Cyber Security for CMOS Image Sensors

Boyd Fowler, Wenshou Chen and Kevin Johnson

OmniVision Technologies, Email: boyd.fowler@ovt.com / Tel.:+1-408-653-2309

***Abstract*—This paper describes the current state of the art in cyber security for CMOS images sensors. It also shows some of the limitations of this technology, such as unidirectional certificate exchange and incomplete message authentication (MAC) of the image. Then an architecture is proposed that can overcome some of these limitations and improve data and command security. We also describe a framework for analyzing the security of these systems and use it to bound the security of both architectures described in this paper.**

I. Introduction

Cyber security is critical for protecting image data but it is a relatively new feature in image sensors. Modern cyber-security systems use standard algorithms in provably secure frameworks [1]. An example of this is transport layer security (TLS) used by HTTPS for internet data transmission. In addition to a provably secure framework, an efficient hardware implementation is also required to accelerate these algorithms, reduce power dissipation and to resist side channel attacks [2].

II. Current Cyber Security Systems

Figure 1 shows a typical cyber security implementation with an image sensor connected to an ASIC. In this implementation two methods are used to secure the system. The first technique uses a cryptographically signed certificate to authenticate the identity of the image sensor. The second is a message authentication code (MAC) used to guarantee the integrity of the image sensor data. The boot-up process between the ASIC and the image sensor starts by having the image sensor transmit a signed certificate, such as an X509.v3 [3], to the ASIC. Then the ASIC uses a global public key, stored in the ASICs non-volatile memory, to verify the certificate's signature is from a trusted source, such as the image sensor vendor. If the signature is valid then it loads the image sensor public key from the certificate into the asymmetric encryption engine. Then the ASIC creates a random secret key, using a pseudo random number generator (PRG), for the MAC process. This key is then asymmetrically encrypted using the sensor's public key and it is transmitted to the image sensor. The image sensor then decrypts the secret MAC key and loads it into the MAC hardware. Then a NONCE, a number that is only used once, is randomly generated in the image sensor and supplied to the MAC hardware and the image data is processed. A unique NONCE is needed for each image to make the process secure. Finally the image, the NONCE and the unique MAC tag are output and sent to the ASIC. After N images are transmitted from the sensor to the ASIC, a new secret key for the MAC process must be generated by the ASIC and sent to the image sensor to keep the system secure. N is typically a function of the systems susceptibility to side channel based key recovery attacks.

The system described in Figure 1 has many security limitations including an ASIC that can be compromised or bypassed, unencrypted image sensor data, and insecure image sensor control. In addition, due to limited computational resources often only a portion of the image is actually used in the MACing process. Although the part of image that is MACed is often randomized using the secret key this still leads to very poor security.

In order to understand the security of this system we need a few definitions. First we define a chosen plaintext attack (CPA) game for a MAC algorithm $\mathcal{J} = (S, V)$, where S is the MAC signing algorithm and V is the verification algorithm. This game is shown in Figure 2. It starts by having the challenger (image sensor) create a secret key k from key space $\mathcal{K}$ ($k \overset{R}{\leftarrow} \mathcal{K}$). Then the adversary $\mathcal{A}$ (the attacker) sends a set of messages $(m_0, m_1, m_2 \ldots m_{q-1})$ from the message space $\mathcal{M}$ to the challenger. The challenges creates a tag $t_i \leftarrow S(k, m_i)$ from the tag space $\mathcal{T}$ for each received message and sends it back to the adversary. Finally the adversary tries to create a valid message tag pair (m_q, t_q), where $m_q \notin (m_0, m_1, m_2\,,\ \ldots\ m_{q-1})$. If the adversary succeeds then the message tag pair is an existential forgery. The security, or advantage, of the MAC is the probability that the adversary creates an existential forgery ($MACadv[\mathcal{A}, \mathcal{J}]$). For practical cyber systems we need $MACadv[\mathcal{A}, \mathcal{J}]$ to be negligible. The definition of negligible does depend on the attacker but it is usually $< 2^{-90}$.

Using the CPA game above, assume that only a randomly selected fraction x/y of the image is MACed, where x is the number of MACed pixels and y is the total number of pixels. We will call this MAC algorithm $\mathcal{J}^{'}$. In this scenario the attacker can send a single message to the challenger and receive a corresponding tag. Then the attacker can change a single pixel in the image and return the modified image and the received tag. The probability of winning this game is

$$MACadv[\mathcal{A}, \mathcal{J}'] \leq MACadv[\mathcal{B}_{\mathcal{J}}, \mathcal{J}] + (1 - x/y) \tag{1}$$

where $MACadv[\mathcal{B}_{\mathcal{J}}, \mathcal{J}]$ is the probability of winning the game assuming every pixel is MACed and $(1 - x/y)$ is the probability that the pixel selected by the attacker is not in the set of pixels that were MACed. Note that $\mathcal{B}_{\mathcal{J}}$ is a sub adversary of $\mathcal{A}$. Even if $x = y - 1$ and $y \sim 2^{27}$ the MAC advantage is $\gg\ 2^{-90}$.

Now we define a chosen cipher text attack (CCA) game. Given a cipher $\mathcal{E} = (E, D)$, where E is encryption algorithm and D is the decryption algorithm, defined over a key space $\mathcal{K}$, a message space $\mathcal{M}$ and a cipher-text space $\mathcal{C}$, the game starts with the challenger randomly selecting a key $k \overset{R}{\leftarrow} \mathcal{K}$ and a binary value $b \in \{0,1\}$. Then the adversary $\mathcal{A}$ makes a series of queries to the challenger. Each query is either an encryption query or a decryption query. An encryption query consists of having the adversary send two messages of the same length $(m_{i0}, m_{i1}) \in \mathcal{M}^2$ to the challenger, then the challenger encrypts message $c_i \leftarrow E(k, m_{ib})$ and returns the cipher-text c_i to the adversary. A decryption query consists of having the adversary send cipher-text $c_j \in \mathcal{C}$ that is not the response of any of the encryption queries. The challenger then computes $m_j \leftarrow D(k, c_j)$ and returns the decrypted message to the adversary. The adversary can initiate as many of these queries as necessary in any order. Then at the end of the game the adversary computes the value of b. $\mathcal{A}$'s advantage with respect to $\mathcal{E}$ is

$$CCAadv[\mathcal{A}, \mathcal{E}] = |P_r[W_0] - P_r[W_1]|, \qquad (2)$$

where $P_r[W_b]$ is the probability that $\mathcal{A}$ calculates 1 given the challenger has selected b at the beginning of the game.

Now we define a pseudo random number generator attack game. Given a pseudo random number generator $\mathcal{G}$ defined over $(\mathcal{S}, \mathcal{R})$. The game starts with the challenger randomly selecting a binary value $b \in \{0,1\}$. If $b = 0$ the challenger generates a random seed $s \overset{R}{\leftarrow} \mathcal{S}$ and bit stream $\mathcal{G}(s) \in \mathcal{R}$ and sends it to the adversary $\mathcal{A}$. Otherwise if $b = 1$ then the challenger creates a truly random bit stream $r \overset{R}{\leftarrow} \mathcal{R}$ and sends it to the adversary $\mathcal{A}$. Finally the adversary computes the value of b. $\mathcal{A}$'s advantage with respect to $\mathcal{G}$ is

$$PRGadv[\mathcal{A}, \mathcal{G}] = |P_r[W_0] - P_r[W_1]|. \qquad (3)$$

To evaluate the total system security we must understand the most likely attacks. There are at least four primary attacks on this system. The first is replacement of the sensor with a bogus device, the second is the replacement of the ASIC with a bogus device, the third is a passive eavesdropping (EA) attack between the sensor and ASIC and the last is the active man in the middle attack (MITMA) between the sensor and ASIC. Note that a passive attack can only read the transmitted data between the sensor and the ASIC, while an active attack can both read and write the transmitted data.

Security against sensor replacement is based on the difficulty of forging a valid certificate (from a trusted source) with an associated secret key. This is determined by the security of the public key signing algorithm used for the certificate such at DSA [4] or ECDSA [5] and the security of the secret key associated with the certificate (stored in ROM).

There is no cryptographic security against replacement of the ASIC by an attacker in this system. The sensor will send data to any receiver that can negotiate a valid connection. Making it physically difficult to replace the ASIC is the only level of security.

There is no cryptographic security against EA in this system. Therefore, image sensor data can be freely collected by an attacker and used for any nefarious purpose.

Security against data modification by a MITMA depends not only on the MAC security, but also the security of the secret key. The security of the secret key is function of the random number generator in the ASIC, the security of the asymmetric encryption algorithm and side channel based key recovery security of the sensor and the ASIC. Using the Union Bound for probabilities the MITMA security can be bounded by

$$\begin{aligned} &MITMAadv[\mathcal{A}, \mathcal{I}, \mathcal{E}_{pk}, \mathcal{G}, \mathbb{S}, \mathbb{C}] \leq \\ &MACadv[\mathcal{B}_{\mathcal{I}}, \mathcal{I}] + CCAadv[\mathcal{B}_{pk}, \mathcal{E}_{pk}] + \\ &PRGadv[\mathcal{B}_{\mathcal{G}}, \mathcal{G}] + SCAadv[\mathcal{B}_{\mathbb{S}}, \mathbb{S}] + \\ &SCAadv[\mathcal{B}_{\mathbb{C}}, \mathbb{C}], \end{aligned} \qquad (4)$$

where $MACadv[\mathcal{B}_{\mathcal{I}}, \mathcal{I}]$ is the advantage of the MAC $\mathcal{I}$. Examples of $\mathcal{I}$ include HMAC [6], CMAC [7] or GMAC [8]. $CCAadv[\mathcal{B}_{pk}, \mathcal{E}_{pk}]$ is the chosen cipher text advantage of the public key encryption algorithm $\mathcal{E}_{pk}$. Examples of $\mathcal{E}_{pk}$ include RSA [9] or ECC [10]. $PRGadv[\mathcal{B}_{\mathcal{G}}, \mathcal{G}]$ is the advantage of the pseudo random number generator $\mathcal{G}$. Examples of $\mathcal{G}$ include Salsa20 [11], ChaCha20 [12] or a true random number generator [13]. $SCAadv[\mathcal{B}_{\mathbb{S}}, \mathbb{S}]$ and $SCAadv[\mathcal{B}_{\mathbb{C}}, \mathbb{C}]$ are the advantages of side channel based key recovery attacks against the image sensor $\mathbb{S}$ and ASIC $\mathbb{C}$ respectively. Each $\mathcal{B}_x$ is a sub adversary of $\mathcal{A}$. Therefore this system can only be secure against MITMA if all of the pixels in each image are MACed, the MAC algorithm is secure, the public key encryption algorithm is secure, the random number generator is secure and the image sensor and ASIC are secure against side channel attacks.

Penetration testing is a critical part of cyber security system design. This process enables designers to empirically determine $SCAadv[\mathcal{B}_{\mathbb{S}}, \mathbb{S}]$ and $SCAadv[\mathcal{B}_{\mathbb{C}}, \mathbb{C}]$ as functions of the amount of data encrypted. Therefore, bounding how often the secret keys must be updated to achieve a desired level of security.

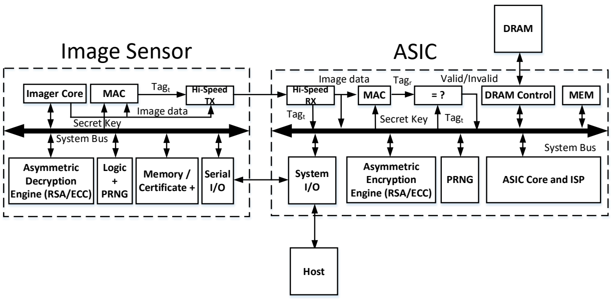

Figure 1

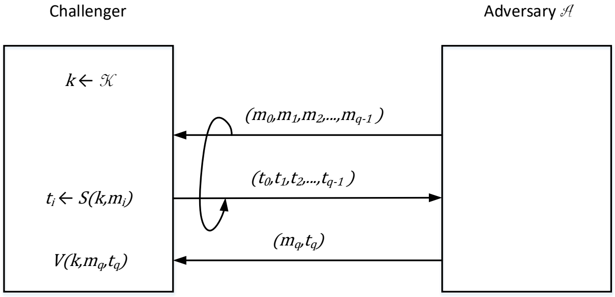
Challenger
Adversary $\mathcal{A}$
$k \leftarrow \mathcal{K}$
$(m_0, m_1, m_2, \ldots, m_{q-1})$
$(t_0, t_1, t_2, \ldots, t_{q-1})$
$t_i \leftarrow S(k, m_i)$
(m_q, t_q)
$V(k, m_q, t_q)$

Figure 2

III. Next Generation Cyber Security

Many of the drawbacks of the current cyber security system described in Section II can be corrected using the system shown in Figure 3. In Figure 3 the image sensor is connected to the ASIC in the same manner as the previous system, but it incorporates symmetric certificate exchange between the sensor and the ASIC. In addition, authenticated encryption (AE) [14] is used for all of the image sensor data and the command data between the image sensor and the ASIC. This enables both security and integrity of the data to and from the image sensor. The improved boot-up sequence between the image sensor and the ASIC starts by having the ASIC and image sensor both exchange certificates. After both of the signatures of the certificates are validated, using global public keys, the public keys from each certificate are loaded into the respective asymmetric encryption hardware. Next the image sensor and the ASIC randomly create secret key materials for the authenticated encryption blocks which are then encrypted using the respective public keys from the exchanged certificates. The encrypted secret key materials are shared between the image sensor and the ASIC. Then both the image sensor and the ASIC combine the key materials to create the final secret key(s). This can be done using a collision resistant hash function [15], such as SHA-256 [16] or using an elliptic curve multiplication depending on the algorithm used to exchange the key materials. Finally the hashed secret keys are loaded into the AE hardware and the encryption process for the image data and the command stream begins. Note that the secret keys used for image data and command data must be separate (therefore we need 4 separaete secret keys).

The proposed system in Figure 3 is not without limitations. First it requires significantly more processing logic and power than the system described in Section II. In addition it also makes command communication between the sensor and the ASIC much more complex. Usually commands between a sensor and an ASIC are a few bytes, but using AE for command data security makes the smallest package size about 48 bytes for a 16 byte or less command. This is because AE algorithms require that each transmission include a NONCE, cyber text and a tag. Typically the NONCE, cyber text and tag are at least 128 bits. The longer the command the lower the overhead, but this is a significant cost for security.

Just like in Section II there are four primary attacks on this system. The first two are replacement of the sensor or the ASIC by an attacker. Security against sensor or ASIC replacement is based on the difficulty of forging a valid certificate (from a trusted source) with an associated secret key. This is determined by the security of the public key signing algorithm used for the certificate and the security of the secret key associated with the certificate.

Security against EA in this system is based on the security of the symmetric encryption algorithm used as a part of AE and the ability of the system to keep the secret keys safe. Since this system uses bi-directional certificates and both the image sensor and the ASIC create parts of the secret keys, the probability of breaking the cipher text from the sensor and from the ASIC is very low. In addition, since both the image sensor and the ASIC create part of the key materials, even if the entropy of the PRGs is low such as ½ bit per bit, after the key materials are combined (using a cryptographic hash like SHA-256) the total entropy of the final secret keys should be very close to 1 bit per bit. Again using the Union Bound we find the EA advantage of the system

$$EAadv[\mathcal{A}, \mathcal{E}_{sk}, \mathcal{E}_{pk}, \mathcal{G}, \mathbb{S}, \mathbb{C}] \leq CCAadv[\mathcal{B}_{\mathcal{E}_{sk}}, \mathcal{E}_{sk}] + CCAadv\left[\mathcal{B}_{\mathcal{E}_{pk}}, \mathcal{E}_{pk}\right]^2 + PRGadv[\mathcal{B}_{\mathcal{G}}, \mathcal{G}]^2 + SCAadv[\mathcal{B}_{\mathbb{S}}, \mathbb{S}] + SCAadv[\mathcal{B}_{\mathbb{C}}, \mathbb{C}], \tag{5}$$

where $CCAadv[\mathcal{B}_{\mathcal{E}_{sk}}, \mathcal{E}_{sk}]$ is the advantage of the symmetric encryption algorithm used in the AE, such as GCM [8].

Just as in Section II security against MITMA depends not only on the MAC security of the AE algorithm, but also the security of the secret keys. The security of the secret keys is a function of the random number generator in the sensor and the ASIC, the security of the asymmetric encryption algorithm and side channel attack security of the sensor and the ASIC. Using the Union Bound for probabilities the MITMA security can be bounded by

$$MITMAadv[\mathcal{A}, \mathcal{I}, \mathcal{E}_{pk}, \mathcal{G}, \mathbb{S}, \mathbb{C}] \leq MACadv[\mathcal{B}_{\mathcal{I}}, \mathcal{I}] + CCAadv\left[\mathcal{B}_{\mathcal{E}_{pk}}, \mathcal{E}_{pk}\right]^2 + PRGadv[\mathcal{B}_{\mathcal{G}}, \mathcal{G}]^2 + SCAadv[\mathcal{B}_{\mathbb{S}}, \mathbb{S}] + SCAadv[\mathcal{B}_{\mathbb{C}}, \mathbb{C}]. \tag{6}$$

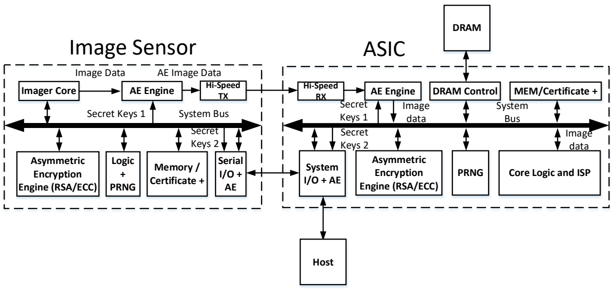

Figure 3

IV. Discussion

Bidirectional authentication and key exchange help to significantly improve the security of the proposed system. First bidirectional certificate exchange validates the identity of the components on both sides of the communication link. Then it reduces the security requirements for both the key generation process in the sensor and in the ASIC. It also reduces the security requirements for the asymmetrically encrypted key materials sent between the devices. This is true because an attacker needs both the sensor and the ASIC key materials to recover the final secret key(s) used for AE.

Since encryption, message authentication and digital signatures are brittle to even a single bit error, data transmission errors cannot be distinguished from cyber-attacks. For example, an automotive image sensor can generate 10^{13} bits/hour, but if the bit error rate in a given video transmission channel is on the order of 10^{-12} then there will be multiple corrupted frames per hour. In order to mitigate this problem, some type of error correction code is required in the final system for at least the video data (note that the command data rate is much lower than the video data). For example a (18,16) Solomon Reed [17] code, i.e. a code with 18 bytes per block including 2 parity bytes, can correct up to one byte per block and detect up to two byte errors per block. If the data transmission error rate is 10^{-12}, and the errors are assumed to be independent, then the probability of having a block that has uncorrected errors, assuming a (18,16) Solomon Reed coded channel, is

$$\begin{aligned} &P_r(> 0 \text{ bit errors in a byte}) \\ &\qquad = 1 - (1 - 10^{-12})^8 = p' \end{aligned} \tag{7}$$

$$\begin{aligned} &P_r(> 1 \text{ bytes in a block have bit errors}) \\ &= 1 - \sum_{i=0}^{1} \binom{18}{i} (1-p')^{18-i} (p')^i \\ &\qquad = 2 * 10^{-20}. \end{aligned} \tag{8}$$

This would increase the expected time between uncorrected errors to

$$\begin{aligned} &\frac{128}{(10^{13} * 2 * 10^{-20}) * 24 * 365.25} \\ &\qquad > 73K\ years. \end{aligned} \tag{9}$$

Using error correction codes clearly reduces the error rate to an acceptable level, but it also increases the computation, power dissipation and chip size.

Cyber security is always a tradeoff between computation and performance. This means that enhanced security increases power dissipation, silicon area and system cost. Therefore, understanding the key attack scenarios, attack consequences, and mitigations is critical for optimizing the system for a required security level.

V. Conclusions

We have shown that current cyber security systems in image sensors can be insecure under certain conditions. These conditions include, eavesdropping attacks between the source and destination, not MACing all of the data in a message, control of the sensor or ASIC and key recovery attacks. We have proposed a next generation cyber security system that tries to mitigate most of the current generation's problems using symmetric certificate exchange and symmetric secret key material exchange in addition to adding authenticated encryption to both the image and command data channels.

Although the proposed cyber security architecture is more secure than current systems, it is still sensitive to key recovery attacks especially against the static private keys associated with the certificates. To further improve security an online certificate status protocol (OCSP) [18] interface could be implemented in the system to check the validity of all of the component certificates. If a given component certificate is invalid then data from that component would also be considered invalid in the system. Finally the proposed cyber security architecture does not include error correction which is necessary to detect active attacks on the system.

References

[1] D. Boneh and V. Shoup, "A Graduate Course in Applied Cryptography", http://crypto.stanford.edu/~dabo/cryptobook/bonehshoup_0_4.pdf.

[2] R. Spreitzer et al., "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices*", IEEE Communications Surveys & Tutorials, Vol. 20, No. 1, 2018.

[3] https://en.wikipedia.org/wiki/X.509.

[4] M. Bellare et al., "The Exact Security of Digital Signatures-How to Signwith RSA in Rabin", Advances in Crytology – EUROCRYPT '96, EUROCRYPT 1996, Lecture Note in Computer Science, vol. 1070, Springer, Berlin, Heidelberg.

[5] D. Johnson et al., "The Elliptic Curve Digital Signature Algorithm (ECDSA), IJIC 1, 36-63 (2001).

[6] P. Gauravaram, S. Hirose and S. Annadurai, "An Update on the Analysis and Design of NMAC and HMAC Functions", International Journal of Network Security, Vol.7, No.1, PP.49–60, July 2008.

[7] C. Baritel-Ruet, F. Dupressoir, P. Fouque and B. Gregoire, "Formal Security Proof of CMAC and its Variants", 2018 IEEE 31st Computer Security Foundations Symposium.

[8] A. Delignat-Lavaud et al., "Implementing and Proving the TLS 1.3 Record Layer", 2017 IEEE Symposium on Security and Privacy.

[9] R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM, Feb. 1978, 21(2): 120-126.

[10] V. Kapoor et al., "Elliptic Curve Cryptography", ACM Ubiquity, Volume 9, Issue 20, May 20 – 26, 2008.

[11] D. J. Bernstein, "The salsa20 family of stream ciphers," eSTREAM, ECRYPT Stream Cipher Project, Report 2005/025, 2005, http://www.ecrypt.eu.org/stream.

[12] D.J. Bernstein, "Chacha, a variant of salsa20," Jan. 2008, http://cr.yp.to/chacha.html.

[13] B. Sunar et al., "A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks", IEEE Transactions on Computers, vol. 56, no. 1, January 2007

[14] M. Bellare et al., "Authenticated encryption in SSH: Provably fixing the SSH binary packet protocol", ACM Conference on Computer and Communications Security (CCS-9) (2002), ACM Press, pp. 1–11.

[15] https://en.wikipedia.org/wiki/Collision_resistance

[16] NIST/NSA, "FIPS 180-2: Secure Hash Standard (SHS)", August 2002 (change notice: February 2004).

[17] S. B. Wicker and V. K. Bhargava, "*Reed-Solomon codes and their applications*", John Wiley & Sons, 1999.

[18] https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol