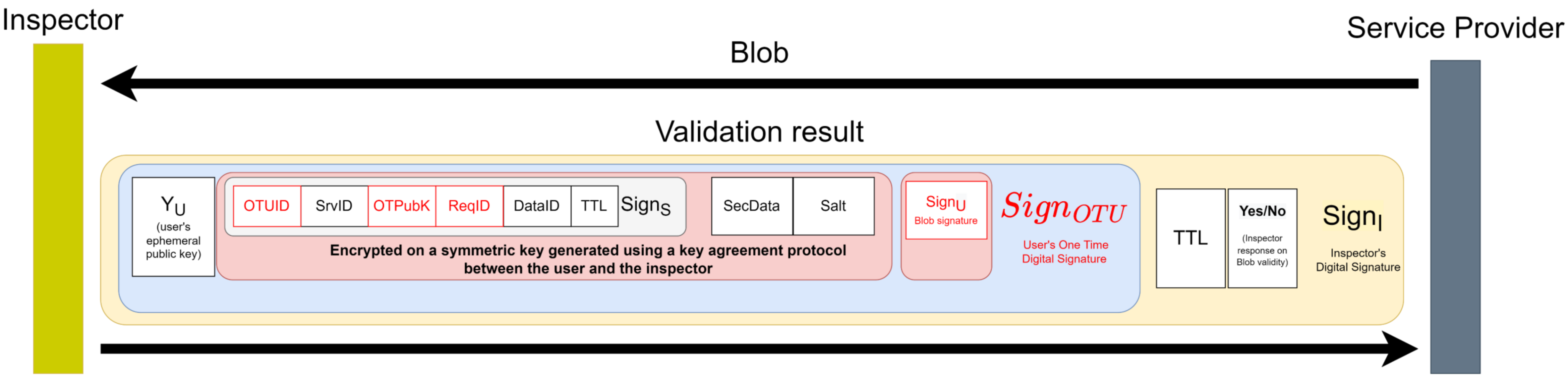
One-time Signature – a digital signature generated by the user using a one-time key pair.



Implementing the solution using X-protocol

The service does not have the ability to decrypt the reply and get the user's authentication data and signature based on the user's registered key pair.

The only thing he can do is to ask the inspector to validate the blob and receive a validation result which contains the blob, TTL and the result signed by the inspector.



Inspector
Service Provider
Blob
Validation result
Y_U (user's ephemeral public key)
OTUID
SrvID
OTPubK
ReqID
DataID
TTL
$Sign_S$
SecData
Salt
Encrypted on a symmetric key generated using a key agreement protocol between the user and the inspector
$Sign_U$
Blob signature
$Sign_{OTU}$
User's One Time Digital Signature
TTL
Yes/No
(Inspector response on Blob validity)
$Sign_I$
Inspector's Digital Signature

How does the implementation affects protocol properties?

The service cannot determine the user from whom the authentication data was received. This property is provided by replacing the original key pair of the user with a one-time key pair. At the same time, the connection of the protocol data with the user is indicated in encrypted form and is available only to the Inspector.

How does the implementation affects protocol properties?

The service cannot determine the user from whom the authentication data was received. This property is provided by replacing the original key pair of the user with a one-time key pair. At the same time, the connection of the protocol data with the user is indicated in encrypted form and is available only to the Inspector.

The user cannot refuse the fact of confirmation of personal data if he really provided them. The correctness of the blob is confirmed by the inspector only if a number of conditions are met, in particular, successful verification of the correctness of the user's encrypted signature under the generated blob. If the user did not provide data, then the encrypted signature under the blob cannot be generated on his behalf by an adversary. The presence of the user's digital signature in the encrypted part of the blob, which contains the requested personal data, makes it impossible for the user to refuse the fact of blob formation. That is, the user knew to whom, for how long and what data he provided.

Further research in terms of zero trust model

X-protocol modification provides user anonymity against the service provider. However, it enhances identity provider involvement as a nonrepudiation property can be validated only by the identity provider.

Further research in terms of zero trust model

X-protocol modification provides user anonymity against the service provider. However, it enhances identity provider involvement as a nonrepudiation property can be validated only by the identity provider.

The possible solution is to create a one-time keypair link to the user's registered public key such that:

- The problem of getting a registered public key by one-time key pair is hard (that is, the service does not have the ability to find out user identity);
- The problem of getting a one-time key pair by registered public key without knowledge of the private key is hard (that is, only users can create one-time key pairs);
- The problem of finding several one-time keys that belong to one registered public key is hard (that is, the service cannot link several blobs to one user).

Conclusion

The provided User-centric model modification solves the problem of message linking and thus mitigates extended attack surface by reducing known points of user activity.

We believe this approach to be a perspective solution for Zero Trust security model and provide a realization based on X-protocol.

Conclusion

The provided User-centric model modification solves the problem of message linking and thus mitigates extended attack surface by reducing known points of user activity.

We believe this approach to be a perspective solution for Zero Trust security model and provide a realization based on X-protocol.

Thank you for listening!

If you have any questions, suggestions or comments, please contact me.

Ilia Gerasimov
i.gerasimov@kryptonite.ru