

Forum: [Hack Forums](#)

Thread: [Why All Encryption Over The Internet Is Completely Useless](#)

HotNoob - January 13, 2015, 3:59 pm

Part 1, goes over the basics; I ignore Asymmetrical encryption to better emphasize and make easier to understand for the average joe.

[Video: <https://www.youtube.com/watch?v=bGYtJHMnvns>]

Part 2, goes into more detail to address why Asymmetrical (RSA/DH/SSL/TLS) encryption does dick all

<https://www.youtube.com/watch?v=0jffZjiL9AU>

Enjoy.

chick3nman - January 13, 2015, 4:51 pm

(Jan 13, 2015 04:59 PM)HotNoob Wrote: [\[CITE\]](#) Part 1, goes over the basics; I ignore Asymmetrical encryption to better emphasize and make easier to understand for the average joe.

-snip-

Part 2, goes into more detail to address why Asymmetrical (RSA/DH/SSL/TLS) encryption does dick all

<https://www.youtube.com/watch?v=0jffZjiL9AU>

Enjoy.

This is why asymmetric encryption exists...? Even in your video about asymmetric, I'm pretty sure you missed the number 1 way we fix the problem you presented, which is known and mirrored public keys. If I know the public key of a server, and I go to connect to it, and the key i pair with is different, the connection is killed and I know that I am being MITM attacked. If i don't know a public key, say this is the first time I go to connect with the server and I haven't traded public keys yet, then i ask multiple third parties for the public key of the server. If there are discrepancies in the responses I get, then i know something is up and that i should be careful. If i get all the same public key back, and I go ahead and connect to the server and it presents a different public key, then I know that something is wrong again.

Now, I know the response will be "But what if the person MITM attacking you replaces their key over the pub key of the server in all the responses you get from third parties? How will you know the pub key then?" Ask another server to contact the server for you, tell them to sign a message to you with their key and against your pub key, and then to verify it and send it over. If the message you get isn't signed with any of the keys third parties claim or the key the server is showing, you know something is wrong again.

The Diffie–Hellman key exchange was written specifically for this purpose and actually addresses the problem of someone who is able to intercept and CHANGE your incoming/outgoing messages. It is known that it is vulnerable to Unkown key-share attacks, the type of attack you are describing, and there are several fixes that are proposed to solve this issue, some include apply an authentication layer using protocols such as STS, others involve using WOT or Web of Trust systems that are used to effectively decentralize the key verification part of pub-key encryption and allow you to confirm new connections with people you already trust.

WOT and similar systems fix this problem under the assumption that at SOME point in time, you are able to trust at least ONE other person fully. If you can not reliably trust one other person at any point in time, you have much much bigger problems than someone decrypting your SSL sessions every time you log in to your email. I can physically walk my pub key to someone on a flash drive and constantly physically verify things with other people near me. In theory, that one physical exchange is all i need to begin a Web of Trust. If i can't trust that physical key exchange, I should probably be hiding in my closet with all the lights off.

TL;DR, Pub key exchange is only broken if you half-ass it and don't check things out correctly.

HotNoob - January 13, 2015, 8:30 pm

(Jan 13, 2015 05:51 PM)chick3nman Wrote: [\[CITE\]](#) This is why asymmetric encryption exists...? Even in your video about asymmetric, I'm pretty sure you missed the number 1 way we fix the problem you presented, which is known and mirrored public keys. If I know the public key of a server, and I go to connect to it, and the key i pair with is different, the connection is killed and I know that I am being MITM attacked. If i don't know a public key, say this is the first time I go to connect with the server and I haven't traded public keys yet, then i ask multiple third parties for the public key of the server. If there are discrepancies in the responses I get, then i know something is up and that i should be careful. If i get all the same public key back, and I go ahead and connect to the server and it presents a different public key, then I know that something is wrong again.

Now, I know the response will be "But what if the person MITM attacking you replaces their key over the pub key of the server in all the responses you get from third parties? How will you know the pub key then?" Ask another server to contact the server for you, tell them to sign a message to you with their key and against your pub key, and then to verify it and send it over. If the message you get isn't signed with any of the keys third parties claim or the key the server is showing, you know something is wrong again.

The Diffie–Hellman key exchange was written specifically for this purpose and actually addresses the problem of someone who is able to intercept and CHANGE your incoming/outgoing messages. It is known that it is vulnerable to Unkown key-share attacks, the type of attack you are describing, and there are several fixes that are proposed to solve this issue, some include apply an authentication layer using protocols such as STS, others involve using WOT or Web of Trust systems that are used to effectively decentralize the key verification part of pub-key encryption and allow you to confirm new connections with people you already trust.

WOT and similar systems fix this problem under the assumption that at SOME point in time, you are able to trust at least ONE other person fully. If you can not reliably trust one other person at any point in time, you have much much bigger problems than someone decrypting your SSL sessions every time you log in to your email. I can physically walk my pub key to someone on a flash drive and constantly physically verify things with other people near me. In theory, that one physical exchange is all i need to begin a Web of Trust. If i can't trust that physical key exchange, I should probably be hiding in my closet with all the lights off.

TL;DR, Pub key exchange is only broken if you half-ass it and don't check things out correctly.

TL;DR u didnt see part 2.

lIn2read, lIn2watch, lIn2listen...

Or are you a government plant?

4:44