Andrew Johnson

CS 427

Final Project Paper

RSI Videofied's Faulty Crypto-System Analysis

The year 2000 was a good year for many, but for one company it was the start of the beginning. This was the year that RSI Video Technologies was founded. They had a vision of wireless security systems being perfected. One idea they added to security systems is video verifications, where they send a video feed to the owner. That way the owner can tell if it's a false alarm or not. Where does this tie in with cryptography? The key factor to their security system is that it is wireless. Therefore, when the user gets their video feed, it must be sent over IP or GPRS (cellular network). You don't want anyone being able to see or block that feed to you, so RSI must encrypt this video to you so only you can see it.

With this knowledge, you would want top security for your system, except that is the opposite you would get when using their crypto-system. Instead, RSI has a security protocol that looks like a security protocol, but any attacker with beginning knowledge of the systems could easily manipulate the alarms.

It seems as if the developers were trying to use the block cipher known as AES for their cryptography. They use the serial number of a panel to prove it's an authentic system, then the server will return the key. The "secure" part of this transition is that when sending the serial number to the server after the initial transaction, the server will not send the key again. This seems like it can be secure, but it is easily worked around. All an attacker has to do you obtain a new key from Videofied is connect to a separate Videofied server.

Conveniently, there a many servers we can connect to. Another thing that is noticed when receiving the "new" key is that it is exactly like the first key. This makes it deterministic and not uniformly random. This already makes the system very vulnerable to attacks due to a very weak authentication process. With further examination of the system and key. The key returned by server has a high resemblance to the serial number that was originally sent to it.

Therefore, the attacker just has to use the serial number to determine the key for authentication. He can do this before sending in the plaintext to the server. With all of this, we know all the attacker needs to access to your system's serial number and he can trivially find the key the server accepts. With this key the attacker is able to manipulate what the server sees. This means the server will never send a flag to the user saying someone is breaking in. Since the attacker can go this far with the system then the cryptography is essentially void.

Another part wrong with this system is that nothing is actually encrypted. It's more of a check to see if the person is authorized to view and then shows it. Although the authentication is abysmal, so anyone can go in and change the messages.

A man named Andrew Tierney found this flawed system. When he found it, he tried contacting the company about their problem. The company never did respond so he reported the finding to CERT/CC (Computer Emergency Response Team/Coordination Center). The company was then notified by them to fix/update their systems. This update was completed and verified by CERT 3 months later. I'd say this was a proper way to help fix this problem. Gibbons gave the company time to try to fix it on their own. When it seemed like nothing was happening he let CERT know so they can take action. That forced the company to put out a verified product to where users aren't at risk.

If I were to try to fix this system myself, we would need to use a secure PRF for encrypting the data that is being sent and using a MAC to verify our authentication after using the PRF to encrypt the necessary data. The PRF would guarantee that the data being sent around is uniformly random, instead of the original numbers moved around, and then easily reversed by the owner. Then the MAC will make sure that attackers can't just send in any server address and they are properly part of the system. With both of these it makes the attacker's ability to get into the system exponentially longer.

In conclusion, I went over a security company that deals with wireless video feeds for security systems. We saw that the company failed to think about encrypting any data they were sending through and neglected to try to give a uniformly random key. This made the cryptographic security of their systems absolutely void. To fix this I gave a way to fix this by using PRF's and MAC's. If these things can be implemented properly then the system would be secure. The company has already fixed the problems that were stated earlier, so the security should be at a proper level.