A Chosen Messages Attack on the ISO/IEC 9796-1 Signature Scheme

(lecture at Eurocrypt 2000)

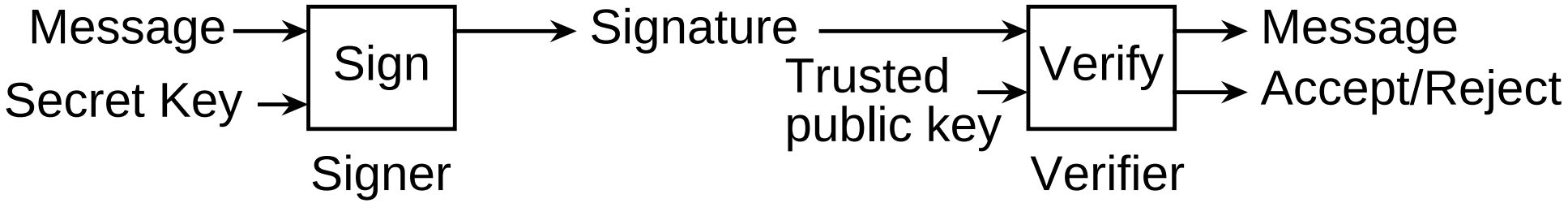
François Grieu

Spirtech

francois.grieu@spirtech.com

The ISO/IEC 9796-1 standard

is a digital signature scheme giving message recovery, using redundancy, built on the RSA or Rabin public key cryptosystems. The signed message, of limited length, is embedded in the signature :



It was designed 1989-1990, approved in 1991 as ISO/IEC 9796

ISO/IEC 9796 = ISO/IEC 9796-1 but ≠ ISO/IEC 9796-2 (hash-based)

A key design criteria was to use no hash function, for which there was no widely accepted standard : the now ubiquitous Secure Hash Algorithm was adopted in 1993 and revised as SHA-1 in 1995

The ISO/IEC 9796 standard was designed to resist known attacks, without formal security analysis. Practical RSA signature schemes with provable security appeared years later [1] and use a hash function.

[1] Bellare, M. and Rogaway, P: The exact security of digital signatures, how to sign with RSA and Rabin, Eurocrypt 1996

RSA signature : the need for redundancy

A signature scheme from the RSA cryptosystem can **not** be as simple as

- signing message M with the secret key by $S = \mathcal{S}(M) = M^d \bmod n$

- recovering M using the public key $M = \mathcal{V}(S) = S^e \bmod n$

- checking M makes sense (how ?)

Problem, it is easy to construct random messages with known signature : just select S arbitrarly, compute the matching message $M = \mathcal{V}(S)$ and with some trial and error a message that “makes sense” may be found.

For example about one attempt out of 10^5 will give a valid C or Pascal string of at least three ASCII letters in a sensible case.

To avoid this and automate the “makes sense” test, a widely used technique is to add some redundant information in the value submitted to the secret function $\mathcal{S}$, and check this redundant information on the verifier side.

The function producing the redundant message $\tilde{M}$ must be an easily invertible injection, in order to provide message recovery on the verifier side.

RSA signature : simple redundancy

One of the first redundancy technique has been straight duplication of a message of size fixed to half the size of the public modulus n

For example the message M = 123456h may have signature S = $S(\tilde{M})$ with $\tilde{M}$ = 000000000000000000000000123456000000000000000000000000123456h

On the verifier side it is checked $V(S)$ is made of two identical halves.

Aside from the fact that the all-zero message has trivial signature, this approach can be attacked using the “multiplicative” property of the RSA cryptosystem :

$$S(A\ B) = (S(A)\ S(B) \bmod n)$$

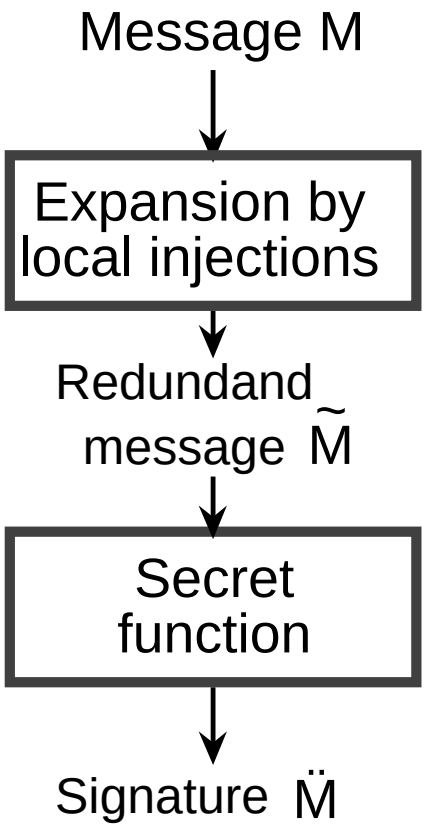
With public exponent e=3 , knowing the signature S of the above message and noticing that $S(1000_h) = 10_h$ one can compute the signature for message 123456000_h as $(10_h.S \bmod n)$

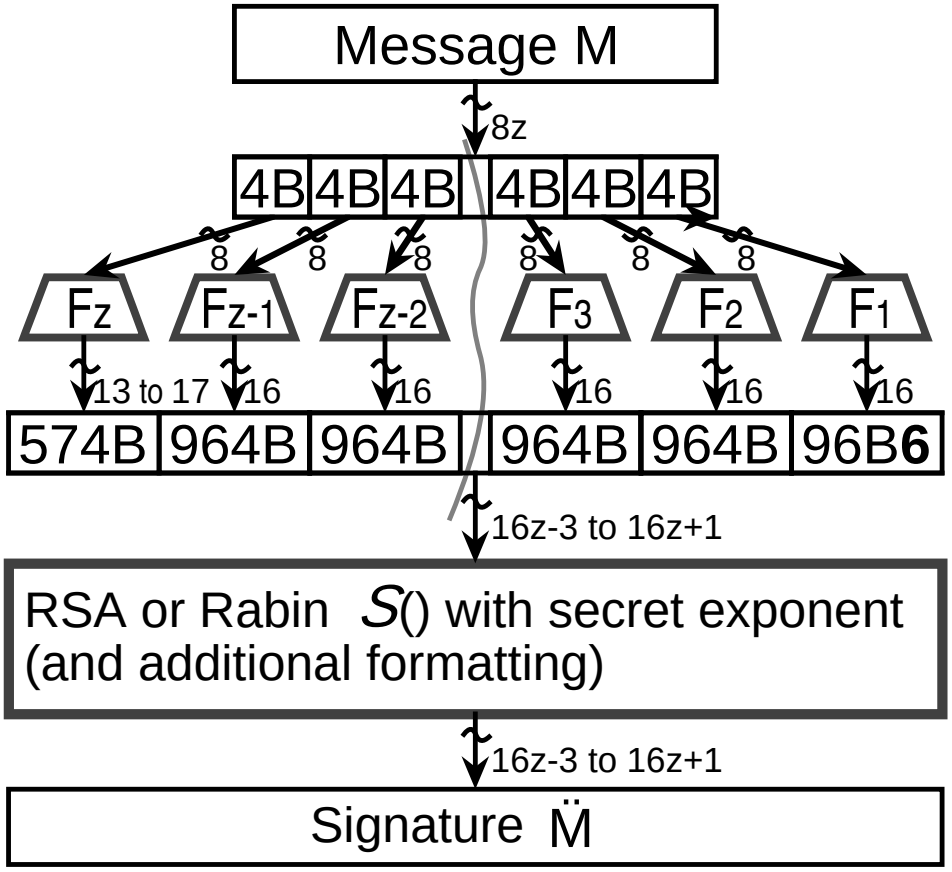
Many other attacks on simple redundancy schemes are known [2]

[2] Misarsky J. F., How (not) to design RSA signature schemes, Public-key cryptography, Springer-Verlag, LNCS 1431.

ISO 9796-1 signature production

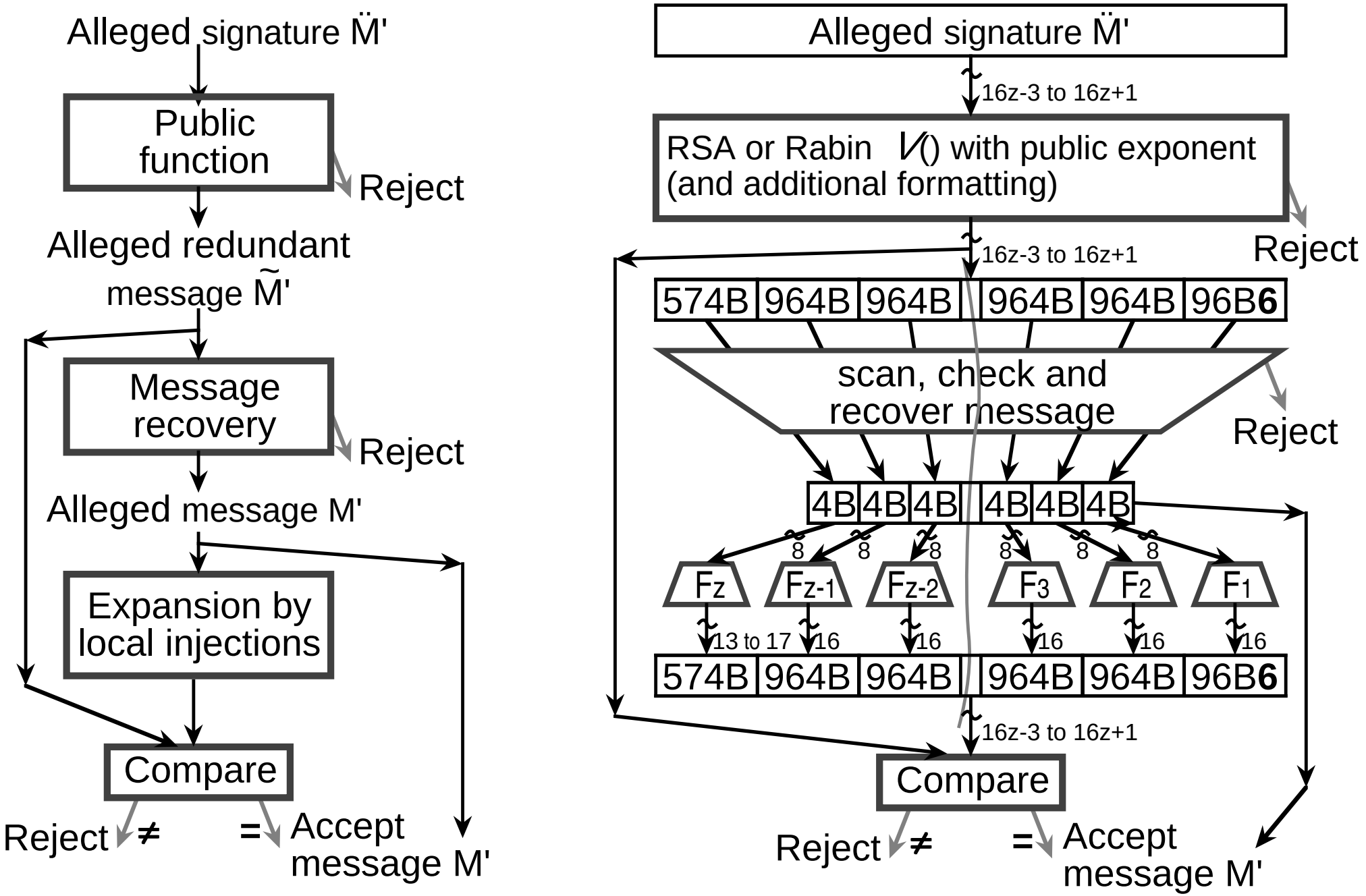
We restrict our description, and the attack, to public modulus n of 16z-2 to 16z+2 bits and messages M of 8z bits (half the key size); these are common parameters.





The additional formatting of ISO 9796-1 makes $\tilde{M}$ 1 bit shorter than n, and for even public exponent e ascertains the Jacobi symbol is +1 before exponentiation; the cryptosystem remains reversible using the fact that $\tilde{M} \equiv 6 \mod 16$

ISO 9796-1 signature verification



The attack plan

We select one pair of small integers a, b and construct the set of message pairs A, B such that corresponding expanded messages verify :

$$\frac{\tilde{A}}{\tilde{B}} = \frac{a}{b} \qquad (2)$$

where $M \longrightarrow \tilde{M}$ is the ISO 9796-1 redundancy function

We'll see how to solve this equation very simply.

When two message pairs A, B and C, D are solution of the above, we have :

$$\frac{\tilde{A}}{\tilde{B}} = \frac{\tilde{C}}{\tilde{D}} = \frac{a}{b} \qquad \text{thus} \quad \tilde{A}\,\tilde{D} = \tilde{B}\,\tilde{C}$$

The multiplicative property of the RSA cryptosystem does the rest :

$$S(\tilde{A}\,\tilde{D}) = S(\tilde{B}\,\tilde{C}) \quad \text{thus} \quad S(\tilde{A})S(\tilde{D}) \equiv S(\tilde{B})S(\tilde{C}) \mod n$$

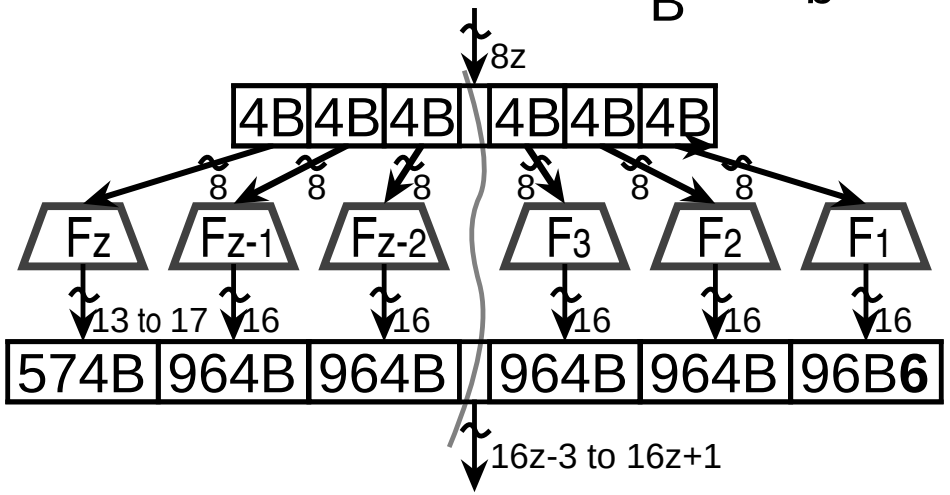
And we get $S(\tilde{D}) = S(\tilde{A})^{-1}S(\tilde{B})\,S(\tilde{C}) \mod n$

$S(\tilde{A})^{-1} \mod n$ can be computed from $S(\tilde{A})$ using the Extended Euclidian algorithm.

With a minor complication due to the formatting prescribed by ISO 9796-1, the signature of message D is deduced from the signature of the three messages A B C and the public modulus.

Finding messages A,B with $\frac{\tilde{A}}{\tilde{B}} = \frac{a}{b}$

where $M \longrightarrow \tilde{M}$
is the injection shown
built from z small injections



Choice of a,b

we can restrict to $a<b$ and a,b relatively prime

the 4 right bits of F_1 are 6 thus solution may exist only for $a \equiv b \mod 8$

the left bit of F_z is always 1, implying $a < 2b$ and finaly $9 \leq a < b < 2a$

Reformulating the problem

For hypothetical solution A,B there is a uniquely defined integer W such that

$$\tilde{A} = a\,W \quad \text{and} \quad \tilde{B} = b\,W \qquad (3)$$

We break $\tilde{A}$, $\tilde{B}$, W into 16 bits segments in the above multiplication, and the problem becomes finding segments of W such that the resulting segments of $\tilde{A}$, $\tilde{B}$ are among the 256 acceptable values defined by the F_i functions.

The search turns into an easy graph traversal

During the multiplications by 16 bits segments $\tilde{A} = a\,W$ and $\tilde{B} = b\,W$ we have *overflows* propagating from a segment to the next left segment, these are noted $\bar{a}_i$, $\bar{b}_i$ in the paper. Whe have $0 \le \bar{a}_i < a$ and $0 \le \bar{b}_i < b$.

For each of the z 16 bits segments, we can easily build the matching values of overflows on the right, on the left, value of segment of W, and corresponding segment in A,B. These z problems can be solved indepently, then assembled into a graph linked by the overflows, and with the additional constraint that the left and right overflows are 0,0. The search for a solution is equivalent to traversing the graph.

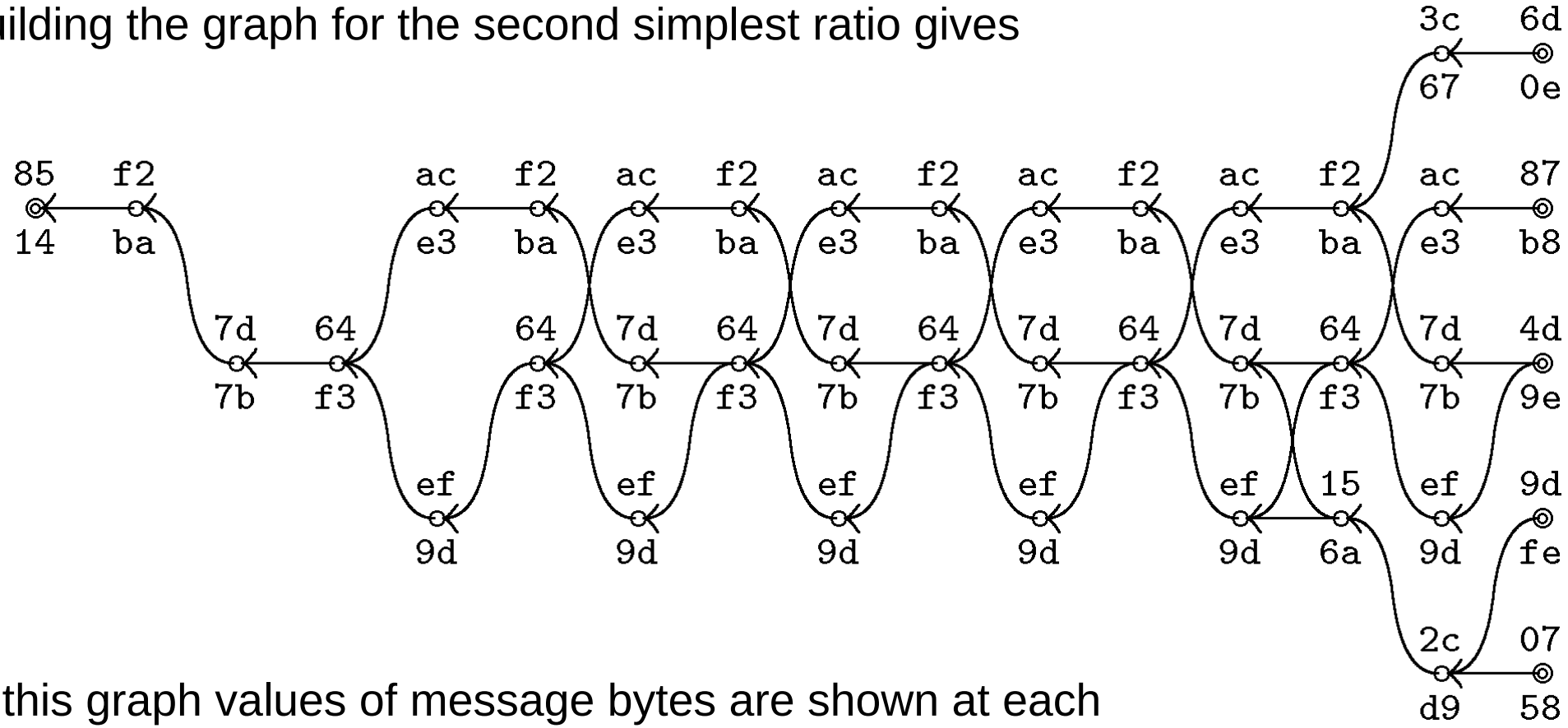
An optimisation, used in the algorithm given in the paper, is to build only these portions of the graph that are connected to one side. The paper also shows how the overflows can conveniently be stored a as single quantity, the link, which can take at most a+b-1 values.

At this point we have found how to solve $\frac{\tilde{A}}{\tilde{B}} = \frac{a}{b}$

if there are solutions, but have not shown there are any.

Solutions for a 256 bits modulus and ratio a/b = 11/19

Building the graph for the second simplest ratio gives



In this graph values of message bytes are shown at each node. For example, the bottom link give the solution

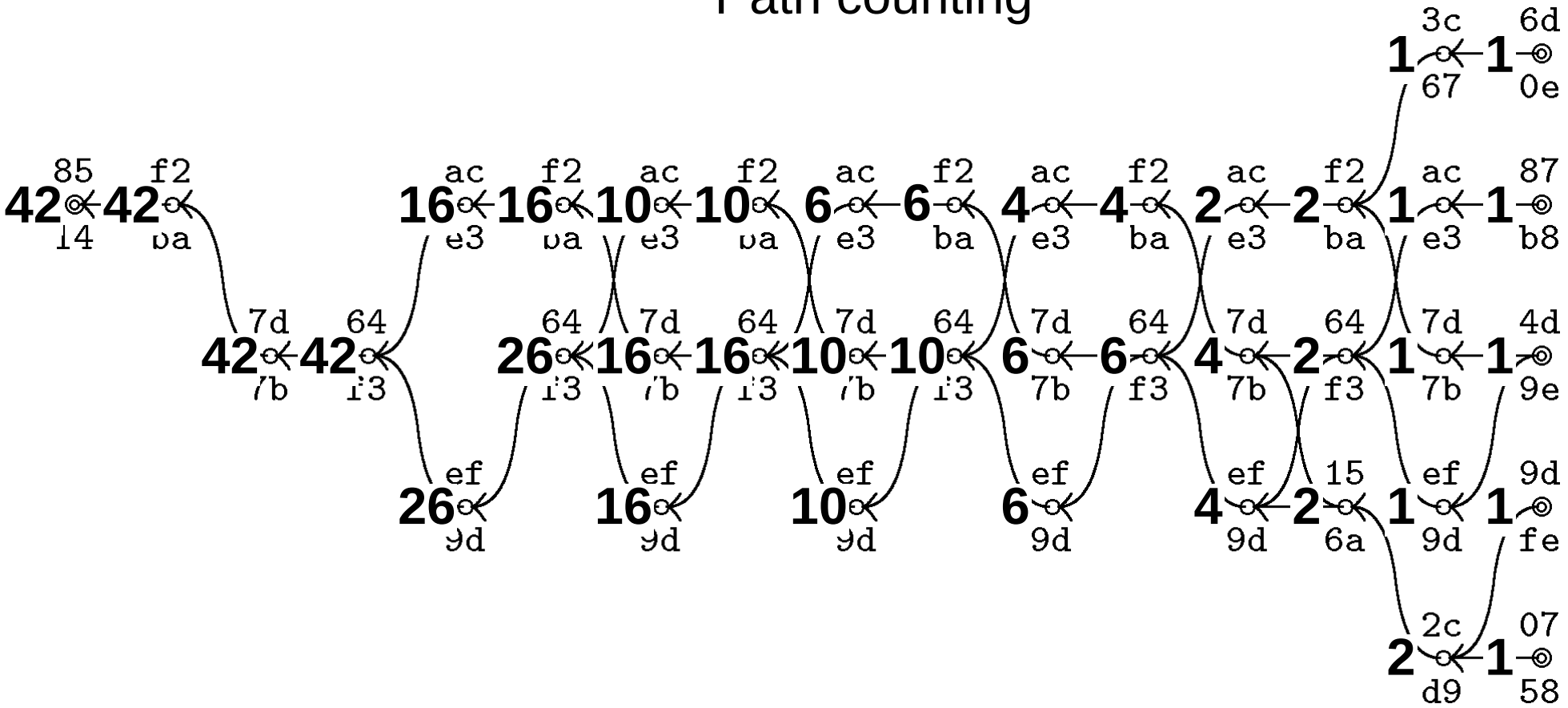
A = `85f27d64ef64ef64ef64ef64ef152c07`

B = `14ba7bf39df39df39df39df39d6ad958`

with indeed $\frac{\tilde{A}}{\tilde{B}} = \frac{11}{19}$

Since this graph is periodic in the center, it can readily be used for larger keys like 1024 bits, provided $k \equiv 0 \bmod 32$. Finding 4 messages suitable for the attack is as simple as finding two distinct solutions for this maze.

Path counting



To count the number of path throught the graph ,we label right nodes with 1, then each node on the left with the sum of it's ancestors. The total number of path is the sum of the left nodes. In this graph, we have 42 distinct message pairs. It is easy to see the number of paths grows as a Fibonnaci sequence when the key size grows by 32 bits increments.

How many messages are suitable for the attack ?

A simple program builds the graph, count solutions, and output some messages in seconds. For k=1024 bits and restricting to b<1024, we have about 9% of the 13264 explored ratios that are usable (give at least two message pairs). We get about $5.7\ 10^{14}$ such pairs, among which 98% are for the ratio 389/525 which gives 2^{49} pairs.

Asymptoticaly, when the modulus bit size k grows by 16 bits, we have twice as many pairs when k = 16z-2, 16z or 16z+2, or about 1.62177 times more pairs when k = 16z-1 or 16z+1.

Constraints on the messages, which occur often in practice, actually make the search easier, and there are solutions even with some degree of constraint, like “nearly entirely ASCII” or “entirely displayable” messages.

Open problems :
- why and how do solutions thin out when a and b increase ?
- could the three injections be modified to block the attack ?
- what if the center injection are not all alike ?

the search algorithm works unmodified and there are solutions for many choices of injections, but in what proportion ?

Is it practical ?

The main result is obtaining the forged signature of 1 message from the signature of 3 other messages, for all usual choices of key size, regardless of the public exponent e.

But the attack works only for messages that are in some narrow subset of the possible messages, and thus in practice works only if the attacker
- can obtain the signature of messages he or she choose in this subset before submitting them for signature,
- and has a benefit in obtaining the signature of another message in this subset.

This classifies the attack as a “Chosen Messages Attack”.

A vulnerable setup would be one where signatures of arbitrary messages are available for a price. The attacker can get 2x signatures for the price of x+2. Things may get even worse if the price is part of the message.

When the public exponent e is even, we can use the attack to find a non-trivial square root modulus n, then factor n (a total break of the system) from the signature of 4 chosen messages. This reduces to 2 signatures when e=2 (the most likely choice of even e). This would be a threat to an off-the-shelf Smart Card implementation of ISO/IEC 9796-1 signature, if it allows signature of externally generated messages.

Can we still use ISO/IEC 9796-1 ?

Because the adversary must be able to obtain signatures of special messages, the attack does **not** appear to be a threat in quite a few scenarios, including
- when the signer builds the whole message,
- or when the message is, or includes, a secure hash.

But resistance to chosen message attacks is a desirable property, and as a result of previous chosen messages attacks [3][4] and the present one, the SC27 group considers withdrawing the standard, which makes another reason to avoid it for new applications.

[3] Coron, J. S. and Naccache, D. and Stern, J. P.: A new signature forgery strategy applicable to ISO 9796-1/2, ECASH™, PKCS #1 V2.0, ANSI X9.31, SSL-3.02. Circulated as ISO/IEC JTC1/SC27 N2329 alias WG2 N429, May 1999.

[4] Coppersmith, D. and Halevi, S. and Jutla, C.: ISO 9796-1 and the new forgery strategy (Working Draft), August 1999.
See <http://grouper.ieee.org/groups/1363/contrib.html>