SoK: Secure Messaging

Nik Unger , et al. Presented by, Pritha.D.N

May 23, 2016

Abstract

This paper takes a systematization of knowledge approach on the domain of Secure Messaging. They propose an evaluation framework for their security, usability, and ease-of-adoption properties of the existing academic and end-user tools. Three key challenges identified are trust establishment, conversation security, and transport privacy.

1 Evaluation

The following is a brief overview of the tools and techniques that were evaluated and the takeaway derived from this method of systematization. This section covers the basic definitions and methodologies of tools currently available.

1.1 Trust Establishment

Trust establishment is the process of users verifying that they are actually communicating with the parties they intend.

1.1.1 Features under consideration

Security and Privacy Features under consideration are:

Network MitM Prevention: Prevents Man-in-the-Middle (MitM) attacks by local and global network adversaries.

Operator MitM Prevention: Prevents MitM attacks executed by infrastructure operators.

Operator MitM Detection: Allows the detection of MitM attacks performed by operators after they have occurred.

Operator Accountability: It is possible to verify that operators behaved correctly during trust establishment.

Key Revocation Possible: Users can revoke and renew keys (e.g., to recover from key loss or compromise).

Privacy Preserving: The approach leaks no conversation metadata to other participants or even service operators. Usability Properties under consideration are:

Automatic Key Initialization: No additional user effort is required to create a long-term key pair.

Low Key Maintenance: Key maintenance encompasses recurring effort users have to invest into maintaining keys. Some systems require that users sign other keys or renew expired keys. Usable systems require no key maintenance tasks.

Easy Key Discovery: When new contacts are added, no additional effort is needed to retrieve key material.

Easy Key Recovery: When users lose long-term key material, it is easy to revoke old keys and initialize new keys (e.g., simply reinstalling the app or regenerating keys is sufficient).

In-band: No out-of-band channels(different from the current channel of communication) are needed that require users to invest additional effort to establish.

No Shared Secrets: Shared secrets require existing social relationships. This limits the usability of a system, as not all communication partners are able to devise shared secrets.

Alert-less Key Renewal: If other participants renew their long-term keys, a user can proceed without errors or warnings.

Immediate Enrollment: When keys are (re-)initialized, other participants are able to verify and use them immediately.

Inattentive User Resistant: Users do not need to carefully inspect information (e.g., key fingerprints) to achieve security.

Adoption Properties under consideration are:

Multiple Key Support: Users should not have to invest additional effort if they or their conversation partners use multiple public keys, making the use of multiple devices with separate keys transparent. While it is always possible to share one key on all devices and synchronize the key between them, this can lead to usability problems.

No Service Provider Required: Trust establishment does not require additional infrastructure (e.g., key servers).

No Auditing Required: The approach does not require auditors to verify correct behavior of infrastructure operators.

No Name Squatting: Users can choose their names and can be prevented from reserving a large number of popular names.

Asynchronous: Trust establishment can occur asynchronously without all conversation participants online.

Scalable: Trust establishment is efficient, with resource requirements growing logarithmically (or smaller) with the total number of participants in the system.

1.1.2 Techniques available

A brief overview of the techniques evaluated:

Opportunistic Encryption (Baseline) Here an encrypted session is established without any key verification.

Key Fingerprint Verification Manual verification requires users to compare some representation of a cryptographic hash of their partners' public keys out-of-band (e.g., in person or via a separate secure channel)

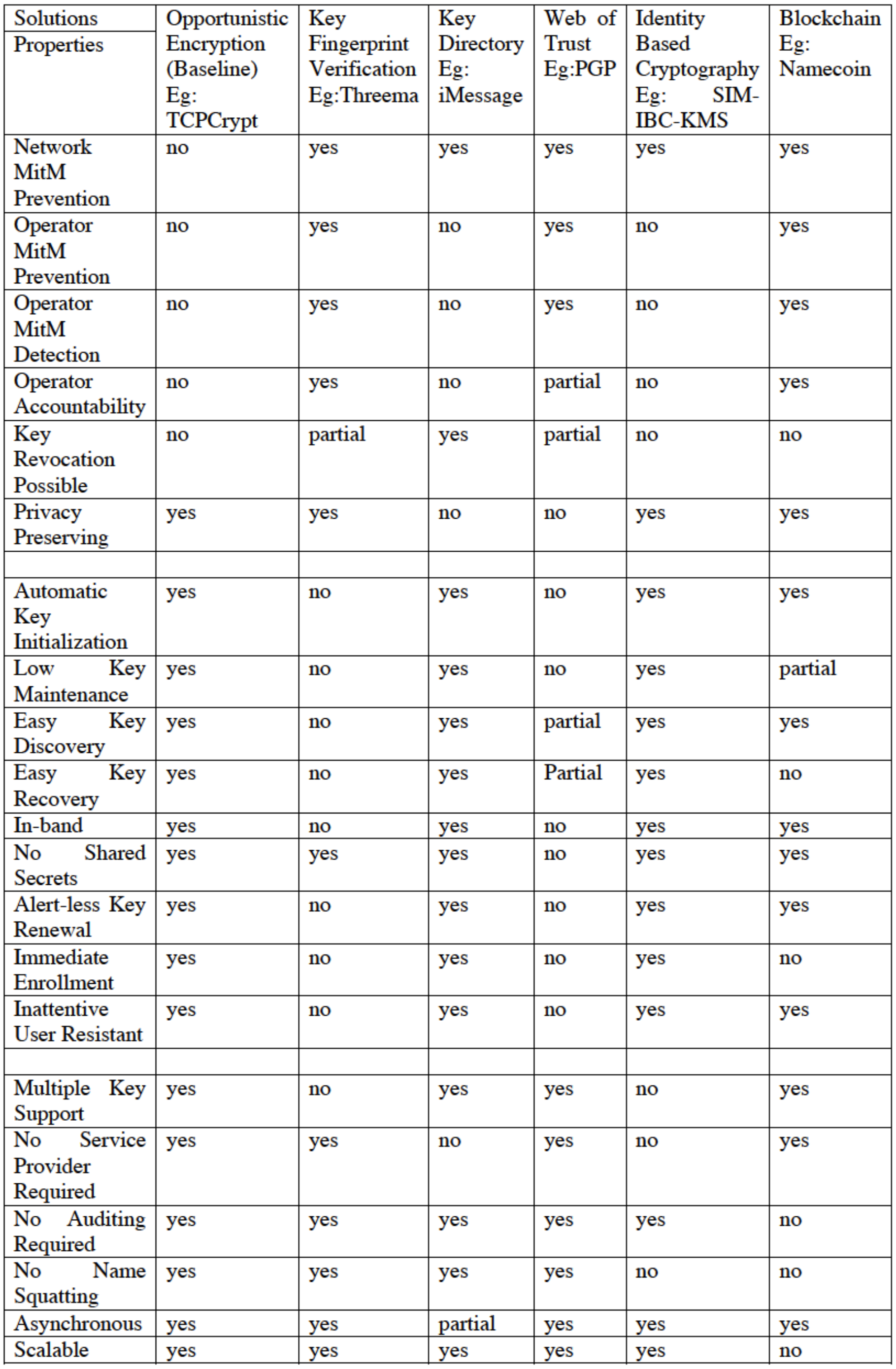
Key Directory One instance of this scheme is in Authority-based trust schemes, public keys must be vouched for by one or more trusted authorities (CA).

A major issue with trusted authorities is that they can vouch for fraudulent keys in an attack. The Certificate Transparency protocol requires that all issued web certificates are included in a public log. This append-only log is implemented using a signed Merkle tree with continual proofs of consistency. Certificates are only trusted if they include cryptographic proof that they are present in the log. This ensures that any keys the authority vouches for will be visible in the log and evidence will exist that the authority singed keys used in an attack.

Web of Trust Users verify each other's keys using manual verification and, once they are satisfied that a public key is truly owned by its claimed owner, they sign the key to certify this. These certification signatures might be uploaded to key servers.

Identity Based Cryptography Here, plaintext identifiers (such as email or IP addresses) are mapped to public keys. A trusted third party, the Private Key Generator (PKG), publishes a PKG public key that is distributed to all users of the system.

Blockchain Namecoin, a bitcoin derivative scheme, allows users to claim identifiers, add arbitrary data (e.g., public keys) as records for those identifiers, and even sell control of their identifiers to others.



Solutions Properties	Opportunistic Encryption (Baseline) Eg: TCPCrypt	Key Fingerprint Verification Eg:Threema	Key Directory Eg: iMessage	Web of Trust Eg:PGP	Identity Based Cryptography Eg: SIM-IBC-KMS	Blockchain Eg: Namecoin
Network MitM Prevention	no	yes	yes	yes	yes	yes
Operator MitM Prevention	no	yes	no	yes	no	yes
Operator MitM Detection	no	yes	no	yes	no	yes
Operator Accountability	no	yes	no	partial	no	yes
Key Revocation Possible	no	partial	yes	partial	no	no
Privacy Preserving	yes	yes	no	no	yes	yes
Automatic Key Initialization	yes	no	yes	no	yes	yes
Low Key Maintenance	yes	no	yes	no	yes	partial
Easy Key Discovery	yes	no	yes	partial	yes	yes
Easy Key Recovery	yes	no	yes	Partial	yes	no
In-band	yes	no	yes	no	yes	yes
No Shared Secrets	yes	yes	yes	no	yes	yes
Alert-less Key Renewal	yes	no	yes	no	yes	yes
Immediate Enrollment	yes	no	yes	no	yes	no
Inattentive User Resistant	yes	no	yes	no	yes	yes
Multiple Key Support	yes	no	yes	yes	no	yes
No Service Provider Required	yes	yes	no	yes	no	yes
No Auditing Required	yes	yes	yes	yes	yes	no
No Name Squatting	yes	yes	yes	yes	no	no
Asynchronous	yes	yes	partial	yes	yes	yes
Scalable	yes	yes	yes	yes	yes	no

Figure 1: The security/privacy and usability properties in the schemes available

1.2 Conversation Security

After trust establishment has been achieved, a conversation security protocol protects the security and privacy of the exchanged messages. This encompasses how messages are encrypted, what data is attached to them, and what cryptographic protocols (e.g., ephemeral key exchanges) are performed.

1.2.1 Features under consideration

Security and Privacy Features under consideration are:

Confidentiality: Only the intended recipients are able to read a message. Specifically, the message must not be readable by a server operator that is not a conversation participant.

Integrity: No honest party will accept a message that has been modified in transit.

Authentication: Each participant in the conversation receives proof of possession of a known long-term secret from all other participants that they believe to be participating in the conversation. In addition, each participant is able to verify that a message was sent from the claimed source.

Participant Consistency: At any point when a message is accepted by an honest party, all honest parties are guaranteed to have the same view of the participant list.

Destination Validation: When a message is accepted by an honest party, they can verify that they were included in the set of intended recipients for the message.

Forward Secrecy: Compromising all key material does not enable decryption of previously encrypted data.

Backward Secrecy: Compromising all key material does not enable decryption of succeeding encrypted data.

Anonymity Preserving: Any anonymity features provided by the underlying transport privacy architecture are not undermined (e.g., if the transport privacy system provides anonymity, the conversation security level does not deanonymize users by linking key identifiers).

Speaker Consistency: All participants agree on the sequence of messages sent by each participant. A protocol might perform consistency checks on blocks of messages during the protocol, or after every message is sent.

Causality Preserving: Implementations can avoid displaying a message before messages that causally precede it.

Global Transcript: All participants see all messages in the same order.

Deniability-related features:

Message Unlinkability: If a judge is convinced that a participant authored one message in the conversation, this does not provide evidence that they authored other messages.

Message Repudiation: Given a conversation transcript and all cryptographic keys, there is no evidence that a given message was authored by any particular user.

Participation Repudiation: Given a conversation transcript and all cryptographic key material for all but one accused participant, there is no evidence that the honest participant was in a conversation with any of the other participants.

1.2.2 Techniques available

A brief overview of the techniques evaluated:

TLS + Trusted Server Using a central server to relay messages and securing connections from clients to the central server using a transport-layer protocol like TLS.

Static Asymmetric Cryptography Uses participants' static long-term asymmetric keypairs for signing and encrypting.

Authenticated Diffie-Hellman The participants generate an ephemeral session key and authenticate the exchange using their long-term keys. The resulting session key is used to derive symmetric encryption and MAC keys.

Key ratchet/evolution A session key ratchet is a simple approach is to use key derivation functions (KDFs) to compute future message keys from past keys.

Diffie-Hellman Ratchet is a different ratcheting approach, introduced by Off The Record messaging, is to attach new DH contributions to messages. With each sent message, the sender advertises a new DH value. Message keys are then computed from the latest acknowledged DH values.

Double-Ratchet (Axolotl): To improve the forward secrecy of a DH ratchet, both ratchet approaches can be combined: session keys produced by DH ratchets are used to seed per-speaker KDF ratchets. Messages are then encrypted using keys produced by the KDF ratchets, frequently refreshed by the DH ratchet on message responses

1.3 Transport Privacy

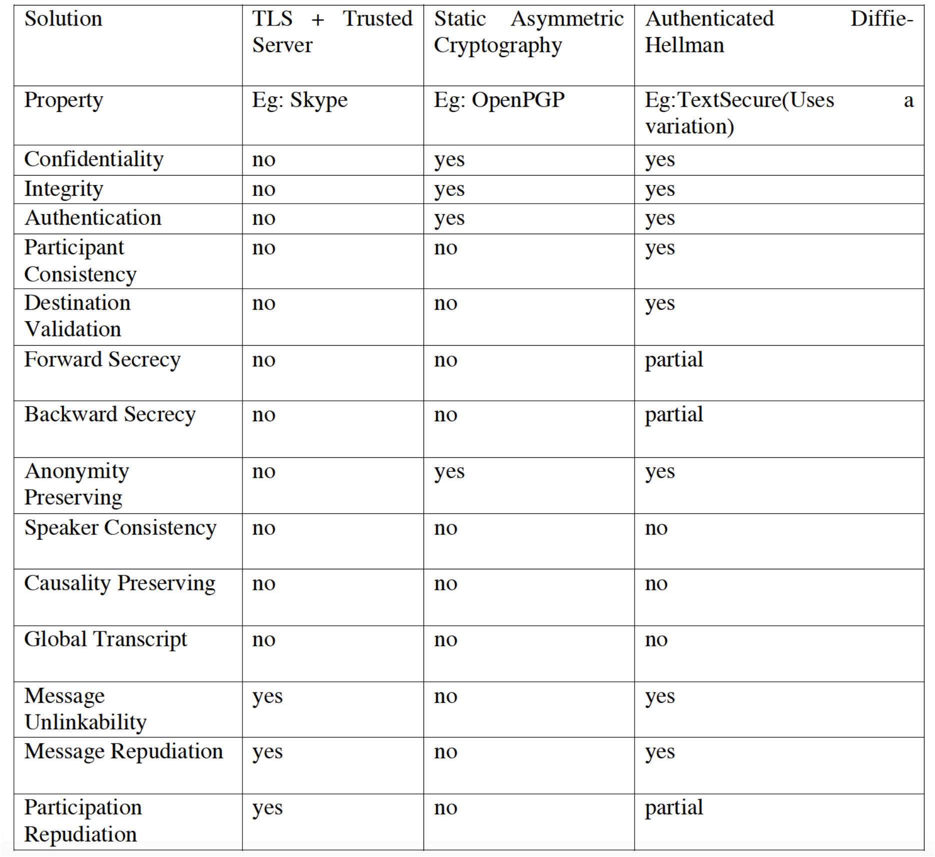
The transport privacy layer defines how messages are exchanged, with the goal of hiding message metadata such as the sender, receiver, and conversation to which the message belongs.

1.3.1 Features under consideration

Privacy Features under consideration are:

Sender Anonymity: When a chat message is received, no non-global entities except for the sender can determine which entity produced the message.

Recipient Anonymity: No non-global entities except the receiver of a chat message know which entity received it.



Solution	TLS + Trusted Server	Static Asymmetric Cryptography	Authenticated Diffie-Hellman
Property	Eg: Skype	Eg: OpenPGP	Eg:TextSecure(Uses a variation)
Confidentiality	no	yes	yes
Integrity	no	yes	yes
Authentication	no	yes	yes
Participant Consistency	no	no	yes
Destination Validation	no	no	yes
Forward Secrecy	no	no	partial
Backward Secrecy	no	no	partial
Anonymity Preserving	no	yes	yes
Speaker Consistency	no	no	no
Causality Preserving	no	no	no
Global Transcript	no	no	no
Message Unlinkability	yes	no	yes
Message Repudiation	yes	no	yes
Participation Repudiation	yes	no	partial

Figure 2: The security/privacy and usability properties in the conversation security schemes available

Participation Anonymity: No non-global entities except the conversation participants can discover which set of network nodes are engaged in a conversation.

Unlinkability: No non-global entities except the conversation participants can discover that two protocol messages belong to the same conversation.

Global Adversary Resistant: Global adversaries cannot break the anonymity of the protocol.

Usability Properties under consideration are:

Contact Discovery: The system provides a mechanism for discovering contact information. No

Message Delays: No long message delays are incurred.

No Message Drops: Dropped messages are retransmitted.

Easy Initialization: The user does not need to perform any significant tasks before starting to communicate.

No Fees Required: The scheme does not require monetary fees to be used.

Adoption Properties under consideration are:

Topology Independent: No network topology is imposed on the conversation security or trust establishment schemes.

No Additional Service: The architecture does not depend on availability of any infrastructure beyond the chat participants.

Spam/Flood Resistant: The availability of the system is resistant to denial-of-service attacks and bulk messaging.

Low Storage Consumption: The system does not require a large amount of storage capacity for any entity.

Low Bandwidth: The system does not require a large amount of bandwidth usage for any entity.

Low Computation: The system does not require a large amount of processing power for any entity.

Asynchronous: Messages sent to recipients who are offline will be delivered when the recipient reconnects, even if the sender has since disconnected.

Scalable: The amount of resources required to maintain system availability scales linearly with the number of users.
A brief overview of the techniques evaluated:

Onion Routing Instead of relying on centralized servers for message storage and forwarding, peer-to-peer based schemes try to establish a direct message exchange between the participants. Since end users frequently change their IP addresses, these systems often use Distributed Hash Tables (DHTs) to map usernames to IP addresses without a central authority.

DC-nets (Dinning Cryptographer) DC-nets are group protocols that execute in rounds. At the start of each round, each participant either submits a secret message or no message. At the end of the round, all participants receive the xor of all secret messages submitted, without knowing which message was submitted by which participant.

Message Broadcast Distributing messages to everyone: This approach provides recipient anonymity, participation anonymity, and unlinkability against all network attackers.

Private Information Retrieval Allows a user to query a database on a server without enabling the server to determine what information was retrieved. It can be used to store databases of message inboxes, as well as databases of contact information. Recipient anonymity is provided because, while the server knows the network node that is connecting to it, the server cannot associate incoming connections with protocol messages that they retrieve. For the same reason, the protocols offer participation anonymity and unlinkability.

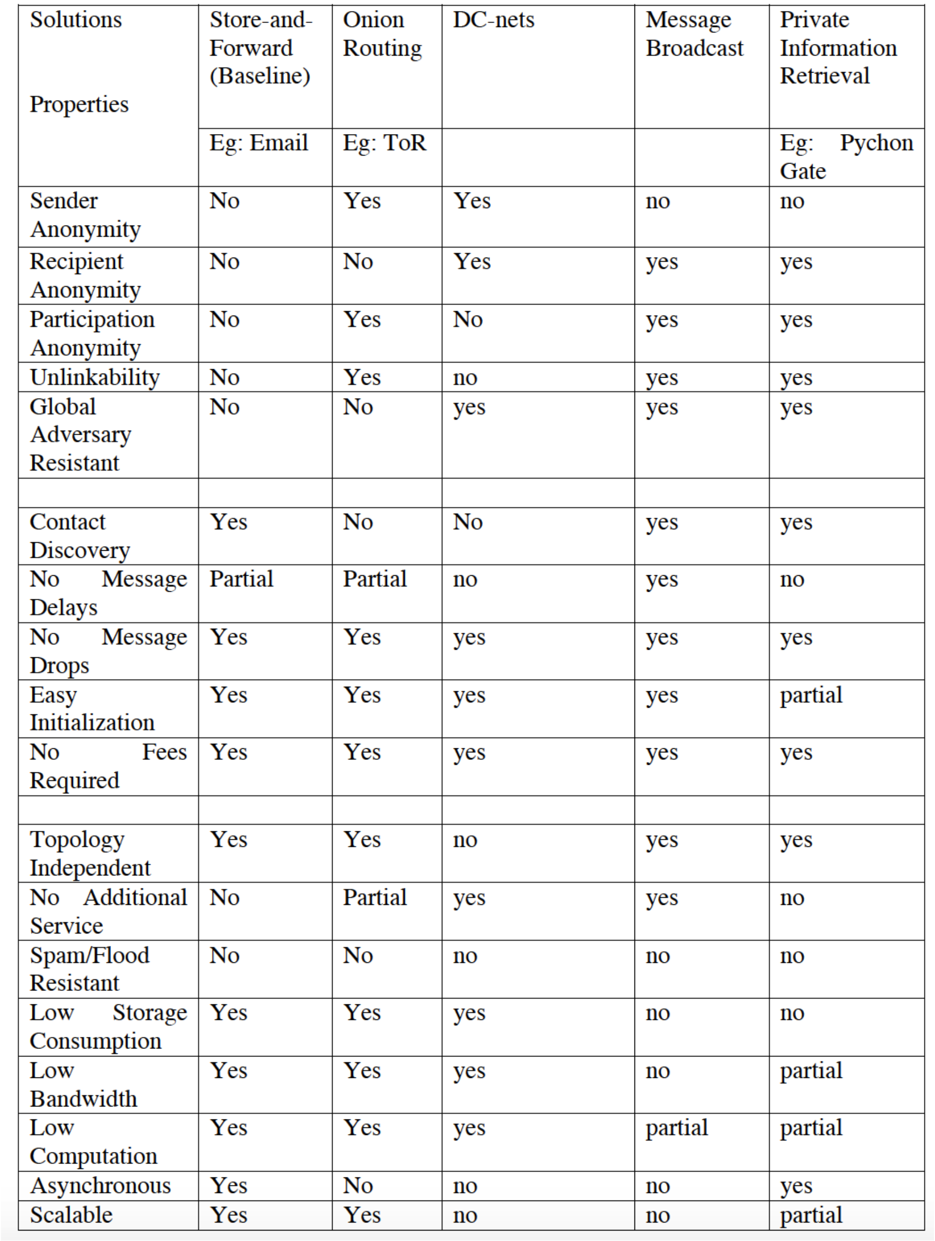
2 Conclusion

Existing knowledge on secure messaging suggests three major problems must be resolved: trust establishment, conversation security and transport privacy.

Secure approaches in trust establishment perform poorly in usability and adoption, while more usable approaches lack strong security guarantees. The most promising approach for trust establishment is a combination of central key directories, transparency logs to ensure global consistency of the key directory's entries, and a variety of options for security-conscious users to verify keys out of band to put pressure on the key directory to remain honest.

The observations on the conversation security layer suggest that asynchronous environments and limited multi-device support are not fully resolved. For two-party conversation security, per-message ratcheting with resilience for out-of-order messages combined with deniable key exchange protocols, as implemented in Axolotl, can be employed today at the cost of additional implementation complexity with no significant impact on user experience.

Finally, transport privacy remains a challenging problem. No suggested approaches managed to provide strong transport privacy properties against global adversaries while also remaining practical.



Solutions Properties	Store-and-Forward (Baseline)	Onion Routing	DC-nets	Message Broadcast	Private Information Retrieval
	Eg: Email	Eg: ToR			Eg: Pychon Gate
Sender Anonymity	No	Yes	Yes	no	no
Recipient Anonymity	No	No	Yes	yes	yes
Participation Anonymity	No	Yes	No	yes	yes
Unlinkability	No	Yes	no	yes	yes
Global Adversary Resistant	No	No	yes	yes	yes
Contact Discovery	Yes	No	No	yes	yes
No Message Delays	Partial	Partial	no	yes	no
No Message Drops	Yes	Yes	yes	yes	yes
Easy Initialization	Yes	Yes	yes	yes	partial
No Fees Required	Yes	Yes	yes	yes	yes
Topology Independent	Yes	Yes	no	yes	yes
No Additional Service	No	Partial	yes	yes	no
Spam/Flood Resistant	No	No	no	no	no
Low Storage Consumption	Yes	Yes	yes	no	no
Low Bandwidth	Yes	Yes	yes	no	partial
Low Computation	Yes	Yes	yes	partial	partial
Asynchronous	Yes	No	no	no	yes
Scalable	Yes	Yes	no	no	partial

Figure 3: The security/privacy and usability properties in the Transpost security schemes available