End-to-end Security with Translation[1]

Kevin E. Foltz
Institute for Defense Analyses
4850 Mark Center Dr.
Alexandria, Virginia 22311

[1] The publication of this paper does not indicate endorsement by the Department of Defense or IDA, nor should the contents be construed as reflecting the official position of these organizations.

ABSTRACT

End-to-end security involves two entities communicating at a distance over an insecure communication channel while maintaining many of the security properties of private, in-person communication. End-to-end security transforms an insecure channel into a secure one, such that the entities know who the other is, their communication is not interpretable by anyone observing the channel, and an active attacker that changes the content communicated will be detected. For digital communication it is desirable to have end-to-end security properties. However, not all entities that wish to communicate share a common language. This may be humans from different countries, web services that encode data in different formats, or applications that communicate with different protocols. The ideas they wish to share are common to both entities, but the representation of them is different. In such a case, end-to-end security limits the ability to use common translation methods that would allow communication. This paper discusses different translation approaches in the context of end-to-end security.

Keywords: Security, End-to-end Security, System Design, Translation, Authentication, Confidentiality, Integrity

1. INTRODUCTION

Adversaries continue to penetrate our network defenses and in many cases already exist within our network perimeter. They have infiltrated the online environment, jeopardizing the confidentiality, integrity, and availability of enterprise information and systems. The fortress model – hard on the outside, soft on the inside – assumes that the boundary can prevent all types of penetration, but this assumption has been proven wrong by a multitude of reported network-related incidents. A wiser assumption for data and information security practitioners is that the adversary exists within the network. The solution is to use end-to-end security, which builds a secure connection between two entities when they wish to communicate.

This paper describes a way to provide translation services within such an end-to-end security framework. Translation services present a unique challenge and a tempting target for embedded malicious entities because translation takes place where data is changed but the normal end-to-end integrity verification methods are not feasible. A malicious entity that compromises a mediation service could selectively feed malicious content to an unsuspecting entity. Detection would be difficult because most entities only understand either the input format or the output format of data and cannot validate the translation. There is no perfect translation approach, and this paper discusses various approaches and their tradeoffs. The following sections describe end-to-end security, translation challenges, and potential current and future solutions.

2. END-TO-END SECURITY

End-to-end security, in the context of this paper, has three main security principles:

- Know the Players – ensure that all entities are vetted, registered, and credentialed;
- Secure the Communication Channel – use end-to-end unbroken authentication, encryption, and integrity validation between data requester and provider;
- Reduce External Trust Requirements – use communication models that focus on the key players involved and eliminate trust of third parties.

A. Know the Players

Before any secure communication can begin it is imperative that the communicating entities know who is participating in the communication. For a standard client-server communication model, the client chooses to initiate a connection to the server. The server is responsible for validating that it is the intended server. This process can occur only if preparations are made in advance:

- Vetting of entities,
- Registration of entities,
- Credentialing of entities.

The vetting process evaluates a proposed real-world entity to be sure it meets certain minimum requirements for registration and credentialing. This includes data collection, personal interviews, and background checks for people. In some cases, the vetting is extensive, but at a minimum it seeks to gain enough information to define unambiguously who someone is. A set of attributes is collected during the vetting process related to identity of the individual.

The registration process collects the information about an entity and stores it in an authoritative location. In some cases, such information is already available in public databases, but in others a separate authoritative store is kept for the purpose of identity registration.

The credentialing process involves the selection of identity information about an individual and the creation of a digital credential. The credential includes information about an entity and information used for authentication. This may be a digital certificate that ties a cryptographic key to an identity. The primary purpose of the credential is to provide a way for entities to validate the identity of

another entity, especially when the two entities may not already know each other.

B. Secure the Communication Channel

With the ability to authenticate the identity of other entities, we simply have a set of secure points. Each entity is secure in its identity and can validate this to other such points. The next step for end-to-end security is to connect these points with secure communication channels. This prevents outsiders from observing or modifying communication between credentialed entities. These communication channels observe three primary security rules:

- End-to-end Two-way Authentication,
- End-to-end Confidentiality,
- End-to-end Integrity.

End-to-end authentication provides assurance that the ensuing communication is between the proper endpoints. This leverages the identity credentials issued to each registered entity. Authentication must be two-way to ensure each entity that the other entity is who it claims to be.

End-to-end confidentiality is provided by encrypting communications between the two authenticated endpoints. The endpoints negotiate an encryption algorithm and a set of keys that are tied to the authenticated identities of the endpoints. This ensures that only the proper endpoint can decrypt the content that is transmitted.

End-to-end integrity provides assurance that the communications are not modified between the two endpoints. Potential modifications include:

- Deleting content,
- Adding content,
- Repeating previous content,
- Modifying content.

When adding content, it may be possible for a third party to properly encrypt content, even if the third party cannot understand or decrypt the real content that is transmitted. This can be used to send arbitrary content to an endpoint. In other cases, the added content would simply serve to disrupt communications.

Integrity can be provided using a message authentication code (MAC), which allows the receiver to validate that the proper endpoint sent the message in the proper order with no modifications by third parties.

All of these security properties of the communication channel can be provided by the Transport Layer Security Protocol (TLS).

C. Reduce External Trust Requirements

With known players and secure communication, we now have a network of points that can communicate with each other. This provides security to entities within the network from entities outside the network. However, it is not always appropriate to involve an entity in the network just because it is in the network. In particular, a two-party communication need not involve third parties if they are not involved in the communication.

It is sometimes necessary to involve external entities in a communication, such as validating credentials, retrieving addresses, or other functions. These are critical functions for the communication and its security, but they do not reveal the data that is communicated between the two endpoints in the communication. The third parties observe the metadata, but not the data, of the communication.

The reduction of external trust is applied only to third parties that have access to the data of the communication. For example, a web application firewall or other gateway security appliance may scan the content of communications to look for attacks or other security threats. This is a common approach for an organization to protect those within the organization from threats embedded in normal communications. However, this introduces another entity into the communication path.

An alternative to the central scanner is a policy of using endpoint-based scanners. These reside on the endpoints and scan content that is sent and received. This does not require the endpoints to trust a third party. They continue to communicate with end-to-end security, with no reliance on third parties.

In general, an approach with fewer trusted third parties is more desirable than one with more trusted third parties. Each third party to a communication is a potential point of compromise and attack on the communication.

3. FUNDAMENTAL TRANSLATION CHALLENGES

When two endpoints wish to communicate ideas these ideas must be represented in some way. This section examines the process of such communication and the challenges of accurately conveying information from one entity to another. The challenges fall into three categories:

- Representation Granularity,
- Repetition Distortion,
- Translation Context.

A. Representation Granularity

The first step to communicate an idea is to represent it. For languages, we choose words to represent ideas. As a starting point, consider the idea of "snow." The description of a winter landscape would often involve the word "snow." This is a single English word that represents white, frozen, flakey precipitation.

However, using "snow" to describe the condition of a ski trail would not be appropriate. Skiers and ski condition reports use terms like

- Powder,
- Packed powder,
- Groomed,
- Moguls,
- Corn,

- Frozen granular

and many others. These many terms describe relevant aspects of snow of interest to skiers, such as the hardness, consistency, depth, and texture of the snow.

Taking this idea one step further, those who live most of their lives on and in snow have developed a rich vocabulary of words to describe snow. Someone who lives in the tropics would not be familiar with these words or the ideas represented by them.

With more words comes higher precision. This allows a richer set of ideas to be expressed without requiring long descriptions of the ideas. However, sometimes the simple word "snow" is appropriate. The answer to a question about what skiers ski on might simply be "snow." Any further refinement is unnecessary and misleading because all types of snow apply to this answer. Precise terms, like "champagne powder" only apply to a small subset of situations involving snow. The challenge for communicating ideas is to choose the word that most closely represents the set of situations that correspond to an idea.

This is an issue of granularity. Words are discreet, so ideas fall between, within, and around them. So, the representation of ideas as words is fundamentally imprecise due to this granularity. It is possible to refine the representation of an idea with additional description, but this simply adds more words, which themselves are imprecise. Conveying ideas often involves an acceptance of this granularity and an acceptance that something is lost when ideas are represented in language.

B. Repetition Distortion

When an idea is represented by words, it can be repeated. The words can be used to convey the idea to someone else. This is useful for spreading ideas from person to person. Rather than fully describing the ideas represented by the words, the words themselves are repeated.

The problem is that repetition is not a perfect reconstruction. A simple example involves the game of "telephone." In this game one person whispers a message into the ear of someone next to them. This person then repeats the message, to the best of their abilities, to the person next to them, and this continues, often around a circle, until the final person announces what they heard. The first person then announces the original message, and everyone laughs at how different the two are.

In the digital age, this may seem quaint, but even email, which in theory is just a digital representation, suffers from distortion as it is repeated. An email that is forwarded through multiple recipients often undergoes changes. One person may indent. Another may change fonts. Another may add special characters to the beginning of the lines of the forwarded email. Others may change text size or tab settings. If the original email contained formatted information, such as a tab-based table of values, the meaning in this table may be lost as these changes occur. In other cases, the text simply stretches out into a vertical strip of words due to all the indents.

As an extreme example, consider a single file, such as a video file. Transfering such a file should be a simple and repeatable process. This was attempted by repeatedly uploading and downloading a video to YouTube. The initial video shows someone talking. After a few iterations, the quality degrades. After 1,000 iterations the sound is unintelligible and the video consists of moving areas of color that barely correspond to the original video.

The underlying representation of an idea may change slightly at each repetition. People choose different words to repeat an idea, email programs choose different formatting when forwarding message, and video encoding and decoding are not perfect matches. These subtle changes in the representation are not intended to change the meaning. They simply choose slightly different ways to represent the idea. However, repeatedly choosing different representations introduces an additive effect on the changes to the representation, which eventually percolates up to the meaning itself. Noise in the representation starts to be explicitly represented in the next iteration, and eventually the original idea is squeezed out by the representation artifacts.

C. Translation Context

When repeating an idea, it is sometimes necessary to change the underlying representation of the idea. This is where translation is used. Translation attempts to represent the original idea, as expressed in one language or representation in another language or representation.

A large challenge in this situation is mapping contexts of the two languages or representations. With language, there is a set of shared context for all who speak the language. With another language, the context is different. There is no perfect translation of one word to another because the words are used in different contexts even if they appear to represent the same idea.

Translating skiing terms to a language spoken only in hot climates would be difficult. The words related to snow would not exist, and the ideas would not have any context or significance for the people speaking the language. The result would be that the communication would not convey information about snow as much as it would simply describe the idea that such snow exists and is part of a different culture. Often languages simply adopt the original words instead of trying to translate them, and they are incorporated into the language as-is. However, this is a slow process, and until that context is established translation is difficult.

The idea of context also applies to computer languages. Some are designed for recursion, while others are designed for arrays or pointers. The level of abstraction in the concepts may be different. The basic types of data may be different. A translation from one programing language to

another may be possible technically, but performance, readability, and maintainability may suffer.

Translation of poetry is especially difficult. Aspects of words other than their meaning, such as alliteration, assonance, word stress, and tone do not translate easily.

In addition to the lost context, any translation involves extracting ideas from one representation, which has granularity issues, and then re-representing in another language, which again introduces granularity. Also, translation involves repetition, which introduces additional distortion.

Combining all of these issues, it is clear that translation is inherently inaccurate. Some translations are better than others, but even a very good translator will introduce some inaccuracies into the original ideas, and these are based on the way representation, repetition, and translation of these ideas takes place.

4. APPROACHES FOR SECURE TRANSLATION

When two entities wish to communicate in different languages, some form of translation is needed. If these two entities desire end-to-end security, this requires careful planning of how to implement such translation.

Four methods are discussed here:

- Go-between,
- Translation Service,
- Translation App,
- Homomorphic Encryption.

These offer tradeoffs among usability, convenience, and security. The primary concern for this paper is improving security, but usability and convenience are considered as well, because they are important for a real-world implementation.

A. Go-Between

The first translation approach involves two entities who wish to communicate and a third entity acting as a go-between. This go-between understands both languages and translates. It listens to one entity and then translates and repeats the ideas to the other entity. The endpoint entities communicate only through the go-between. They send communications destined for the other endpoint to the go-between and listen for responses from the go-between.

This is a common approach used in real life with translators, and it is available in digital form through Google Translate and other such services.

The security of this approach has some weaknesses. First, the listener must trust the go-between with the translated information. The go-between knows all the information that the listener receives and actually is responsible for generating this content. Unless the endpoint trusts the go-between to perform such operations, this method does not work. Not only the content is revealed, but also the source of the content, the fact that the listener wishes to receive in a particular language, and the fact the the listener does not wish to receive the original language. The metadata may be more revealing than the data in some cases.

Second, the speaker must trust the go-between. On the Internet, in particular, this can be a problem. Websites often require a login or other form of authentication of a requester, and a translator may not have the proper credentials for access even if the original requester does.

Third, this method does not work for private data. A service such as Google Translate cannot access private data. It can only translate what is reachable on the public Internet.

Fourth, there is no ability to limit or redact the information in the content that the go-between receives. For a website with personal information, it would be desirable to eliminate the personal information prior to translation, but the go-between translates before the recipient can decide what to redact. The go-between either sees everything or nothing, but no finer-grained sharing is possible with this method.

Fifth, it is difficult to determine whether the go-between is showing a real translation or not. If the translator is implemented as part of a firewall, the firewall may block certain content, which would lead to an incomplete translation. If the translator wishes to mis-translate a competitor's web sites in a malicious way, there is no way to detect this. If a mistake is made, there is no way to fix this. Any guarantees of integrity of the translation must be provided through additional means. This approach does not provide any.

B. Translation Service

One of the problems with the go-between is that it breaks the end-to-end connection between the endpoints. It inserts another potentially untrusted entity in the middle. Even if this is a credentialed entity, which eliminates outside eavesdroppers, this is still a trust issue because this introduces a new entity into the conversation.

To attempt to fix this and its associated security issues, the go-between is moved out of the position between the two entities and instead put on the side. An entity wishing to translate may now communicate directly and then request translation directly from a translation service through a separate end-to-end secure connection.

This fixes the second and third weaknesses of the go-between approach. Only one endpoint must trust the translation service, because the other no longer interacts with it, and private data may be shared directly by the entity requesting translation.

The first and fourth weaknesses remain. The entity requesting translation must still trust the translation service, and the translation service sees all content that is translated. However, these can be mitigated better with a translation service than with a go-between.

The entity requesting translation must trust the translation service, but the level of trust is reduced. The requester no longer indicates the source of the content, so concerted attempts by the translator to deceive are more difficult. Mis-translating a competitor's website would be difficult if done page-by-page, paragraph-by-paragraph, sentence-by-sentence, or even word-by-word. By choosing the appropriate content to translate, the requester to the translation service can compare word-by-word translations to translations of larger sections. Any attempts to deceive would likely show up as inconsistencies between the word-level translations, which would not be aware of the context from which the words came, and the translations of entire pages or documents, which would have more context.

The problem of revealing all information to the translator can also be mitigated in some cases. If a document has structure that allows identification of sensitive information, even without its translation, this sensitive information can be removed prior to translation by the requester. This would not apply to all cases, such as pure text, but a confirmation email from a foreign hotel, for example, would likely highlight personal information, reservation numbers, and credit card information. The rest could safely be copied and submitted for translation.

The final weakness of the go-between can be significantly mitigated through the translation service approach. The correlation between the input and output can be established by the same approach as mentioned above for mitigating trust. This reduces the translation problem to that of word translation, which is easier to fix than higher-level attempts to deceive.

C. Translation App

The translation service preserves end-to-end security, but it still requires trust of additional endpoints. To improve this solution requires eliminating any extra trusted entities. To do this, the translation service is replaced with an entity that provides a translation application. Instead of doing the translation itself, this entity now provides all the knowledge needed for the endpoint to do the translation itself. This could be implemented as a mobile app that is downloaded and installed on a device that is used for communication. Now the device can intercept and translate communications instead of requiring a third party.

This fixes the fourth weakness that requires the third party to see the translated information. It also reduces the level of trust in the translation service. The translation process is now static, which means it does not depend on who is requesting translation, when, how, or any other context outside of the content itself.

The fifth weakness can be even further mitigated from the translation service by using reviews of the app from others and external testing and validation against known translations. This does not provide full assurance, but it offers a strong mitigation against intentionally misleading translations for particular types of content.

The problem with this approach is related to its strength. The translation now happens on the endpoint doing the communication. For mobile devices in particular, translation may be a computationally intensive process that affects performance of the primary communication, it may require additional storage, and it may reduce battery life.

The exact impacts of the approach depend on the situation. A large application may reduce storage on the device and consume network bandwidth to download. A computationally intensive translation process may take a long time, consume power, and reduce the performance of other applications on the device.

Another issue to consider is whether the app will be downloaded and used once or repeatedly. If used repeatedly, only a single download is needed, which may actually consume less bandwidth than repeated requests to a third-party translator.

It is possible for either the requester or provider of information to do the translation. It may actually be more convenient for the information provider to translate, since there are likely to be fewer providers than consumers and the providers often have more computation resources available. This would be similar to websites that offer different language choices. The same issues are relevant, but the final requirement on the device resources is likely to be easier to address. A potential complication is where a requester knows what they want, but a provider does not provide that translation capability. Putting the translation app on the requester makes the approach more flexible.

D. Homomorphic Encryption

A final approach using a particular type of encryption offers some different trade-offs. The idea of this approach is to use encryption that preserves some structure of the original data. Using this structure, operations can be performed on the encrypted data that correspond to related operations on the original data. For example, addition of numbers may correspond to multiplication of their encrypted values.

This enables a requester to use a translator without revealing the data itself to the translator. The process is as follows:

1) Transform the translation service to operate on homomorphically encrypted data;
2) Encrypt the data to be translated at the requester;
3) Send the encrypted data to the translator;
4) Translate the encrypted data using the homomorphic transformation of the translation process;
5) Return the result to the requester, still encrypted;
6) Decrypt the result at the requester to retrieve the translation of the original data.

Because of the complexity of translation, the homomorphic encryption must be full homomorphic encryption (FHE), which enables arbitrary operations on the original data by corresponding operations on the encrypted data.

Using homomorphic encryption combined with either the go-between or the translation service approach offers significant improvements in security. The main problem with this approach is that FHE is very slow. The encryption, decryption, and homomorphic computation introduce significant computational burden and associated delays on the entities involved. Current implementations are simply not practical.

Research into homomorphic computing will likely improve its performance, but it it not likely to reach mainstream use soon. A potential solution is the use of partial homomorphic encryption, which is faster, but allows only a single operation on the data, such as addition or multiplication. This could be useful, for example, for simple translation problems, such as converting a number of miles to a number of kilometers.

5. IMPORTANCE OF SECURE TRANSLATION

The problem of secure translation is important for any two people trying to communicate without a shared language, especially in regard to sensitive information. However, this problem has much wider applicability.

With the Internet of Things (IoT) growing rapidly, a large number of different entities on the Internet use different, often proprietary, communication protocols and data formats, and they often share personal or other sensitive information with other entities. To enable all of these entities to work together, we need a way to translate all the protocols and data formats, and we need to secure the data from end to end. This maps directly to the secure translation problem.

Another importance of end-to-end secure translation is the problem of automated exploits of translation failures. If an external entity can view data and its translation by eavesdropping or other means, it is easier to find and exploit translation errors between endpoints. With the amount of such translation between IoT and other endpoints, a single exploit can rapidly escalate to create significant damage.

With new protocols and formats, the old ones do not disappear, so the translation challenge will always exist. Legacy equipment and systems must use translators to talk to newer equipment, and this challenge will grow as more equipment (future legacy equipment) is produced.

The inherent inacurracy of translation, with granularity, repetition distortion, and translation context mis-matches, means that with increasing need for translation, endpoints will learn to expect and tolerate higher levels of distortion and lower levels of fidelity in their communications. A non-native language speaker understands that misunderstandings happen. A malicious individual can expoit these to blame intentionally misleading information on the translation process, thereby avoiding retribution for the harm they cause.

In the digital world, the problem is similar. Increasing noise makes intentional malicious activity blend into the noise. Attacks on digital systems are often identified by something that is abnormal. When the base level of noise is high, abnormal things blend in with this noise and go undetected.

6. CONCLUSION

Translation is difficult due to basic challenges in representing and repeating information. The different contexts of different languages and representations introduces another source of errors and inaccuracies. In order to reduce the ability of malicious entities from exploiting these inaccuracies to do harm, end-to-end security prevents, to the extent possible, any external entities from interfering with a two-party communication. Different approaches trade off among convenience, availability, performance, and security. This paper analyzes different approaches for secure translation with respect to security. Currently common approaches, such as go-betweens or translation services, lack basic security properties. The use of local translation through an application mitigates security problems if the local device can handle the resource requirements. Homomorphic encryption offers hope for the future—that it will patch up some of the current security weaknesses of using cryptography, but current implementations of FHE are too slow and PHE too limited for general trranslation needs.

REFERENCES

[1] Virgil Gligor, “Homomorphic Computations in Secure System Design,” Final Report Carnegie Mellon University, Pittsburgh, PA 15213, July 10, 2014.