VISA Format Preserving Encryption

1. Cover Sheet

Name of submitted mode	VISA Format Preserving Encryption (VFPE)
Principal Submitter	VISA USA Inc. Attention: Kim R. Wagner or John Sheets P.O.Box 8999 San Francisco, CA 94128-8999 Phone: 650-432-3489 (kw) or 650-432-2984 (js) Fax: 650-554-4097 Email: kwagner@visa.com or jsheets@visa.com
Auxiliary Submitter	N/A
Mode's inventors	John Sheets, Kim R. Wagner
Name of Owner	VISA USA Inc.

2. Notation

0xNN	Hexadecimal notation, NN in base 16.
$\lceil x \rceil$	The real number *x* rounded up to the nearest integer larger than or equal to *x*.
$\lfloor x \rfloor$	The real number *x* rounded down to the nearest integer smaller than or equal to *x*.
a div *b*	$\left\lfloor \frac{a}{b} \right\rfloor$ for non-negative integers *a* and *b*
$\log_r$	Base-*r* logarithm, where *r* is a natural number
s[*i*]	When *s* is a string, *s*[*i*] is the *i*'th element of s. Indexing starts at 1, hence, *s* consists of *s*[1], *s*[2], …, *s*[len(*s*)]. *s*[*i*] is defined for all *i* such that $1 \le i \le \text{len}(s)$, and undefined for all other *i*. In particular, if *s* is the empty string, *s*[*i*] is undefined for all *i*.
len(*s*)	The length of the string *s*
AsInteger	For a string of bits, *s*, we write AsInteger(*s*) for the corresponding integer value. That is, if *s* = *s*[1]…*s*[len(*s*)] with *s*[1] the most significant bit, then AsInteger (*s*)= $\sum_{i=1}^{len(s)} [(s[t](2]^{len(s)-i)}$.
s||*t*	The concatenation of the strings *s* and *t*.

3. Mode Specification

The Visa Format Preserving Encryption mode (VFPE) operates as a stream cipher that is format preserving.

The easiest way to think of VFPE is as Counter Mode (CTR) from NIST SP800-38A generalized to modulo-*n* addition instead of modulo-2 addition.

Let *A* be an alphabet with *n* different characters, where *n* is a natural number greater than 1. We denote by *A** the set of strings with elements from *A*, including the empty string. In this description it is assumed that the alphabet *A* is the set {0, …,*n*-1}. If this is not the case, a translation is needed, based simply on the number of different characters in the alphabet *A*. The translation would happen prior to encryption, and again, after decryption, so that encryption and decryption will always work on alphabets of the form {0, …,*n*-1} for some positive integer n, greater than 1.

VFPE uses Counter (CTR) mode as defined in [5] with a block cipher CIPH (AES or TDEA) with block size *b* bits, and encryption key *K* for CIPH, and a sequence of counter blocks (called counters in [5]) T_1, T_2, … , to produce a sequence of output blocks, one for each counter block. Each output block consists of *k* base-*n* digits, where *k* is a configurable parameter which must be chosen from the interval $\{1, \ldots, \lfloor \log_n 2^b \rfloor\}$. For reasons explained below, each counter block is *b*-7 bits, rather than *b* bits as in [5]. The mechanism for how to produce the output blocks is also described below.

To encipher a plaintext *P* of length *L*, with $1 \leq L$, as many output blocks as necessary (but no more) are generated, so that the total number of base-*n* digits in the output blocks is at least *L*, that is, we calculate the unique integers *p* and *r* such that $\frac{L}{k} \leq p < \frac{L}{k} + 1$ and $0 \leq r < k$, such that $L = pk - r$, and generate output blocks G_1,…, G_p . Then each plaintext base-*n* digit *P*[*i*] is added, modulo-*n*, to the *i*'th base-*n* digit from the concatenation of the output blocks, $G_1 \| G_2 \| \ldots \| G_p$, to form the *i*'th digit of the ciphertext:

$$C[i] = (P[i] + (G_1 \| \ldots \| G_p)[i]) \mod n \ .$$

Since *k* may not divide *L*, some digits of the last output block, G_p may be ignored. In fact, the last *r* base-*n* digits of G_r are not used.

To decipher a ciphertext *C* of length *L*, with $1 \leq L$, as many output blocks as necessary (but no more) are generated, so that the total number of base-*n* digits in the output blocks exceed *L*, which is done in the same way as for encryption. Then from each ciphertext base-*n* digit *C*[*i*] is subtracted, modulo-*n*, the *i*'th base-*n* digit from the concatenation of the output blocks, $G_1 \| \ldots \| G_p$, to form the *i*'th digit of the plaintext:

$$P[i] = (C[i] - (G_1 1 \| \ldots \| G_1 p)[i]) \mod n .$$

For VFPE, as for Counter mode itself, the sequence of counter blocks must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single encryption: across all of the messages that are encrypted under a given key K, all counters must be distinct. See [5] for methods for generating counters.

Given a block cipher CIPH with block length *b*, a key *K* for CIPH, a *b*-7 bit counter *T*, a natural number *n*>1, which is the base of the plaintext to be enciphered, and an integer *k* with $0 < k \leq \lfloor \log_n(2^b) \rfloor$, an output block consisting of *k* base-*n* digits is produced in the following way:

A 7-bit counter, *S*, is initialized to 0. Then $CIPH_K$ is applied to *S*||*T* to produce a block *B* of *b* bits. B is interpreted as an integer in the interval $\{0, \ldots, 2^b - 1\}$, and if $B < n^k \left\lfloor \frac{2^b}{n^k} \right\rfloor$, then it is accepted, otherwise *S* is incremented and $CIPH_K$ is applied again to *S*||*T, etc.,* until *B* is accepted or *S* equals 127. If S = 127, an error is raised, otherwise *B* is converted to base-*n* and is the *k*-digit base-*n* output block, possibly with leading zeros . Under the assumption that $CIPH_K$ is a pseudorandom permutation, the probability in each iteration that *B* is accepted is at least 0.5, and the probability that an error is raised, is at most 2^{-128}. The pseudocode below describes this algorithm:

```
i = 0;

Input_Block = Si || T ;

max_B = (n^k)*((2^b) div (n^k));
B = CIPH(K, Input_Block);
while ( (AsInteger(B) ≥ max_B) AND (i < 127)) {
       i = i+1;

       Input_Block = Si || T ;

       B = CIPH(K, Input_Block);
};
if (i=127) return ERROR;
Output_Block = Convert(B, k, n);
return Output_Block;
```

Here it is assumed that S_0, S_1, ..., S_{127} enumerate the 128 different 7-bit combinations, that "AsInteger" takes a string of b bits $B[1]$, ..., $B[b]$ and converts it to the integer $\sum_{i=1}^{b} B[i](2)^{b-i}$, and that "Convert" converts B to k base-n digits, with leading zeros if necessary:

```
Convert(B, k, n) {
        M = AsInteger(B);
        for (i=1; i≤k; i++){
                D[i] = M mod n;
                M = M div n;
        };
        return D;
}
```

The maximum value for L, that is, the bit length of the longest plaintext that can be enciphered is $2^{b/2}$.

The upper bound $n^k \left\lfloor \frac{2^b}{n^k} \right\rfloor$ for B interpreted as an integer is chosen as the largest possible whole multiple of n^k, that makes it possible to extract a k-digit base-n number uniformly from it, assuming the distribution of B is uniform.

Table 1 below shows as an example the values of B which are acceptable and those which are not, when n = 10 (decimal digits), b = 8 (numbers from 0 to 256) and k = 1 or 2 (extracting 1 or 2 decimal digits).

k = 1				k = 2		
B	Status	B mod n^k		B	Status	B mod n^k
0	OK	0		0	OK	0
…	OK	…		…	OK	…
249	OK	9		199	OK	99
200	Try again	n/a		250	Try again	n/a
…	Try again	n/a		…	Try again	n/a
255	Try again	n/a		255	Try again	n/a

Table 1 – Extracting decimal digits from 8 bits

Compared with other format preserving encryption schemes the VFPE is approximately one order of magnitude faster, being able on average to encipher more than 36.9 decimal digits per invocation of AES. Additionally being a stream cipher it is not vulnerable to small plaintext attacks contrary to the other format preserving encryption schemes, where dictionary attacks are often an issue in practice, and where tweaks and other mechanisms are needed to attempt to mitigate this vulnerability. Hence, VFPE works even in settings where tweak data is not available or reliable.

VFPE is fully parallelizable and the key stream can be precomputed, since it does not depend on the plaintext (unlike other FPEs where tweak data is derived from the plaintext).

As a stream cipher, VFPE is malleable, that is, an attacker can by modifying ciphertext effect predictable modulo-n additions on the underlying plaintext. Hence, if integrity is required, MAC'ing or other integrity preserving mechanisms need to be used. This vulnerability mirrors that of any stream cipher, e.g. Counter Mode (CTR).

One important property of stream ciphers is that if an adversary were to obtain two ciphertexts encrypted with the same key stream, digit-wise subtraction would yield the key stream used for those two ciphertexts, and hence both plaintexts as well. It is therefore critical that key streams are not reused, which is the reason behind not reusing counter blocks through the lifetime of a key.

In the retail payments area the technique that is used currently for encrypting PIN blocks, known as Derived Unique Key Per Transaction (DUPKT, see [8]) is very suitable for this purpose, since it ensures that a new key is used for each encryption. With a new key (for the underlying block cipher, CIPH), the concern about repeating key streams has been eliminated. Each instantiation of DUKPT provides up to 5 keys, a PIN key, a MAC key in either direction, and a data encipherment key ENC in either direction, and for VFPE the data encryption key ENC for the appropriate direction must be used.

Performance Analysis

To ensure that there are enough bits for the *k* base-*n* digits and a uniform distribution can be achieved, there are $n^k \left\lfloor \frac{2^b}{n^k} \right\rfloor$ different acceptable possibilities for the *b*-bit number *B*, so that we successfully can extract the *k*-digit base-*n* number from *b* bits with equal likelihood for each *k*-digit base-*n* number, which are the cases where B < $n^k \left\lfloor \frac{2^b}{n^k} \right\rfloor$. In this case the extracted number is *B* mod n^k. The likelihood of success (of extracting a *k*-digit base-*n* number from *b* bits) is $p(n,k,b) = \frac{n^k}{2^b} \cdot \left\lfloor \frac{2^b}{n^k} \right\rfloor$. If we don't have success, CIPH is called again with an updated counter, using the same key, *K*, and we try again the output from CIPH_K. This gives the average number of calls to the block cipher to extract *k* base-*n* digits as $a(n,k,b) = \frac{1}{p(n,k,b)}$, since in general $\sum_{i=0}^{\infty} p(1-p)^i(i+1) = p^{-1}$ for $0 < p \leq 1$, ignoring the possibility of error. We have probability *p* of iterating once (success the first iteration), and thus probability (1-*p*) of iterating again. In that case, we have conditional probability *p* of terminating for a total of 2 iterations, giving the total probability for this case as (1-*p*)*p*, and in general the probability of iterating *i*+1 times and terminating thereafter is $(1-p)^i p$, which explains the sum above. Thus the average yield of base-*n* digits from one call to CIPH_K is $y(n,k,b) = \frac{k}{p(n,k,b)}$.

Table 2 below shows the values of $p(n,k,b)$, $a(n,k,b)$, and $y(n,k,b)$ for various typical and/or illustrative values of n, k, and b. The value 95 for the base *n* is chosen as an example because the number of printable ascii characters is 95 (when space is considered a printable ascii character).

Base, *n*	Number of bits, *b*	Number of Base-*n* digits to extract, *k*	Likelihood of success per iteration, *p*(*n*,*k*,*b*)	Average number of iterations, *a*(*n*,*k*,*b*)	Average yield of base-*n* digits per iteration, *y*(*n*,*k*,*b*)
10	4	1	0.6250	1.600	0.6250
10	13	3	0.9766	1.024	2.9297
10	64	19	0.5421	1.845	10.2999
10	64	18	0.9758	1.025	17.5641
10	128	38	0.8816	1.134	33.5016

10	128	37	0.9992	1.001	36.9693
10	128	36	0.9992	1.001	35.9701
95	64	9	0.9908	1.009	8.9173
95	64	8	0.9998	1.000	7.9984
95	128	19	0.9980	1.002	18.9629
95	128	18	0.9992	1.001	17.9859

Table 2 - Yields of base-*n* numbers from *b* bits of key material

The two cases that are likely to be most important in practice are highlighted in the table above. The table shows that the highest possible yield of decimal numbers from one round of AES on average (when extracting 37 digits per AES block) is in excess of 36.97 decimal digits, and that the similar number for TDEA is in excess of 17.56 decimal digits achieved when extracting 18 decimal digits from a TDEA block.

It can be seen from the table that the value *k* (number of base-*n* digits extracted from each block of bits output by CIPH) can sometimes beneficially be set to one less than its maximally allowable value, in order to optimize the average number of base-*n* digits generated per call to CIPH.

4. Limitations and other security considerations

The maximum allowable value for the base, *n*, is 2^b, where *b* is the block size of the underlying block cipher. If $n = 2^t$, for a *t* such that $1 \leq t \leq b$, and where *t* divides *b*, VFPE degenerates essentially to CTR since every possibility of the *b* bits in the bit stream are acceptable as a $\frac{b}{t}$-digit base-*t* number.

The minimum allowable value for *n* is 2, since it normally does not make practical sense to work in base-1, and in particular it does not make sense to talk about a *random* digit in base-1.

Given *n* and *b*, the maximum number of base-*n* digits that can be enciphered using only one counter value is $k = \lfloor \log_n(2^b) \rfloor$.

We refer to [5] Appendix B for generation of counter blocks, and to *ibid*. Section 6.5 for the definition of CTR. In particular the following quote highlights the requirement for unique counters across the lifetime of a key: "The sequence of counters must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single message: across all of the messages that are encrypted under the given key, all of the counters must be distinct."

In order to mitigate against precomputation attacks as described in [6], it is recommended that each initial counter block incorporates information unique to the encrypting party. That way the amortization that is crucial for Hellman's time-memory tradeoff attack is prevented.

Following the analysis in [6], the plaintext length is limited by the condition that if (for a given encrypting party) the counter blocks T_i have *d* variable bits each, excluding bits used for redundancy or to identify uniquely the encrypting party, then no more than $2^{d/2}$ blocks of length *b* should be generated with one key. Since *k* base-*n* digits are encrypted with each block of *b* bits, the limit for the plaintext is then $\frac{k}{b} \cdot 2^{d/2}$ base-*n* digits. For example, if AES is used as CIPH, *b*=128. The counter blocks are 128-7 = 121 bits each. Assuming 16 bits are used to uniquely

identify the encrypting party, *d*=105. When extracting *k*=37 decimal digits from one AES block, this amounts to a limit of $37 \cdot 2^{48}$ decimal digits enciphered with any one key.

For TDEA *b*=64. The counter blocks are 64-7 = 57 bits each. Assuming 16 bits are used to uniquely identify the encrypting party, *d*=41. When extracting 18 decimal digits from one TDEA block, this amounts to a limit of $9 \cdot 2^{18} = 294{,}912$ decimal digits enciphered with any one key.

These limits are more conservative than those defined in [5] for CTR.

5. Summary of Properties

Security Function	Encryption
Error Propagation	None, since key stream is operating on plaintext digit-by-digit
Synchronization	None built-in. Has to be done outside of mode if required. For example Derived Unique Key Per Transaction (DUKPT) can be used. (See X9.24-1).
Parallelizability	Fully parallelizable. Each block of the key stream can be generated independently, and encryption of each block can happen independently.
Keying Material Requirements	One AES-128 (or TDEA) key required
Counter/IV/Nonce Requirements	8 or 16 byte counter required for key stream generation to maintain a state, for TDEA and AES respectively
Memory Requirements	Very low: 8 or 16 byte counter (for TDEA or AES), 16 byte key, plaintext block, ciphertext block, plus executable
Pre-processing Capability	Key stream can be pre-computed
Message Length Requirements	If (for a given encrypting party) a counter block *T* has *d* variable bits, excluding bits used for redundancy or to identify uniquely the encrypting party, then no more than $\frac{k}{b} \cdot 2^{d/2}$ base-n digits should be encrypted with any one key.
Other Characteristics	Similar to CTR mode, the decryption of any ciphertext block is vulnerable to the introduction of specific bit errors into that ciphertext block if its integrity is not protected. This malleability may be mitigated by protecting the integrity of the ciphertext in other ways, e.g. with a MAC independent of the encipherment.

6. References

[1] M. Bellare, A. Desai, E. JokiPii, and P. Rogaway, *A concrete security treatment of symmetric encryption*, Sept 2000.

[2] M. Bellare, P. Rogaway, T. Spies, *The FFX mode of operation for format-preserving encryption*, Draft 1.1, Feb 20, 2010.

[3] J. Black and P. Rogaway, *Ciphers with arbitrary finite domains*, 2002.

[4] E. Brier, T. Peyrin, J. Stern, *BPS: a format-preserving encryption proposal*, Ingenico, France, 2010.

[5] M. Dworkin, NIST SP 800-38A: *Recommendation for block cipher modes of operation – Methods and techniques*, 2001.

[6] D. McGrew, *Counter mode security: Analysis and recommendations*, Cisco Systems, Nov 15, 2002.

[7] J. Vance, *VAES3 scheme for FFX – An addendum to "The FFX mode of operation for format-preserving encryption*", Draft 1.0, May 2011.

[8] X9.24 Part 1: 2009, *Retail financial services: Symmetric key management Part 1: Using symmetric techniques*.

7. Performance Estimates

The VFPE can encrypt more than 36.97 decimal digits on average per call of AES. It is fully parallelizable and the key stream is precomputable.

8. Intellectual Property Statements / Agreements / Disclosures

Visa has filed a patent application (USPTO-20090063354, 12/202,978, Account transaction fraud detection) for parts of the technology described in this submission. Should the patent be granted and the technology included in the forthcoming NIST document on format preserving encryption, Visa intends to provide licence to the technology on reasonable and non-discriminatory (RAND) terms.