Attacks and Countermeasures on AES and ECC

Henrik Tange
Center for Wireless Systems and Applications / CTIF-Copenhagen
Technical University of Denmark
DTU Ballerup Campus
DK-2750 Ballerup, Denmark
heta@ihk.dk

Birger Andersen
Center for Wireless Systems and Applications / CTIF-Copenhagen
Technical University of Denmark
DTU Ballerup Campus
DK-2750 Ballerup, Denmark
bia@ihk.dk

Abstract: AES (Advanced Encryption Standard) is widely used in LTE and Wi-Fi communication systems. AES has recently been exposed to new attacks which have questioned the overall security of AES. The newest attack is a so called biclique attack, which is using the fact that the content of the state array is foreseeable while the rounds are performed. ECC (Elliptic Curve Cryptography) is used as a public key crypto system with the key purpose of creating a private shared between two participants in a communication network. Attacks on ECC include the Pohlig-Hellman attack and the Pollard's rho attack. Furthermore side-channels attacks can be applied to ECC. This paper reflects an ongoing research in the field of countermeasures against the attacks mentioned above.

Keywords: AES, ECC, attacks, countermeasures.

I. INTRODUCTION

AES (Advanced Encryption Standard) is used widely as symmetric encryption scheme in LTE and Wi-Fi. In LTE it is part of the EPS [1, p. 168] (Evolved Packet System) cryptography algorithms and is preferred over KASUMI which is widely used in 2G and 3G. In Wi-Fi AES is used in WPA2 (Wi-Fi Protected Access II)[2].

AES was announced by the NIST (National Institute of Standards and Technology) the 26th of November 2001 (FIPS-197). AES is created by Joan Daemen and Vincent Rijmen.
AES does not use a Feistel network. A Feistel network is using the same operations for encryption and decryption. AES has special algorithms for encryption and decryption. The AES algorithm uses a fixed block size of 128 bits and different key sizes of 128, 192 or 256 bits. The four operations are: AddRoundKey, SubBytes, ShiftRows and MixColumns. The use of four operations follows a well-known described scheme in the main algorithm consisting of rounds: In the initial round AddRoundKey is performed. In the following rounds (call them center-rounds) SubBytes, ShiftRows, MixColumns and AddRoundKey are performed. In the last round only SubBytes, ShiftRows and AddRoundKey are performed. If the key size is 128 bits 10 center-rounds are executed, if the key size is 192 bits the number of center-rounds is 12 and finally if the key size is 256 bits the number of center-rounds is 14.
There exists no mathematical based proof for the security of AES. AES is as described above iterating over a SPN (Substitution - Permutation Network) structure. The AES algorithm has been verified using the Mizar Proof Checker [3].

ECC (Elliptic Curve Cryptography) is a fairly new public key cryptography system. ECC can for instance be used in small radio systems because of its small footprint and because a high security level can be achieved using smaller keys. It can be proved that ECC, which is based on the discrete logarithm problem, can offer the same level of security as a 1024-bit RSA key[4] with a less than 200-bit elliptic curve key. The footprint can be very small: For example has ECC been implemented on the micaZ platform from Crossbow with only 128 kB of flash memory and 4 kB of RAM [5]. Elliptic curves are algebraic/geometric curves. These curves have been studied extensively for the past 150 years. They have also been used in a for instance number theory and in the proof of Fermat last theorem. Elliptic Curves for cryptography was discovered in 1985 by Victor Miller and Neil Koblitz. As stated above, ECC is used as public key cryptography. In public key cryptography, each user have a public known base point on the curve and a private key which is used to create a common shared - also known as a private symmetric key. The security of the ECC is based on the elliptic curve discrete logarithm problem (ECDLP).

The discrete logarithm problem is described as follows: "Let P and Q be two points on an elliptic curve that KP = Q, where K is a scalar. Given P and Q, it is computationally infeasible to obtain K, if K is sufficiently large. K is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication i.e. multiplication of a scalar K with any point P on the curve to obtain another point Q on the curve [6, p.12].
Koblitz Curves with TNAF (τ -adic Non-adjacent Form) [6, p. 116] are very often used in ECC systems. A Koblitz curve (Anomalous Binary Curve) E_a is an elliptic curve which is defined over F_2^m, possessing some special properties making it computationally easier to make the point multiplication explained above. The general Koblitz curve is described by the equation:

$$E_a : y^2+xy = x^3+ ax^2+b \text{ [6,p. 114],}$$

This paper reflects an ongoing research project in the field of countermeasures against attacks on AES and ECC.

II. ATTACKS ON AES

It is well known that brute force attacks on AES are not considered as the biggest problem regarding security while breaking a 256 bit key would take more than 2^{200} operations. One possible attack on AES is the side-channel attack. Side-channel attack is based on the knowledge of the algorithm implementation and measurements of for instance power consumptions or timing. A side channel attack is not an attack on the mathematical background for AES but an attack on the implementation of the cipher or the knowledge of the main algorithm. The side channel attack concerns about measurements like power consumptions, timing information or electromagnetic leaks. D. J. Bernstein has described cache-timing attacks on AES [7]. He is using an exact copy of hardware and software as used by the victim. In this attack the time for a look-up in the S-Box is measured. The time used for this look-up is actually depending on the array index. The time is a function of the array indexing which leads to an easy way to back substitute the look-up. Therefore AES leaks information about the key and the key can be deduced exactly. In summary the biggest problem first of all is that AES is fully predictably – it is well known how key scheduling is performed. Second, the problem is that a static S-Box is used.

Another well-known attack is the related-subkey attack. The related-subkey attack is described in Advances in Cryptography – EUROCRYPT 2010 by Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir [8]. In this article an 11 round attack on the AES-256 code is done by:

1. The attack starts with an odd round. The attack looks at the output difference between three active S-Boxes in round 10. Data and time complexity is described to be 2^{70}.
2. One more active S-Box is restricted making it easier to discard the wrong pairs.
3. Now start with an even round and look at the output difference between three active S-Boxes in round 9.
4. Start from an even round and restrict two S-Boxes.

In the conclusion the problem is well described: The key schedule does not have "industrial strength" and the initial key is not sufficiently mixed. Also it is concluded that the key schedule algorithm is too linear causing "unusually long key differentials of probability 1" [8, p. 314]. Another problem is that it is possible cancel data differences with key differences over a set of rounds.

A newer, and more sophisticated, attack is known as the biclique attack[9]. The biclique refers to bipartite graph. This attack type is based on the "Meet-in-the-middle" attacks. The attacker chooses a key space partition and places it into groups of keys with cardinality 2^{2d}. The key is indexed as an element into a 2^d x 2^d matrix: K[i,j]. From the data transformation of the cipher a variable V can now be chosen such that:

$$P \xrightarrow{K[i,-]} V$$

This is a function of the plaintext and a key identical for all keys in a row. For all keys in a column the function

$$V \leftarrow \xrightarrow{K[-,j]} C$$

is selected. This is a function of the ciphertext C and a key for all keys in a column. The parts $f_1$and f_2 corresponds to the same part of the ciphertext.
Now having this pair (P,C) the attacker can now compute 2^d possible values of V← and →V from the plaintext part and the ciphertext part. The "Meet-in-the-middle" attack is more effective than the brute force attack with a factor of 2^d.
The background of biclique attack is defined as follows: In AES a number of keys K[I,j] will be calculated in the key schedule function. At any time during encryption algorithm the state will have 2^d internal states S. The ciphertext C_i can be seen as a function of a key K[I,j] and a specific state S_i[9, p.12]. The attacker forms a set of 2^{2d} keys from the key space and regards the block cipher as a combination of two sub cipher as seen in figure 1.

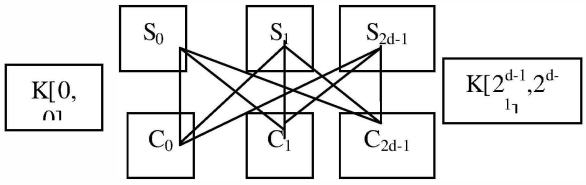

Fig. 1

A data transform of a cipher consists of two parts:

1. The attacker constructs a structure of 2^d ciphertext parts C_i and also 2^d intermediate states S_j in connection with the key group K[I,j]. Then a partial decryption of C_i results in S_j given K[I,j].
2. The attacker uses an oracle to decrypt ciphertext C_i with the key K_{secret} . If K_{secret} is found in K[I,j] the state S_j maps to the plaintext P_i which propose a key candidate.

Round transformation of the AES is not designed to have strong resistance against several classes of attacks for a smaller number of rounds. The splitting of the cipher into three parts makes the AES weak. Another problem is that the lack of execution the MixColumn transformation method in the last round is useful for the design of attacks for more

rounds. Also the key schedule transformation makes it easier to attack the AES taking advantage of the relatively slow backwards diffusion.

III. ATTACKS ON ECC

The security of ECC is based on ECDLP as described above. The elliptic curve parameters have to be carefully chosen to resist attacks. With a proper set of parameters the exhaustive search (simple computation of points {P, 2P, 3P, *4P, ..., nP*} will use n/2 steps in average. With a large n it will take a great amount of computations, which is not considered as possible threat.

There are two well-known attack types: The Pohlig-Hellman attack and the Pollard's rho attack. The Pohlig-Hellman attack is built on the the prime factorization of

$$n = p_1^{e1}, p_1^{e2}, p_2^{e3}, p_n^{en} \quad .$$

The idea is to compute $l_i = l \bmod p_i^{ei}$ for $1 \leq l \leq r$. Now a system of congruences can be solved using the Chinese Remainder Theorem:

$$\begin{aligned} l &= l_1 \ (mod\ p_1^{e1}) \\ l &= l_2 \ (mod\ p_2^{e2}) \\ &\vdots \\ l &= l_n \ (mod\ p_n^{en}) \end{aligned}$$

The Pollard's rho attack is based on the idea of finding two pairs (*c',d'*) (*c'',d''*) of the integers modulo n such:

$$c' P + d'Q = c''P + d''Q$$

This equation can be reordered:

$$(c'-c'') P = (d''-d') Q = (d'-d'') lP \text{ and}$$

$$(c'-c'') \quad (d''-d') \ l \ (mod\ n)$$

Because $l = \log_P Q$, l can be found:

$$l = (c'-c'')(d''-d')^{-1} \bmod n$$

The expected number of iterations in order to find is $\sqrt{\pi n}/2$.

Also a parallelized Pollard's rho attack exists concerning with attacks performed on a server with M processors. The expected numbers of elliptic curve operations, performed by each processor, is calculated to be $3\sqrt{n}/M$ [6, p. 160]. The expected speedup is a factor of $\sqrt{M}$ for prime elliptic curves and $\sqrt{2M}$ for Koblitz curves compared to the normal Pollard's rho attack. For 113 bits ECDLP instance the expected running time is calculated to be 1045 days [6, p. 164].

Another group of attacks are isomorphism attacks. Isomorphism attacks are reducing Elliptic Curve Discrete Logarithm Problem to a simple discrete logarithm problem thereby making it a simpler attack type. In fact if E is an elliptic curve defined over a finite field F_q and $P \in EF_q$ having the prime order n. Now define G as a group of order n. Because n is a prime P and G are both cyclic and isomorphic. The isomorphism can be calculated:

$$\varphi: (P) \rightarrow G$$

The ECDLP of P can now be reduced to DLP in G, given P and:

$$\log_P Q = \log_{\varphi(P)} \varphi(Q)$$

At the time three kinds of isomorphism attack types are known[6, p.168]:

1. Attack on prime-field-anomalous curves
2. Weil and Tate pairings attacks
3. The Guadry-Hess-Smart (GHS) Weil descent attacks

The attack on prime-field anomalous curves can be performed since the group $E(F_p)$ is cyclic and $E(F_p)$ is isomorphic to the additive group F_p^+. By using the extended Euclidean algorithm it is possible to compute $a^{-1} \bmod p$ given $l = ba^{-1} \bmod p$.

The Weil and Tate pairings attack is based on the Weil pairing attack that constructs an isomorphism from $(P)\ to\ G$with the additional constraints, $!\ n\ /\ (q-1)$ while the Tate pairing attack constructs an isomorphism without the constraint. For elliptic curves with a small embedding degree k the Weil and Tate pairing reduces the solving to a sub-exponential-time problem.

The Weil descent attack is based on fact that the Weil restriction $W_{K/k}$ can be intersected with n-1 hyperplanes and thereby reduce the ECDLP to an instance of Hyper Elliptic Curve Discrete Logarithm Problem (HCDLP). The HCDLP yields a sub-exponential-time algorithm.

Also invalid- curve attacks are known in ECC. According to the Group Laws for

$$\{E|K\}: y^2 = x^3 + ax + b$$

and

$$\left\{ E\middle|F_{2^m}\right\}: y^2 + xy = x^3 + ax + b \ ,$$

defined as Elliptic curves in the simplified Weierstrass form, Point Addition does not involve the coefficient b. If *E'* is an elliptic curve which only differ from an elliptic curve E in the coefficient b the addition laws for both are the same. This can result in an invalid curve attack. The problem will arise if the receiver (A) does not calculate that the point is on the curve. The attacker can select an invalid curve *E'* where $E'(F_q)$ contains a point Q_B of small order l and send this point to A. A computes $Q_A = dQ_B$, where d is the private key of A. B can now find $d_l = d \bmod l$. Repeating this with other invalid curves B can extract d.

Side-channel attacks can also be performed on ECC[6, p.238]. Side-channel attacks utilize information which is leaked during operation. This group of attacks is named power analysis attacks (measuring the power consumed during operation). There are two well known ways of performing power analysis attacks, namely Simple Power Attacks (SPA)[6, p.240] and Differential Power Attacks (DPA)[6, p.242]. In SPA it is possible to deduce secret key material by a simple power measurement. In fact a private key used in TNAF can be deduced by measure the related action. In DPA the system is attacked by attacking using a selection function and the attacker collects a sample of measurements and calculates the difference of averages – also called a differential trace.

IV. COUNTERMEASURES

In AES side-channel (timing) attacks are a problem. One solution could be a constant-time implementation of AES. Another solution is to remove the predictability in AES. The timing attack is only possible due to a well-known key scheduling and the use of static S-Boxes. Removal of the static S-Box could prevent timing attacks along with a private key dependent key scheduling algorithm.

The related-subkey attack is made possible due to an insufficient mix of the private key. Also the problem that it is possible cancel data differences with key differences over a set of rounds is a severe problem. A solution is to change the key scheduling algorithm to provide a more careful mix of the keys. This can be done by making the key scheduling algorithm dependent of the actual private key.

The biclique attack uses the advantage that the round transformation in AES is not designed to resist an attack where the cipher can be split up in several parts. The security of AES does not rely on a mathematical proof. AES is secured by permutation and substitution. The design flaw is that AES is based on a static main algorithm, where the run through is foreseeable. Bogdanov, Khovratovich and Rechberger concludes [9, p. 27] that especially the fact that the MixColumn transformation is omitted in the last round of the AES implementation is helpful for attacks against AES. They also conclude that the key schedule algorithm is suffering of relatively slow backward diffusion. The most obvious solution to protect against this attack type is to change the static implementation of the AES main algorithm. This can be done in several ways. One possibility is to provide a – secret - knowledge about a dynamic implementation of the AES main algorithm. Another possibility is to make the main algorithm flow dependent of the private key. Because the security of AES relies on the number of permutations and substitutions – not on the order of execution of ShiftRows, MixColumn, SubBytes and AddRoundKey - this could be a possible solution. Regarding the key scheduling algorithm this algorithm also can be made dynamic instead of the static key scheduling in the AES. This can be done by introducing a key scheduling algorithm depending on another secret input.

In ECC the Pohlig- Hellman attack is as mentioned a well-known attack type. A solution to this attack is to choose elliptic curve parameters in a way that n is divisible with a prime number p very large so $O\sqrt{p}$ where p is the largest prime divisor of n.

The Pollard's rho attack in ECC can again be avoided by a large chosen key that will make this attack method infeasible. The ECC standard implementation allows for instance a Koblitz curve with the size of 571 bits.

The isomorphic attack group contains: The prime-field anomalous curves attack, the Weil and Tate pairing attacks and the Weil descent attack. The attack on prime-field anomalous attack can be prevented by checking whether $\#E(F_p) = p$. In the case of Weil and Tate pairing attacks it must be ensured that the embedded degree k > 6 [6, p. 169]. For an elliptic curve E defined over F_P it is sufficient to check that the order of base point n does not divide q^k -1 for all small k. The Weil descent attack can be avoided by using elliptic curves over F_{2^m} where m is composite, for example $F_{2^{169}}$.

Invalid curve attack can be avoided by implementing an algorithm that checks that the received point does lie on the expected elliptic curve.

Side-channel attack is also a severe problem in ECC since both NAF and TNAF algorithms represent the private key k as a sequence $\{0,\pm 1\}$. It is important that all operations are hidden for measurements as for instance SPA or DPA. This can be done by optimize the code as much as possible and at the same time add some dummy operations where the price for execution time is low. Another possibility is to effective shield the hardware.

V. CONCLUSIONS

The countermeasures above can be divided into two groups: Some of the attack types can be prevented by a careful implementation and some attack types can only be prevented by a change in algorithms.

In AES the problems are centered on a weak implementation of the key scheduling algorithm and the use of static S-Boxes. A solution mentioned in this paper is to remove the static S-Box and instead use a dynamic implementation. The biclique attack in AES shows that the static implementation and thereby foreseeable execution is a major problem in AES. This paper gives some proposals for changes to the AES algorithm to prevent the biclique attack. In our ongoing research we are following the approach to make the AES main algorithm and the key scheduling algorithm dynamic. The main research is on changing the order of permutation and substitution sub-algorithms according to a dynamic variable.

In ECC a great part of the attacks can be avoided by a careful implementation and careful selection of the domain

parameters. A common problem for AES and ECC is the side-channel attack types. In software implementations it is important level out the power consumptions caused by the different algorithms. In ECC the main research area is side-channel attacks.

Research results are expected to be available for presentation at the conference.

REFERENCES

[1] Dan Forsberg et al., "LTE Security", Wiley, 2010.

[2] A.S.Rumale, D. N. Chaudhari, "IEEE 802.11x, and WEP, EAP,WPA/WPA2", IJCTA NOV-DEC 2011.

[3] Hiroyuki Okazaki, Kenichi Arai, and Yasunari Shidama, "Formal Verification of AES Using the Mizar Proof Checker", Shinshu University / Tokyo University of Science, Technische Universität Freiberg, 2011, http://elrond.informatik.tu-freiberg.de/papers/WorldComp2012/FCS3246.pdf

[4] Michael Rosing, "Implementing Elliptic Curve Cryptography", Manning, 1999.

[5] Samant Khajuria, Henrik Tange, "Implementation of Diffie-Hellman Key Exchange on Wireless Sensor Using Elliptic Curve Cryptography", Wireless Vitae 2009.

[6] Hankerson, Menezes and Vanstone, "Guide to Elliptic Curve Cryptography", Springer, 2004.

[7] Daniel J. Bernstein, "Cache-timing attacks on AES", Department of Mathematics, Statistics, and Computer Science, The University of Illinois at Chicago, 2005.

[8] Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir, "Key Recovery Attacks of Practical Complexity on AES-256 Variants with Up to 10 Rounds", Advances in Cryptography – EUROCRYPT 2010, LNCS 6110, Springer, 2010.

[9] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger, "Biclique Cryptanalysis of the Full AES", ASIACRYPT'11, 2011.