generates several peaks (shown in Fig. 3), and it is difficult to distinguish which peaks are generated by which images of the input. However, the peak values and their corresponding positions shown in Table 1 indicate only four significant peaks. To obtain the images corresponding to these four peaks, the DRFT-based correlator is applied. To see which image in both the reference and the input generates the first peak, we move Fig. 2 40 pixels in the X-direction and 124 pixels in the Y-direction. The moved image is correlated with the input object by the DRFT-based correlator, resulting in the displayed image shown in Fig. 4a, which is the 'A and B' image in the top righthand corner of Fig. 1. By applying the same procedures to the second, third, and fourth peaks, we obtain two single Bs and one single A, shown in Fig. 4b-d. The conclusion of the DRFT-based correlation between Figs. 1 and 2 and B' image in the top righthand corner of Fig. 1. By applying the DRFT contains both time and frequency information (name, address, etc.) and computes the secret key shared with U, as shown in Table 1. The Table 1: Seven larger peak values of Fig. 3

<table>
<thead>
<tr>
<th>Peak number</th>
<th>Peak value</th>
<th>Co-ordinate (x, y)</th>
<th>Corresponding input object</th>
<th>Matching reference object</th>
<th>Position</th>
<th>Areas (pixels)</th>
<th>120-peak value</th>
<th>DRFT-based correlator output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7969</td>
<td>(40, 124)</td>
<td>Upper right A, B</td>
<td>A, B</td>
<td>A(73.30) B(82.46)</td>
<td>102</td>
<td>Fig. 4a</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.5078</td>
<td>(116, 40)</td>
<td>Lower left B</td>
<td>A</td>
<td>B(32.98)</td>
<td>65</td>
<td>Fig. 4b</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.5078</td>
<td>(117, 91)</td>
<td>Upper left B</td>
<td>A</td>
<td>B(38.18)</td>
<td>65</td>
<td>Fig. 4c</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.2891</td>
<td>(116, 18)</td>
<td>Left A</td>
<td>A</td>
<td>B(17.50)</td>
<td>37</td>
<td>Fig. 4d</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.2578</td>
<td>(8, 8)</td>
<td>Z</td>
<td>B</td>
<td>B(55.59)</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.2188</td>
<td>(54, 100)</td>
<td>X</td>
<td>B</td>
<td>B(97.20)</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.2031</td>
<td>(31, 43)</td>
<td>Y</td>
<td>B</td>
<td>B(73.94)</td>
<td>26</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Correlation: The DRFT contains both time and frequency information. Thus, the DRFT-based correlator can not only generate correlation peaks, but also display the corresponding correlated images. With this property, the DRFT-based correlator can be used to recognise multiple different objects.

References

Improvement of Saeednia's self-certified key exchange protocols

Tzong-Chen Wu, Yuh-Shihng Chang and Tzouh-Yi Lin

In 1997, two self-certified key exchange protocols were proposed by Saeednia. It is shown that Saeednia's self-certified key exchange protocols are insecure in that an adversary may impersonate any legitimate user in key exchange. An improvement against the impersonation attack is described.

Introduction: Saeednia [1] presented two key exchange protocols based on Girault's self-certified public key system [2]. In Saeednia's key exchange protocols, there exists a trusted third party (TTP) for system setup and user registration; however, the TTP does not know the secret key of any user during user registration. Saeednia's key exchange protocols preserve the merits inherent in both the identity-based system and the self-certified system, and hence allow a considerable reduction in communication complexity. In this Letter, we first show that Saeednia's self-certified key exchange protocols are insecure. An adversary may impersonate any legitimate user in running these protocols. We also present an improvement that can withstand the impersonation attack.

Saeednia's key exchange protocol: In the setup of this system, the TTP chooses an integer n as the product of two large distinct primes p and q of almost the same size such that \( p = 2q' + 1 \) and \( q = 2q'' + 1 \), where \( p' \) and \( q' \) are also primes, a base \( g \) of \( 1 \) or \( 0 \) and \( r = q' \), a large integer \( u < r \), and a one-way hash function \( f \). The TTP makes \( g, u, f, n, p, q, r, \) and \( n \) public, keeps \( r \) secret and discards \( p \) and \( q \) afterwards. Next, any user \( U_i \) can register with the TTP by performing the following steps:

(i) \( U_i \) randomly chooses a secret key \( x_i \in Z_p \) computes the public key \( y_i = g^{x_i} \mod n \) and gives it to the TTP.

(ii) The TTP prepares a string \( I \) associated with \( U_i \)'s personal information (name, address, etc.) and computes \( U_i \)'s identity \( I_D = f(I) \).

(iii) The TTP computes \( w_i = y_i^{u/2} \mod n \) as a witness and sends \( \{I, w_i\} \) to \( U_i \).

(iv) \( U_i \) verifies the identity and the witness by checking that \( y_i = w_i^{u/2} \mod n \). Saeednia claimed that forging a valid witness \( w_i \) for \( U_i \) is equivalent to breaking an instance of the RSA cryptosystem [3]. Suppose that \( U_i \) and \( U_j \) want to exchange a secret key to be used for secure communication. They can perform the following protocols.

Protocol 1

(i) \( U_i \) sends \( \{I, w_i\} \) to \( U_j \).

(ii) \( U_j \) sends \( \{I, w_j\} \) to \( U_i \).

(iii) \( U_i \) computes the secret key shared with \( U_j \) as \( k = w_i^{u/2} \mod n \).

(iv) \( U_j \) computes the secret key shared with \( U_i \) as \( k = w_j^{u/2} \mod n \).

Protocol 2

(i) \( U_i \) randomly chooses a secret integer \( t_i \in Z_q \), computes \( v_i = g^{t_i} \mod n \) and sends \( \{I, w_i, v_i\} \) to \( U_j \).

(ii) \( U_j \) randomly chooses a secret integer \( t_j \in Z_q \), computes \( v_j = g^{t_j} \mod n \) and sends \( \{I, w_j, v_j\} \) to \( U_i \).

(iii) \( U_i \) computes the secret key shared with \( U_j \) as \( k = w_i^{u/2} \cdot t_j \cdot v_j \mod n \).

(iv) \( U_j \) computes the secret key shared with \( U_i \) as \( k = w_j^{u/2} \cdot t_i \cdot v_i \mod n \).
Note that the secret key shared between $U_i$ and $U_j$ can be regarded as

$$k = y_j^{x_i} = y_i^{x_j} = g^{x_i x_j} \pmod{n}$$

(mod n) (in protocol 1)

or

$$k = y_j^{x_i} \cdot v_j^{x_j} = y_i^{x_j} \cdot v_i^{x_i} = g^{x_j(x_i + x_j)} \pmod{n}$$

(mod n) (in protocol 2)

**Attack on Saeednia’s key exchange protocols:** Consider the case that an adversary pretends to act as $U_i$ and tries to exchange a secret key with $U_j$, such that $U_i$ will indeed share the secret key with $U_j$. First of all, the adversary randomly chooses an integer $a \in Z_n$. Then he sets $x_j = a \cdot f(U_j)$ as a false secret key for $U_i$ and replaces $U_j$’s original public key $y_j$ (maintained by the TTP) with $y_j' = g^{y_j} \pmod{n}$. Thereafter, the adversary can easily compute a valid witness $w_j' \equiv g^{x_j} \pmod{n}$ for $U_j$, since the substituted public key $y_j' = g^{y_j(a)} = w_j' \pmod{n}$ is also self-certified. In protocol 1, if the adversary sends $(y_i, w_i')$ to $U_i$ in step (i) and intercepts the message $(y_i, w_i')$ that is intended to be sent to $U_i$ in step (ii), both the adversary and $U_j$ will share the same secret key $k' = w_i'^{y_j(a)} = w_i'^{y_j} \equiv g^{x_j} \pmod{n}$. Moreover, $U_j$ will believe that he indeed communicates with $U_i$ because the pair $(y_j', w_i')$ will be authenticated successfully. It can be seen that protocol 2 is also vulnerable to such an attack.

**Our improvement:** The weakness of Saeednia’s key exchange protocols is that the witness $w_i$ computed by the TTP is not self-certified, although to forge a valid $w_i$ is equivalent to breaking an instance of the RSA cryptosystem. We can easily remove this weakness by replacing step (iii) of the user registration phase with

(iii*) The TTP computes $w_i = (y_i \cdot ID)^{y_j^{-1}} \pmod{n}$ as a witness and sends $(y_i, w_i)$ to $U_i$.

Our improvement only requires one more subtraction than the original step. Note that $y_i = g^{x_i} = (w_j^{y_j} + ID) \pmod{n}$. This implies that, without knowing $x_i$, any adversary can easily compute an authenticated pair $(w_i, y_i)$ for $U_i$ satisfying the check in step (iv) of the user registration phase. However, the adversary still does not know $U_i$’s secret key $x_i$ unless he can solve the problem of computing discrete logarithms modulo a large composite $[5, 6]$. Since $x_i$ is unknown, the adversary cannot pretend to act as $U_i$ to share a secret key with $U_j$. That is, in steps (iii) and (iv) of protocols 1 and 2, the adversary and $U_j$ cannot obtain the same secret key.

References


---

**40Gbit/s EAM driver IC in SiGe bipolar technology**

R. Schmid, T.F. Meister, M. Rest and H.-M. Rein

An SiGe bipolar IC for directly driving a differential electroabsorption modulator in a 40Gbit/s fibre optic TDM system is presented. An adjustable modulator bias voltage (0 to −2V) is generated on-chip by a novel active network in the output stage. Clear eye diagrams at 40Gbit/s and output swings up to 2.5V_{pp} (nominal 2V_{pp}) were measured on mounted chips.

**Introduction:** Today, there is a worldwide trend towards increasing the data rate in fibre optic TDM systems from 10 to 40Gbit/s (cf. [1]). A severe bottleneck in the electronics of such a system is the modulator driver on the transmitting side. This is because of the contradicting demands on high voltage swing and high operating speed [2]. Electroabsorption modulators (EAM) need a lower voltage swing compared to Mach-Zehnder interferometers (MZI) and, therefore, be more suitable to be driven by monolithic integrated circuits (rather than by expensive hybrid amplifiers). However, as one disadvantage, the EAM represents a capacitive load, in contrast to the 50Ω input of an MZI. As a consequence, at 40Gbit/s it would be extremely difficult to drive the modulator via a 50Ω transmission line. In this case, due to insufficient matching, double-reflections would occur which increase time jitter. To solve these problems, we prefered (in contrast to the usual practice) to bond the driver outputs directly to the EAM chip. High-speed performance is further improved and mounting costs are reduced by implementing the (low-ohmic) load resistors as well as the EAM biasing on the driver chip.

Promising results at 40Gbit/s have been published for ICs in III-V compound semiconductors which show single-ended voltage swings of 2.2V_{pp} [3, 4], while in the presentation corresponding to [4], even 40Gbit/s at 2.9V_{pp} swing is claimed. However, all these results were achieved for on-chip measurements only. Moreover, capacitive loading by an EAM and the resulting problems, mentioned above, are not considered.

To reliably achieve 40Gbit/s with the SiGe bipolar technology available here, we must reduce the voltage swing at the driver output. For this, a symmetrical EAM configuration was used which can be driven by differential signals with a voltage swing of 2V_{pp} (2 × 1V_{pp}).

**Circuit design:** Fig. 1 shows the circuit concept of the driver IC, which consists of two current switches (CS1, CS2) each driven by three emitter follower (EF) pairs, as in our previously published designs [2, 5]. However, due to the direct coupling of driver and EAM chip, the output stage must now be much more sophisticated.

First, the output current switch (CS2) is extended to a cascade configuration by a grounded-base stage (GBS), which mitigates potential breakdown problems of the output transistors and, moreover, slightly increases the current driving speed. Owing to the strong capacitive loading by the EAM, the output time constant must be reduced by low output resistances (nominal R_{out} = 25Ω), requiring a high switching current in the output stage (nominal IA = 40mA). For generating the EAM bias, the output resistors are connected to cascaded EFs (EF1, EF2). The bias voltage V_{bias} equals the...