On the Security of Digital Signature Schemes Based on Error–Correcting Codes

Sheng–bo Xu** Jeroen Doumen* Henk van Tilborg*
* Department of Mathematics and Computing Science
Eindhoven University of Technology, P.O. Box 513
5600 MB Eindhoven, the Netherlands
E–mail: {doumen; henkvt} @win.tue.nl
** Pijnenburg Securealink B.V., Boxtelseweg 26
5261 NE Vught, The Netherlands
E–mail: s.xu@securealink.com

Abstract

In this paper we discuss the security of digital signature schemes based on error–correcting codes. Several attacks to the Xinmei scheme are surveyed, and some reasons given to explain why the Xinmei scheme failed, such as the linearity of the signature and the redundancy of public keys. Another weakness is found in the Alabbadi–Wicker scheme, which results in a universal forgery attack against it. This attack shows that the Alabbadi–Wicker scheme fails to implement the necessary property of a digital signature scheme: *it is infeasible to find a false signature algorithm $D^\star$ from the public verification algorithm E such that $E(D^\star(\underline{m})) = \underline{m}$ for all messages $\underline{m}$*. Further analysis shows that this new weakness also applies to the Xinmei scheme.

Keywords: Digital Signatures, Error–Correcting Codes.

1 Introduction

The concept of digital signatures was proposed by Diffie and Hellman when they introduced the public–key cryptography in their pioneering paper [10]. When Alice wants to send a signed message $\underline{m}$ to Bob, she sends the pair $(\underline{m}, \underline{s})$ where $\underline{s}$ is the signature $\underline{s} = D(\underline{m})$. Bob can then verify the signature by applying Alice's public verification algorithm E to $\underline{s}$ (i.e. the relation $E(\underline{s}) = E(D(\underline{m})) = \underline{m}$ must hold). See for instance [22] for a more detailed introduction to this subject.

Digital signatures are playing an important role in electronic commerce as they can replace a written signature. Several digital signature schemes are based on the discrete logarithm problem (e.g. ElGamal [11]) and the large integer factorization problem (e.g. RSA [17]).

People are now trying to design new digital signature schemes based on other mathematic hard problems. The problem of decoding general linear codes is one of those, which has been proved to be NP–complete by Berlekamp, McEliece and Van Tilborg [7]. McEliece [14] first proposed a public–key cryptosystem based on linear error–correcting codes, which derives its security from the above general decoding problem. No efficient attack on McEliece's cryptosystem has been found until now, though several computationally intensive attacks have been discussed in the literature [6, 8, 9]. Since then, several other cryptosystems based on error–correcting codes have been proposed, such as the Rao–Nam private–key cryptosystem [16], the Xinmei signature scheme [25] and the Stern identification scheme [18]. These schemes are either used to protect the secrecy or to provide the authenticity of the message according to different needs. In this paper, we will discuss the security of digital signature schemes based on error–correcting codes.

Some public–key cryptosystems are also digital signature schemes, such as RSA [17]. Unlike RSA, McEliece's public–key cryptosystem cannot be used directly as a digital signature scheme because its encryption function maps binary k–tuples to binary n–tuples and this mapping is not surjective [22]. In 1990, Xinmei Wang proposed the first digital signature scheme based on error–correcting codes [25]. Here we will refer to it as the Xinmei scheme. In the Xinmei scheme, the signature is generated in a manner similar to the way plaintext is encrypted in the Rao–Nam private–key cryptosystem [16]. The Xinmei scheme was claimed to have its security rely on the property of error–correcting codes (we will discuss this in Subsection 2.2) and the difficulty of factoring large matrices. In 1992, several methods were proposed to attack the Xinmei scheme. Harn and Wang [12] first proposed a homomorphism attack on the Xinmei Scheme without factoring large matrices, and they presented an improved scheme (we will refer to it as the Harn–Wang scheme) in which a nonlinear function is introduced to defeat the homomorphism attack. Then, Alabbadi and Wicker [1] showed that the Xinmei scheme is vulnerable to a chosen plaintext attack with complexity $O(n^3)$, where n is the length of codeword. They [2] also showed that the Harn–Wang scheme can be broken completely by a known plaintext attack with complexity $O(k^3)$, where k is the message length in the error–correcting code. Later, Van Tilburg [19] showed that one can directly obtain the signature key from public keys in both the Xinmei scheme and the Harn–Wang scheme. In 1993, Alabbadi and Wicker [3] proposed a new digital signature scheme based on error–correcting codes. At a later time in the same year, Van Tilburg [20] showed that this new scheme is not secure if one is able to verify n signatures (with linearly independent error vectors). In 1994, Alabbadi and Wicker [4] proposed a universal forgery attack to the Xinmei scheme and their own scheme. Later that year, Alabbadi and Wicker [5] presented another digital signature scheme based on error–correcting codes. In this paper we refer to it as the Alabbadi–Wicker scheme. They claimed that *the proposed scheme is resistant to the attacks that proved successful when used against the aforementioned digital signature schemes.*

The subsequent sections are organized as follows: in the second section, we will survey some attacks against the Xinmei scheme and explain why it is susceptible to them. In the third section we will give a simple description of the Alabbadi–Wicker scheme. In the fourth section a universal forgery attack against the Alabbadi–Wicker scheme is introduced. Finally, we present some comments about the security of digital signature schemes based on error–correcting codes.

2 Security analysis of the Xinmei Scheme

2.1 Description of the Xinmei Scheme

Setup phase: The signer takes a $(k \times n)$ generator matrix G of a binary Goppa code with an error–correcting capability of t errors of which t' errors can be corrected efficiently. The signer also chooses a right inverse matrix G^{-R} of G that satisfies $GG^{-R} = I_k$ where I_k is the $k \times k$ identity matrix.

The signer then publishes his public keys $t, t'(< t), H, J, W$ and T, which are given by:

$$\begin{aligned} J &= P^{-1}G^{-R}S^{-1} \\ W &= G^{-R}S^{-1} \\ T &= P^{-1}H^T \end{aligned}$$

where H is the parity check matrix of the Goppa code in standard form (so the Berlekamp–Massey algorithm can be applied) and P is an $n \times n$ full rank random matrix[1] and S is a $k \times k$ full rank matrix, called the scrambling matrix. The signer's secret keys are SG and P.

Signature phase: The signature $\underline{s}$ of a k–bit message $\underline{m}$ is obtained by computing

$$\underline{s} = (\underline{e} + \underline{m}SG)P \tag{1}$$

where $\underline{e}$ is a random n–bit error vector of Hamming weight $w(\underline{e}) \leq t'$, chosen by the signer. After the signature $\underline{s}$ is calculated, the signer sends the pair $(\underline{m}, \underline{s})$ to the verifier.

Verification phase: The authenticity of the message can be checked in the following way:

1. Calculate the syndrome $\underline{s}T = [(\underline{e} + \underline{m}SG)P]P^{-1}H^T = \underline{e}H^T$.
2. Use the Berlekamp–Massey algorithm to calculate $\underline{e}'$ from the above syndrome. If $t' < w(\underline{e}') < t$, [2] the verifier stops the verification procedure and requests a retransmission[3] of the pair $(\underline{m}, \underline{s})$ from the signer. Otherwise, the verifier takes $\underline{e}'$ as $\underline{e}$ and continues the verification.
3. Calculate $\underline{s}J$.
4. Verify whether $\underline{m} = \underline{s}J - \underline{e}W$. If this is the case, the signature has been verified and is valid.

2.2 Some weaknesses in the Xinmei Scheme

As mentioned in the introduction, the Xinmei scheme is vulnerable to several types of attacks. In the following, we will survey these attacks and analyze why the Xinmei scheme is susceptible to them.

[1] P cannot be a permutation matrix. If it is, the Xinmei Scheme reduces to the Rao–Nam private–key cryptosystem. It has been shown in [16] that the matrix SGP can be determined through majority voting if the Hamming weight of the n–bit error vector e is not in the neighbourhood of $n/2$.

[2] The original inequations are $2t' - t > w(\underline{e}') > t$. Obviously, they are wrong because $t > t'$.

[3] Allowing retransmissions in the protocol can be dangerous due to Sloppy Alice attacks [23]

- *Homomorphism attack [12].* Since the error vectors $\underline{e}$ are revealed during the verification, an attacker can choose two message–signature pairs satisfying $w(\underline{e}_1 + \underline{e}_2) \leq t'$. Then $\underline{s}_1 + \underline{s}_2$ will be a valid signature for the message $\underline{m}_1 + \underline{m}_2$. Obviously, it is the linearity of the signature in the Xinmei scheme that results in this homomorphism attack. To thwart this attack, Harn and Wang suggested modifying the Xinmei scheme with a hash function by setting $\underline{s} = h(\underline{m})SGP$.

- *Chosen–plaintext attacks [1].* If the cryptanalyst is able to get $n + 1$ different pairs of signatures and error patterns for the same message m in which n signatures are linearly independent[4], he can obtain the secret matrix P using the relation $S = EP$ where S and E are the $n \times n$ matrices with as i^{th} row $\underline{s}_i + \underline{s}_{n+1}$ respectively $\underline{e}_i + \underline{e}_{n+1}$.

  Once P is known, the cryptanalyst can obtain another secret key SG through the following chosen–plaintext attack: suppose he has obtained the k message–signature pairs for a set of linearly independent messages. Using the error patterns from the verification procedure, he can calculate SG from the equation $E' = MSG$ where E' and M are the $k \times n$ matrices with as i^{th} row $\underline{s}_i P^{-1} + \underline{e}_i$ respectively m_i.

  The linearity of the signature enables the cryptanalyst to successfully recover the secret keys P and SG in the above chosen–plaintext attacks. Besides the linearity, the knowledge of the error patterns plays an important role in this chosen–plaintext attack. In the Xinmei Scheme, the random error pattern is used to improve the security of scheme. Unfortunately, its leakage results in the failure of the Xinmei scheme. To defeat the above attack, Alabbadi and Wicker suggested to introduce a nonlinear function (hash function) $f(x, y)$ into the signature scheme [3]. In their scheme, $f(x, y)$ is used to hash the k–bit message $\underline{m}$ and the n–bit error vector $\underline{e}$ to replace the k–bit message in the Xinmei scheme.

  In addition, Alabbadi and Wicker proved that the Harn–Wang scheme is susceptible to a known–plaintext attack [2].

- *Directly recovering the secret keys from the public keys.* In the above attacks, the cryptanalyst can calculate the signer's secret keys from some triplets of messages, signatures and error patterns. Van Tilburg [19] also showed that the secret keys in the Xinmei scheme can be recovered directly from the public keys. Since G and H^T are orthogonal matrices, one can find a so–called analogous generator matrix $\tilde{G} = QG$ where Q is an unknown nonsingular $k \times k$ matrix. Following this, an analogous scrambling matrix $\tilde{S}$ can be obtained by inverting $\tilde{G}W = QGG^{-R}S^{-1} = QS^{-1} = (\tilde{S})^{-1}$. The original secret key SG then follows from $\tilde{S}\tilde{G} = SQ^{-1}QG = SG$. Finally P can be recovered from the equation $[J\tilde{S}\tilde{G}|T] = P^{-1}[W\tilde{S}\tilde{G}|H^T]$. Van Tilburg proved that $[W\tilde{S}\tilde{G}|H^T]$ has rank n. Thus the Xinmei scheme can be totally broken and the same also applies to the Harn–Wang scheme.

  Alabbadi and Wicker also tried to recover G from H and they estimated that the search is infeasible because it has complexity $O(k!)$ [1].

[4] There is an error in the original article [1], namely, *the cryptanalyst is supposed to have the ability to induce the signer to generate $n + 1$ linearly independent signatures for the same message $\underline{m}$.* This is impossible because the length of signature is only n, and unnecessary because the attack only needs n linearly independent signatures for a message $\underline{m}$

Without question, it is the redundancy in the public keys that results in the above attack. However, in order for the verifier to check the signature's validity, the signer has to publish some necessary public keys. Firstly the verifier needs to have the ability to decode the signature. The public key has to include some information about the parity check matrix[5]. Furthermore, the verifier needs to recover the message, whether in hashed form or not, (in order to defeat forgery attacks by checking whether the recovered message is equal to the received message or not) from the signature using some public keys. These public keys and the verification procedure undoubtedly leak information about the secret keys.

- *Potential threats from analogous matrices.* Is it possible for the verifier to completely defeat forgery attacks by recovering the message and checking if it is equal to the received message in the Xinmei scheme and other schemes [3, 5] ? In the following, we will explore some potential threats from analogous matrices of secret keys.

  Firstly the generator matrix G is the most important secret key in the Xinmei scheme and other schemes [3, 5]. Even though the cryptanalyst knows the parameters (length n, dimension k and minimum distance d) of the code used in these schemes, it is still difficult for the cryptanalyst to find G. For each (n, k) binary linear code, there are $(2^k-1)(2^k-2)\cdots(2^k-2^{k-1})$ different generator matrices. As a secret key the generator matrix G is protected by two nonsingular random matrices S and P against direct calculation by the attacker (by using the public keys and the verification procedure).

  Different generator matrices define different mappings from messages to signatures. But it is difficult to design a verification procedure which can check whether the signature satisfies the real mapping. This is because the real mapping is not known by the verifier or the attacker (or he could break the scheme). However, the attacker can obtain an analogous generator matrix $\tilde{G}$ from $\tilde{G}H^T = 0_{k\times(n-k)}$ because he knows the parity check matrix H. This analogous matrix $\tilde{G}$ can be found in polynomial time. Then the cryptanalyst can use $\tilde{G}$ to forge a signature. It is possible for the forged signature to pass the verification procedure because all items related to $\tilde{G}$ in the signature usually can be cancelled in the procedure of calculating the syndrome. In addition, this can also happen to other secret keys. Thus, it is possible for the cryptanalyst to forge a signature which can pass other checks in the verification procedure.

  We will show in section 4 that this method can be used to break the Alabbadi–Wicker scheme. We will first give a description of the Alabbadi–Wicker scheme.

3 The Alabbadi–Wicker scheme

In the initialization phase, each user chooses a t–error correcting binary irreducible Goppa code C with length $n = 2^m$ and dimension k. The code is described by an irreducible polynomial $G(z)$ of degree t and coefficients in $GF(2^m)$. The user then selects a $k \times n$ binary generator matrix G and a $(n-k) \times n$ binary parity check matrix H for the chosen code.

[5]In the Xinmei and other schemes, the verifier is supposed to have the ability to recover the error pattern from the signature by means of the Berlekamp–Massey algorithm. However, the Berlekamp–Massey algorithm requires the parity check matrix be in a standard form [15]. Thus the parity check matrix has to be a public key.

The user then chooses two $k \times n$ binary matrices W and V such that

$$G = W + V \tag{2}$$

and the rank of W is k. This means that there exists an $n \times k$ binary right–inverse matrix W^{-R} such that

$$WW^{-R} = I_k \tag{3}$$

where I_k is the $k \times k$ identity matrix. The matrix W^{-R} is chosen such that GW^{-R} has nonzero rank $k' < k$. Then the signer generates a nonsingular $n \times n$ binary matrix P. The final step of initializing the signature scheme is the computation of the following matrices:

$$\begin{aligned} H' &= P^{-1}H^T & (4)\\ W' &= P^{-1}W^{-R} & (5)\\ W'' &= W^{-R}GW^{-R}. & (6) \end{aligned}$$

The user publishes $G(z)$, W^{-R}, H', W', W'', t and t', where t' is an integer such that $t' < t$. The private key consists of the matrices V, W, G, $W^{-R}G$, and P.

In addition, a nonlinear one–way function $f(x, y) : GF(2^k) \times GF(2^n) \rightarrow GF(2^k)$ (hash function) is selected by a system operator and made available to all users of the system.

Suppose a user wants to sign a k–bit message $\underline{m}$. He then selects two binary vectors at random: a n–bit vector $\underline{e}$ of weight t', and a k–bit vector $\underline{r}$ of arbitrary (but nonzero) weight. The signature pair $(\underline{s}, \underline{x})$ of the message $\underline{m}$ is then computed as follows:

$$\begin{aligned} \underline{x} &= \{\underline{r}G + f(\underline{m}, \underline{e})V\}P & (7)\\ \underline{s} &= \{\underline{e} + f(\underline{m}, \underline{e})W + \underline{x}W^{-R}G\}P. & (8) \end{aligned}$$

The user transmits the triplet $\underline{x}$, $\underline{s}$, and $\underline{m}$. The receiver gets a signature pair $(\underline{x}', \underline{s}')$ along with the message $\underline{m}'$ (all these might be corrupted by noise). The signature validation is then performed the following five steps:

1. The following expression is computed:

$$\begin{aligned} \underline{x}' + \underline{s}' &= [\{\underline{r}G + f(\underline{m}, \underline{e})V + \underline{e} + f(\underline{m}, \underline{e})W + \underline{x}W^{-R}G\}P]'\\ &= [\{\underline{r}G + f(\underline{m}, \underline{e})G + \underline{e} + \underline{x}W^{-R}G\}P]' \end{aligned}$$

2. The syndrome is calculated:

$$\begin{aligned} (\underline{x}' + \underline{s}')H' &= [\{\underline{r}G + f(\underline{m}, \underline{e})G + \underline{e} + \underline{x}W^{-R}G\}P]'P^{-1}H^T\\ &= \underline{e}'H^T. \end{aligned}$$

3. The Berlekamp–Massey algorithm is applied to the above syndrome [6] to obtain the error vector $\underline{e}'$. If $w(\underline{e}') \neq t'$, the receiver requests a retransmission[7] of $\underline{x}$, $\underline{s}$ and $\underline{m}$. Otherwise the receiver continues the validation process under the assumption that $\underline{e}' = \underline{e}$, $\underline{x}' = \underline{x}$ and $\underline{s}' = \underline{s}$.

[6] This is actually an error in the scheme: applying the Berlekamp–Massey algorithm requires the parity check matrix to be in standard form [15]. As the parity check matrix is clearly not in standard form, the Berlekamp–Massey algorithm will not give the correct error vector.

[7] Allowing retransmissions in the protocol can be dangerous due to Sloppy Alice attacks [23]

4. The hash of the message and the error vector $f(\underline{m}, \underline{e})$ is recovered from $\underline{x}$, $\underline{e}$, and $\underline{s}$ by computing the following expression:

$$\begin{aligned} \underline{s}W' + \underline{x}W'' + \underline{e}W^{-R} &= \underline{s}P^{-1}W^{-R} + \underline{x}W^{-R}GW^{-R} + \underline{e}W^{-R} \\ &= \underline{e}W^{-R} + f(\underline{m}, \underline{e}) + \underline{x}W^{-R}GW^{-R} + \underline{x}W^{-R}GW^{-R} + \underline{e}W^{-R} \\ &= f(\underline{m}, \underline{e}). \end{aligned}$$

5. Finally, the verifier compares $f(\underline{m}, \underline{e})$ is compared with $f(\underline{m}', \underline{e})$. If they are equal, the signature pair is accepted as a valid signature.

4 Security cryptanalysis of the Alabbadi–Wicker scheme

Alabbadi and Wicker claimed that *the proposed scheme is resistant to the attacks that proved successful when used against the aforementioned digital signature schemes as well as other attacks.* First we will analyze how the Alabbadi–Wicker scheme defeats the attacks described in Section 2.

4.1 How the Alabbadi–Wicker scheme defeats the above attacks

The Alabbadi–Wicker scheme looks similar to the Xinmei Scheme if we combine the signatures x and s as follows:

$$\begin{aligned} \underline{x} + \underline{s} &= \{\underline{r}G + f(\underline{m}, \underline{e})G + \underline{e} + \underline{x}W^{-R}G\}P \\ &= \{\underline{e} + [\underline{r} + f(\underline{m}, \underline{e}) + \underline{x}W^{-R}]G\}P \\ &= \{\underline{e} + \underline{m}'S'G\}P \end{aligned}$$

However, Alabbadi and Wicker adopted different methods to defeat the attacks which are successful against the Xinmei scheme. First a nonlinear (hash) function f is applied to the message $\underline{m}$ and the error vector $\underline{e}$ to prevent the homomorphism attack in 2.2. Furthermore a k–bit vector $\underline{r}$ of arbitrary (but nonzero) weight has been introduced to the signature $\underline{x}$. The verifier cannot solve $\underline{r}$ from the signature $\underline{x}$ and only the signer knows it. Thus the Alabbadi–Wicker scheme can defeat both the chosen–plaintext and the known–plaintext attack in 2.2. Lastly, the generator matrix G has been split into two matrices W and V and some of the public keys (namely W^{-R}, W' and W'') include only partial information about G. So it is at least difficult to derive the secret key G from the public keys directly. A total break appears to be infeasible, primarily because the public keys do not completely describe G (this is true because the matrix $W'' = W^{-R}GW^{-R}$ is not of full rank). The cryptanalyst thus seems to be forced to perform an exhaustive search through all possible generator matrices for the code C.

However, the Alabbadi–Wicker scheme is not as secure as they claimed, *their digital signature scheme derives its security from the complexity of three problems: the decoding of general linear error–correcting block codes, the factoring of large matrices, and the derivation of a matrix from its right inverse.* In the following sections, we will present a universal forgery attack against it.

4.2 A universal forgery of the Alabbadi–Wicker scheme

In [5], Alabbadi and Wicker analyzed the possibility of a universal forgery, i.e. being able to sign an arbitrary message given only the public keys. Even though their attack did not succeed, it did motivate the following attack using analogous matrices.

4.2.1 Recovering the parity check matrix H

As was discussed in section 2.2, the parity check matrix H should be in a standard form and be published to the public so that anyone can check the signature if the Berlekamp–Massey algorithm is to be used. But the Alabbadi–Wicker scheme does not publish H. Even though the verifier can calculate the correct syndrome using H', he cannot calculate the correct error vector by means of the Berlekamp–Massey algorithm if H is not in standard form.

In addition, it is possible for the cryptanalyst to recover H from the public keys and the verification procedure. From the second and the third step in the validation of a signature we can get the following equation:

$$(\underline{x} + \underline{s})H' = \underline{e}H^T \tag{9}$$

where $\underline{x}$, $\underline{s}$ and $\underline{e}$ and H' are known to the cryptanalyst.

Suppose the cryptanalyst is able to obtain signatures with n independent error vectors $\underline{e}_i$ and the corresponding $(\underline{x}_i + \underline{s}_i)H'$ $(1 \leq i \leq n)$. Then he can solve the parity check matrix H^T from the n equations (9) by setting $H^T = E^{-1}(X + S)H'$ where

$$E = \begin{pmatrix} \underline{e}_1 \\ \underline{e}_2 \\ \vdots \\ \underline{e}_n \end{pmatrix} \quad , \quad S = \begin{pmatrix} \underline{s}_1 \\ \underline{s}_2 \\ \vdots \\ \underline{s}_n \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_n \end{pmatrix}.$$

The complexity of solving H^T in this way is only $O(n^3)$.

4.2.2 Calculating an analogous matrix $\tilde{P}$

Now let us turn our attention to trying to recover P. After the cryptanalyst has successfully recovered the parity check matrix H, he can try to find the nonsingular matrix P according to the following method. From (4) and (5) we can get the following expression:

$$[H'|W'] = P^{-1}[H^T|W^{-R}] \tag{10}$$

where H' and H^T are $n \times (n-k)$ matrices and W' and W^{-R} are $n \times k$ matrices. So $[H'|W']$, $[H^T|W^{-R}]$ and P^{-1} are $n \times n$ matrices. Alabaddi and Wicker proved that $[H^T|W^{-R}]$ is a singular matrix, so the cryptanalyst cannot find P^{-1} from equation (10). Even so, he can obtain an analogous matrix $\tilde{P}^{-1}$ which can also be used to forge a signature.

Even though $[H^T|W^{-R}]$ is not a full rank matrix, the cryptanalyst can obtain a nonsingular row transformation matrix $\tilde{P}^{-1}$ from (10), which satisfies the following equations:

$$H' = \tilde{P}^{-1}H^T \quad (11)$$
$$W' = \tilde{P}^{-1}W^{-R}. \quad (12)$$

Of course, it would be best if the matrix $\tilde{P}^{-1}$ is equal to P^{-1}. However, the cryptanalyst has no way of knowing this. The forgery is still possible, even if the two matrices are not equal. The cryptanalyst may calculate the inverse matrix $\tilde{P}$ from $\tilde{P}^{-1}$ in polynomial time. The matrix $\tilde{P}$ will play an important role in the following universal forgery.

4.2.3 Universal Forgery

Here we will use other methods to calculate an an analogous generator matrix $\tilde{G}$,

$$\tilde{G}H^T = 0_{k\times(n-k)}. \quad (13)$$

Note that $\tilde{G}$ is in general not equal to G, the generator matrix used by the signer.

Since W^{-R} is public key, the cryptanalyst can calculate a left inverse $\tilde{W}$ of W^{-R} which satisfies

$$\tilde{W}W^{-R} = I_k. \quad (14)$$

Then the cryptanalyst calculates $\tilde{V} = \tilde{G} + \tilde{W}$. Again, in general $V \neq \tilde{V}$ and $W \neq \tilde{W}$.

Since W'' and W^{-R} are public keys, the cryptanalyst can calculate a Y which satisfies the following equation.

$$W'' = W^{-R}GW^{-R} = YW^{-R}. \quad (15)$$

Now the cryptanalyst can forge the signature of any message $\underline{m}$. According to Alabbadi–Wicker scheme, an n–bit error vector $\underline{e}$ of weight t' is chosen at random. Since $\underline{r}$ is only used to protect G from the attacks in Section 2, we discard it (after all, we are trying to forge a signature, not to obscure G).

To obtain a signature for the message $\underline{m}$ the cryptanalyst first calculates $\underline{x}$ of the signature pair $(\underline{x}, \underline{s})$ from the implicit equation

$$\underline{x} = \{f(\underline{m}, \underline{e})\tilde{V} + \underline{x}Y\}\tilde{P}. \quad (16)$$

Then he can calculate $\underline{s}$ from

$$\underline{s} = \{\underline{e} + f(\underline{m}, \underline{e})\tilde{W} + \underline{x}Y\}\tilde{P}. \quad (17)$$

The cryptanalyst can now send the triple $(\underline{m}, \underline{x}, \underline{s})$ as a signed message. We now show that this triple will pass the signature validation of the Alabbadi–Wicker scheme. We will not consider the channel noise.

$$\begin{aligned}(\underline{x} + \underline{s})H' &= \{f(\underline{m}, \underline{e})\tilde{V} + \underline{e} + f(\underline{m}, \underline{e})\tilde{W}\}\tilde{P}H' \\ &= \{f(\underline{m}, \underline{e})\tilde{G} + \underline{e}\}\tilde{P}\tilde{P}^{-1}H^T \\ &= \underline{e}H^T.\end{aligned}$$

It is obvious that the signature $(\underline{x}, \underline{s})$ will pass the first three steps of the check. Now we will calculate the fourth check.

$$\begin{aligned} \underline{s}W' + \underline{x}W'' + \underline{e}W^{-R} &= \underline{s}P^{-1}W^{-R} + \underline{x}W'' + \underline{e}W^{-R} \\ &= \underline{e}W^{-R} + f(\underline{m}, \underline{e}) + \underline{x}YW^{-R} + \underline{x}W'' + \underline{e}W^{-R} \\ &= \underline{e}W^{-R} + f(\underline{m}, \underline{e}) + \underline{x}W'' + \underline{x}W'' + \underline{e}W^{-R} \\ &= f(\underline{m}, \underline{e}). \end{aligned}$$

So the forged signature has passed all steps of the check and will be accepted as valid.

4.2.4 Example

We will now present an example which shows how the above attack works. We will use the $(6, 3, 3)$–code that Alabbadi and Wicker chose for their example in [5]. The public and private keys for their scheme are:

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}, H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, W = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, P^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$W^{-R} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, H' = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, W' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, W'' = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Suppose we have recovered the parity check matrix H as described in Section 4.2.1. We will now show how to calculate the analogous matrices $\tilde{P}, \tilde{G}, \tilde{W}$ and Y from the public keys of the Alabbadi–Wicker scheme.

We first calculate $\tilde{P}^{-1}$ from equation (10) and arrive at

$$\tilde{P}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Note that many choices are possible here since $\left(H^T|W^{-R}\right)$ is not a full–rank matrix. Now we can invert this matrix to get $\tilde{P}$ (note that $\tilde{P}^{-1}$ is a full–rank matrix, so this is possible).

$$\tilde{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The matrices $\tilde{P}$ and $\tilde{P}^{-1}$ are really different from P and P^{-1}. Later we will see that this does not effect the ability to forge a signature.

Similarly, we can calculate $\tilde{G}, \tilde{W}$ and Y from equations (13,14,15) and arrive at

$$\tilde{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \tilde{W} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, Y = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The matrix $\tilde{V}$ follows from the equation $\tilde{G} = \tilde{W} + \tilde{V}$.

$$\tilde{V} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Suppose we want to sign the message $\underline{m} = (001)$ and we select the error vector $\underline{e} = (000001)$. As in [5] we take $f(\underline{m}, \underline{e}) = (011)$. According to the forgery steps (16,17) in section 4.2.3 we can calculate the signature pair $(\underline{x}, \underline{s})$ of the message $\underline{m}$ as $\underline{x} = (101111)$ and $\underline{s} = (111100)$.

We leave it to the reader to verify that this signature pair passes the verification of the Alabbadi–Wicker scheme (again supposing no errors occurred in the channel) for the message $\underline{m}$.

5 Discussion

The above universal forgery makes use of analogous matrices such as $\tilde{G}, \tilde{W}$ and Y. There exist other possible drawbacks in the Alabbadi–Wicker scheme which we shall not discuss here. Our aim is to find the reason behind the above universal forgery. This may help to improve the Alabbadi–Wicker scheme or design a new signature schemes using error–correcting codes.

From the description of the universal forgery, it seems very difficult to prevent this kind of forgery. The designers hope to hide the real secret key G in its analogous matrices because there are many analogous matrices. Thus it will be infeasible for the cryptanalyst to find

the original map between the message and signature, which is defined by the secret key G. However, they did not realize that the verifier does not have the ability to check this original map between the message and its signature. Even though the analogous matrices $\tilde{G}, \tilde{W}$ and Y define a different map, the verifier cannot detect it because he does not know the secret key.

In addition, the universal forgery in section 4.2.3 also shows that neither the Alabbadi–Wicker scheme nor the Xinmei scheme have the important property that it should be infeasible for an attacker to find a signature algorithm that passes the verification step given only this verification algorithm. In fact, there are many such signature algorithms an attacker can come up with. It is very difficult for the designer to defeat them when he designs a digital signature scheme using the same method as Xinmei scheme.

Up to now all attempts to design a secure digital signature scheme based on error–correcting codes have failed. Why is it so hard to design such a scheme? This is because these signature schemes do not really depend on the hardness of the decoding problem of general error–correcting codes.

Van Tilburg showed in [21] that *signature schemes, where the security is based only on the bounded, hard–decision decoding problem for linear codes, do not exist.* However Kabatianskii, Krouk and Smeets proposed a digital signature scheme based on random error–correcting codes in 1997 [13]. They exploited a probably unknown fact that for every linear code the set of its correctable syndromes contains a linear subspace of relatively large dimension L. Unfortunately their scheme can be used only once. Even so it does give us some new ideas to further explore the use of this intractability feature of the decoding problem.

6 Conclusion

In this paper we discussed the security of digital signature schemes based on error–correcting codes and surveyed some existing weaknesses in the Xinmei Scheme. We also explored potential threats from matrices that have the same properties as some of the secret matrices, which we called analogous matrices. As an example we presented a universal forgery of the Alabbadi–Wicker scheme.

Acknowledgements

The first author would like to thank Professor Wang for his valuable and kindly help. The second author would like to thank STW for their financial support in the project *Strong Authentication Methods*, number EWI.4536.

References

[1] M. Alabbadi and S.B. Wicker, *Security of Xinmei digital signature scheme*, Electronic Letters 28(9), 1992, pp.890–891.

[2] M. Alabbadi and S.B. Wicker, *Cryptoanalysis of the Harn and Wang modification of the Xinmei digital signature scheme*, Electronic Letters 28(18), 1992, pp.1756–1758.

[3] M. Alabbadi and S.B. Wicker, *Digital signature scheme based on error–correcting codes*, Proc. of 1993 IEEE International Symposium on Information Theory, San Antonio, USA, 1993, pp.199.

[4] M. Alabbadi and S.B. Wicker, *Susceptibility of digital signature scheme based on error–correcting codes to universal forgery*, Proc. of 1994 IEEE International Symposium on Information Theory, Trondheim, Norway, 1994, pp.494.

[5] M. Alabbadi and S.B. Wicker, *A digital signature scheme based on linear error–correcting block codes*, Advance in Cryptology, ASIACRYPT'94, pp.238–248.

[6] T. A. Berson, *Failure of the McEliece Public–key Cryptosystem under message–resent and related message attack*, Advance in Cryptology, Crypto'97, pp.213–220.

[7] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Transactions on Information Theory, Vol.24, No.3, 1978, pp.384–386.

[8] F. Chabaud, *on the security of some cryptosystems based on error–correcting codes*, Advance in cryptology, Eurocrypt'94, pp.131–139.

[9] A. Canteaut and N. Sendrier, *Cryptanalysis of the original McEliece cryptosystem*, Advance in Cryptology, Aisacrypt'98, pp.187–199.

[10] W. Diffie and M.E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory, Vol.22, No.6, 1976, pp.644–654.

[11] T. ElGamal, *A public–key cryptosystem and a signature scheme based on discrete logarithms*, Advances in Cryptography, Crypto'84, pp.10–18, 1985.

[12] L. Harn and D.C. Wang, *Cryptoanalysis and modification of digital signature scheme based on error–correcting codes*, Electronic Letters 28(2), 1992, pp.157–159.

[13] G. Kabatianskii, E. Krouk and B. Smeets, *A digital signature scheme based on random error–correcting codes*, the 6th IMA International Conference Cirencester, UK, December 1997, pp.161–177.

[14] R.J. McEliece, *A public–key Cryptosystem Based on Algebraic Coding Theory*, DSN progress report 42–44, pp.114–116, 1978.

[15] J. Macwilliams and N.J. Sloane, *The theory of error–correcting codes*, New York: North–Holland Publishing Company, 1978.

[16] T.R.N. Rao and K.H. Nam, *Private–key algebraic–code encryptions*, IEEE Transactions on Information Theory, Vol.35, No.4, pp.445–457, 1989.

[17] R.L. Rivest, A. Shamir and L.M. Adleman, *A method for obtaining digital signatures and public–key cryptosystems*, Communications of the ACM, Vol.21, No.2, pp.120–126, 1978.

[18] J. Stern, *A New Identification Scheme Based on Syndrome Decoding*, Advances in Cryptology: CRYPTO'93, Springer–Verlag, Berlin, 1994, pp.13–21.

[19] J. van Tilburg, *Cryptanalysis of Xinmei digital signature scheme*, Electronic Letters 28(20), 1992, pp.1935–1936.

[20] J. van Tilburg, *Cryptanalysis of the Alabbadi–Wicker digital signature scheme*, Proc. of Fourteenth Symposium on Information Theory in the Benelux, Veldhoven, Netherlands, May 1993, pp.114–119.

[21] J. van Tilburg, *Security–analysis of a class of cryptosystems based on linear error–correcting codes*, Ph.D thesis, Eindhoven University of Technology, 1994.

[22] H.C.A. van Tilborg, *An interactive introduction to cryptology*, Eindhoven University of Technology, 1999.

[23] E. Verheul, J.M. Doumen and H.C.A. van Tilborg, *Sloppy Alice Attacks! Adaptive Chosen Ciphertext Attacks on the McEliece cryptosystem*, to be published.

[24] S.B. Xu and J.M. Doumen, *An attack against the Alabbadi–Wicker scheme*, The 20th symposium on information theory in the Benelux, Haasrode, Belgium, 27–28 May, 1999.

[25] X.M. Wang, *Digital signature scheme based on error–correcting codes*, Electronics Letters, Vol.26, No.13, pp.898–899, June 21st, 1990.