Man-in-the-middle attacks evolved... but our security models didn't

Hugo Jonker[1], Sjouke Mauw[2], and Rolando Trujillo-Rasua[2]

[1] Open University of the Netherlands
hugo.jonker@ou.nl
[2] University of Luxembourg
{rolando.trujillo,sjouke.mauw}@uni.lu

Abstract. The security community seems to be thoroughly familiar with man-in-the-middle attacks. However, the common perception of this type of attack is outdated. It originates from when network connections were fixed, not mobile, before 24/7 connectivity became ubiquitous. The common perception of this attack stems from an era before the vulnerability of the protocol's context was realised. Thanks to revelations by Snowden and by currently available man-in-the-middle tools focused on protocol meta-data (such as so-called "Stingrays" for cellphones), this view is no longer tenable. Security protocols that only protect the contents of their messages are insufficient. Contemporary security protocols must also take steps to protect their context: who is talking to whom, where is the sender located, etc. In short: the attacker has evolved. It's high time for our security models and requirements to catch up.

1 Introduction

Man-in-the-Middle (MitM) attacks are well-known to the security protocols community. Indeed, a security protocol will typically be proved secure against a Dolev-Yao [DY83] attacker, a powerful definition of man-in-the-middle from 1983. However, the use of systems has evolved since 1983 – as have threats against systems. The current use of MitM models by the security protocols community does not account for this evolution and, as such, is outdated. Two key ways in which systems and their use have evolved are:

- Ubiquitous usage coupled with a dependence on connectivity,
- Personalisation of the connection.

This has led to numerous MitM attacks on deployed protocols such as SSL/TLS and GSM (cf. Section 2). Such attacks are generally treated as individual occurrences. In this paper, we advocate that such attacks are part of a larger pattern: the current notion of man-in-the-middle as used in security research no longer matches real world MitM-attacks. We propose to address this by evolving the notion of man-in-the-middle to account for a protocol's context, that is, those data that influence the protocol or are inherently used or disclosed by the protocol (cf. Section 2). We distinguish two classes of solutions, based on whether or not

the communication partner is trusted. We call these classes *context agreement* and *context verification*, which are discussed in Sections 3.2 and 3.1, respectively. We illustrate how context may be leveraged to strengthen protocols by showing a simplified extension for GSM/UMTS in Section 4.

2 How man-in-the-middle attacks outgrew security models

As mentioned, networks and the use of networks has evolved. The dependence on connectivity ensures that most protocols have fallbacks to ensure backwards compatibility. The mobility of the endpoints means that the context is no longer fully known a priori. Moreover, as connected devices have become small and capable, the connection has become far more personal.

2.1 Backwards compatibility

Computers have become ubiquitous, and are used in many if not most jobs in the western world. Moreover, computers have become vastly interconnected: a button pressed here affects a display there, a file saved here is updated there, a withdrawal here changes a balance sheet there, etc. Nowadays, many systems even depend on such interconnectivity. This interconnectivity is only possible if the computers on either end understand each other - computers must be using the same communication protocol.

As our understanding of security increased, flaws in communication protocols were found and repaired. Since systems depend on interconnectivity, there inevitably has to be a fallback mechanism to communicate with computers that are not yet updated. Typically, this leads to an initialisation phase, during which the computers agree on version, cipher suite, and other parameters of the protocol. However, in general, this phase is not considered part of the protocol and omitted from security proofs. This has already caused multiple vulnerabilities, e.g. the DROWN [ASS+16], LOGJAM [ABD+15], FREAK [BBD+15], and POODLE [MDK14] attacks on TLS.

The solutions for such attacks are typically rather ad hoc, in that the solution addresses the attack and little beyond that. However, we should learn more from these attacks than just how to prevent one instance.

2.2 Personalisation of the connection

With the rise of the smart phone and wireless connectivity, users no longer need to be stationary to communicate. Connections have become mobile. As such, the user's context is no longer fixed. Moreover, the small form factor, ubiquitous connectivity and the mature capabilities of current devices allow users to take their personal computing platform with them at all times, and communicate at any time using wireless communication. Thereby, these devices have become far more intwined with personal life than any computing platform that came before.

Of course, the wireless communication signals can be picked up by any nearby antenna, and so security protocols can ensure that the contents of their communication remains confidential from an eavesdropper. However, an eavesdropper can still learn information, such as the approximate location of the communicating party. Since the communication endpoint has become far more personal, determining the location of the communicating device has become synonymous with determining a person's location. Nowadays, there exist commercially-available MitM devices for intercepting and tracking GSM devices[3] and WiFi/Bluetooth devices[4]. These devices perform typical man-in-the-middle attacks, yet such attacks are not considered when proving security of the protocol.

Existing solutions against such tracking are based on detecting the trackers, for instance by analysing protocol properties of the communication signal. Currently, research into detecting cell site simulators has just begun (e.g. [DPK+14]), and a few smart phone apps claim to detect such shenanigans. These all rely on particular details of the used protocol (e.g., base station parameter fingerprinting, network operator fingerprinting, etc), and the simulator's imperfection in replicating such details. However, we hold the view that this will only lead to an arms race, while not addressing the crucial underlying point: that there is, indeed, a man-in-the-middle. A more thorough, generic solution must be developed.

3 How to determine context

Remark that all these attacks arise from the outdated perception of man-in-the-middle as currently held by the security community. While the aforementioned attacks are all classified as MitM attacks once discovered, our current design processes and evaluation tools are insufficient to prevent these attacks. We argue that the security community must start researching contextual physical properties in a formal way as to make it part of the protocol. An example of such a property is signal strength.

We distinguish two categories of possible solutions:

- **context verification:** without a trusted communication partner,
- **context agreement:** with a trusted communication partner.

If there exists a trusted communication partner for the user (e.g., a trusted base station), the user and her partner can exchange their observations on the perceived context, and determine whether their combined observations indicate an anomaly. Thus, user and partner agree on some properties of the context, which is reminiscent of the security requirement "data agreement". Hence we label this *context agreement*.

If the user does not trust any communication partner, she is determining the validity of the context by herself. We label this *context verification*.

[3] For an overview, see EFF's cell site simulator FAQ.
[4] E.g. the Navizon indoor triangulation system.

3.1 Context verification

Security claims are proven based on assumptions on the adversary capabilities. An important assumption is whether the adversary is able to decrypt an encrypted message without knowing the encryption key, also known as perfect cryptography, without which it is impossible to prove security in current security protocols. Such assumption is based on computationally hard problems and our inability to solve them. It is thus apparent that physical claims can also be proven based on physical laws that constrain the capabilities of the adversary.

So, what do a secret key and the speed of light have in common? The theory of relativity states that a message or a signal cannot travel at a speed faster than the speed of light. Thus, an adversary cannot manipulate a message traveling at the speed of light without inevitably causing a delay, just as he cannot forge a signature without knowing the secret key. This provides a means for context verification, in particular for proximity verification, which has been studied since 1993 when Brands and Chaum introduced *distance bounding* protocols [BC93].

Distance bounding protocols are cryptographic protocols that exchange messages at (nearly) the speed of light. By measuring the round-trip-time (RTT) of a message exchange, a verifier can securely compute a tight upper bound on its distance to a prover. Therefore, the verifier can ensure that the prover is within a given radius.

While there exists a rich literature in distance bounding protocols [AMTR15], the secure verification of other physical properties (e.g. location and signal strength) is still an open challenge. There exist heuristics approaches, such as [DPK+14], which measures the deviation of the attacker's signal from the expected signal. The challenge is thus how to bring this heuristic approaches into a formal framework that allows for security proofs.

Definition 1 (Context verification). *A party A achieves* context verification *of her observation $obs_A(C_B)$ of the context C_B of party B if, whenever A completes a run of the protocol (apparently with B) then $obs_A(C_B)$ is correct with respect to C_B.*

In contrast to standard security properties, context verification relies on properties that may change with time, e.g., location and meteorological conditions. Remark that context verification does not require B to execute the protocol with A. As such, it captures contextual information about the context of the communication partner that can be verified without his involvement.

3.2 Context agreement

We introduce context agreement as a notion weaker than context verification. To formalise it, we build on the notion of *data agreement* in security protocols, whereby two parties securely agree on the values of a set of variables [Low97]. This is used, for example, to secretly agree on a session key for subsequent communications.

Definition 2 (Context agreement). *A party A achieves* context agreement *with another party B on B's context C_B if, whenever A completes a run of the protocol (apparently with B) then B has been previously running the protocol (apparently with A) and the observation of A on B's context is the same as B's observation of his context in that run, that is: $obs_A(C_B) = obs_B(C_B)$.*

Note that in context agreement the trusted parties agree on the *observed* context, not on the actual context. A sufficiently powerful attacker or a lying party can ensure that a fake context is agreed to. For example, a device B may determine its location based on a fake GPS signal. Upon agreement on B's location, neither A nor B are aware that a false location has been agreed upon.

We observe that context agreement has already been used to prevent downgrade attacks. Downgrade attacks are a type of MitM attack that exploit backward compatibility. For example, a renegotiation vulnerability[5] was uncovered in some implementations of SSL/TLS. The vulnerability was soon fixed by including and verifying information about previous handshakes so that the client and server agree on their views of the negotiation state.

Note that context verification requires that the context observed by A is indeed B's actual context, while context agreement may be an agreement on a context that is different from B's actual context. Therefore these properties are incomparable.

4 An illustration: Detecting a GSM/UMTS MitM

A recent hot topic is the availability of so-called cell-site simulators [MW04] (also known as IMSI catchers or Stingrays). These devices act as plug-and-play man-in-the-middle devices for GSM/UMTS traffic. Because phones favor towers with strong signal strength, reducing security is simple for a cell-site simulator: just ensure that the lowest-security base station has the best signal, or impersonate a base station and switch to low-security mode. In some cases, the cell-site simulator itself can easily turn off encryption completely. A cell-site simulator can thus collect identifying information such as the International Mobile Subscriber Identity (IMSI), metadata about calls like the telephone number dialed, the date, time, and duration of the call, or even learn the content of the calls if none or weak encryption is used.

There exist several underlying security protocols in the UMTS/GSM standard. Figure 1 shows a simplification (omitting most details) of the standard handshake in UMTS. Every phone shares its own long term secret key[6] k with a dedicated home network. To further simplify the depiction, we let this key be

[5] CVE-2009-3555.

[6] In the GSM standard, the tower may choose unilaterally to stop encryption, and the client has to follow. An attacker can therefore simply shut down encryption (e.g. by using a downgrading attack to fall back to the old standard, and then to stop encryption). Thus, this shared key alone cannot ensure secure communication.

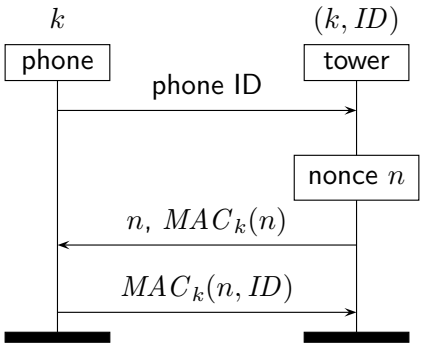

Fig. 1. Simplified UMTS protocol.

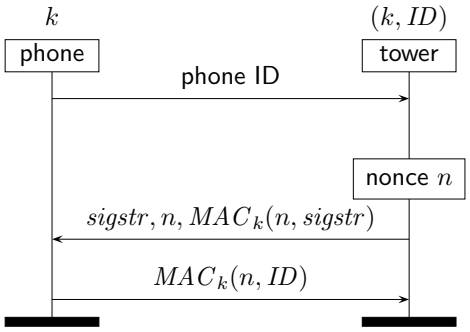

Fig. 2. Inclusion of context.

known to the base station. Once connected to a base station, the phone sends its identity. The base station then uses the secret key to authenticate the phone.

Note that the protocol in Figure 1 does not perform context authentication. Therefore, this protocol is unable to prevent MitM attacks as the one in [MW04].

We propose to leverage this shared key in order to achieve context agreement as follows: The phone and the tower communicate securely on the tower's signal strength (*sigstr* in Figure 2). Using the authenticated value of signal strength, the phone can determine whether the perceived signal strength is "realistic", and not far higher or lower than the rest of the communication warrants. This is depicted in Figure 2. Obviously, this approach can be strengthened by including further details of the context. An obvious extension is inclusion of the tower's GPS coordinates, as this will allow the phone to determine a much narrower range of acceptable values of the signal strength.

The point of the sketched solution is to illustrate how context can be leveraged to detect a man-in-the-middle. A MitM attacker can either relay encrypted messages from the client, or drop them. If messages are dropped, after the tower has announced its capability to engage in this exchange (which will be nothing more than announcing supported protocols), then clearly there is a MitM attacker. If the messages do come encrypted, but the tower finds the client's perceived signal strength unrealistic, then likely there is a MitM attacker.

5 Conclusions

In this paper, we highlighted that more and more man-in-the-middle attacks are of a type that are not addressed by current security models. The overall principle is that security models do not take sufficient context into account. Context includes details such as the setup/initialisation phase, in which protocol parameters are decided, but also physical parameters, such as location.

We presented examples of either case where the lack of consideration of context led to attacks. To address this, we advocated research of contextual physical

properties in a formal way. Finally, we sketched two solution directions for considering contextual properties. We plan to expand upon the following avenues for further research:

- Context can be both subjective and objective. The possession of a secret key can be regarded as an objective contextual information, either you have it or not. Location however can be both objective and subjective. A fixed communication tower has an objective view on its location, but a mobile phone does not. This distinction between subjective or objective contextual data requires further clarification and formalization.
- Context may change. Therefore, distinguishing between older context and current context can provide a way to satisfy security properties. For example, if a phone is communicating with a fixed base station, it is sufficient to verify whether the (observation of the) current context of the base station matches the previously determined context, as this context should not change.

References

ABD+15. D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Z. Béguelin, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proc. 22nd SIGSAC Conference on Computer and Communications Security*, pages 5–17. ACM, 2015.

AMTR15. G. Avoine, S. Mauw, and R. Trujillo-Rasua. Comparing distance bounding protocols: A critical mission supported by decision theory. *Computer Communications*, 67:92–102, 2015.

ASS+16. N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. A. Halderman, V. Dukhovni, E. Ksper, S. Cohney, S. Engels, C. Paar, and Y. Shavitt. DROWN: Breaking TLS using SSLv2. 2016.

BBD+15. B. Beurdouche, K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, A. Pironti, P.-Y. Strub, and J. K. Zinzindohoue. A messy state of the union: Taming the composite state machines of TLS. In *Proc. 36th Symposium on Security and Privacy, S&P'15*, pages 535–552. IEEE Computer Society, 2015.

BC93. S. Brands and D. Chaum. Distance-Bounding Protocols. In *Advances in Cryptology – EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 344–359. Springer-Verlag, 1993.

DPK+14. A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. R. Weippl. IMSI-catch me if you can: IMSI-catcher-catchers. In *Proc. 30th Annual Computer Security Applications Conference, ACSAC'14*, pages 246–255. ACM, 2014.

DY83. D. Dolev and A.C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(12):198–208, 1983.

Low97. G. Lowe. A hierarchy of authentication specifications. In *Proc. 10th Workshop on Computer Security Foundations, CSFW'97*, pages 31–43. IEEE Computer Society, 1997.

MDK14. B. Möller, T. Duong, and K. Kotowicz. This POODLE bites: exploiting the SSL 3.0 fallback, 2014.

MW04. U. Meyer and S. Wetzel. A man-in-the-middle attack on UMTS. In *Proc. 3rd ACM Workshop on Wireless Security, WiSE'04*, pages 90–97, New York, NY, USA, 2004. ACM.