June 29, 2005

Title: Galois MAC with forgery probability close to ideal

Source: Kaisa Nyberg, Nokia Research Center
Henri Gilbert and Matt Robshaw, France Telecom R&D

1.Truncation and Short Tags

In this note we would like to address an issue brought into discussion in [F05]. In their response [MV05] the authors successfully defended their design against Ferguson's concerns. However, there is one issue which still remains: for longer messages the forgery probability of Galois MACs is far from being ideal. In the GCM specification [MV04] this problem is taken care of by setting an upper bound to the message length and discouraging the use of tags less than 64 bits long. While this removes the problem in practice, the range of application of GCM mode will be limited and the theoretical problem remains unsolved.

An example of a practical system where the tag length is set to 32 bits is the 3GPP UMTS system for the third generation mobile networks. This year 3GPP initiated a project [E05] to design a new set of encryption and integrity algorithms for UMTS. For many reasons, Galois MAC construction was considered an attractive solution for the new message integrity function. The only problem was that the forgery probability grows as the messages get longer. The message length in the UMTS system is currently upper-bounded by 20 000 bits, but longer messages may need to be allowed in the future. Using the upper bound of 2^{16} for the message length, computing in GF(2^{64}) and truncating to 32-bit MAC, the forgery probability becomes 2^{-22}. This means an unacceptable degradation in security compared to other more traditional computationally secure MAC constructions. Note that the forgery probability is only doubled, if the MAC is computed in the Galois field GF(2^{32}). With the current construction, the higher security offered by the larger field is lost in truncation.

This drawback can be avoided using a two-stage MAC construction. The idea is to concatenate two universal hash families as proposed by Stinson [S92], Bierbrauer et al [BJKS93] and Nevelsteen and Preneel [NP99]. Their technique was recently referred to as "secure truncation" by D. Bernstein in [B05].

2. A Two-Stage MAC

Let us recall the definitions of ε-AXU and ε-AU hash families.

Definition: [NP99] Let $H = \{h_k : A \rightarrow B, k \in K\}$ be a family of hash functions mapping elements of set A to set B. The family H is ε-Almost-Xor-Universal **(ε-AXU)** if,

$$\forall\, x, x' \in A, \quad x \neq x' \;\Rightarrow\; \forall \delta \in B, \quad \mathbf{Pr}_k\,\{h_k(x) \oplus h_k(x') = \delta\} \leq \varepsilon.$$

If the condition holds only for $\delta = 0$, that is,

$$\forall\, x, x' \in A, \quad x \neq x' \;\Rightarrow\; \mathbf{Pr}_k\,\{h_k(x) \oplus h_k(x') = \; 0\} \leq \varepsilon,$$

the family H is called ε-Almost-Universal **(ε-AU).**

To compute a 32-bit MAC we adopt a two-stage construction. In the first phase we work over GF(2^{64}) and we use an ε_1-AU hash function family with longer hash codes. In stage two we use an ε_2-AXU hash function family, with shorter hash codes.

There is a composition theorem due to Stinson [BJKS93, S92], which allows different universal hash function families to be combined. The most useful result for us is the following:

> *If there exists an ε_1-AU family H_1 of hash functions from A to B and an ε_2-AXU family H_2 of hash functions from B to C, then there exists an ε-AXU family H of hash functions from A to C where $H = H_1 \times H_2$, and $\varepsilon = \varepsilon_1 + \varepsilon_2 - \varepsilon_1\varepsilon_2 \leq \varepsilon_1 + \varepsilon_2$.*

For the MAC construction, we use an $L{\cdot}2^{-64}$-AU hash function family in the first stage, where L is the length of the message in 64-bit blocks (after padding and length appending). In the second stage we have a 2^{-32}-AXU hash function family and so our two-stage construction provides an ($L{\cdot}2^{-64} + 2^{-32}$)-AXU hash function family, that is, an 2^{-31}-AXU hash function family.

3. Practical Instantiation

The authentication function of GMAC is a practical instantiation of the Wegman-Carter MAC. Let $H = \{h_k : A \rightarrow B,\ k{\in}K\}$ be a family of hash functions mapping elements of set A to set B. Then the Wegman-Carter MAC associated with H is defined as follows:

- Key inputs: Index $k{\in}K$ and a one-time value $k' \in B$.
- Data input: Message x from A.
- Output: MAC tag $\tau \in B$ where $\tau = h_k(x) \oplus k'$.

Then by [S96, NP99] for example, it is known that if the family of hash functions $H = \{h_k : A \rightarrow B,\ k{\in}K\}$ is ε-AXU, then the worst case forgery probability for the Wegman-Carter MAC associated with H, for an adversary provided with one known or chosen message m and the associated MAC tag τ, is upper bounded by ε.

Many practical instantiations of a two-stage MAC can now be constructed based on the various constructions of ε-AXU and ε-AU hash families known from literature. Below we will describe a polynomial MAC, which could be easily adapted to the GMAC authentication function to make its forgery probabilities closer to ideal. The description is given for 32-bit tag with computations in GF(2^{64}), but is easily generalised to any field sizes GF(2^n) and tag lengths $t < n$, to achieve forgery probability close to 2^{-t} for messages up to about 2^{n-t} blocks.

Suppose that the message to be authenticated (after appropriate formatting) consists of L 64-bit blocks $M_{L-1}, \ldots, M_1, M_0$. Given a 64-bit quantity k, a 64-bit intermediate tag T_{in} is computed using a $L{\cdot}2^{-64}$-AU hash family as

$$T_{in} = M_{L-1}k^{L-1} + \ldots + M_1 k + M_0 \text{ over GF}(2^{64}).$$

Given a second 64-bit quantity λ the 32-bit message authentication tag is computed by computing first the product $\lambda \cdot T_{in}$ over GF(2^{64}), and truncating the result to the least significant 32 bits. This is an instantiation of an 2^{-32}-AXU hash family (secure truncation), see Lemma 10 of [BJKS93]. The authentication tag is obtained by xor-ing this result with a 32-bit one time pad. For this construction the worst case forgery probability is bounded by $L\cdot 2^{-64}+2^{-32} \approx 2^{-32}$ for messages with length up to 2^{38} bits.

4. Some Alternatives

The inner tag can also be computed using any $L\cdot 2^{-64}$–AXU family such as $M_{L-1}k^{L} + ... + M_1\, k^{2} + M_0\, k$ used for the GMAC authentication function. But with the selection above in Sec.3 we can save one multiplication over GF(2^{64}) for further use in the secure truncation stage.

The secure truncation can be replaced by some other ε-AXU hash family. Perhaps the most useful alternatives are the following.

1. View T_{in} as two 32-bit quantities N_2 and N_1 and select a 32-bit quantity λ. The 32-bit hash code is computed as $N_2\lambda^2 + N_1\lambda$ over GF(2^{32}). For this construction the key is only 32 bits, and the worst case forgery probability is upper-bounded by $\mathrm{L}\cdot 2^{-64}+2^{-31} \approx 2^{-31}$.
2. View T_{in} as two 32-bit values N_2 and N_1 and select two 32-bit quantities λ_1 and λ_2. Compute $N_2\lambda_2 + N_1\lambda_1$ over GF(2^{32}). This gives a 2^{-32}-AXU hash family, and the same forgery probability as with the construction given in Sec.3.

5. Multiple forgeries

Finally, we would like to comment on multiple forgeries. They can be efficiently prevented by deriving a new key for each message. The GMAC specifies how a one time pad is derived for each tag from the key and the IV using the AES counter mode keystream generator. Almost at the same effort more keystream could be generated to be used as one time key in the computation of the hash code. For example, in the practical instantiation described above this would mean that fresh keys k and λ are selected for each new message.

References

[B05] D. Bernstein. Stronger security bounds for Wegman-Carter-Shoup authenticators. Proceedings of Eurocrypt'05, LNCS 3494, 164-180, Springer-Verlag, 2005. Also available via http://cr.yp.to/antiforgery/securitywcs-20050227.pdf.

[BJKS93] J. Bierbrauer, T. Johansson, G. Kabatianskii, and B. Smeets. On families of hash functions via geometric codes and concatenation. Proceedings of CRYPTO '93, LNCS 773, 331-342, Springer-Verlag, 1993.

[E05] ETSI SAGE. Design of the 3GPP confidentiality and integrity algorithms UEA2 and UIA2. Ongoing.

[F05] N. Ferguson. Authentication weaknesses in GCM. *Comments submitted to NIST Modes of Operation Process*, May 20, 2005.

[MV04] D. McGrew and J. Viega. The security and performance of the Galois/Counter mode of operation. Available via the IACR eprint server at http://eprint.iacr.org/2004/193.pdf.

[MV05] D. McGrew and J. Viega. GCM Update. *Comments submitted to NIST Modes of Operation Process*, May 31, 2005.

[NP99] W. Nevelsteen and B. Preneel. Software performance of universal hash functions. Proceedings of EUROCRYPT '99, LNCS 1592, 24-41, Springer-Verlag, 1999.

[S92] D. Stinson. Universal hashing and authentication codes. Proceedings of CRYPTO '91, LNCS 576, 74-85, Springer-Verlag, 1992.

[S96] V. Shoup. On fast and provably secure message authentication based on universal hashing. Proceedings of CRYPTO '96, LNCS 1109, 313-328, Springer-Verlag, 1996.