IEICE TRANS. FUNDAMENTALS, VOL.E94–A, NO.8 AUGUST 2011

LETTER

A Note on Practical Key Derivation Functions

Shoichi HIROSE[†a)], ***Member***

SUMMARY In this article, we first review key derivation functions specified in NIST SP 800-108 and one proposed by Krawczyk. Then, we propose parallelizable key derivation functions obtained by modifying or using the existing schemes. We also define two measures of efficiency of key derivation functions, and evaluate their performance in terms of the two measures.

key words: *key derivation, pseudorandom function, counter mode, output feedback mode*

1. Introduction

Key derivation functions (KDFs) are widely used in cryptography. A KDF is an algorithm to produce cryptograhpically secure secret keys from input of an initial keying material, which is sufficiently random but not uniform in general.

Although KDFs were not treated very much in detail until recently, Krawczyk provided a thorough study on KDFs [5]. He presented the extract-then-expand paradigm for multi-purpose KDFs. It is illustrated in Fig. 1. The extractor first extracts a short pseudorandom key from the initial keying material. Then, using the short key, the expand part produces as many secret keys as needed. An HMAC-based KDF is also proposed in [5]. The HMAC and, more generally, pseudoradom functions are suitable for both the extractor and the expand part.

It is often the case that only the expand part is focused on and it is simply called a KDF. We will also follow this convention in the remaining part.

The KDF (expand part) in [5] is a kind of output feedback mode of a PRF with a counter. It is called the feedback mode PRF. It is a carefully designed practical KDF. However, it is essentially sequential since it is based on a feedback mode. Three other KDFs based on a PRF are also presented in NIST SP 800-108 [2]. One is a simple counter mode and another is a simple feedback mode. The other is a somewhat complex mode, which is called a double-pipe iteration mode. It is also essentially sequential. Some other hash-based KDFs are also presented in [1], [3], [6].

In this article, we will review the PRF-based KDFs listed above, and propose a few parallelizable KDFs based on them. Then, we will define two measures of efficiency and evaluate the performance of the KDFs.

The rest of the article is organized as follows. Existing schemes by Krawczyk and of NIST SP 800-108 are reviewed in Sect. 2. Proposed schemes are presented in Sect. 3. Performance of the schemes given in Sects. 2 and 3 is summarized in Sect. 4. A concluding remark is given in Sect. 5.

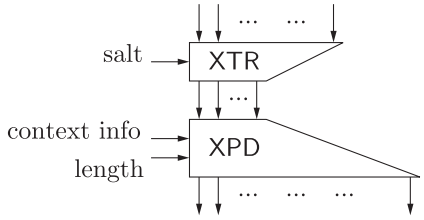

Fig. 1 Extract-then-expand paradigm.

2. Some Existing Schemes

Throughout the article, F_K is assumed to be a variable-input-length pseudorandom function with a key K. It is also assumed that the range of F_K is $\{0,1\}^n$. Typical examples of F are HMAC [7] and CMAC [4].

For a KDF, x represents an input of context information, and ℓ represents the length of the output. For simplicity, we will assume that ℓ is a multiple of n, and $\ell = Nn$. An output string of a KDF will be represented by $K_1||K_2||\cdots||K_N$, where $K_i \in \{0,1\}^n$ for $i = 1, 2, \ldots, N$. The symbol || represents concatenation.

2.1 KDFs in NIST SP 800-108

NIST SP 800-108 specifies three KDF modes using a PRF: the counter mode, the feedback mode, and the double-pipeline iteration mode.

(1) The counter mode.

The counter mode is defined as follows:

$$\mathsf{CTR}(K, x, \ell) = K_1||K_2||\cdots||K_N,$$

where $K_i = F_K(x||i)$ for $1 \le i \le N$. It is also illustrated in Fig. 2. This mode is fully parallelizable: All K_i's can be computed in parallel.

(2) The feedback mode.

The feedback mode is defined as follows:

$$\mathsf{FB}(K, x, \ell) = K_1||K_2||\cdots||K_N,$$

Manuscript received April 5, 2011.

[†]The author is with the Graduate School of Engineering, University of Fukui, Fukui-shi, 910-8507 Japan.

a) E-mail: hrs_shch@u-fukui.ac.jp

DOI: 10.1587/transfun.E94.A.1764

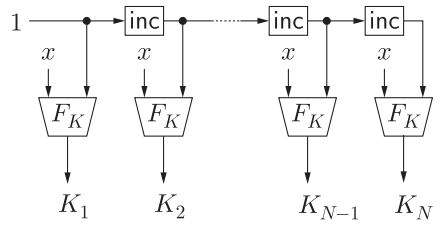

Fig. 2 KDF in counter mode in NIST SP 800-108.

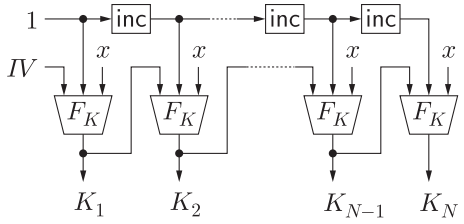

Fig. 3 KDF in feedback mode in NIST SP 800-108.

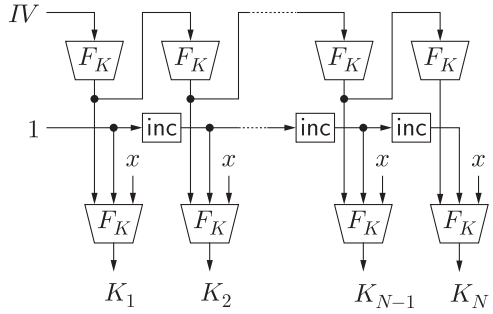

Fig. 4 KDF in double-pipeline iteration mode in NIST SP 800-108.

where $K_0 = IV$ and $K_i = F_K(K_{i-1}||i||x)$ for $1 \le i \le N$. This mode is also illustrated in Fig. 3.

For this mode, the counter values are optional for the inputs to F_K's. However, if the counter values are omitted, then a birthday-type distinguishing attack works for the mode because of the cycle of the output feedback. The feedback mode is essentially sequential.

(3) The double-pipeline iteration (DPI) mode.

The DPI mode is defined as follows:

$$\mathsf{DPI}(K, x, \ell) = K_1||K_2||\cdots||K_N,$$

where $K_i = F_K(F_K{}^i(IV)||i||x)$ for $1 \le i \le N$. The DPI mode is also illustrated in Fig. 4.

For this mode, actually, the counter values are also optional for the inputs to F_K's of the second (outer) pipeline. However, if the counter values are omitted, then a birthday-type distinguishing attack also works for the mode because of the cycle in the output feedback of the first (inner) pipeline.

The DPI mode is essentially sequential since the output feedback mode is applied to the inner F_K's.

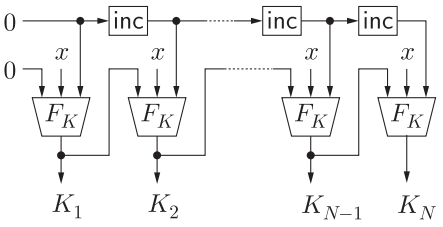

Fig. 5 Feedback mode PRF in [5].

2.2 Feedback Mode PRF

The feedback mode PRF is specified in [5]. It is defined as follows:

$$\mathsf{FBP}(K, x, \ell) = K_1||K_2||\cdots||K_N,$$

where $K_0 = 0$ and $K_i = F_K(K_{i-1}||x||i-1)$ for $1 \le i \le N$. It is also illustrated in Fig. 5.

The feedback mode PRF is very sophisticated. The counter values of inputs to F_K's are mandatory, which avoids the cycle of output feedback. Notice that an input to F_K of this scheme is different from that of the similar feedback mode in NIST SP 800-108. It does not matter in theory, that is, on the assumption that F is a pseudorandom function. However, the input $K_{i-1}||x||i-1$ seems better than $K_{i-1}||i||x$ in practice. F is often instantiated by HMAC or CMAC, which have iterated structure. If $K_{i-1}||i||x$ is given to this type of function, then the variable parts of the input, K_{i-1} and i, are fed only into the first part of the iteration.

The possible disadvantage of this scheme is that it is essentially sequential.

3. Proposed Schemes

3.1 The First Scheme

The first scheme is inspired by the KDF in DPI mode in NIST SP 800-108. The DPI mode seems too cautious. Our idea is quite simple: The output feedback is not necessary for the inner F_K's in the DPI mode. We call the first scheme encapsulated counter (EC) mode. It is defined as follows:

$$\mathsf{EC}(K, x, \ell) = K_1||K_2||\cdots||K_N,$$

where $K_i = F_K(F_K(i-1)||x||i-1)$. It is also illustrated in Fig. 6.

The EC mode is fully parallelizable: All K_i's can be computed in parallel.

The inputs to inner F_K's are counter values. However, the corresponding outputs are only fed into outer F_K's, and are not disclosed. The inputs to outer F_K's are distinct due to the counter values, and they are quite uncorrelated with each other due to the pseudorandomness of $F_K(i-1)$'s. Thus, the EC mode is expected to be more secure than the simple counter mode in practice.

The EC mode can be more efficient than the double

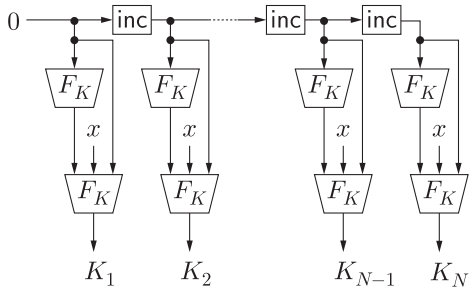

Fig. 6 Encapsulated counter (EC) mode.

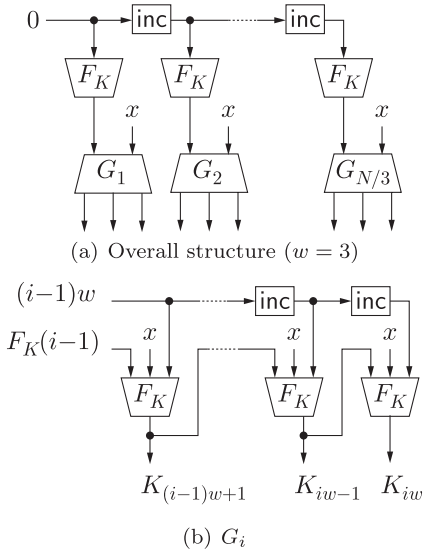

Fig. 7 Generalized encapsulated counter (GEC) mode.

pipeline iteration mode of NIST SP 800-108. Moreover, it seems as secure as the double pipeline iteration mode. The inputs to the outer F_K's in the EC mode are $F_K(i-1)\|x\|i-1$, which follows the feedback mode PRF by Krawczyk.

3.2 The Second Scheme

The second scheme can be regarded as a generalized version of the EC mode. We call it generalized EC (GEC) mode. It is defined as follows:

$$\mathsf{GEC}(K, x, \ell) = K_1\|K_2\|\cdots\|K_N,$$

where

$$K_{(i-1)w+1}\|\cdots\|K_{iw-1}\|K_{iw} = G_i(F_K(i-1)\|x)$$

for $i = 1, 2, \ldots, N/w$. G_i is defined as follows:

$$K_{(i-1)w+1} = F_K(F_K(i-1)\|x\|(i-1)w)$$

and

$$K_{(i-1)w+j} = F_K(K_{(i-1)w+j-1}\|x\|(i-1)w + j - 1)$$

for $j = 2, 3, \ldots, w$. The GEC mode is also illustrated in Fig. 7.

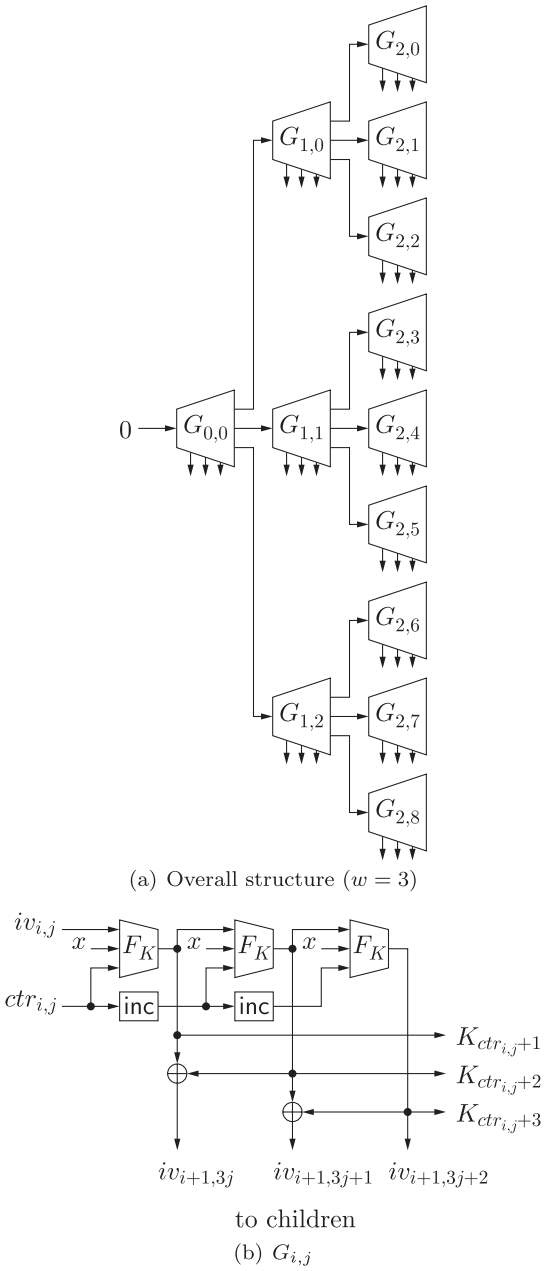

Fig. 8 Tree of the feedback mode PRF (TFP) mode.

For the GEC mode, G_i's can be computed in parallel. G_i's have the same structure as the feedback mode PRF. Thus, G_i's inherit its properties. All the inputs to F_K's are distinct due to the counter values.

3.3 The Third Scheme

The third scheme is a tree mode of the feedback mode PRF. We call it TFP mode. It is defined as follows:

$$\mathsf{TFP}(K, x, \ell) = K_1\|K_2\|\cdots\|K_N.$$

For integers $i \geq 0$ and $0 \leq j \leq w^i - 1$, let

$$ctr_{i,j} = w\left(\frac{w^i - 1}{w - 1} + j\right).$$

Then,

$$K_{ctr_{i,j}+1} || K_{ctr_{i,j}+2} || \cdots || K_{ctr_{i,j}+w} = G_{i,j}(iv_{i,j} || x || ctr_{i,j}),$$

where

$$iv_{i,j} = \begin{cases} 0 & \text{if } i = 0 \\ K_{d(i,j)+w} & \text{if } j \equiv -1 \pmod{w} \\ K_{d(i,j)+1} \oplus K_{d(i,j)+2} & \text{otherwise} \end{cases}$$

and

$$d(i, j) = ctr_{i-1,\lfloor j/w \rfloor} + j - w\lfloor j/w \rfloor.$$

$G_{i,j}$ is defined as follows:

$$K_{ctr_{i,j}+1} = F_K(iv_{i,j} || x || ctr_{i,j})$$

and, for $2 \le k \le w$,

$$K_{ctr_{i,j}+k} = F_K(K_{ctr_{i,j}+k-1} || x || ctr_{i,j} + k - 1).$$

The TFP mode is illustrated in Fig. 8.

$G_{i,j}$'s on distinct paths from the root ($G_{0,0}$) can be computed in parallel. All the inputs to F_K's are distinct due to the counter values.

4. Discussion

The performance of the proposed schemes and the existing schemes are shown in this section. First, two measures of the performance are defined:

$$\text{rate} = \frac{N}{\text{number of invocations of } F_K}$$

and

$$\text{sequentiality} = \text{maximum number of } F_K\text{'s in a cascade.}$$

The sequentiality represents the time required to compute the entire output by as many processors as possible.

Table 1 Performance of KDFs.

schemes	rate	sequentiality
FBP	1	N
DPI	1/2	$N + 1$
EC	1/2	2
GEC	$w/(w + 1)$	$w + 1$
TFP	1	$w \log_w N$

Table 1 summarizes the performance of the proposed schemes and the existing schemes given in Sect. 2. For GEC, there is a trade-off between the rate and the sequentiality.

5. Conclusion

We have presented three parallelizable KDFs based on the existing KDFs by Krawczyk and in NIST SP 800-108. We also have evaluated their performance in terms of two measures, rate and sequentiality.

Acknowledgements

This work was partially supported by KAKENHI 21240001.

References

[1] C. Adams, G. Kramer, S. Mister, and R. Zuccherato, "On the security of key derivation functions," Proc. 7th Information Security Conference (ISC 2004), Lect. Notes Comput. Sci., 3225, pp.134–145, 2004.
[2] L. Chen, "Recommendation for key derivation using pseudorandom functions (revised)," NIST Special Publication 800-108, 2009.
[3] Q. Dang and T. Polk, "Hash-based key derivation (HKD)," draft-dang-nistkdf-01.txt, 2006.
[4] M. Dworkin, "Recommendation for block cipher modes of operation: The CMAC mode for authentication," NIST Special Publication 800-38B, 2005.
[5] H. Krawczyk, "On extract-then-expand key derivation functions and an HMAC-based KDF," http://webee.technion.ac.il/~hugo/kdf/, 2008.
[6] National Institute of Standards and Technology (NIST), "Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography," NIST Special Publication 800-56A, 2006.
[7] National Institute of Standards and Technology (NIST), "The keyed-hash message authentication code (HMAC)," Federal Information Processing Standards Publication 198-1, 2008.