Eavesdropping proh

Sometimes, when two people or software applications are communicating via the Internet, a third party is listening. Cryptographic protocols could prevent this situation, but software developers often find it difficult to correctly integrate them into applications. This is the reason why researchers at the TU Darmstadt want to automate encryption.

By Boris Hänßler

Andrea wants to send her friend Stefan a message via the Internet. To prevent anyone else from reading it, she communicates with Stefan and agrees on a secret code with him that only he and she will be able to decipher. Should the message fall into the wrong hands, it will consist of nothing more than an incomprehensible string of characters that cannot be deciphered without the key. However, what Andrea and Stefan don't know is that a spy has inserted himself between them. The secret code that they believe they established between themselves was actually generated by the spy. He sent it to both participants by pretending to be the other one in each case. Now he can read all of their messages and, for example, ask Andrea for an important password. She feels safe, as she thinks she's sending it to her friend.

This scenario gives a rough idea of a man-in-the-middle attack, during which an attacker manipulates Internet communication. In information technology, the relevant key or code is part of a so-called encryption algorithm. It is designed to protect data that is transmitted back and forth either within a single application or between different applications. Encryption protocols usually run in the background without the user noticing. If, for example, we visit an online shop, our browsers and the online shop automatically negotiate a unique key that is used to mathematically encrypt the data, be it details of the order or bank details. A third party could not do anything useful with the data without the key.

„Scientists from the fields of cryptography and software engineering are working closely together here; something that is still happening much too rarely around the world".

We encounter cryptography constantly: when using Apps, when sending emails, when communicating via messenger services or during in-house corporate communications – whenever sensitive data is involved. Encryption protocols ensure a certain level of security under certain assumptions concerning the abilities of the potential attackers, but only if the designer of an encryption protocol implements it correctly and the application developers have used and integrated it correctly in their code. And, it is in this context that one finds problems that should not be ignored. Configuring cryptographic components is difficult and one cannot expect software developers to be cryptography experts: they make mistakes. Between 2013 and 2015 alone, 1769 security vulnerabilities registered in the USA's „National Vulnerability Database" (http://nvd.nist.gov) were the result of such mistakes – as such, issues involving the integration of encryption protocols in applications were the fourth most frequent source of such registered vulnerabilities.

This is not surprising, if one considers several studies presented the most renowned scientific conferences relating to the area of cyber security over the past few years. These demonstrate that software integration is an important point of weakness, especially since the use of components of cryptographic libraries requires knowledge of too many details that application programmers often do not posses. For example, software developers need to ensure that the individual steps of an encryption protocol are executed in a specific order, for which concrete recommendations are available depending on what is to be protected. However, software developers frequently lack the time to read the relevant manuals. Another source of errors are so-called digital certificates that verify the validity of a given key. Developers sometimes disable the verification process for their software certificates, in order to speed up testing, but then forget to re-enable it for the production system. According to Mira Mezini, Head of the Software Engineering Research Group at the TU Darmstadt: „these are both common errors, even among serious software providers; and those are just two examples among many". Intruders always have the advantage that they only need to discover a single security vulnerability to be able to steal data, while developers are faced with the enormous challenge of closing all possible security gaps.

It is neither possible nor appropriate for all software developers to double up as encryption experts, especially when considering that according to a 2013 IDC report (available at: http://bit.ly/2a86Mju), 7.5 million out of 18.5 million software developers world wide are hobby programmers. According to Mira Mezini, this trend is increasing. „This is why", she explains, „cryptographic experts and software developers need to better augment each other's skills". This is precisely the approach of the „CROSSING" collaborative research center at the TU Darmstadt, funded by the German Research Foundation (DFG). An interdisciplinary team from the TU Darmstadt not only develops novel encryption protocols in the context of „CROSSING", but also new methods and tools to simplify the integration of encryption protocols into application codes and, thereby, to increase trust in computer technology in the long term.

There are three areas of focus at „CROSSING": One of these involves the development of new cryptographic primitives, which constitute the smallest building blocks in cryptographic protocols. The second research priority is concerned with creating new cryptographic systems from these primitives. In the research area, in which Mira Mezini is

Information

Specialism: Software Engineering
Prof. Dr.-Ing. Mira Mezini
Phone: ++49 (0)6151/16-21360
E-mail:
mezini@st.informatik.tu-darmstadt.de
www.stg.tu-darmstadt.de

ibited

involved, the scientists support software developers in the integration of cryptographic primitives and systems in their applications without errors. It is precisely this kind of research that has received too little attention in cyber security research until now. „Till now“, Mezini explains, „cyber security research has focused, if at all, on software end-users, to provide them with software tools with which they could protect themselves from harmful software“. „CROSSING“, she adds, is investigating issues concerning software developers.

Yet, how can the researchers help software developers? „Our assumption“ says Mezini, „is that the errors they make when trying to properly integrate cryptographic protocols could be significantly reduced if we were to engage more with the requirements of the developers and provide them with active support in the integration of encryption protocols in application codes by way of intelligent tools. This means intelligent encryption libraries the components of which would, to a significant degree, automatically install themselves in the application codes“. In an initial step, Mezini's team has approached developers themselves to ask for their views on the core question as part of an empirical study. „Developers work in a task-oriented manner“, she says. „They want encryption libraries that tell them: this encryption component is available for the task you wish to achieve, and this is how to configure, combine and use it to ensure the security of this particular task. It would be best if the libraries could take over these tasks, i.e., do them automatically for the developers“. For example, a developer might wish to encrypt a communication channel. The library would then suggest the best components to use for the encryption and explain how to configure them to prevent them from negatively impacting one another. The objective is to develop appropriate software-oriented methods and procedures that would be built into encryption libraries. In a follow-up step, the process would be further automated. Ultimately, the intelligent encryption library would include two interfaces: one interface for the designers of cryptographic protocols to check that they had been correctly implemented, and another interface for the software developers to ensure that correctly implemented encryption protocols would also be correctly integrated into and used in application software. In this way, the procedures would be checked twice – when first integrated into the encryption library and then during the implementation of the application software. Ideally, the latter would take place directly within a software development environment.

Today, software is programmed in this type of development environment. This involves integrated tools that simplify software programming, checking and testing. The most popular development environment for the widely adopted programming language JAVA is „Eclipse“. According to Mezini: „we first want to enhance the functionality of Eclipse by linking it to our encryption library such that, even during programming, continuous checks would be running in the background to ensure that cryptographic algorithms are placed at the right points

Photo: Katrin Binner

Professor of Computer Science Mira Mezini, Head of the Software Engineering Research Group at the TU Darmstadt.

and are correctly configured and deployed. Other development environments could also be enhanced in a similar way“. In addition, this approach would ensure that an application would remain secure even if the programmer were to upgrade it at some point, or if certain encryption procedures that had been integrated within the encryption library were suddenly to be deemed no longer secure. This would save the developers time, and they would always be able to be confident that they had implemented the best possible procedure currently available.

In addition to Mezini and her colleague Eric Bodden, who moved to the University of Paderborn a few months ago, two doctoral students and a post-doctoral researcher are working on the project. „What we hope“, says Mezini, „is that an active community will emerge in the long term that will adopt and enrich our encryption library with more and new cryptographic components. „We have created an important basis by at least having brought cryptographers and researchers working on software methods and languages closer together at the TU Darmstadt, something that is still happening much too rarely around the world“.

The author is a technology journalist.